18.6 CCM Memory Map/Register Definition

NOTE

CCM Register reset values may not be the same in ROM. See for more information.

CCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_4000	CCM Control Register (CCM_CCR)	32	R/W	0401_16FFh	18.6.1/843
20C_4004	CCM Control Divider Register (CCM_CCDR)	32	R/W	0000_0000h	18.6.2/845
20C_4008	CCM Status Register (CCM_CSR)	32	R	0000_0010h	18.6.3/846
20C_400C	CCM Clock Switcher Register (CCM_CCSR)	32	R/W	0000_0100h	18.6.4/847
20C_4010	CCM Arm Clock Root Register (CCM_CACRR)	32	R/W	0000_0000h	18.6.5/849
20C_4014	CCM Bus Clock Divider Register (CCM_CBCDR)	32	R/W	0001_8D00h	18.6.6/850
20C_4018	CCM Bus Clock Multiplexer Register (CCM_CBCMR)	32	R/W	0002_0324h	18.6.7/852
20C_401C	CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)	32	R/W	00F0_0000h	18.6.8/855
20C_4020	CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)	32	R/W	02B9_2F06h	18.6.9/858
20C_4024	CCM Serial Clock Divider Register 1 (CCM_CSCDR1)	32	R/W	0049_0B00h	18.6.10/ 859
20C_4028	CCM SSI1 Clock Divider Register (CCM_CS1CDR)	32	R/W	0EC1_02C1h	18.6.11/ 862
20C_402C	CCM SSI2 Clock Divider Register (CCM_CS2CDR)	32	R/W	0007_36C1h	18.6.12/ 864
20C_4030	CCM D1 Clock Divider Register (CCM_CDCDR)	32	R/W	33F7_1F92h	18.6.13/ 866
20C_4034	CCM HSC Clock Divider Register (CCM_CHSCCDR)	32	R/W	0002_A150h	18.6.14/ 868
20C_4038	CCM Serial Clock Divider Register 2 (CCM_CSCDR2)	32	R/W	0002_A150h	18.6.15/ 870
20C_403C	CCM Serial Clock Divider Register 3 (CCM_CSCDR3)	32	R/W	0001_0841h	18.6.16/ 872
20C_4048	CCM Divider Handshake In-Process Register (CCM_CDHIPR)	32	R	0000_0000h	18.6.17/ 874
20C_4054	CCM Low Power Control Register (CCM_CLPCR)	32	R/W	0000_0079h	18.6.18/ 877
20C_4058	CCM Interrupt Status Register (CCM_CISR)	32	w1c	0000_0000h	18.6.19/ 880
20C_405C	CCM Interrupt Mask Register (CCM_CIMR)	32	R/W	FFFF_FFFFh	18.6.20/ 883
20C_4060	CCM Clock Output Source Register (CCM_CCOSR)	32	R/W	000A_0001h	18.6.21/ 886
20C_4064	CCM General Purpose Register (CCM_CGPR)	32	R/W	0000_FE62h	18.6.22/ 889

Table continues on the next page...

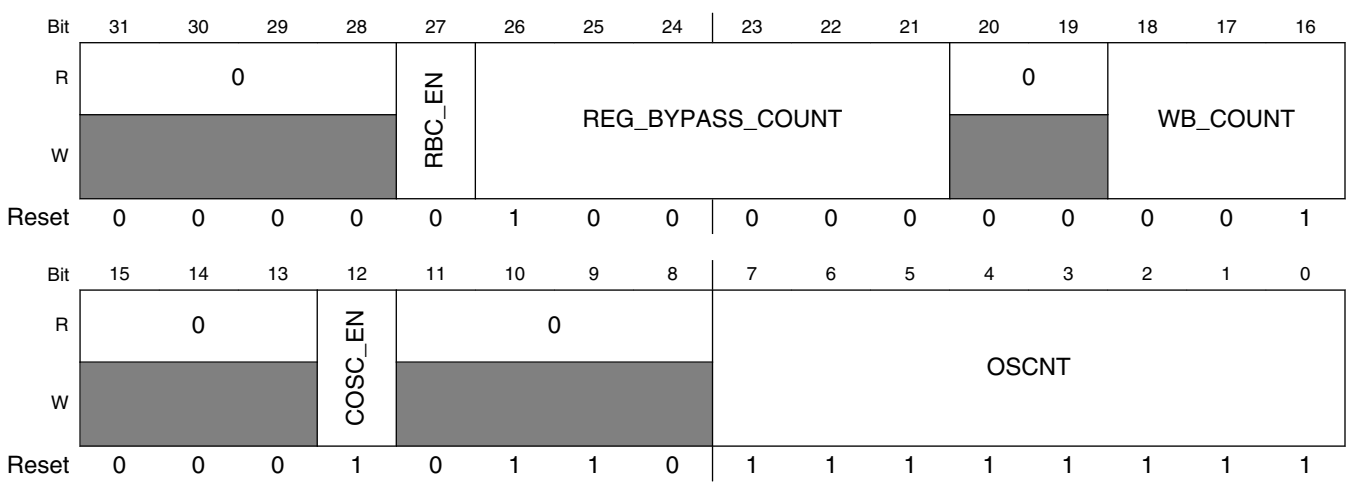
CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_4068	CCM Clock Gating Register 0 (CCM_CCGR0)	32	R/W	FFFF_FFFFh	18.6.23/ 890
20C_406C	CCM Clock Gating Register 1 (CCM_CCGR1)	32	R/W	FFFF_FFFFh	18.6.24/ 892
20C_4070	CCM Clock Gating Register 2 (CCM_CCGR2)	32	R/W	FC3F_FFFFh	18.6.25/ 893
20C_4074	CCM Clock Gating Register 3 (CCM_CCGR3)	32	R/W	FFFF_FFFFh	18.6.26/ 895
20C_4078	CCM Clock Gating Register 4 (CCM_CCGR4)	32	R/W	FFFF_FFFFh	18.6.27/ 896
20C_407C	CCM Clock Gating Register 5 (CCM_CCGR5)	32	R/W	FFFF_FFFFh	18.6.28/ 897
20C_4080	CCM Clock Gating Register 6 (CCM_CCGR6)	32	R/W	FFFF_FFFFh	18.6.29/ 899
20C_4088	CCM Module Enable Overide Register (CCM_CMEOR)	32	R/W	FFFF_FFFFh	18.6.30/ 900

18.6.1 CCM Control Register (CCM_CCR)

The figure below represents the CCM Control Register (CCR), which contains bits to control general operation of CCM. The table below provides its field descriptions.

Address: 20C_4000h base + 0h offset = 20C_4000h



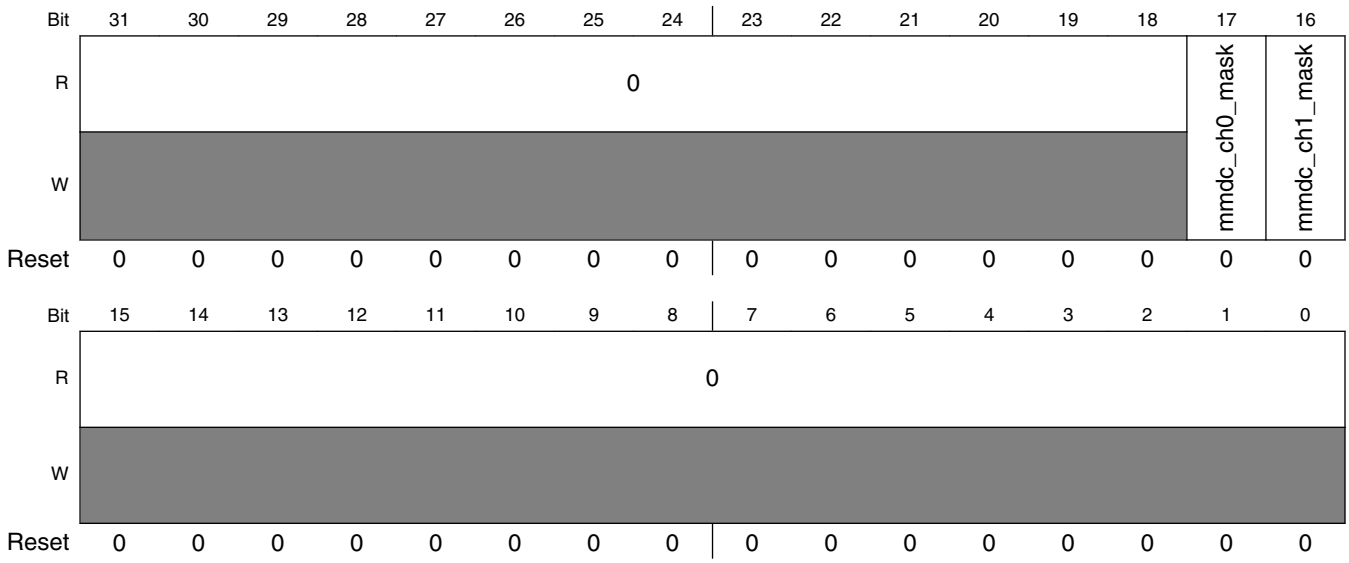
CCM_CCR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27 RBC_EN	Enable for REG_BYPASS_COUNTER. If enabled, analog_reg_bypass signal will be asserted after REG_BYPASS_COUNT clocks of CKIL, after standby voltage is requested. If standby voltage is not requested analog_reg_bypass won't be asserted, event if counter is enabled. 1 REG_BYPASS_COUNTER enabled. 0 REG_BYPASS_COUNTER disabled
26–21 REG_BYPASS_ COUNT	Counter for analog_reg_bypass signal assertion after standby voltage request by PMIC_STBY_REQ. Should be zeroed and reconfigured after exit from low power mode. REG_BYPASS_COUNT can also be used for holding off interrupts when the PGC unit is sending signals to power gate the core. 000000 no delay 000001 1 CKIL clock period delay 111111 63 CKIL clock periods delay
20–19 Reserved	This read-only field is reserved and always has the value 0.
18–16 WB_COUNT	Well Bias counter. Delay, defined by this value, counted by CKIL clock will be applied till well bias is enabled at exit from wait or stop low power mode. Counter will be used if wb_core_at_lpm or wb_per_at_lpm bits are set. Should be zeroed and reconfigured after exit from low power mode. 000 no delay 001 1 CKIL clock delay 111 7 CKIL clocks delay
15–13 Reserved	This read-only field is reserved and always has the value 0.
12 COSC_EN	On chip oscillator enable bit - this bit value is reflected on the output cosc_en. The system will start with on chip oscillator enabled to supply source for the PLLs. Software can change this bit if a transition to the bypass PLL clocks was performed for all the PLLs. In cases that this bit is changed from '0' to '1' then CCM will enable the on chip oscillator and after counting oscnt ckil clock cycles it will notify that on chip oscillator is ready by a interrupt cosc_ready and by status bit cosc_ready. The cosc_en bit should be changed only when on chip oscillator is not chosen as the clock source. 0 disable on chip oscillator 1 enable on chip oscillator
11–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
OSCNT	Oscillator ready counter value. These bits define value of 32KHz counter, that serve as counter for oscillator lock time. This is used for oscillator lock time. Current estimation is ~5ms. This counter will be used in ignition sequence and in wake from stop sequence if sbyos bit was defined, to notify that on chip oscillator output is ready for the dpll_ip to use and only then the gate in dpll_ip can be opened. 00000000 count 1 ckil 11111111 count 256 ckil's

18.6.2 CCM Control Divider Register (CCM_CCDR)

The figure below represents the CCM Control Divider Register (CCDR), which contains bits that control the loading of the dividers that need handshake with the modules they affect. The table below provides its field descriptions.

Address: 20C_4000h base + 4h offset = 20C_4004h



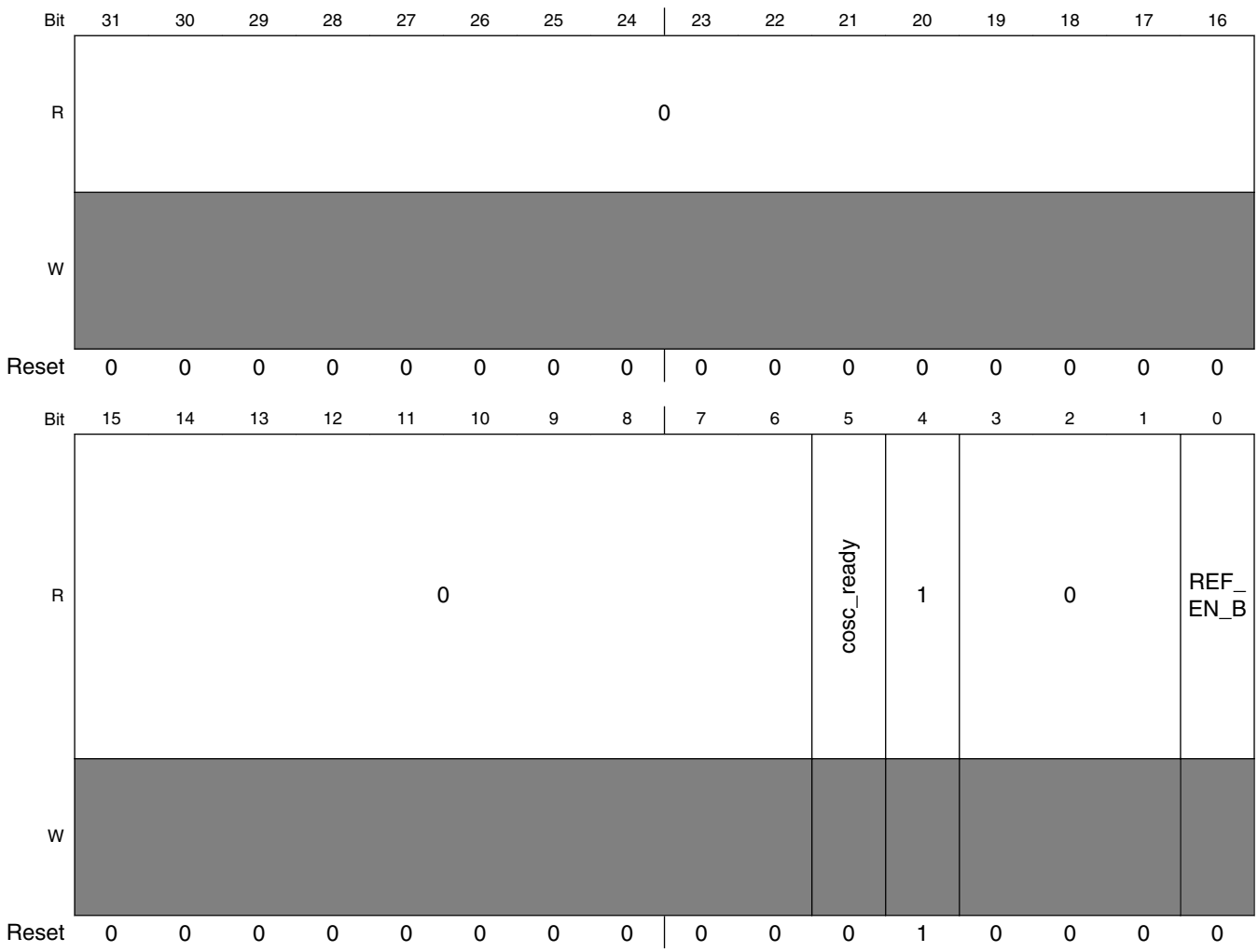
CCM_CCDR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 mmdc_ch0_mask	During divider ratio mmdc_ch0_axi_podf change or sync mux periph_clk_sel change (but not jtag) or SRC request during warm reset, mask handshake with mmdc_ch0 module. 0 allow handshake with mmdc_ch0 module 1 mask handshake with mmdc_ch0. Request signal will not be generated.
16 mmdc_ch1_mask	During divider ratio mmdc_ch1_axi_podf change or sync mux periph2_clk_sel change (but not jtag) or SRC request during warm reset, mask handshake with mmdc_ch1 module. 0 allow handshake with mmdc_ch1 module 1 mask handshake with mmdc_ch1. Request signal will not be generated.
Reserved	This read-only field is reserved and always has the value 0.

18.6.3 CCM Status Register (CCM_CSR)

The figure below represents the CCM status Register (CSR). The status bits are read-only bits. The table below provides its field descriptions.

Address: 20C_4000h base + 8h offset = 20C_4008h



CCM_CSR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5 cosc_ready	Status indication of on board oscillator. This bit will be asserted if on chip oscillator is enabled and on chip oscillator is not powered down, and if oscnt counter has finished counting.

Table continues on the next page...

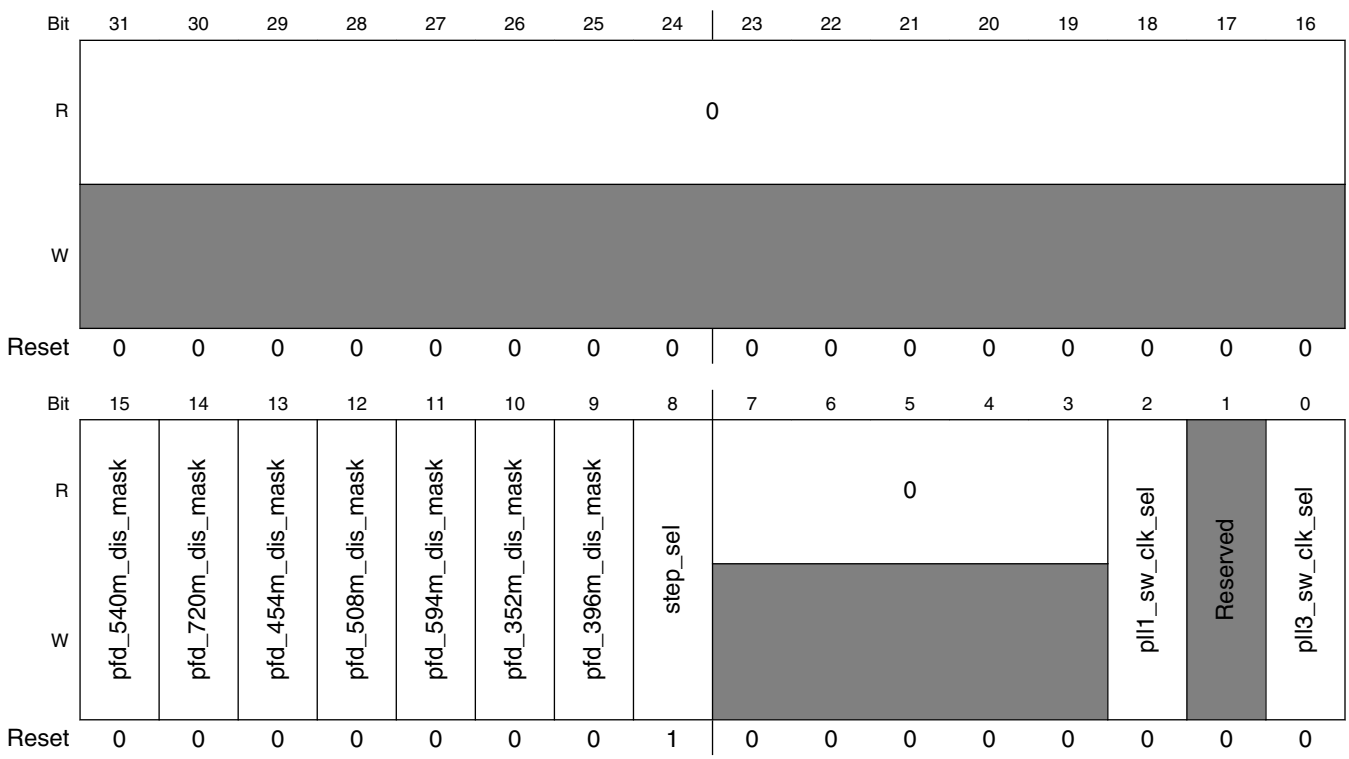
CCM_CSR field descriptions (continued)

Field	Description
	0 on board oscillator is not ready. 1 on board oscillator is ready.
4 Reserved	This read-only field is reserved and always has the value 1.
3–1 Reserved	This read-only field is reserved and always has the value 0.
0 REF_EN_B	Status of the value of CCM_REF_EN_B output of ccm 0 value of CCM_REF_EN_B is '0' 1 value of CCM_REF_EN_B is '1'

18.6.4 CCM Clock Switcher Register (CCM_CCSR)

The figure below represents the CCM Clock Switcher register (CCSR). The CCSR register contains bits to control the switcher sub-module dividers and multiplexers. The table below provides its field descriptions.

Address: 20C_4000h base + Ch offset = 20C_400Ch



CCM_CCSR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 pfd_540m_dis_mask	Mask of 540M PFD auto-disable. 0 - 540M PFD disable=0 (PFD always on) 1 540M PFD disable is managed by associated dividers disable. If all 540M-driven dividers are closed, PFD is disabled.
14 pfd_720m_dis_mask	Mask of 720M PFD auto-disable. 0 720M PFD disable=0 (PFD always on) 1 720M PFD disable is managed by associated dividers disable. If all 720M-driven dividers are closed, PFD is disabled.
13 pfd_454m_dis_mask	Mask of 454M PFD auto-disable. 0 454M PFD disable=0 (PFD always on) 1 454M PFD disable is managed by associated dividers disable. If all 454M-driven dividers are closed, PFD is disabled.
12 pfd_508m_dis_mask	Mask of 508M PFD auto-disable. 0 508M PFD disable=0 (PFD always on) 1 508M PFD disable is managed by associated dividers disable. If all 508M-driven dividers are closed, PFD is disabled.
11 pfd_594m_dis_mask	Mask of 594M PFD auto-disable. 0 594M PFD disable=0 (PFD always on) 1 594M PFD disable is managed by associated dividers disable. If all 594M-driven dividers are closed, PFD is disabled.
10 pfd_352m_dis_mask	Mask of 352M PFD auto-disable. 0 352M PFD disable=0 (PFD always on) 1 352M PFD disable is managed by associated dividers disable. If all 352M-driven dividers are closed, PFD is disabled.
9 pfd_396m_dis_mask	Mask of 396M PFD auto-disable. 0 396M PFD disable=0 (PFD always on) 1 396M PFD disable is managed by associated dividers disable. If all 396M-driven dividers are closed, PFD is disabled.
8 step_sel	Selects the option to be chosen for the step frequency when shifting ARM frequency. This will control the step_clk. **NOTE:** This mux is allowed to be changed only if its output is not used, i.e. ARM uses the output of pll1, and step_clk is not used. 0 osc_clk (24M) - source for lp_apm. 1 PLL2 PFD2 clock
7–3 Reserved	This read-only field is reserved and always has the value 0.
2 pll1_sw_clk_sel	Selects source to generate pll1_sw_clk. 0 pll1_main_clk 1 step_clk

Table continues on the next page...

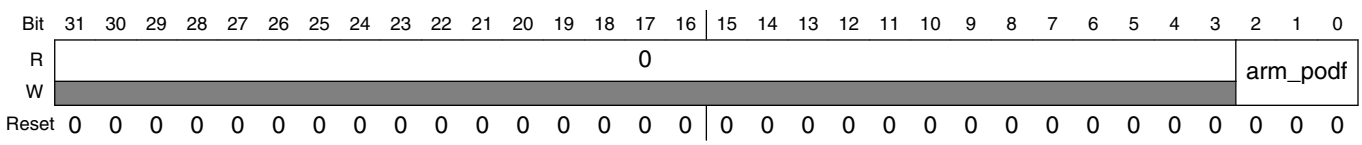
CCM_CCSR field descriptions (continued)

Field	Description
1 -	This field is reserved. Reserved
0 pll3_sw_clk_sel	Selects source to generate pll3_sw_clk. This bit should only be used for testing purposes. 0 pll3_main_clk 1 pll3 bypass clock

18.6.5 CCM Arm Clock Root Register (CCM_CACRR)

The figure below represents the CCM Arm Clock Root register (CACRR). The CACRR register contains bits to control the ARM clock root generation. The table below provides its field descriptions.

Address: 20C_4000h base + 10h offset = 20C_4010h



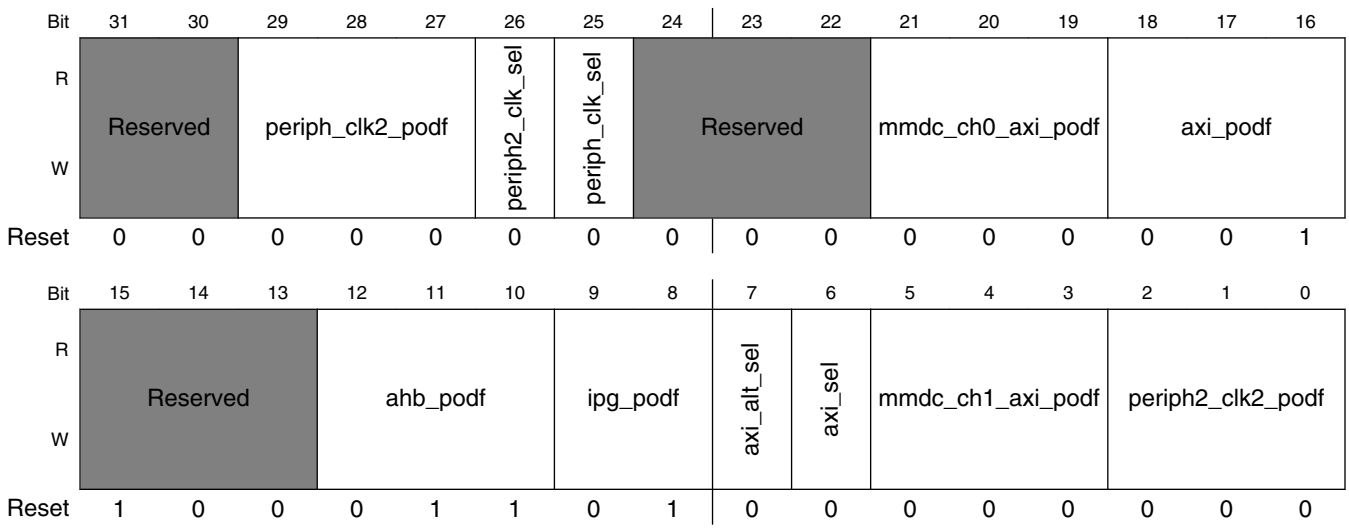
CCM_CACRR field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
arm_podf	Divider for ARM clock root. **NOTE:** If arm_freq_shift_divider is set to '1' then any new write to arm_podf will be held until arm_clk_switch_req signal is asserted. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

18.6.6 CCM Bus Clock Divider Register (CCM_CBCDR)

The figure below represents the CCM Bus Clock Divider Register (CBCDR). The CBCDR register contains bits to control the clock generation sub module dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 14h offset = 20C_4014h



CCM_CBCDR field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–27 periph_clk2_podf	Divider for periph2 clock podf. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
26 periph2_clk_sel	Selector for peripheral2 main clock (source of mmdc_ch1_clk_root). **NOTE:** Any change of this mux select will involve handshake with the MMDC. Refer to the CCDR and CDHIPR registers for the handshake bypass and busy bits. 0 PLL2 (pll2_main_clk) 1 derive clock from periph_clk2_clk clock source.

Table continues on the next page...

CCM_CBCDR field descriptions (continued)

Field	Description
25 periph_clk_sel	Selector for peripheral main clock (source of MMDC_CH0_CLK_ROOT). **NOTE:** Alternative clock source should be used when PLL is relocked. For PLL relock procedure pls refer to the PLL chapter. **NOTE:** Any change of this sync mux select will involve handshake with the MMDC. Refer to the CCDR and CDHIPR registers for the handshake bypass and busy bits. 0 PLL2 (pll2_main_clk) 1 derive clock from periph_clk2_clk clock source.
24–22 -	This field is reserved. Reserved
21–19 mmdc_ch0_axi_podf	Divider for mmdc_ch0_axi podf. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
18–16 axi_podf	Divider for axi podf. **NOTE:** Any change of this divider might involve handshake with EMI and IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–13 -	This field is reserved. Reserved
12–10 ahb_podf	Divider for AHB PODF. **NOTE:** Any change of this divider might involve handshake with EMI and IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

Table continues on the next page...

CCM_CBCDR field descriptions (continued)

Field	Description
9–8 ipg_podf	Divider for ipg podf. **NOTE:** SDMA module will not support ratio of 1:3 and 1:4 for ahb_clk:ipg_clk. In case SDMA is used, then those ratios should not be used. 00 divide by 1 01 divide by 2 10 divide by 3 11 divide by 4
7 axi_alt_sel	AXI alternative clock select 0 pll2 396MHz PFD will be selected as alternative clock for AXI root clock 1 pll3 540MHz PFD will be selected as alternative clock for AXI root clock
6 axi_sel	AXI clock source select 0 Periph_clk output will be used as AXI clock root 1 AXI alternative clock will be used as AXI clock root
5–3 mmdc_ch1_axi_podf	Divider for mmdc_ch1_axi podf. **NOTE:** This design implementation does not use MMDC_CH1_CLK_ROOT as a clock source to the MMDC. Only MMDC_CH0_CLK_ROOT is used. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
periph2_clk2_podf	Divider for periph2_clk2 podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

18.6.7 CCM Bus Clock Multiplexer Register (CCM_CBCMR)

The figure below represents the CCM Bus Clock Multiplexer Register (CBCMR). The CBCMR register contains bits to control the multiplexers that generate the bus clocks. The table below provides its field descriptions.

NOTE

Any change on the above multiplexer will have to be done while the module that its clock is affected is not functional and the respective clock is gated in LPCG. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

The change for arm_clk_sel should be done through sdma so that ARM will not use this clock during the change and the clock will be gated in LPCG.

Address: 20C_4000h base + 18h offset = 20C_4018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	gpu3d_shader_podf			gpu3d_core_podf			gpu2d_core_clk_podf		pre_periph2_clk_sel			periph2_clk2_sel	pre_periph_clk_sel		gpu2d_core_clk_sel	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	vpu_axi_clk_sel		periph_clk2_sel		vdoaxi_clk_sel	pcie_axi_clk_sel	gpu3d_shader_clk_sel		Reserved		gpu3d_core_clk_sel		Reserved		gpu3d_axi_clk_sel	gpu2d_axi_clk_sel
Reset	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0

CCM_CBCMR field descriptions

Field	Description
31–29 gpu3d_shader_podf	Divider for gpu3d_shader clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
28–26 gpu3d_core_podf	Divider for gpu3d_core clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2

Table continues on the next page...

CCM_CBCMR field descriptions (continued)

Field	Description
	010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
25–23 gpu2d_core_clk_podf	Divider for gpu2d_core clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
22–21 pre_periph2_clk_sel	Selector for pre_periph2 clock multiplexer 00 derive clock from PLL2 main 528MHz clock 01 derive clock from 396MHz PLL2 PFD 10 derive clock from 352M PFD 11 derive clock from 198MHz clock (divided 396MHz PLL2 PFD)
20 periph2_clk2_sel	Selector for periph2_clk2 clock multiplexer 0 derive clock from pll3_sw_clk 1 derive clock from PLL2 Main
19–18 pre_periph_clk_sel	Selector for pre_periph clock multiplexer 00 derive clock from PLL2 main 528MHz clock 01 derive clock from 396MHz PLL2 PFD 10 derive clock from 352M PFD 11 derive clock from 198MHz clock (divided 396MHz PLL2 PFD)
17–16 gpu2d_core_clk_sel	Selector for open vg (GPU2D Core) clock multiplexer 00 derive clock from axi 01 derive clock from pll3_sw_clk 10 352M PFD 11 derive clock from 396M PFD
15–14 vpu_axi_clk_sel	Selector for VPU axi clock multiplexer 00 derive clock from AXI 01 derive clock from 396M PFD 10 derive clock from 352M PFD 11 Reserved
13–12 periph_clk2_sel	Selector for peripheral clk2 clock multiplexer 00 derive clock from pll3_sw_clk

Table continues on the next page...

CCM_CBCMR field descriptions (continued)

Field	Description
	01 derive clock from OSC_CLK (pll1_ref_clk) 10 derive clock from PLL2 (pll2_main_clk) 11 reserved
11 vdoaxi_clk_sel	Selector for vdoaxi clock multiplexer 0 derive clock from axi clk 1 derive clock from 132M clock
10 pcie_axi_clk_sel	Selector for pcie_axi clock multiplexer 0 derive clock from axi clk 1 derive clock from system_133M clk
9–8 gpu3d_shader_clk_sel	Selector for gpu3d_shader clock multiplexer 00 derive clock from mmdc_ch0 clk 01 derive clock from pll3_sw_clk 10 derive clock from PFD 594M 11 derive clock from 720M PFD
7–6 -	This field is reserved. Reserved
5–4 gpu3d_core_clk_sel	Selector for gpu3d_core clock multiplexer 00 derive clock from mmdc_ch0 01 derive clock from pll3_sw_clk 10 derive clock from 594M PFD 11 derive clock from 396M PFD
3–2 -	This field is reserved. Reserved
1 gpu3d_axi_clk_sel	Selector for gpu3d_axi clock multiplexer 0 derive clock from axi 1 derive clock from system_133M_clk
0 gpu2d_axi_clk_sel	Selector for gpu2d_axi clock multiplexer 0 derive clock from axi 1 derive clock from system_133M_clk

18.6.8 CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)

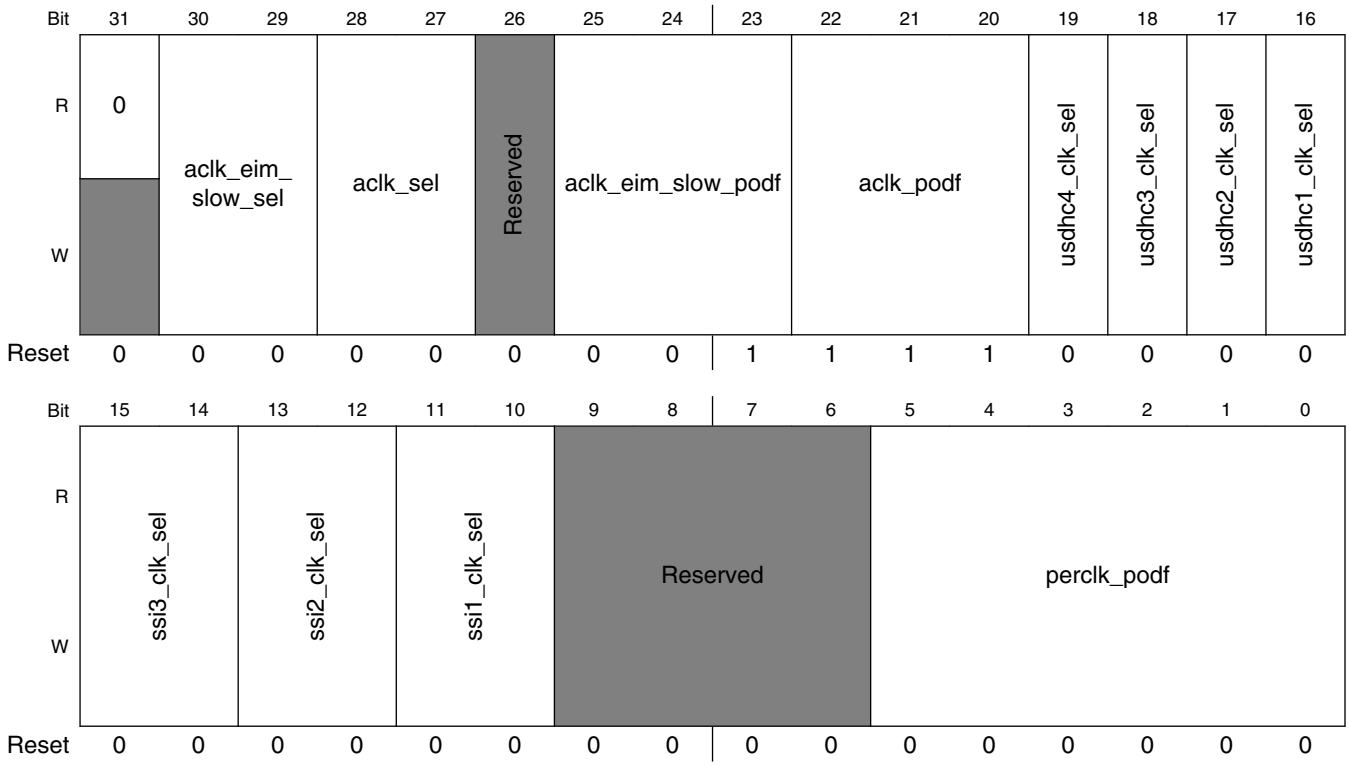
The figure below represents the CCM Serial Clock Multiplexer Register 1 (CSCMR1). The CSCMR1 register contains bits to control the multiplexers that generate the serial clocks. The table below provides its field descriptions.

NOTE

Any change on the above multiplexer will have to be done while the module that its clock is affected is not functional and

the clock is gated. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

Address: 20C_4000h base + 1Ch offset = 20C_401Ch



CCM_CSCMR1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–29 aclk_eim_slow_sel	Selector for aclk_eim_slow root clock multiplexer 00 derive clock from AXI clk root 01 derive clock from pll3_sw_clk 10 derive clock from 396M PFD 11 derive clock from 352M PFD
28–27 aclk_sel	Selector for aclk root clock multiplexer 00 derive clock from 396M PFD 01 derive clock from pll3_sw_clk 10 derive clock from AXI clk root 11 derive clock from 352M PFD
26 -	This field is reserved. Reserved
25–23 aclk_eim_slow_podf	Divider for aclk_eim_slow clock root. 000 divide by 1

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
22–20 aclk_podf	Divider for aclk clock root. **NOTE:** These bits are inverted between R/W and are not sequential. 000 divide by 7 (Read value 110) 001 divide by 8 (Read value 111) 010 divide by 5 (Read value 100) 011 divide by 6 (Read value 101) 100 divide by 3 (Read value 010) 101 divide by 4 (Read value 011) 110 divide by 1 (Read value 000) 111 divide by 2 (Read value 001)
19 usdhc4_clk_sel	Selector for usdhc4 clock multiplexer 0 derive clock from 396M PFD 1 derive clock from 352M PFD
18 usdhc3_clk_sel	Selector for usdhc3 clock multiplexer 0 derive clock from 396M PFD 1 derive clock from 352M PFD
17 usdhc2_clk_sel	Selector for usdhc2 clock multiplexer 0 derive clock from 396M PFD 1 derive clock from 352M PFD
16 usdhc1_clk_sel	Selector for usdhc1 clock multiplexer 0 derive clock from 396M PFD 1 derive clock from 352M PFD
15–14 ssi3_clk_sel	Selector for ssi3 clock multiplexer 00 derive clock from 508.2M PFD 01 derive clock from 454.7M PFD 10 derive clock from pll4 11 Reserved
13–12 ssi2_clk_sel	Selector for ssi2 clock multiplexer 00 derive clock from 508.2M PFD 01 derive clock from 454.7M PFD 10 derive clock from pll4 11 Reserved
11–10 ssi1_clk_sel	Selector for ssi1 clock multiplexer

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	00 derive clock from 508.2M PFD 01 derive clock from 454.7M PFD 10 derive clock from pll4 11 Reserved
9–6 -	This field is reserved. Reserved
perclk_podf	Divider for perclk podf. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

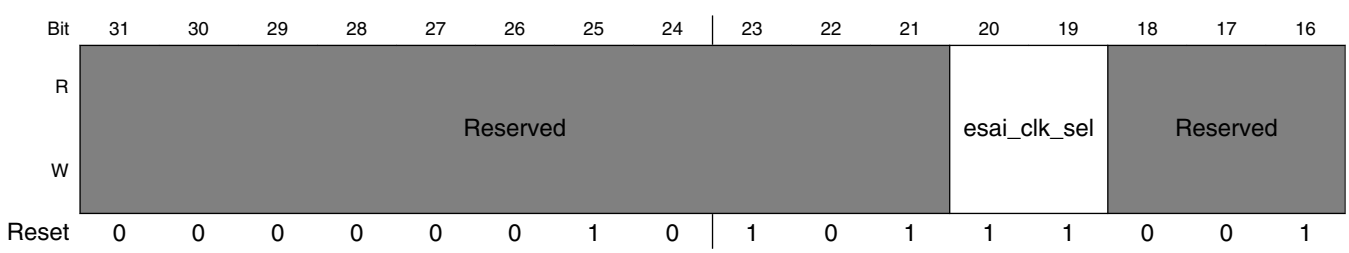
18.6.9 CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)

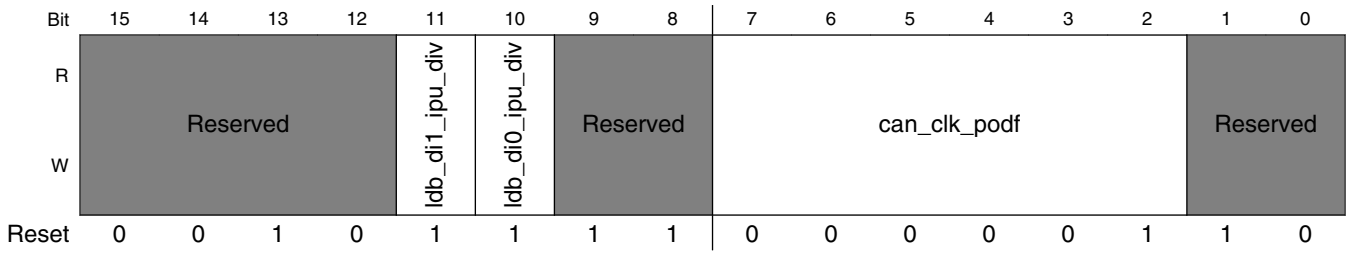
The figure below represents the CCM Serial Clock Multiplexer Register 2 (CSCMR2). The CSCMR2 register contains bits to control the multiplexers that generate the serial clocks. The table below provides its field descriptions.

NOTE

Any change on the above multiplexer will have to be done while the module that its clock is affected is not functional and the clock is gated. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

Address: 20C_4000h base + 20h offset = 20C_4020h





CCM_CSCMR2 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20–19 esai_clk_sel	Selector for esai clock multiplexer 00 derive clock from pll4 divided clock 01 derive clock from 508M PFD clock 10 derive clock from 454M PFD clock 11 derive clock from pll3_sw_clk
18–12 -	This field is reserved. Reserved
11 ldb_di1_ipu_div	Control for divider of ldb clock for IPU di1 0 divide by 3.5 1 divide by 7
10 ldb_di0_ipu_div	Control for divider of ldb clock for IPU di0 0 divide by 3.5 1 divide by 7
9–8 -	This field is reserved. Reserved
7–2 can_clk_podf	Divider for can clock podf. 000000 divide by 1 000111 divide by 8 111111 divide by 2^6
-	This field is reserved. Reserved

18.6.10 CCM Serial Clock Divider Register 1 (CCM_CSCDR1)

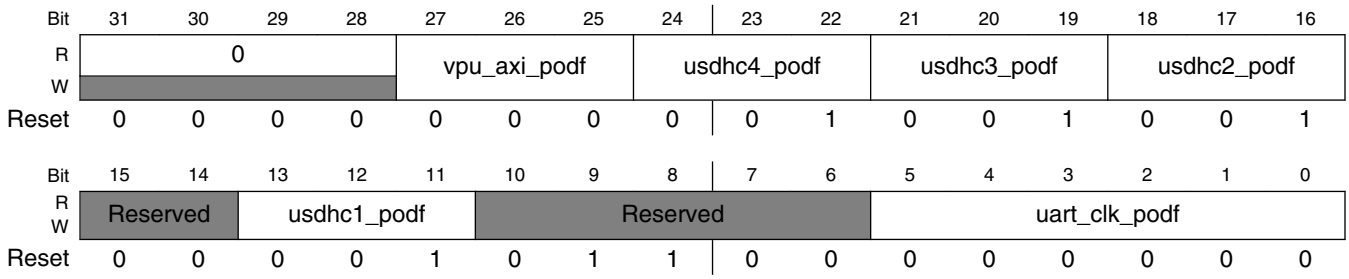
The figure below represents the CCM Serial Clock Divider Register 1 (CSCDR1). The CSCDR1 register contains bits to control the clock generation sub-module dividers. The table below provides its field descriptions.

NOTE

Any change on the above dividers will have to be done while the module that its clock is affected is not functional and the

affected clock is gated. If the change will be done during operation of the module, then it is not guaranteed that the modules operation will not be harmed.

Address: 20C_4000h base + 24h offset = 20C_4024h



CCM_CSCDR1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–25 vpu_axi_podf	Divider for vpu axi clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
24–22 usdhc4_podf	Divider for esdhc4 clock pred. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–19 usdhc3_podf	Divider for usdhc3 clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4

Table continues on the next page...

CCM_CSCDR1 field descriptions (continued)

Field	Description
	100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
18–16 usdhc2_podf	Divider for usdhc2 clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–14 -	This field is reserved. Reserved
13–11 usdhc1_podf	Divider for usdhc1 clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
10–6 -	This field is reserved. Reserved.
uart_clk_podf	Divider for uart clock podf. 000000 divide by 1 111111 divide by 2^6

18.6.11 CCM SSI1 Clock Divider Register (CCM_CS1CDR)

The figure below represents the CCM SSI1, SSI3, ESAI Clock Divider Register (CS1CDR). The CS1CDR register contains bits to control the ssi1 clock generation dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 28h offset = 20C_4028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				esai_clk_podf			ssi3_clk_pred			ssi3_clk_podf					
W																
Reset	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				esai_clk_pred			ssi1_clk_pred			ssi1_clk_podf					
W																
Reset	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1

CCM_CS1CDR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27–25 esai_clk_podf	Divider for esai clock podf. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
24–22 ssi3_clk_pred	Divider for ssi3 clock pred. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–16 ssi3_clk_podf	Divider for ssi3 clock podf. The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this.

Table continues on the next page...

CCM_CS1CDR field descriptions (continued)

Field	Description
	000000 divide by 1 111111 divide by 2^6
15–12 Reserved	This read-only field is reserved and always has the value 0.
11–9 esai_clk_pred	Divider for esai clock pred. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
8–6 ssi1_clk_pred	Divider for ssi1 clock pred. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
ssi1_clk_podf	Divider for ssi1 clock podf. The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6

18.6.12 CCM SSI2 Clock Divider Register (CCM_CS2CDR)

The figure below represents the CCM SSI2, LDB Clock Divider Register (CS2CDR). The CS2CDR register contains bits to control the ssi2 clock generation dividers, and ldb serial clocks select. The table below provides its field descriptions.

Address: 20C_4000h base + 2Ch offset = 20C_402Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					enfc_clk_podf						enfc_clk_pred			enfc_clk_sel	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	ldb_di1_clk_sel			ldb_di0_clk_sel			ssi2_clk_pred			ssi2_clk_podf					
W																
Reset	0	0	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1

CCM_CS2CDR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26–21 enfc_clk_podf	Divider for enfc clock divider. 000000 divide by 1 000001 divide by 2 111111 divide by 2^6
20–18 enfc_clk_pred	Divider for enfc clock pred divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
17–16 enfc_clk_sel	Selector for enfc clock multiplexer **NOTE:** Multiplexor should be updated when output clock is gated. 00 pll2 352M PFD 01 pll2 clock

Table continues on the next page...

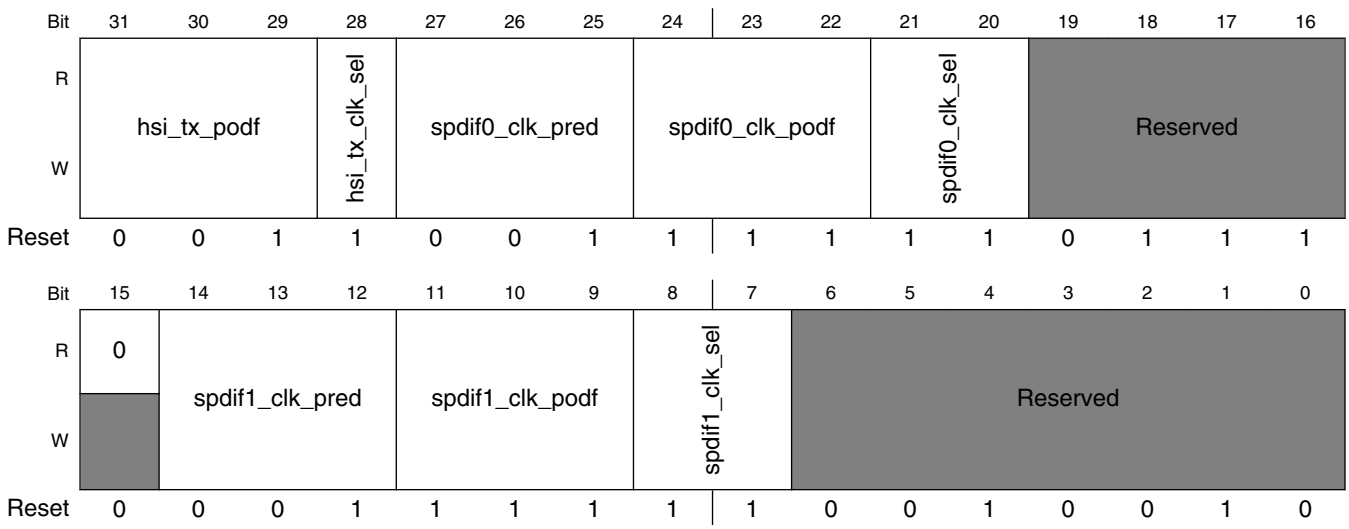
CCM_CS2CDR field descriptions (continued)

Field	Description
	10 pll3_sw_clk 11 pll2 396M PFD
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 ldb_di1_clk_sel	Selector for ldb_di1 clock multiplexer **NOTE:** Multiplexor should be updated when both input and output clocks are gated. 000 pll5 clock 001 pll2 352M PFD 010 pll2 396M PFD 011 MMDC_CH1 clock 100 pll3_sw_clk 101-111 Reserved
11–9 ldb_di0_clk_sel	Selector for ldb_di1 clock multiplexer **NOTE:** Multiplexor should be updated when both input and output clocks are gated. 000 pll5 clock 001 pll2 352M PFD 010 pll2 396M PFD 011 MMDC_CH1 clock 100 pll3_sw_clk 101-111 Reserved
8–6 ssi2_clk_pred	Divider for ssi2 clock pred. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
ssi2_clk_podf	Divider for ssi2 clock podf. The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this. **NOTE:** Divider should be updated when output clock is gated. 000000 divide by 1 111111 divide by 2^6

18.6.13 CCM D1 Clock Divider Register (CCM_CDCDR)

The figure below represents the CCM DI Clock Divider Register (CDCDR). The table below provides its field descriptions.

Address: 20C_4000h base + 30h offset = 20C_4030h



CCM_CDCDR field descriptions

Field	Description
31–29 hsi_tx_podf	Divider for hsi_tx clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
28 hsi_tx_clk_sel	Selector for hsi_tx clock multiplexer 0 derive from pll3 120M clock 1 derive from pll2 396M PFD
27–25 spdif0_clk_pred	Divider for spdif0 clock pred. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2

Table continues on the next page...

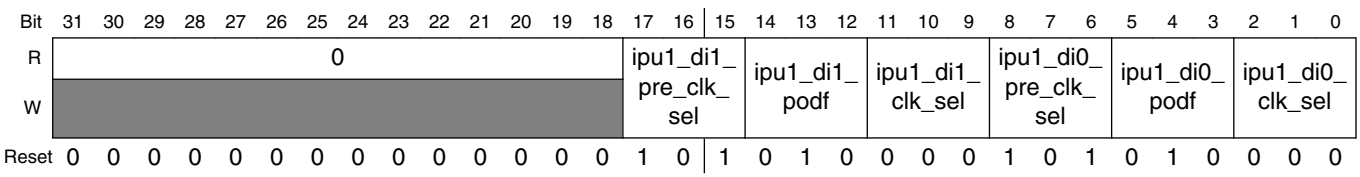
CCM_CDCDR field descriptions (continued)

Field	Description
	010 divide by 3 111 divide by 8
24–22 spdif0_clk_podf	Divider for spdif0 clock podf. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 111 divide by 8
21–20 spdif0_clk_sel	Selector for spdif0 clock multiplexer 00 derive clock from pll4 divided clock 01 derive clock from 508M PFD clock 10 derive clock from 454M PFD clock 11 derive clock from pll3_sw_clk
19–16 -	This field is reserved. Reserved
15 Reserved	This read-only field is reserved and always has the value 0.
14–12 spdif1_clk_pred	Divider for spdif1 clock pred. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** Used for ASRC clock, not related to SPDIF module 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3 111 divide by 8
11–9 spdif1_clk_podf	Divider for spdif1 clock podf. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** Used for ASRC clock, not related to SPDIF module 000 divide by 1 111 divide by 8
8–7 spdif1_clk_sel	Selector for spdif1 clock multiplexer **NOTE:** Used for ASRC clock, not related to SPDIF module 00 derive clock from pll4 divided clock 01 derive clock from 508M PFD clock 10 derive clock from 454M PFD clock 11 derive clock from pll3_sw_clk
-	This field is reserved. Reserved

18.6.14 CCM HSC Clock Divider Register (CCM_CHSCCDR)

The figure below represents the CCM HSC Clock Divider Register (CHSCCDR). The CHSCCDR register contains bits to control the ipu di clock generation dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 34h offset = 20C_4034h



CCM_CHSCCDR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17–15 ipu1_di1_pre_ clk_sel	Selector for ipu1 di1 root clock pre-multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from mmdc_ch0 clock 001 derive clock from pll3_sw_clk 010 derive clock from pll5 011 derive clock from 352M PFD 100 derive clock from 396M PFD 101 derive clock from 540M PFD 110-111 Reserved
14–12 ipu1_di1_podf	Divider for ipu1_di clock divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
11–9 ipu1_di1_clk_sel	Selector for ipu1 di1 root clock multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits.

Table continues on the next page...

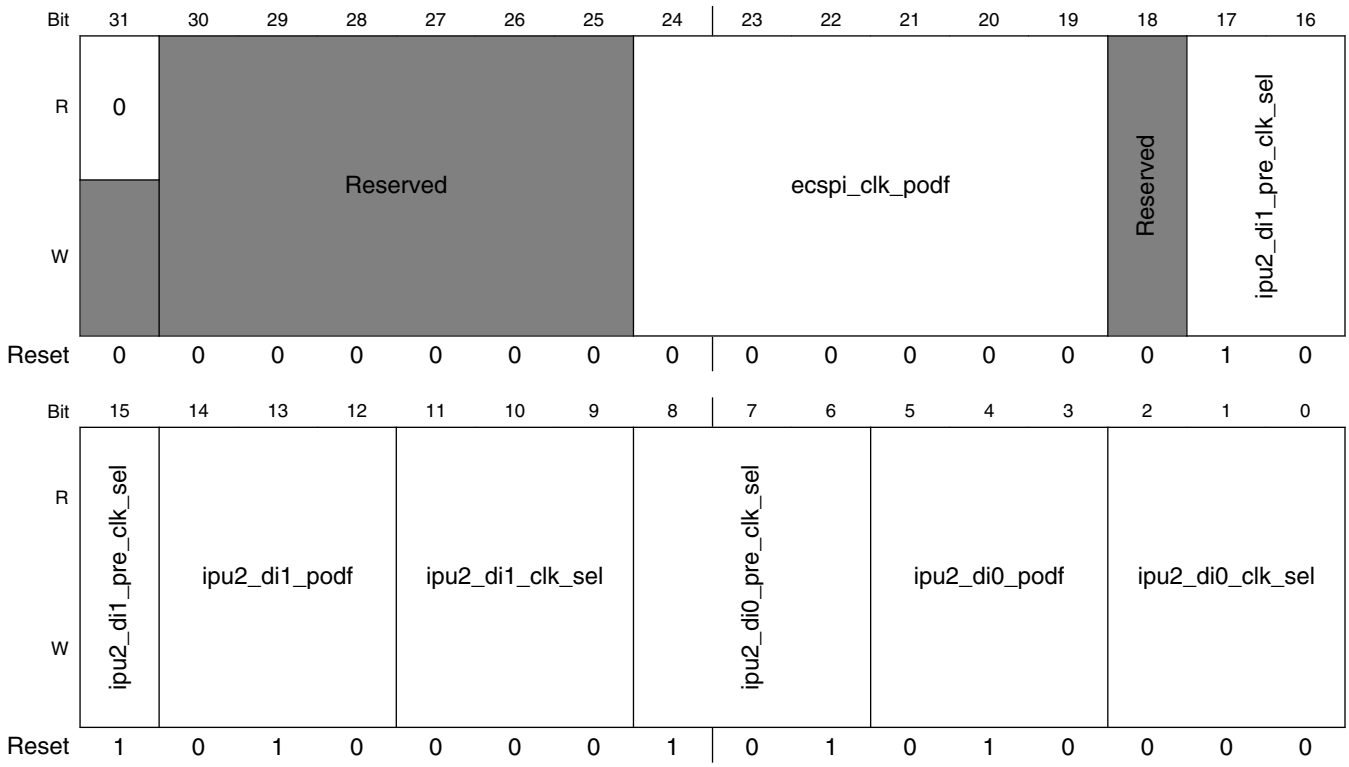
CCM_CHSCCDR field descriptions (continued)

Field	Description
	000 derive clock from divided pre-muxed ipu1 di1 clock 001 derive clock from ipp_di0_clk 010 derive clock from ipp_di1_clk 011 derive clock from ldb_di0_clk 100 derive clock from ldb_di1_clk 101-111 Reserved
8–6 ipu1_di0_pre_clk_sel	Selector for ipu1 di0 root clock pre-multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from mmdc_ch0 clock 001 derive clock from pll3_sw_clk 010 derive clock from pll5 011 derive clock from 352M PFD 100 derive clock from 396M PFD 101 derive clock from 540M PFD 110-111 Reserved
5–3 ipu1_di0_podf	Divider for ipu1_di0 clock divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
ipu1_di0_clk_sel	Selector for ipu1 di0 root clock multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from divided pre-muxed ipu1 di0 clock 001 derive clock from ipp_di0_clk 010 derive clock from ipp_di1_clk 011 derive clock from ldb_di0_clk 100 derive clock from ldb_di1_clk 101-111 Reserved

18.6.15 CCM Serial Clock Divider Register 2 (CCM_CSCDR2)

The figure below represents the CCM Serial Clock Divider Register 2(CSCDR2). The CSCDR2 register contains bits to control the clock generation sub-module dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 38h offset = 20C_4038h



CCM_CSCDR2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30–25 -	This field is reserved. Reserved
24–19 ecspi_clk_podf	Divider for ecspi clock podf. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** The input clock to this divider should be lower than 300Mhz, the predivider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6

Table continues on the next page...

CCM_CSCDR2 field descriptions (continued)

Field	Description
18 - Reserved	This field is reserved.
17–15 ipu2_di1_pre_clk_sel	Selector for ipu2 di1 root clock pre-multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from mmdc_ch0 clock 001 derive clock from pll3_sw_clk 010 derive clock from pll5 011 derive clock from 352M PFD 100 derive clock from 396M PFD 101 derive clock from 540M PFD 110-111 Reserved
14–12 ipu2_di1_podf	Divider for ipu2_di1 clock divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
11–9 ipu2_di1_clk_sel	Selector for ipu1 di2 root clock multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from divided pre-muxed ipu1 di1 clock 001 derive clock from ipp_di0_clk 010 derive clock from ipp_di1_clk 011 derive clock from ldb_di0_clk 100 derive clock from ldb_di1_clk 101-111 Reserved
8–6 ipu2_di0_pre_clk_sel	Selector for ipu2 di0 root clock pre-multiplexer **NOTE:** Multiplexor should only be updated when the output clock is gated. Please refer to System Clocks for the respective output clock gating bits. 000 derive clock from mmdc_ch0 clock 001 derive clock from pll3_sw_clk 010 derive clock from pll5 011 derive clock from 352M PFD 100 derive clock from 396M PFD

Table continues on the next page...

CCM_CSCDR2 field descriptions (continued)

Field	Description
	101 derive clock from 540M PFD 110-111 Reserved
5–3 ipu2_di0_podf	Divider for ipu2_di0 clock divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
ipu2_di0_clk_sel	Selector for ipu2 di0 root clock multiplexer **NOTE:** Multiplexor should be updated only when the output clock is gated. 000 derive clock from divided pre-muxed ipu1 di0 clock 001 derive clock from ipp_di0_clk 010 derive clock from ipp_di1_clk 011 derive clock from ldb_di0_clk 100 derive clock from ldb_di1_clk 101-111 Reserved

18.6.16 CCM Serial Clock Divider Register 3 (CCM_CSCDR3)

The figure below represents the CCM Serial Clock Divider Register 3(CSCDR3). The CSCDR3 register contains bits to control the clock generation sub-module dividers. The table below provides its field descriptions.

Address: 20C_4000h base + 3Ch offset = 20C_403Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													ipu2_hsp_podf		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ipu2_hsp_ clk_sel		ipu1_hsp_podf			ipu1_hsp_ clk_sel		Reserved								
W																
Reset	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1

CCM_CSCDR3 field descriptions

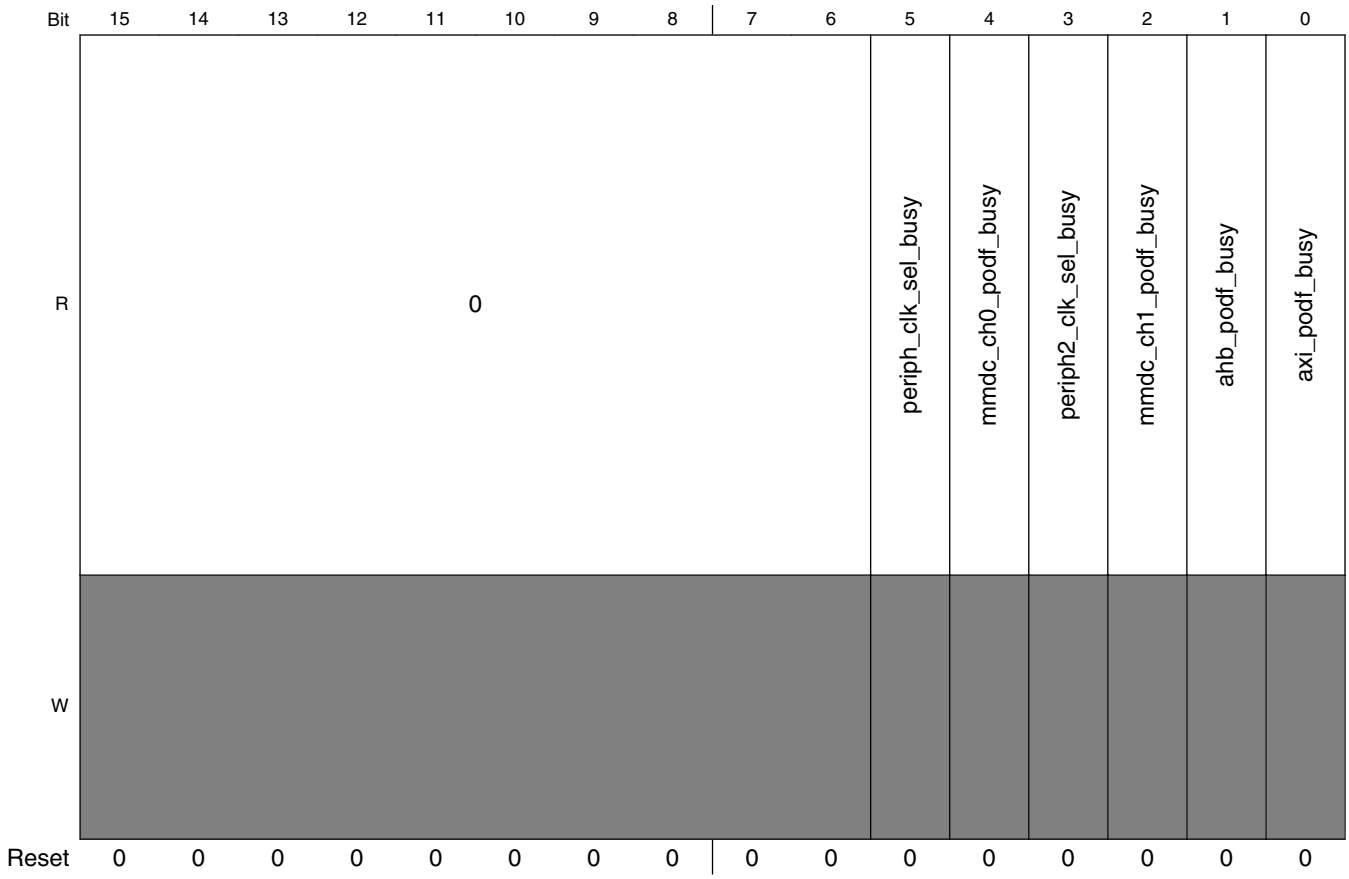
Field	Description
31–19 Reserved	This read-only field is reserved and always has the value 0.
18–16 ipu2_hsp_podf	Divider for ipu2_hsp clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–14 ipu2_hsp_clk_sel	Selector for ipu2_hsp clock multiplexer 00 derive clock from mmdc_ch0 clock 01 derive clock from 396M PFD 10 derive clock from 120M 11 derive clock from 540M PFD
13–11 ipu1_hsp_podf	Divider for ipu1_hsp clock. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
10–9 ipu1_hsp_clk_sel	Selector for ipu1_hsp clock multiplexer 00 derive clock from mmdc_ch0 clock 01 derive clock from 396M PFD 10 derive clock from 120M 11 derive clock from 540M PFD
-	This field is reserved. Reserved

18.6.17 CCM Divider Handshake In-Process Register (CCM_CDHIPR)

The figure below represents the CCM Divider Handshake In-Process Register (CDHIPR). The CDHIPR register contains read-only bits that indicate that CCM is in the process of updating dividers or muxes that might need handshake with modules.

Address: 20C_4000h base + 48h offset = 20C_4048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															arm_podf_busy
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



CCM_CDHIPR field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value 0.
16 arm_podf_busy	Busy indicator for arm_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the arm_podf will be applied.
15–6 Reserved	This read-only field is reserved and always has the value 0.
5 periph_clk_sel_busy	Busy indicator for periph_clk_sel mux control. 0 mux is not busy and its value represents the actual division. 1 mux is busy with handshake process with module. The value read in the periph_clk_sel represents the previous value of select, and after the handshake periph_clk_sel value will be applied.
4 mmdc_ch0_podf_busy	Busy indicator for mmdc_ch0_axi_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the mmdc_ch0_axi_podf will be applied.

Table continues on the next page...

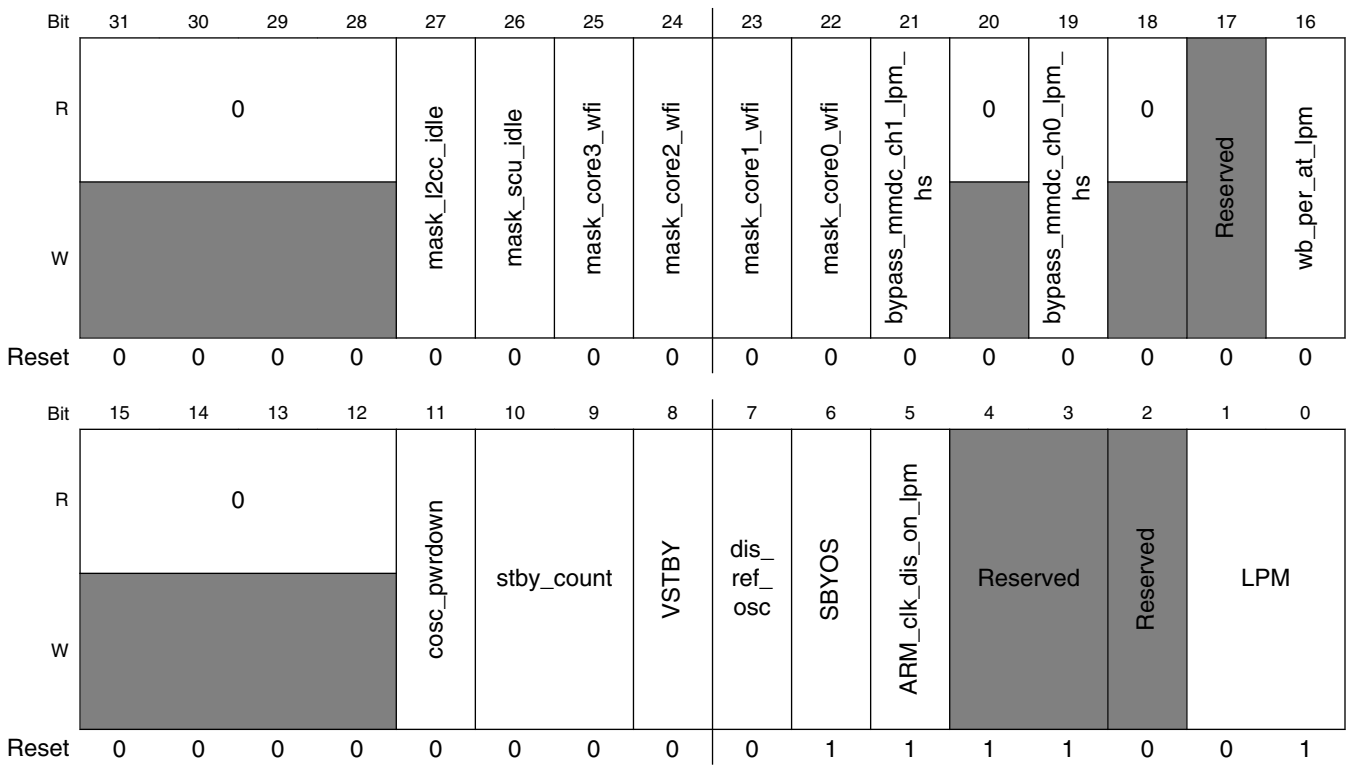
CCM_CDHIPR field descriptions (continued)

Field	Description
3 periph2_clk_sel_busy	Busy indicator for periph2_clk_sel mux control. 0 mux is not busy and its value represents the actual division. 1 mux is busy with handshake process with module. The value read in the periph2_clk_sel represents the previous value of select, and after the handshake periph2_clk_sel value will be applied.
2 mmdc_ch1_podf_busy	Busy indicator for mmdc_ch1_axi_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the mmdc_ch1_axi_podf will be applied.
1 ahb_podf_busy	Busy indicator for ahb_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the ahb_podf will be applied.
0 axi_podf_busy	Busy indicator for axi_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with module. The value read in the divider represents the previous value of the division factor, and after the handshake the written value of the axi_podf will be applied.

18.6.18 CCM Low Power Control Register (CCM_CLPCR)

The figure below represents the CCM Low Power Control Register (CLPCR). The CLPCR register contains bits to control the low power modes operation.The table below provides its field descriptions.

Address: 20C_4000h base + 54h offset = 20C_4054h



CCM_CLPCR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value 0.
27 mask_l2cc_idle	Mask L2CC IDLE for entering low power mode. **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 L2CC IDLE is masked 0 L2CC IDLE is not masked
26 mask_scu_idle	Mask SCU IDLE for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 SCU IDLE is masked 0 SCU IDLE is not masked

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
25 mask_core3_wfi	Mask WFI of core3 for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 WFI of core3 is masked 0 WFI of core3 is not masked
24 mask_core2_wfi	Mask WFI of core2 for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 WFI of core2 is masked 0 WFI of core2 is not masked
23 mask_core1_wfi	Mask WFI of core1 for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 1 WFI of core1 is masked 0 WFI of core1 is not masked
22 mask_core0_wfi	Mask WFI of core0 for entering low power mode **NOTE:** Assertion of all bits[27:22] will generate low power mode request 0 WFI of core0 is not masked 1 WFI of core0 is masked
21 bypass_mmdc_ch1_lpm_hs	Bypass handshake with mmdc_ch1 on next entrance to low power mode (STOP or WAIT). CCM doesn't wait for the module's acknowledge. 0 Handshake with mmdc_ch1 on next entrance to low power mode will be performed. 1 Handshake with mmdc_ch1 on next entrance to low power mode will be bypassed.
20 Reserved	This read-only field is reserved and always has the value 0.
19 bypass_mmdc_ch0_lpm_hs	Bypass handshake with mmdc_ch0 on next entrance to low power mode (STOP or WAIT). CCM doesn't wait for the module's acknowledge. Handshake will also be bypassed, if CGR3 CG10 is set to gate fast mmdc_ch0 clock. 0 Handshake with mmdc_ch0 on next entrance to low power mode will be performed. . 1 Handshake with mmdc_ch0 on next entrance to low power mode will be bypassed.
18 Reserved	This read-only field is reserved and always has the value 0.
17 -	This field is reserved. Reserved
16 wb_per_at_lpm	Enable periphery charge pump for well biasing at low power mode (stop or wait) 0 Periphery charge pump won't be enabled at STOP or WAIT low power modes 1 Periphery charge pump will be enabled at STOP or WAIT low power modes
15–12 Reserved	This read-only field is reserved and always has the value 0.
11 cosc_pwrdown	In run mode, software can manually control powering down of on chip oscillator, i.e. generating '1' on cosc_pwrdown signal. If software manually powered down the on chip oscillator, then sbyos functionality for on chip oscillator will be bypassed.

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
	The manual closing of onchip oscillator should be performed only in case the reference oscilator is not the source of all the clocks generation. 0 On chip oscillator will not be powered down, i.e. cosc_pwrdown = '0'. 1 On chip oscillator will be powered down, i.e. cosc_pwrdown = '1'.
10–9 stby_count	Standby counter definition. These two bits define, in the case of stop exit (if VSTBY bit was set). **NOTE:** Clock cycles ratio depends on pmic_delay_scaler, defined by CGPR[0] bit. 00 CCM will wait (1*pmic_delay_scaler)+1 ckil clock cycles 01 CCM will wait (3*pmic_delay_scaler)+1 ckil clock cycles 10 CCM will wait (7*pmic_delay_scaler)+1 ckil clock cycles 11 CCM will wait (15*pmic_delay_scaler)+1 ckil clock cycles
8 VSTBY	Voltage standby request bit. This bit defines if PMIC_STBY_REQ pin, which notifies external power management IC to move from functional voltage to standby voltage, will be asserted in STOP mode. 0 Voltage will not be changed to standby voltage after next entrance to STOP mode. (PMIC_STBY_REQ will remain negated - '0') 1 Voltage will be requested to change to standby voltage after next entrance to stop mode. (PMIC_STBY_REQ will be asserted - '1').
7 dis_ref_osc	dis_ref_osc - in run mode, software can manually control closing of external reference oscillator clock, i.e. generating '1' on CCM_REF_EN_B signal. If software closed manually the external reference clock, then sbyos functionality will be bypassed. The manual closing of external reference oscilator should be performed only in case the reference oscilator is not the source of any clock generation. **NOTE:** When returning from stop mode, the PMIC_STBY_REQ will be deasserted (if it was asserted when entering stop mode). See stby_count bits. 0 external high frequency oscillator will be enabled, i.e. CCM_REF_EN_B = '0'. 1 external high frequency oscillator will be disabled, i.e. CCM_REF_EN_B = '1'
6 SBYOS	Standby clock oscillator bit. This bit defines if cosc_pwrdown, which power down the on chip oscillator, will be asserted in STOP mode. This bit is discarded if cosc_pwrdown='1' for the on chip oscillator. 0 On-chip oscillator will not be powered down, after next entrance to STOP mode. (CCM_REF_EN_B will remain asserted - '0' and cosc_pwrdown will remain de asserted - '0') 1 On-chip oscillator will be powered down, after next entrance to STOP mode. (CCM_REF_EN_B will be deasserted - '1' and cosc_pwrdown will be asserted - '1'). When returning from STOP mode, external oscillator will be enabled again, on-chip oscillator will return to oscillator mode, and after oscnt count, CCM will continue with the exit from the STOP mode process.
5 ARM_clk_dis_on_lpm	Define if ARM clocks (arm_clk, soc_mxclk, soc_pclk, soc_dbg_pclk, vl_wrck) will be disabled on wait mode. This is useful for debug mode, when the user still wants to simulate entering wait mode and still keep ARM clock functioning. **NOTE:** Software should not enable ARM power gating in wait mode if this bit is cleared. 0 ARM clock enabled on wait mode. 1 ARM clock disabled on wait mode. .
4–3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
LPM	Setting the low power mode that system will enter on next assertion of dsm_request signal. **NOTE:** Set CCM_CGPR[INT_MEM_CLK_LPM] and CCM_CGPR[1] bits to 1 when setting CCM_CLPCR[LPM] bits to 01 (WAIT Mode) or 10 (STOP mode) without power gating. CCM_CGPR[INT_MEM_CLK_LPM] and CCM_CGPR[1] bits do not have to be set for STOP mode entry. 00 Remain in run mode 01 Transfer to wait mode 10 Transfer to stop mode 11 Reserved

18.6.19 CCM Interrupt Status Register (CCM_CISR)

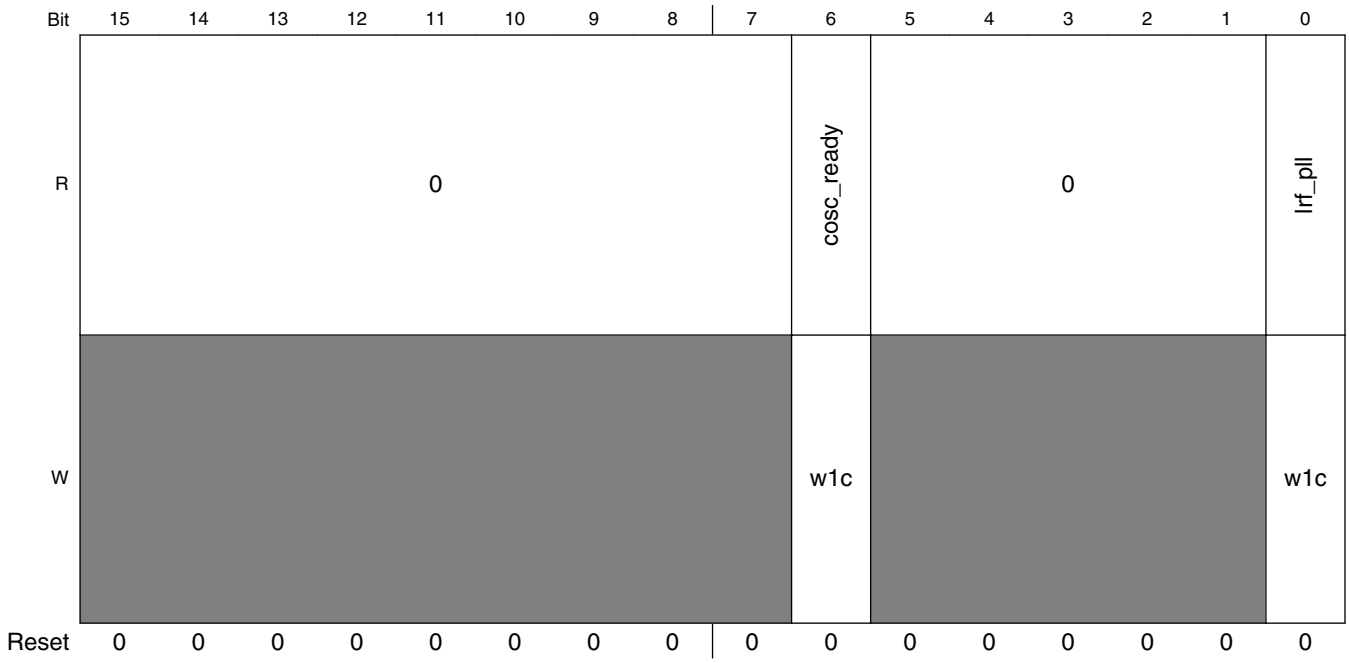
The figure below represents the CCM Interrupt Status Register (CISR). This is a write one to clear register. Once a interrupt is generated, software should write one to clear it. The table below provides its field descriptions.

NOTE

CCM interrupt request 1 can be masked by CCM interrupt request 1 mask bit. CCM interrupt request 2 can be masked by CCM interrupt request 2 mask bit.

Address: 20C_4000h base + 58h offset = 20C_4058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					arm_podf_loaded	0		mmdc_ch0_podf_loaded	periph_clk_sel_loaded	mmdc_ch1_podf_loaded	ahb_podf_loaded	periph2_clk_sel_loaded	0	axi_podf_loaded	0
W						w1c			w1c	w1c	w1c	w1c	w1c		w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



CCM_CISR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26 arm_podf_loaded	CCM interrupt request 1 generated due to frequency change of arm_podf. The interrupt will commence only if arm_podf is loaded during a arm dvfs operation. 0 interrupt is not generated due to frequency change of arm_podf 1 interrupt generated due to frequency change of arm_podf
25–24 Reserved	This read-only field is reserved and always has the value 0.
23 mmdc_ch0_ podf_loaded	CCM interrupt request 1 generated due to update of mmdc_ch0_axi_podf. 0 interrupt is not generated due to update of mmdc_ch0_axi_podf. 1 interrupt generated due to update of mmdc_ch0_axi_podf*
22 periph_clk_sel_ loaded	CCM interrupt request 1 generated due to update of periph_clk_sel. 0 interrupt is not generated due to update of periph_clk_sel. 1 interrupt generated due to update of periph_clk_sel.
21 mmdc_ch1_ podf_loaded	CCM interrupt request 1 generated due to frequency change of mmdc_ch0_podf_ **loaded** 0 interrupt is not generated due to frequency change of mmdc_ch0_podf_ **loaded** 1 interrupt generated due to frequency change of mmdc_ch0_podf_ **loaded**
20 ahb_podf_loaded	CCM interrupt request 1 generated due to frequency change of ahb_podf 0 interrupt is not generated due to frequency change of ahb_podf 1 interrupt generated due to frequency change of ahb_podf
19 periph2_clk_sel_ loaded	CCM interrupt request 1 generated due to frequency change of periph2_clk_sel

Table continues on the next page...

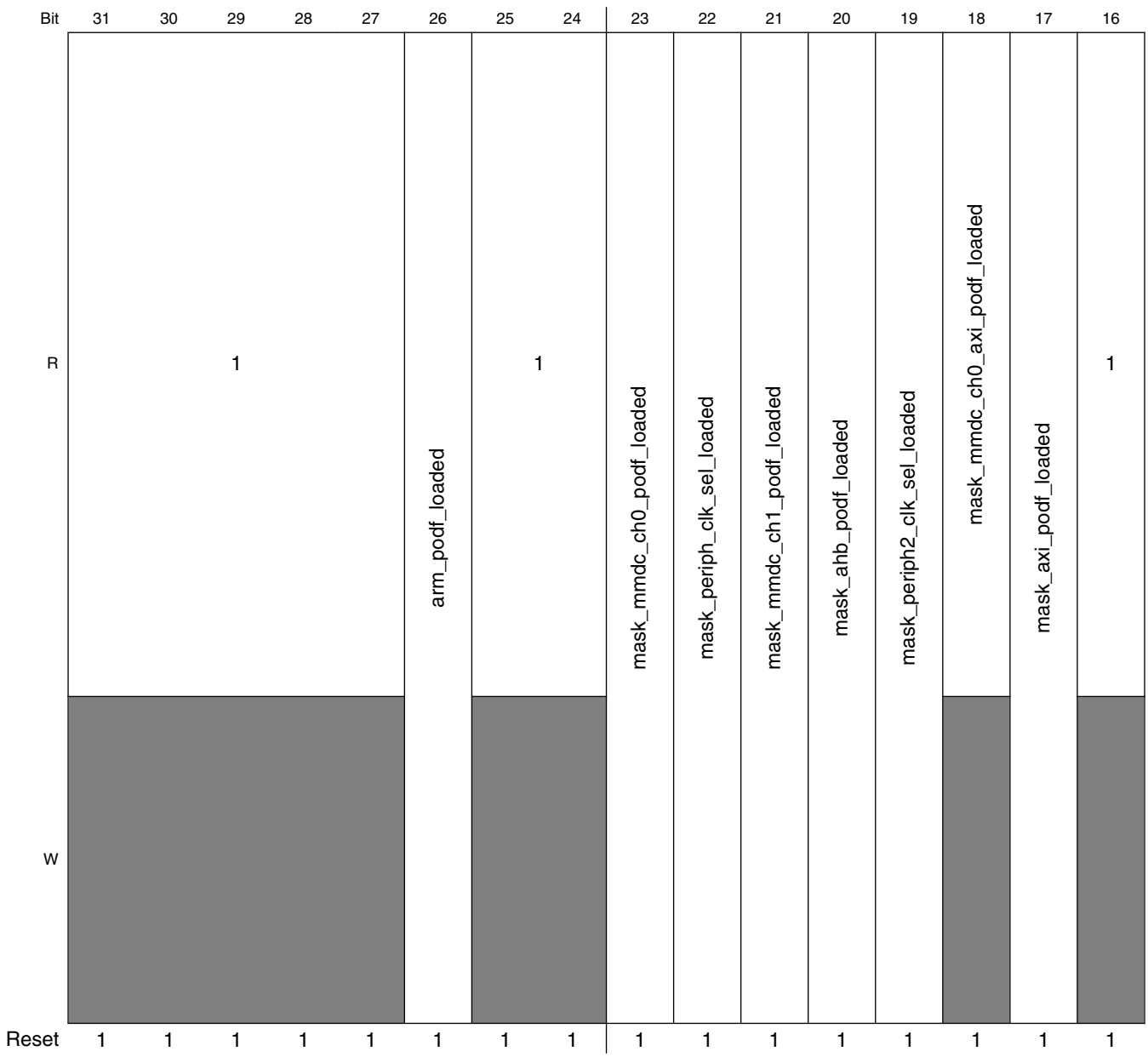
CCM_CISR field descriptions (continued)

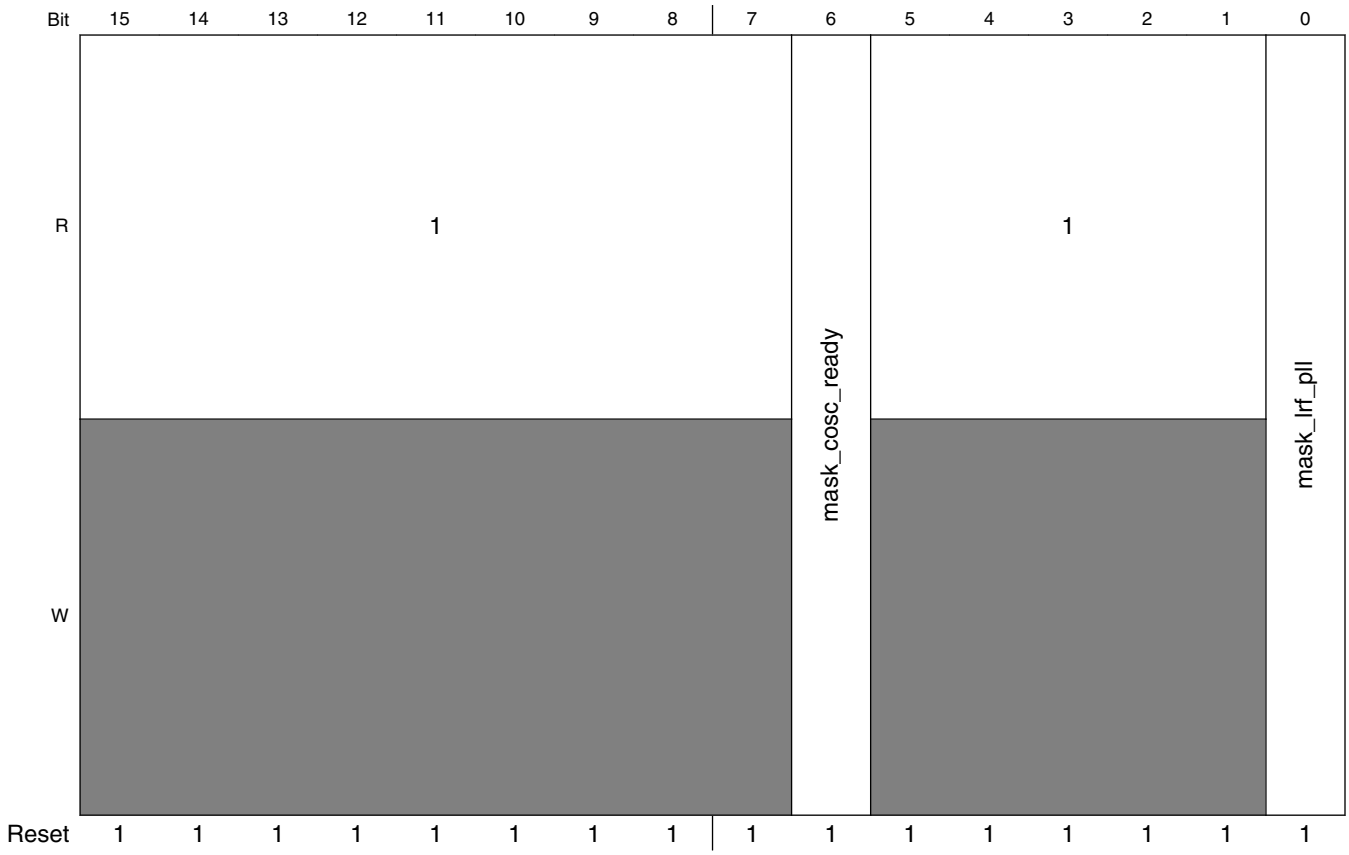
Field	Description
	0 interrupt is not generated due to frequency change of periph2_clk_sel 1 interrupt generated due to frequency change of periph2_clk_sel
18 Reserved	This read-only field is reserved and always has the value 0. 0 interrupt is not generated due to frequency change of mmdc_ch0_axi_podf 1 interrupt generated due to frequency change of mmdc_ch0_axi_podf
17 axi_podf_loaded	CCM interrupt request 1 generated due to frequency change of axi_podf 0 interrupt is not generated due to frequency change of axi_podf 1 interrupt generated due to frequency change of axi_podf
16–7 Reserved	This read-only field is reserved and always has the value 0.
6 cosc_ready	CCM interrupt request 2 generated due to on board oscillator ready, i.e. oscnt has finished counting. 0 interrupt is not generated due to on board oscillator ready 1 interrupt generated due to on board oscillator ready
5–1 Reserved	This read-only field is reserved and always has the value 0.
0 lrf_pll	CCM interrupt request 2 generated due to lock of all enabled and not bypaseed PLLs 0 interrupt is not generated due to lock ready of all enabled and not bypaseed PLLs 1 interrupt generated due to lock ready of all enabled and not bypaseed PLLs

18.6.20 CCM Interrupt Mask Register (CCM_CIMR)

The figure below represents the CCM Interrupt Mask Register (CIMR). The table below provides its field descriptions.

Address: 20C_4000h base + 5Ch offset = 20C_405Ch





CCM_CIMR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 1.
26 arm_podf_loaded	mask interrupt generation due to frequency change of arm_podf 0 don't mask interrupt due to frequency change of arm_podf - interrupt will be created 1 mask interrupt due to frequency change of arm_podf
25–24 Reserved	This read-only field is reserved and always has the value 1.
23 mask_mmdc_ ch0_podf_loaded	mask interrupt generation due to update of mask_mmdc_ch0_podf 0 don't mask interrupt due to update of mask_mmdc_ch0_podf - interrupt will be created 1 mask interrupt due to update of mask_mmdc_ch0_podf
22 mask_periph_ clk_sel_loaded	mask interrupt generation due to update of periph_clk_sel. 0 don't mask interrupt due to update of periph_clk_sel - interrupt will be created 1 mask interrupt due to update of periph_clk_sel
21 mask_mmdc_ ch1_podf_loaded	mask interrupt generation due to update of mask_mmdc_ch1_podf 0 don't mask interrupt due to update of mask_mmdc_ch1_podf - interrupt will be created 1 mask interrupt due to update of mask_mmdc_ch1_podf

Table continues on the next page...

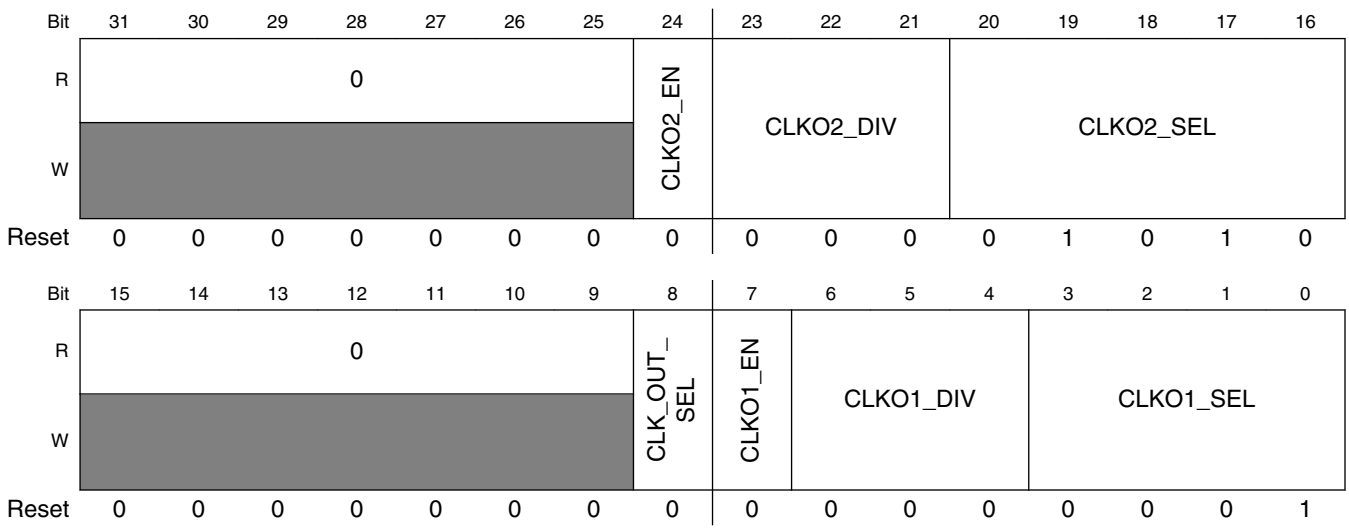
CCM_CIMR field descriptions (continued)

Field	Description
20 mask_ahb_podf_loaded	mask interrupt generation due to frequency change of ahb_podf 0 don't mask interrupt due to frequency change of ahb_podf - interrupt will be created 1 mask interrupt due to frequency change of ahb_podf
19 mask_periph2_clk_sel_loaded	mask interrupt generation due to update of periph2_clk_sel. 0 don't mask interrupt due to update of periph2_clk_sel - interrupt will be created 1 mask interrupt due to update of periph2_clk_sel
18 mask_mmdc_ch0_axi_podf_loaded	mask interrupt generation due to frequency change of mmdc_ch0_axi_podf 0 don't mask interrupt due to frequency change of mmdc_ch0_axi_podf - interrupt will be created 1 mask interrupt due to frequency change of mmdc_ch0_axi_podf
17 mask_axi_podf_loaded	mask interrupt generation due to frequency change of axi_podf 0 don't mask interrupt due to frequency change of axi_podf - interrupt will be created 1 mask interrupt due to frequency change of axi_podf
16–7 Reserved	This read-only field is reserved and always has the value 1.
6 mask_cosc_ready	mask interrupt generation due to on board oscillator ready 0 don't mask interrupt due to on board oscillator ready - interrupt will be created 1 mask interrupt due to on board oscillator ready
5–1 Reserved	This read-only field is reserved and always has the value 1.
0 mask_lrf_pll	mask interrupt generation due to lrf of PLLs 0 don't mask interrupt due to lrf of PLLs - interrupt will be created 1 mask interrupt due to lrf of PLLs

18.6.21 CCM Clock Output Source Register (CCM_CCOSR)

The figure below represents the CCM Clock Output Source Register (CCOSR). The CCOSR register contains bits to control the clocks that will be generated on the output ipp_do_clko1 (CCM_CLKO1) and ipp_do_clko2 (CCM_CLKO2).The table below provides its field descriptions.

Address: 20C_4000h base + 60h offset = 20C_4060h



CCM_CCOSR field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 CLKO2_EN	Enable of CCM_CLKO2 clock 0 CCM_CLKO2 disabled. 1 CCM_CLKO2 enabled.
23–21 CLKO2_DIV	Setting the divider of CCM_CLKO2 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
20–16 CLKO2_SEL	Selection of the clock to be generated on CCM_CLKO2

Table continues on the next page...

CCM_CCOSR field descriptions (continued)

Field	Description
	00000 mmdc_ch0_clk_root 00001 mmdc_ch1_clk_root 00010 usdhc4_clk_root 00011 usdhc1_clk_root 00100 gpu2d_axi_clk_root 00101 wrck_clk_root 00110 ecspi_clk_root 00111 gpu3d_axi_clk_root 01000 usdhc3_clk_root 01001 125M_clk_root 01010 arm_clk_root 01011 ipu1_hsp_clk_root 01100 ipu2_hsp_clk_root 01101 vdo_axi_clk_root 01110 osc_clk 01111 gpu2d_core_clk_root 10000 gpu3d_core_clk_root 10001 usdhc2_clk_root 10010 ssi1_clk_root 10011 ssi2_clk_root 10100 ssi3_clk_root 10101 gpu3d_shader_clk_root 10110 vpu_axi_clk_root 10111 can_clk_root 11000 ldb_di0_serial_clk_root 11001 ldb_di1_serial_clk_root 11010 esai_clk_root 11011 aclk_eim_slow_clk_root 11100 uart_clk_root 11101 spdif0_clk_root 11110 spdif1_clk_root 11111 hsi_tx_clk_root
15–9 Reserved	This read-only field is reserved and always has the value 0.
8 CLK_OUT_SEL	CCM_CLKO1 output to reflect CCM_CLKO1 or CCM_CLKO2 clocks 0 CCM_CLKO1 output drives CCM_CLKO1 clock 1 CCM_CLKO1 output drives CCM_CLKO2 clock
7 CLKO1_EN	Enable of CCM_CLKO1 clock 0 CCM_CLKO1 disabled. 1 CCM_CLKO1 enabled.
6–4 CLKO1_DIV	Setting the divider of CCM_CLKO1 000 divide by 1 001 divide by 2 010 divide by 3

Table continues on the next page...

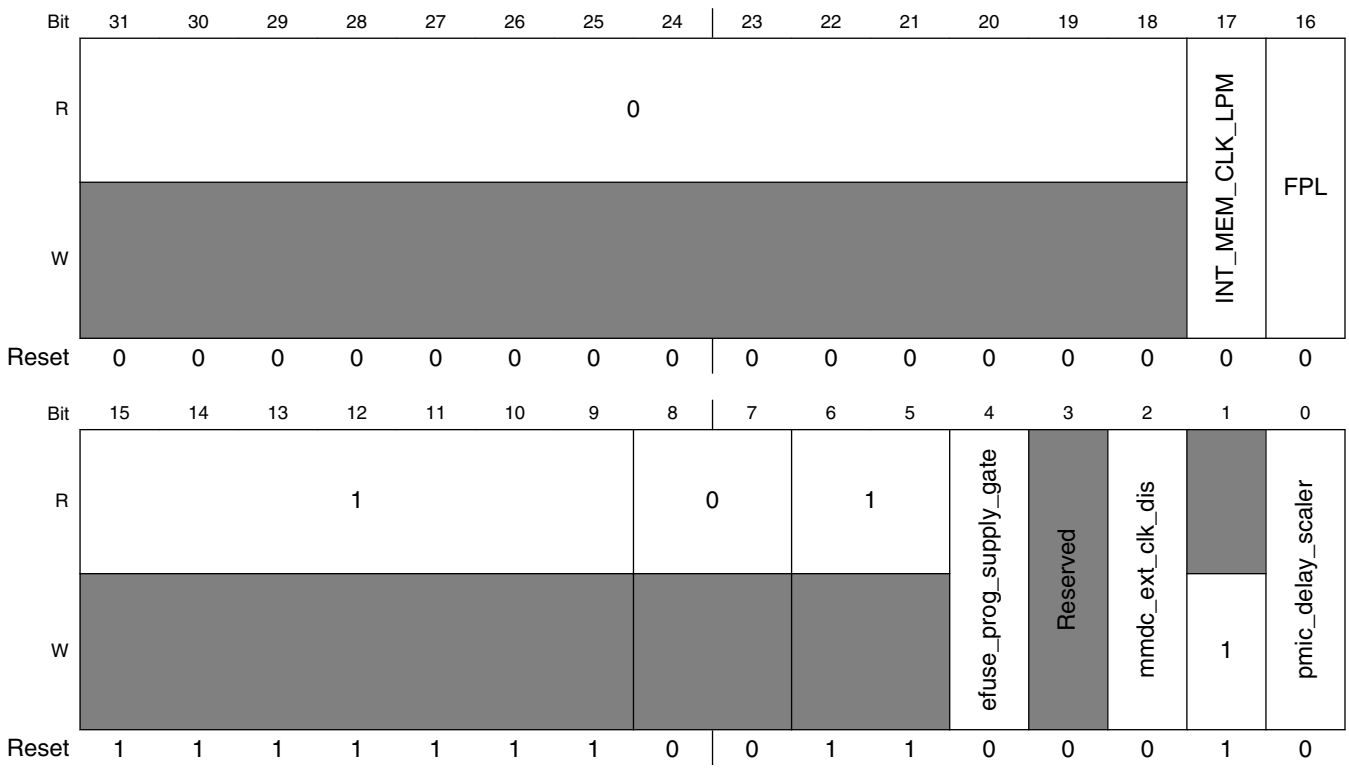
CCM_CCOSR field descriptions (continued)

Field	Description
	011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
CLKO1_SEL	Selection of the clock to be generated on CCM_CLKO1 0000 pll3_sw_clk (this inputs has additional constant division /2) 0001 pll2_main_clk (this inputs has additional constant division /2) 0010 pll1_main_clk (this inputs has additional constant division /2) 0011 pll5_main_clk (this inputs has additional constant division /2) 0100 video_27M_clk_root 0101 axi_clk_root 0110 enfc_clk_root 0111 ipu1_di0_clk_root 1000 ipu1_di1_clk_root 1001 ipu2_di0_clk_root 1010 ipu2_di1_clk_root 1011 ahb_clk_root 1100 ipg_clk_root 1101 perclk_root 1110 ckil_sync_clk_root 1111 pll4_main_clk

18.6.22 CCM General Purpose Register (CCM_CGPR)

Fast PLL enable. Can be used to engage PLL faster after STOP mode, if 24MHz OSC was active

Address: 20C_4000h base + 64h offset = 20C_4064h



CCM_CGPR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value 0.
17 INT_MEM_CLK_LPM	Control for the Deep Sleep signal to the ARM Platform memories with additional control logic based on the ARM WFI signal. Used to keep the ARM Platform memory clocks enabled if an interrupt is pending when entering low power mode. **NOTE:** This bit should always bet set when the CCM_CLPCR_LPM bits are set to 01(WAIT Mode) or 10 (STOP mode) without power gating. This bit does not have to be set for STOP mode entry. 0 Disable the clock to the ARM platform memories when entering Low Power Mode 1 Keep the clocks to the ARM platform memories enabled only if an interrupt is pending when entering Low Power Modes (WAIT and STOP without power gating)
16 FPL	Fast PLL enable.

Table continues on the next page...

CCM_CGPR field descriptions (continued)

Field	Description
	0 Engage PLL enable default way. 1 Engage PLL enable 3 CKIL clocks earlier at exiting low power mode (STOP). Should be used only if 24MHz OSC was active in low power mode.
15–9 Reserved	This read-only field is reserved and always has the value 1.
8–7 Reserved	This read-only field is reserved and always has the value 0.
6–5 Reserved	This read-only field is reserved and always has the value 1.
4 efuse_prog_supply_gate	Defines the value of the output signal cgpr_dout[4]. Gate of program supply for efuse programing 0 fuse programing supply voltage is gated off to the efuse module 1 allow fuse programing.
3 -	This field is reserved. Reserved
2 mmdc_ext_clk_dis	Disable external clock driver of MMDC during STOP mode 1 disable during stop mode 0 don't disable during stop mode.
1 -	Reserved. Keep default value set to '1' for proper operation.
0 pmic_delay_scaler	Defines clock dividion of clock for stby_count (pmic delay counter) 0 clock is not divided 1 clock is divided /8

18.6.23 CCM Clock Gating Register 0 (CCM_CCGR0)

CG(i) bits CCGR 0-6

These bits are used to turn on/off the clock to each module independently. The following table details the possible clock activity conditions for each module.

CGR value	Clock Activity Description
00	Clock is off during all modes. Stop enter hardware handshake is disabled.
01	Clock is on in run mode, but off in WAIT and STOP modes
10	Not applicable (Reserved).
11	Clock is on during all modes, except STOP mode.

Module should be stopped, before set its bits to "0"; clocks to the module will be stopped immediately.

The tables above show the register mapings for the different CGRs. The clock connectivity table should be used to match the "CCM output affected" to the actual clocks going into the modules.

The figure below represents the CCM Clock Gating Register 0 (CCM_CCGR0). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 7 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: 20C_4000h base + 68h offset = 20C_4068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR0 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	dtcp clocks (dtcp_clk_enable)
27–26 CG13	dcic2 clocks (dcic2_clk_enable)
25–24 CG12	dcic 1 clocks (dcic1_clk_enable)
23–22 CG11	CPU debug clocks (arm_dbg_clk_enable)
21–20 CG10	can2_serial clock (can2_serial_clk_enable)
19–18 CG9	can2 clock (can2_clk_enable)
17–16 CG8	can1_serial clock (can1_serial_clk_enable)
15–14 CG7	can1 clock (can1_clk_enable)
13–12 CG6	caam_wrapper_ipg clock (caam_wrapper_ipg_enable)
11–10 CG5	caam_wrapper_aclk clock (caam_wrapper_aclk_enable)
9–8 CG4	caam_secure_mem clock (caam_secure_mem_clk_enable)
7–6 CG3	asrc clock (asrc_clk_enable)

Table continues on the next page...

CCM_CCGR0 field descriptions (continued)

Field	Description
5–4 CG2	apbhdma hclk clock (apbhdma_hclk_enable)
3–2 CG1	aips_tz2 clocks (aips_tz2_clk_enable)
CG0	aips_tz1 clocks (aips_tz1_clk_enable)

18.6.24 CCM Clock Gating Register 1 (CCM_CCGR1)

The figure below represents the CCM Clock Gating Register 1(CCM_CCGR1). The clock gating registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 6Ch offset = 20C_406Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR1 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved
27–26 CG13	gpu3d clock (gpu3d_clk_enable)
25–24 CG12	gpu2d clock (gpu2d_clk_enable) **NOTE:** GPU2D clock cannot be gated without gating OPENVG clock as well. Configure both CG bits (CCM_ANALOG_CCGR1[CG12] and CCM_ANALOG_CCGR3[CG15]), to gate GPU2D.
23–22 CG11	gpt serial clock (gpt_serial_clk_enable)
21–20 CG10	gpt bus clock (gpt_clk_enable)
19–18 CG9	Reserved

Table continues on the next page...

CCM_CCGR1 field descriptions (continued)

Field	Description
17–16 CG8	esai clocks (esai_clk_enable)
15–14 CG7	epit2 clocks (epit2_clk_enable)
13–12 CG6	epit1 clocks (epit1_clk_enable)
11–10 CG5	enet clock (enet_clk_enable)
9–8 CG4	ecspi5 clocks (ecspi5_clk_enable)
7–6 CG3	ecspi4 clocks (ecspi4_clk_enable)
5–4 CG2	ecspi3 clocks (ecspi3_clk_enable)
3–2 CG1	ecspi2 clocks (ecspi2_clk_enable)
CG0	ecspi1 clocks (ecspi1_clk_enable)

18.6.25 CCM Clock Gating Register 2 (CCM_CCGR2)

The figure below represents the CCM Clock Gating Register 2 (CCM_CCGR2). The clock gating registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 70h offset = 20C_4070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR2 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved

Table continues on the next page...

CCM_CCGR2 field descriptions (continued)

Field	Description
27–26 CG13	ipsync_vdoa_ipg clocks (ipsync_vdoa_ipg_master_clk_enable)
25–24 CG12	ipsync_ip2apb_tzasc2_ipg clocks (ipsync_ip2apb_tzasc2_ipg_master_clk_enable) >
23–22 CG11	ipsync_ip2apb_tzasc1_ipg clocks (ipsync_ip2apb_tzasc1_ipg_master_clk_enable)
21–20 CG10	ipmux3 clock (ipmux3_clk_enable)
19–18 CG9	ipmux2 clock (ipmux2_clk_enable)
17–16 CG8	ipmux1 clock (ipmux1_clk_enable)
15–14 CG7	iomux_ipt_clk_io clock (iomux_ipt_clk_io_enable)
13–12 CG6	OCOTP_CTRL clock (iim_clk_enable)
11–10 CG5	i2c3_serial clock (i2c3_serial_clk_enable)
9–8 CG4	i2c2_serial clock (i2c2_serial_clk_enable)
7–6 CG3	i2c1_serial clock (i2c1_serial_clk_enable)
5–4 CG2	hdmi_tx_isfrclk clock (hdmi_tx_isfrclk_enable)
3–2 CG1	Reserved
CG0	hdmi_tx_iahbclk, hdmi_tx_ihclk clock (hdmi_tx_enable)

18.6.26 CCM Clock Gating Register 3 (CCM_CCGR3)

The figure below represents the CCM Clock Gating Register 3 (CCM_CCGR3). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 74h offset = 20C_4074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR3 field descriptions

Field	Description
31–30 CG15	openvgaxiclk clock (openvgaxiclk_clk_root_enable) **NOTE:** OPENVG clock cannot be gated without gating GPU2D clock as well. Configure both CG bits (CCM_ANALOG_CCGR1[CG12] and CCM_ANALOG_CCGR3[CG15]) to gate OPENVG.
29–28 CG14	ocram clock (ocram_clk_enable)
27–26 CG13	Reserved
25–24 CG12	mmdc_core_ipg_clk_p0 clock (mmdc_core_ipg_clk_p0_enable)
23–22 CG11	Reserved
21–20 CG10	mmdc_core_aclk_fast_core_p0 clock (mmdc_core_aclk_fast_core_p0_enable)
19–18 CG9	mlb clock (mlb_clk_enable)
17–16 CG8	mipi_core_cfg clock (mipi_core_cfg_clk_enable)
15–14 CG7	ldb_di1 clock (ldb_di1_clk_enable)
13–12 CG6	ldb_di0 clock (ldb_di0_clk_enable)
11–10 CG5	ipu2_di1 clock and pre-clock (ipu2_ipu_di1_clk_enable)
9–8 CG4	ipu2_di0 clock and pre-clock (ipu2_ipu_di0_clk_enable)

Table continues on the next page...

CCM_CCGR3 field descriptions (continued)

Field	Description
7–6 CG3	ipu2_ipu clock (ipu2_ipu_clk_enable)
5–4 CG2	ipu1_di1 clock and pre-clock (ipu1_ipu_di1_clk_enable)
3–2 CG1	ipu1_di0 clock and pre-clock (ipu1_ipu_di0_clk_enable)
CG0	ipu1_ipu clock (ipu1_ipu_clk_enable)

18.6.27 CCM Clock Gating Register 4 (CCM_CCGR4)

The figure below represents the CCM Clock Gating Register 4 (CCM_CCGR4). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 78h offset = 20C_4078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR4 field descriptions

Field	Description
31–30 CG15	rawnand_u_gpmi_input_apb clock (rawnand_u_gpmi_input_apb_clk_enable)
29–28 CG14	rawnand_u_gpmi_bch_input_gpmi_io clock (rawnand_u_gpmi_bch_input_gpmi_io_clk_enable)
27–26 CG13	rawnand_u_gpmi_bch_input_bch clock (rawnand_u_gpmi_bch_input_bch_clk_enable)
25–24 CG12	rawnand_u_bch_input_apb clock (rawnand_u_bch_input_apb_clk_enable)
23–22 CG11	pwm4 clocks (pwm4_clk_enable)
21–20 CG10	pwm3 clocks (pwm3_clk_enable)
19–18 CG9	pwm2 clocks (pwm2_clk_enable)

Table continues on the next page...

CCM_CCGR4 field descriptions (continued)

Field	Description
17–16 CG8	pwm1 clocks (pwm1_clk_enable)
15–14 CG7	pl301_mx6qper2_mainclk_enable (pl301_mx6qper2_mainclk_enable)
13–12 CG6	pl301_mx6qper1_bch clocks (pl301_mx6qper1_bchclk_enable)
11–10 CG5	Reserved
9–8 CG4	pl301_mx6qfast1_s133 clock (pl301_mx6qfast1_s133clk_enable)
7–6 CG3	Reserved.
5–4 CG2	Reserved.
3–2 CG1	Reserved.
CG0	pcie clock (pcie_root_enable)

18.6.28 CCM Clock Gating Register 5 (CCM_CCGR5)

The figure below represents the CCM Clock Gating Register 5 (CCM_CCGR5). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 7Ch offset = 20C_407Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR5 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved

Table continues on the next page...

CCM_CCGR5 field descriptions (continued)

Field	Description
27–26 CG13	uart_serial clock (uart_serial_clk_enable)
25–24 CG12	uart clock (uart_clk_enable)
23–22 CG11	ssi3 clocks (ssi3_clk_enable)
21–20 CG10	ssi2 clocks (ssi2_clk_enable)
19–18 CG9	ssi1 clocks (ssi1_clk_enable)
17–16 CG8	Reserved
15–14 CG7	spdif clock (spdif_clk_enable)
13–12 CG6	spba clock (spba_clk_enable)
11–10 CG5	Reserved
9–8 CG4	Reserved
7–6 CG3	sdma clock (sdma_clk_enable)
5–4 CG2	sata clock (sata_clk_enable)
3–2 CG1	Reserved
CG0	rom clock (rom_clk_enable)

18.6.29 CCM Clock Gating Register 6 (CCM_CCGR6)

The figure below represents the CCM Clock Gating Register 6 (CCM_CCGR6). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is determined by the number of peripherals in the system.

Address: 20C_4000h base + 80h offset = 20C_4080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR6 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved
27–26 CG13	Reserved
25–24 CG12	Reserved
23–22 CG11	Reserved
21–20 CG10	Reserved
19–18 CG9	Reserved
17–16 CG8	Reserved
15–14 CG7	vpu clocks (vpu_clk_enable)
13–12 CG6	vdoaxiclk root clock (vdoaxiclk_clk_enable)
11–10 CG5	eim_slow clocks (eim_slow_clk_enable)
9–8 CG4	usdhc4 clocks (usdhc4_clk_enable)

Table continues on the next page...

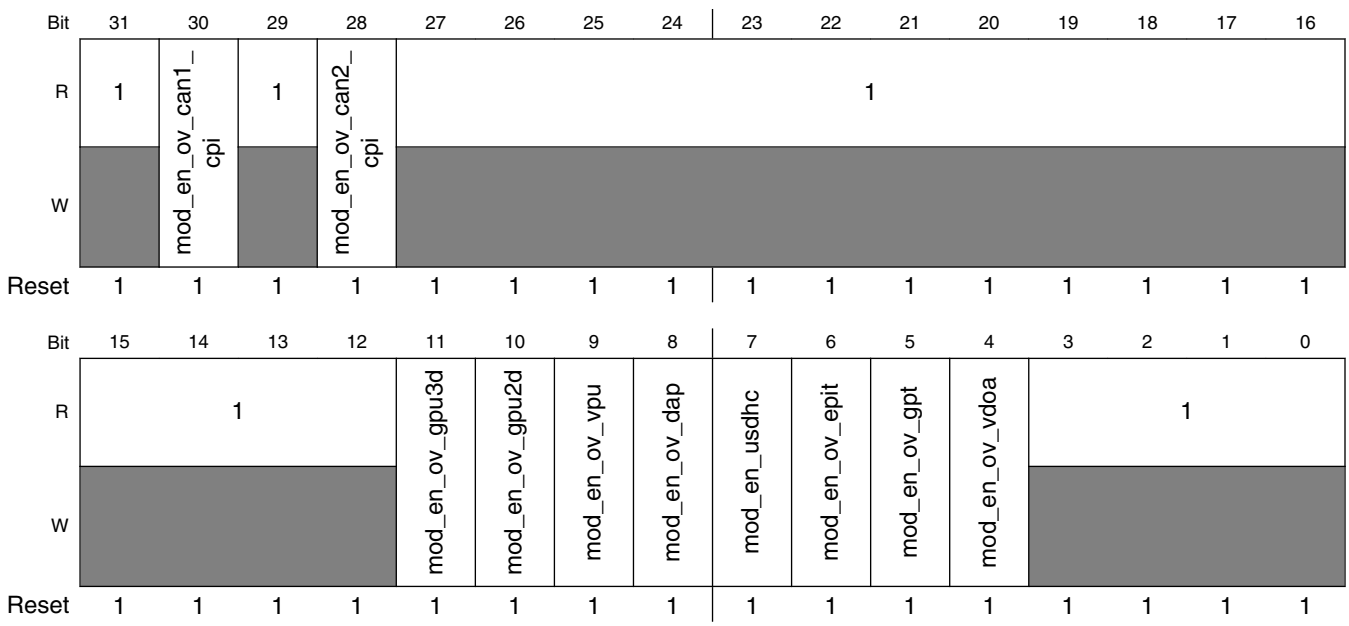
CCM_CCGR6 field descriptions (continued)

Field	Description
7–6 CG3	usdhc3 clocks (usdhc3_clk_enable)
5–4 CG2	usdhc2 clocks (usdhc2_clk_enable)
3–2 CG1	usdhc1 clocks (usdhc1_clk_enable)
CG0	usboh3 clock (usboh3_clk_enable)

18.6.30 CCM Module Enable Overide Register (CCM_CMEOR)

The following figure represents the CCM Module Enable Override Register (CMEOR). The CMEOR register contains bits to override the clock enable signal from the module. This bit is applicable only for modules whose clock enable signals are used. The following table provides its field descriptions.

Address: 20C_4000h base + 88h offset = 20C_4088h



CCM_CMEOR field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 1.

Table continues on the next page...

CCM_CMEOR field descriptions (continued)

Field	Description
30 mod_en_ov_ can1_cpi	Overide clock enable signal from CAN1 - clock will not be gated based on CAN's signal 'enable_clk_cpi'. 0 don't overide module enable signal 1 overide module enable signal
29 Reserved	This read-only field is reserved and always has the value 1.
28 mod_en_ov_ can2_cpi	Overide clock enable signal from CAN2 - clock will not be gated based on CAN's signal 'enable_clk_cpi'. 0 don't override module enable signal 1 override module enable signal
27–12 Reserved	This read-only field is reserved and always has the value 1.
11 mod_en_ov_ gpu3d	Overide clock enable signal from GPU3D - clock will not be gated based on GPU3D's signal. 0 don't override module enable signal 1 override module enable signal
10 mod_en_ov_ gpu2d	Overide clock enable signal from GPU2D - clock will not be gated based on GPU's signal 'gpu2d_busy' . 0 don't override module enable signal 1 override module enable signal
9 mod_en_ov_vpu	Overide clock enable signal from VPU- clock will not be gated based on VPU's signal 'vpu_idle' . 0 don't override module enable signal 1 override module enable signal
8 mod_en_ov_dap	Overide clock enable signal from DAP- clock will not be gated based on DAP's signal 'dap_dbgen' . 0 don't override module enable signal 1 override module enable signal
7 mod_en_usdhc	Overide clock enable signal from USDHC. 0 don't override module enable signal 1 override module enable signal
6 mod_en_ov_epit	Overide clock enable signal from EPIT - clock will not be gated based on EPIT's signal 'ipg_enable_clk' . 0 don't override module enable signal 1 override module enable signal
5 mod_en_ov_gpt	Overide clock enable signal from GPT - clock will not be gated based on GPT's signal 'ipg_enable_clk' . 0 don't override module enable signal 1 override module enable signal
4 mod_en_ov_ vdoa	Overide clock enable signal from vdoa - clock will not be gated based on vdoa signal. 0 don't override module enable signal 1 override module enable signal
Reserved	This read-only field is reserved and always has the value 1.