MCAL Configuration Verification Manual for Mcu

32-bit TriCore™ AURIX™ TC3xx microcontroller family

About this document

Scope and purpose

This Configuration Data Reference document is applicable to all TC3xx devices in the TriCore™ AURIX™ family of 32-bit microcontrollers.

The purpose of this document is to facilitate the integrator to verify the generated code based on the input configuration parameters. This document describes details of structures, defines, macros and variables generated from the configuration parameters.

Intended audience

This document is intended for integrators who need to understand the logic of the generated configuration code of AURIX™ AUTOSAR MCAL.

Reference documents

This document should be read in conjunction with the following documents:

- AURIX™ TC3xx MCAL User Manual Mcu

Table of contents

1 Mcu driver

This chapter describes the details of the configuration data generated from the MCU driver.

1.1 File: Mcu_Cfg.h

The generated header file contains all pre-compile configuration parameters. Pre-compile time configuration allows decoupling of the static configuration from implementation. The file is generated in 'inc' folder.

1.1.1 Macro: MCU_AR_RELEASE_MAJOR_VERSION

Table 1 MCU_AR_RELEASE_MAJOR_VERSION

Name	MCU_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which MCU implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h file with ArMajorVersion 4	`#define MCU_AR_RELEASE_MAJOR_VERSION (4U)`

1.1.2 Macro: MCU_AR_RELEASE_MINOR_VERSION

Table 2 MCU_AR_RELEASE_MINOR_VERSION

Name	MCU_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the Mcu implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h file with ArMinorVersion 2	`#define MCU_AR_RELEASE_MINOR_VERSION (2U)`

1.1.3 Macro: MCU_AR_RELEASE_REVISION_VERSION

Table 3 MCU_AR_RELEASE_REVISION_VERSION

Name	MCU_AR_RELEASE_REVISION_VERSION
Description	Revision version number of AUTOSAR release on which the Mcu implementation is based on.

Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArPatchVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h file with ArPatchVersion 2	`#define MCU_AR_RELEASE_REVISION_VERSION (2U)`

1.1.4 Macro: MCU_SW_MAJOR_VERSION

Table 4 MCU_SW_MAJOR_VERSION

Name	MCU_SW_MAJOR_VERSION	
Description	Major version number of the Mcu module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h file with SwMajorVersion 10	`#define MCU_SW_MAJOR_VERSION (10U)`

1.1.5 Macro: MCU_SW_MINOR_VERSION

Table 5 MCU_SW_MINOR_VERSION

Name	MCU_SW_MINOR_VERSION	
Description	Minor version number of the Mcu module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h file with SwMinorVersion 10	`#define MCU_SW_MINOR_VERSION (10U)`

1.1.6 Macro: MCU_SW_PATCH_VERSION

Table 6 MCU_SW_PATCH_VERSION

Name	MCU_SW_PATCH_VERSION
Description	Patch level version number of the Mcu module.
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwPatchVersion'.

	Note: *The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h file with SwPatchVersion 0	`#define MCU_SW_PATCH_VERSION (0U)`

1.1.7 Macro: MCU _SAFETY_ENABLE

Table 7 MCU_SAFETY_ENABLE

Name	MCU_SAFETY_ENABLE	
Description	Enables/disables safety features	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/McuSafetyEnable' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/McuSafetyEnable = True	`#define MCU_SAFETY_ENABLE (STD_ON)`
	McuGeneralConfiguration/McuSafetyEnable = False	`#define MCU_SAFETY_ENABLE (STD_OFF)`

1.1.8 Macro: MCU_INITCHECK_API

Table 8 MCU_INITCHECK_API

Name	MCU_INITCHECK_API	
Description	Enables/disables Mcu_InitCheck API	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/McuInitCheckApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuInitCheckApi = True	`#define MCU_INITCHECK_API (STD_ON)`
	McuGeneralConfiguration/ McuInitCheckApi = False	`#define MCU_INITCHECK_API (STD_OFF)`

1.1.9 Macro: MCU_RUNTIME_API_MODE

Table 9 MCU_RUNTIME_API_MODE

Name	MCU_RUNTIME_API_MODE	
Description	Decides the mode of execution of Run Time API's	
Verification method	The macro is generated as MCU_MCAL_SUPERVISOR if 'McuGeneralConfiguration/ McuRuntimeApiMode' configuration parameter is set to 'MCU_MCAL_SUPERVISOR' else the macro is generated as MCU_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuRuntimeApiMode =	`#define MCU _RUNTIME_API_MODE`

	MCU_MCAL_SUPERVISOR	(MCU_MCAL_SUPERVISOR)
	McuGeneralConfiguration/ McuRuntimeApiMode = MCU_MCAL_USER1	#define MCU_RUNTIME_API_MODE (MCU_MCAL_USER1)

1.1.10 Macro: MCU_INIT_DEINIT_API_MODE

Table 10 MCU_INIT_DEINIT_API_MODE

Name	MCU _INIT_DEINIT_API_MODE	
Description	Decides the mode of execution of Init and DeInit API's.	
Verification method	The macro is generated as MCU_MCAL_SUPERVISOR if 'McuGeneralConfiguration/ McuInitDeInitApiMode' configuration parameter is set to 'MCU_MCAL_SUPERVISOR' else the macro is generated as MCU_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuInitDeInitApiMode = MCU_MCAL_SUPERVISOR	#define MCU_INIT_DEINIT_API_MODE (MCU_MCAL_SUPERVISOR)
	McuGeneralConfiguration/ McuInitDeInitApiMode = MCU_MCAL_USER1	#define MCU_INIT_DEINIT_API_MODE (MCU_MCAL_USER1)

1.1.11 Macro: MCU_DEV_ERROR_DETECT

Table 11 MCU_DEV_ERROR_DETECT

Name	MCU_DEV_ERROR_DETECT	
Description	Enables/disables the Development Error Detection.	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuDevErrorDetect' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuDevErrorDetect = True	#define MCU_DEV_ERROR_DETECT (STD_ON)
	McuGeneralConfiguration/ McuDevErrorDetect = False	#define MCU_DEV_ERROR_DETECT (STD_OFF)

1.1.12 Macro: MCU_MULTICORE_ERROR_DETECT

Table 12 MCU_MULTICORE_ERROR_DETECT

Name	MCU_MULTICORE_ERROR_DETECT
Description	Enables/disables MultiCore DET Check
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuMultiCoreErrorDetect' configuration parameter is set to 'True' else the macro is generated as STD_OFF.

Example(s)	Action	Generated output
	McuGeneralConfiguration/ McuMultiCoreErrorDetect = True	`#define MCU_MULTICORE_ERROR_DETECT (STD_ON)`
	McuGeneralConfiguration/ McuMultiCoreErrorDetect = False	`#define MCU_MULTICORE_ERROR_DETECT (STD_OFF)`

1.1.13 Macro: MCU_VERSION_INFO_API

Table 13 MCU_VERSION_INFO_API

Name	MCU_VERSION_INFO_API	
Description	Enables/disables the Mcu_GetVersionInfo API.	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuVersionInfoApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuVersionInfoApi = True	`#define MCU_VERSION_INFO_API (STD_ON)`
	McuGeneralConfiguration/ McuVersionInfoApi = False	`#define MCU_VERSION_INFO_API (STD_OFF)`

1.1.14 Macro: MCU_DEINIT_API

Table 14 MCU_DEINIT_API

Name	MCU_DEINIT_API	
Description	Enables/disables Mcu_DeInit API.	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuIfxDeInitApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuIfxDeInitApi = True	`#define MCU_DEINIT_API (STD_ON)`
	McuGeneralConfiguration/ McuIfxDeInitApi = False	`#define MCU_DEINIT_API (STD_OFF)`

1.1.15 Macro: MCU_DISTRIBUTE_PLL_CLOCK_API

Table 15 MCU_DISTRIBUTE_PLL_CLOCK_API

Name	MCU_DISTRIBUTE_PLL_CLOCK_API
Description	Enables/disables Mcu_DistributePllClock API
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/McuNoPll' configuration parameter is set to 'False' else the macro is generated as STD_OFF.

Example(s)	Action	Generated output
	McuGeneralConfiguration/ McuNoPll = False	`#define MCU_DISTRIBUTE_PLL_CLOCK_API (STD_ON)`
	McuGeneralConfiguration/ McuNoPll = True	`#define MCU_DISTRIBUTE_PLL_CLOCK_API (STD_OFF)`

1.1.16 Macro: MCU_INIT_CLOCK_API

Table 16 MCU_INIT_CLOCK_API

Name	MCU_INIT_CLOCK_API	
Description	Enables/disables Mcu_InitClock API	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/McuInitClock' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuInitClock = True	`#define MCU_INIT_CLOCK_API (STD_ON)`
	McuGeneralConfiguration/ McuInitClock = False	`#define MCU_INIT_CLOCK_API (STD_OFF)`

1.1.17 Macro: MCU_PERFORM_RESET_API

Table 17 MCU_PERFORM_RESET_API

Name	MCU_PERFORM_RESET_API	
Description	Enables/disables Mcu_PerformReset API	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuPerformResetApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuPerformResetApi = True	`#define MCU_PERFORM_RESET_API (STD_ON)`
	McuGeneralConfiguration/ McuPerformResetApi = False	`#define MCU_PERFORM_RESET_API (STD_OFF)`

1.1.18 Macro: MCU_GET_RAM_STATE_API

Table 18 MCU_GET_RAM_STATE_API

Name	MCU_GET_RAM_STATE_API	
Description	Enables/disables Mcu_GetRamState API	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuGetRamStateApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**

	McuGeneralConfiguration/ McuGetRamStateApi = True	`#define MCU_GET_RAM_STATE_API (STD_ON)`
	McuGeneralConfiguration/ McuGetRamStateApi = False	`#define MCU_GET_RAM_STATE_API (STD_OFF)`

1.1.19 Macro: MCU_CLR_COLD_RESET_STAT_API

Table 19 MCU_CLR_COLD_RESET_STAT_API

Name	MCU_CLR_COLD_RESET_STAT_API	
Description	Enables/disables Mcu_ClearColdResetStatus API	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuClearColdResetStatusApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuClearColdResetStatusApi = True	`#define MCU_CLR_COLD_RESET_STAT_API (STD_ON)`
	McuGeneralConfiguration/ McuClearColdResetStatusApi = False	`#define MCU_CLR_COLD_RESET_STAT_API (STD_OFF)`

1.1.20 Macro: MCU_TRAP_API

Table 20 MCU_TRAP_API

Name	MCU_TRAP_API	
Description	Enable/disable following APIs: Mcu_GetTrapCause Mcu_SetTrapRequest Mcu_ClearTrapRequest	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/McuIfxTrapApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuIfxTrapApi = True	`#define MCU_TRAP_API  (STD_ON)`
	McuGeneralConfiguration/ McuIfxTrapApi = False	`#define MCU_TRAP_API (STD_OFF)`

1.1.21 Macro: MCU_CPU_CCUCON_UPDATE_API

Table 21 MCU_CPU_CCUCON_UPDATE_API

Name	MCU_CPU_CCUCON_UPDATE_API
Description	Enables/disables Mcu_UpdateCpuCcuconReg API

Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuIfxCpuCcuconApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuIfxCpuCcuconApi = True	`#define MCU_CPU_CCUCON_UPDATE_API (STD_ON)`
	McuGeneralConfiguration/ McuIfxCpuCcuconApi = False	`#define MCU_CPU_CCUCON_UPDATE_API (STD_OFF)`

1.1.22 Macro: MCU_LOW_POWER_MODE_API

Table 22 MCU_LOW_POWER_MODE_API

Name	MCU_LOW_POWER_MODE_API	
Description	Enable/disable following APIs: Mcu_GetCpuIdleModeInitiator Mcu_GetCpuState Mcu_GetWakeupCause Mcu_ClearWakeupCause	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/McuIfxLpmApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuIfxLpmApi = True	`#define MCU_LOW_POWER_MODE_API (STD_ON)`
	McuGeneralConfiguration/ McuIfxLpmApi = False	`#define MCU_LOW_POWER_MODE_API (STD_OFF)`

1.1.23 Macro: MCU_CLK_SRC_FAILURE_NOTIF_API

Table 23 MCU_CLK_SRC_FAILURE_NOTIF_API

Name	MCU_CLK_SRC_FAILURE_NOTIF_API	
Description	Enables/disables Mcu_ClockFailureNotification API	
Verification method	The macro is generated as STD_ON if 'McuGeneralConfiguration/ McuClockSourceFailureNotification' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuClockSourceFailureNotification = True	`#define MCU_CLK_SRC_FAILURE_NOTIF_API (STD_ON)`
	McuGeneralConfiguration/ McuClockSourceFailureNotification = False	`#define MCU_CLK_SRC_FAILURE_NOTIF_API (STD_OFF)`

1.1.24 Macro: MCU_GTM_USED

Table 24 MCU_GTM_USED

Name	MCU_GTM_USED	
Description	Specifies whether GTM is available in hardware or not. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as STD_ON if GTM peripheral is available in the hardware else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	GTM is available	`#define MCU_GTM_USED (STD_ON)`
	GTM is not available	`#define MCU_GTM_USED (STD_OFF)`

1.1.25 Macro: MCU_GTM_NO_OF_TOM_AVAILABLE

Table 25 MCU_GTM_NO_OF_TOM_AVAILABLE

Name	MCU_GTM_NO_OF_TOM_AVAILABLE	
Description	Specifies the number of TOM modules available in hardware. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on the number of TOMs available in device.	
Example(s)	**Action**	**Generated output**
	Number of TOMs available = 5	`#define MCU_GTM_NO_OF_TOM_AVAILABLE (5U)`
	Number of TOMs available = 8	`#define MCU_GTM_NO_OF_TOM_AVAILABLE (8U)`

1.1.26 Macro: MCU_GTM_NO_OF_ATOM_AVAILABLE

Table 26 MCU_GTM_NO_OF_ATOM_AVAILABLE

Name	MCU_GTM_NO_OF_ATOM_AVAILABLE	
Description	Specifies the number of ATOM modules available in hardware. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on the number of ATOMs available in device.	
Example(s)	**Action**	**Generated output**
	Number of ATOMs available = 8	`#define MCU_GTM_NO_OF_ATOM_AVAILABLE (8U)`

	Number of ATOMs available = 12	`#define MCU_GTM_NO_OF_ATOM_AVAILABLE (12U)`

1.1.27 Macro: MCU_GTM_NO_OF_TIM_AVAILABLE

Table 27 MCU_GTM_NO_OF_TIM_AVAILABLE

Name	MCU_GTM_NO_OF_TIM_AVAILABLE	
Description	Specifies the number of TIM modules available in hardware. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on the number of TIMs available in device.	
Example(s)	**Action**	**Generated output**
	Number of TIMs available = 8	`#define MCU_GTM_NO_OF_TIM_AVAILABLE (8U)`
	Number of TIMs available = 12	`#define MCU_GTM_NO_OF_TIM_AVAILABLE (12U)`

1.1.28 Macro: MCU_GTM_NO_OF_TBU_CH_AVAILABLE

Table 28 MCU_GTM_NO_OF_TBU_CH_AVAILABLE

Name	MCU_GTM_NO_OF_TBU_CH_AVAILABLE	
Description	Specifies the number of TBU channels available in GTM. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on the number of TBU channels available in device.	
Example(s)	**Action**	**Generated output**
	Number of TBU channels available = 1	`#define MCU_GTM_NO_OF_TBU_CH_AVAILABLE (1U)`
	Number of TBU channels available = 3	`#define MCU_GTM_NO_OF_TBU_CH_AVAILABLE (3U)`

1.1.29 Macro: MCU_GTM_NO_OF_CCM_AVAILABLE

Table 29 MCU_GTM_NO_OF_CCM_AVAILABLE

Name	MCU_GTM_NO_OF_CCM_AVAILABLE
Description	Specifies the number of clusters available in GTM. *Note: This macro is not configurable by the user.*
Verification method	The macro is generated based on the number of clusters available in device.

Example(s)	Action	Generated output
	Number of clusters available = 5	`#define MCU_GTM_NO_OF_CCM_AVAILABLE (5U)`
	Number of clusters available = 12	`#define MCU_GTM_NO_OF_CCM_AVAILABLE (12U)`

1.1.30 Macro: MCU_GTM_NO_OF_TOUTSEL_AVAILABLE

Table 30 MCU_GTM_NO_OF_TOUTSEL_AVAILABLE

Name	MCU_GTM_NO_OF_TOUTSEL_AVAILABLE	
Description	Specifies the number of TOUTSEL registers available in GTM. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on the number of Toutsel registers available in device.	
Example(s)	**Action**	**Generated output**
	Number of Toutsel registers available = 34	`#define MCU_GTM_NO_OF_TOUTSEL_AVAILABLE (34U)`
	Number of Toutsel registers available = 17	`#define MCU_GTM_NO_OF_TOUTSEL_AVAILABLE (17U)`

1.1.31 Macro: MCU_GTM_CLS_CLK_CFG_RESET_VAL

Table 31 MCU_GTM_CLS_CLK_CFG_RESET_VAL

Name	MCU_GTM_CLS_CLK_CFG_RESET_VAL	
Description	Specifies the reset value of GTM cluster clock configuration register	
Verification method	The macro is generated based on the number of clusters available in device.Steps involved for generation of macro are:1. ResetVal = 0x2 2. A loop is run for the number of clusters available.3. ResetVal = ((ResetVal << Index of Loop) | ResetVal).	
Example(s)	**Action**	**Generated output**
	Number of clusters available = 4	`#define MCU_GTM_CLS_CLK_CFG_RESET_VAL (0x000000aaU)`
	Number of clusters available = 12	`#define MCU_GTM_CLS_CLK_CFG_RESET_VAL (0x00aaaaaaU)`

1.1.32 Macro: MCU_GTM_TO_DSADC_TRIG_AVAILABLE

Table 32 MCU_GTM_TO_DSADC_TRIG_AVAILABLE

Name	MCU_GTM_TO_DSADC_TRIG_AVAILABLE
Description	Specifies whether GTM to DSADC trigger is available in hardware or not.

	Note: *This macro is not configurable by the user.*	
Verification method	The macro is generated as STD_ON if even one Dsadc channel is available in the hardware, else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Number of Dsadc channels available in Hardware = 8	`#define MCU_GTM_TO_DSADC_TRIG_AVAILABLE (STD_ON)`
	Number of Dsadc channels available in Hardware = 0	`#define MCU_GTM_TO_DSADC_TRIG_AVAILABLE (STD_OFF)`

1.1.33 Macro: MCU_GTM_TO_DSADC_TRIG1

Table 33 MCU_GTM_TO_DSADC_TRIG1

Name	MCU_GTM_TO_DSADC_TRIG1	
Description	Specifies whether GTM to DSADC trigger 1 is available in hardware or not. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro is generated as STD_ON if more than 8 Dsadc channels are available in the hardware is else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Number of Dsadc channels available in Hardware = 10	`#define MCU_GTM_TO_DSADC_TRIG1 (STD_ON)`
	Number of Dsadc channels available in Hardware = 5	`#define MCU_GTM_TO_DSADC_TRIG1 (STD_OFF)`

1.1.34 Macro: MCU_TBU_CH_EN_MSK

Table 34 MCU_TBU_CH_EN_MSK

Name	MCU_TBU_CH_EN_MSK	
Description	Specifies the mask for available TBU channels	
Verification method	The macro is generated based on the number of TBU channels available in the hardware. The value is generated based on following algorithm: 1. A loop runs for the number of TBU channels 2. MaskVal = MaskVal | (3 << (2 * Loopindex))	
Example(s)	**Action**	**Generated output**
	Number of TBU channels available in Hardware = 4	`#define MCU_TBU_CH_EN_MSK (0x000000ffU)`
	Number of TBU channels available in Hardware = 2	`#define MCU_TBU_CH_EN_MSK (0x0000000fU)`

1.1.35 Macro: MCU_MAIN_OSC_FREQ

Table 35 MCU_MAIN_OSC_FREQ

Name	MCU_MAIN_OSC_FREQ	
Description	Specifies the frequency of main oscillator used for clock generation in MHz	
Verification method	The macro is generated based on the value assigned in configuration parameter 'McuGeneralConfiguration/McuMainOscillatorFrequency'.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuMainOscillatorFrequency = 16 MHz	`#define MCU_MAIN_OSC_FREQ (16U)`
	McuGeneralConfiguration/ McuMainOscillatorFrequency = 20 MHz	`#define MCU_MAIN_OSC_FREQ (20U)`

1.1.36 Macro: MCU_SYSCLK_FREQ

Table 36 MCU_SYSCLK_FREQ

Name	MCU_SYSCLK_FREQ	
Description	Specifies the frequency of SYSCLK used for clock generation in MHz	
Verification method	The macro is generated based on the value assigned in configuration parameter 'McuGeneralConfiguration/McuSysClkFrequency'.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuSysClkFrequency = 16 MHz	`#define MCU_SYSCLK_FREQ (16U)`
	McuGeneralConfiguration/ McuSysClkFrequency = 20 MHz	`#define MCU_SYSCLK_FREQ (20U)`

1.1.37 Macro: MCU_BACKUP_FREQ

Table 37 MCU_BACKUP_FREQ

Name	MCU_BACKUP_FREQ	
Description	Specifies the frequency of backup clock. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on the value of backup clock used (in MHz).	
Example(s)	**Action**	**Generated output**
	Backup clock Frequency = 100 MHz	`#define MCU_BACKUP_FREQ (100U)`

1.1.38 Macro: MCU_CCU60_USED

Table 38 MCU_CCU60_USED

Name	MCU_CCU60_USED	
Description	Specifies the module which reserves CCU6 kernel 0	
Verification method	The macro is generated as STD_OFF if configuration parameter 'McuCcu6ModuleAllocationConf_0/McuCcu6ModuleAllocationConf' is set to 'CCU6_MODULE_NOT_USED' else is generated as STD_ON.	
Example(s)	**Action**	**Generated output**
	McuCcu6ModuleAllocationConf_0/ McuCcu6ModuleAllocationConf = CCU6_MODULE_NOT_USED	`#define MCU_CCU60_USED (STD_OFF)`
	McuCcu6ModuleAllocationConf_0/ McuCcu6ModuleAllocationConf = CCU6_MODULE_USED_BY_ICU_DRIVER	`#define MCU_CCU60_USED (STD_ON)`

1.1.39 Macro: MCU_CCU61_USED

Table 39 MCU_CCU61_USED

Name	MCU_CCU61_USED	
Description	Specifies the module which reserves CCU6 kernel 1	
Verification method	The macro is generated as STD_OFF if configuration parameter 'McuCcu6ModuleAllocationConf_1/McuCcu6ModuleAllocationConf' is set to 'CCU6_MODULE_NOT_USED' else is generated as STD_ON.	
Example(s)	**Action**	**Generated output**
	McuCcu6ModuleAllocationConf_1/ McuCcu6ModuleAllocationConf = CCU6_MODULE_NOT_USED	`#define MCU_CCU61_USED (STD_OFF)`
	McuCcu6ModuleAllocationConf_1/ McuCcu6ModuleAllocationConf = CCU6_MODULE_USED_BY_ICU_DRIVER	`#define MCU_CCU61_USED (STD_ON)`

1.1.40 Macro: MCU_GPT1_USED

Table 40 MCU_GPT1_USED

Name	MCU_GPT1_USED

Description	Specifies the module which reserves GPT block 1.	
Verification method	The macro is generated as STD_ON if any of the configuration parameter 'McuHardwareResourceAllocationConf/ McuGpt12ModuleAllocationConf/ GPT_TIMER_x' where x=2, 3, 4 is not set to 'GPT_TIMER_NOT_USED' else is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuHardwareResourceAllocationConf/ McuGpt12ModuleAllocationConf_0/ GPT_TIMER_2 = GPT_TIMER_NOT_USED	`#define MCU_GPT1_USED (STD_OFF)`
	McuHardwareResourceAllocationConf/ McuGpt12ModuleAllocationConf_0/ GPT_TIMER_3 = GPT_TIMER_USED_BY_ICU_DRIVER	`#define MCU_GPT1_USED (STD_ON)`

1.1.41 Macro: MCU_GPT2_USED

Table 41 MCU_GPT2_USED

Name	MCU_GPT2_USED	
Description	Specifies the module which reserves GPT block 2.	
Verification method	The macro is generated as STD_ON if any of the configuration parameter 'McuHardwareResourceAllocationConf/ McuGpt12ModuleAllocationConf/ GPT_TIMER_x' where x=5, 6 is not set to 'GPT_TIMER_NOT_USED' else is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McuHardwareResourceAllocationConf/ McuGpt12ModuleAllocationConf_3/ GPT_TIMER_5= GPT_TIMER_NOT_USED	`#define MCU_GPT2_USED (STD_OFF)`
	McuHardwareResourceAllocationConf/ McuCcu6ModuleAllocationConf _3/ GPT_TIMER_6 = GPT_TIMER_USED_BY_ICU_DRIVER	`#define MCU_GPT2_USED (STD_ON)`

1.1.42 Macro: MCU_OSCVAL_REG_VALUE

Table 42 MCU_OSCVAL_REG_VALUE

Name	MCU_OSCVAL_REG_VALUE
Description	Specifies the oscillator value to be configured in Oscillator configuration register
Verification method	The macro is generated with a value only when configuration parameter 'McuGeneralConfiguration/McuOscillatorMode' is configured as

	'EXT_CRYSTAL_CERAMIC_RES_MODE_SEL0'. The value is calculated based on following formula: Oscillator value = McuGeneralConfiguration/McuMainOscillatorFrequency - 15	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuMainOscillatorFrequency = 20 MHz McuGeneralConfiguration/ McuOscillatorMode = EXT_CRYSTAL_CERAMIC_RES _MODE_SEL0	`#define MCU_OSCVAL_REG_VALUE (0x5U)`
	McuGeneralConfiguration/McuMainOscillatorFrequency = 25 MHz McuGeneralConfiguration/ McuOscillatorMode = EXT_CRYSTAL_CERAMIC_RES _MODE_SEL0	`#define MCU_OSCVAL_REG_VALUE (0xAU)`

1.1.43 Macro: MCU_SYSCLK_OSCVAL

Table 43 MCU_SYSCLK_OSCVAL

Name	MCU_SYSCLK_OSCVAL	
Description	Specifies the value to be configured in Oscillator configuration register in order to enable the oscillator watchdog monitoring for SYSCLK	
Verification method	The value of macro is calculated based on following formula: Oscillator value = McuGeneralConfiguration/McuSysClkFrequency - 15	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuSysClkFrequency = 20 MHz	`#define MCU_SYSCLK_OSCVAL (0x5U)`
	McuGeneralConfiguration/McuSysClkFrequency = 25 MHz	`#define MCU_SYSCLK_OSCVAL (0xAU)`

1.1.44 Macro: MCU_OSC_CAPACITANCE_EN

Table 44 MCU_OSC_CAPACITANCE_EN

Name	MCU_OSC_CAPACITANCE_EN
Description	Specifies the oscillator value to be configured in Oscillator configuration register
Verification method	The macro is generated with a value only when configuration parameter ‘McuGeneralConfiguration/McuOscAmpRegulationEnable’ is set to ‘True’. Bit 23 is set if ‘McuGeneralConfiguration/McuOscAmpRegulationEnable’ is set to ‘True’ else 0.

	Bit 24 is set if 'McuGeneralConfiguration/McuOscCapacitance0Enable' is set to 'True' else 0. Bit 25 is set if 'McuGeneralConfiguration/McuOscCapacitance1Enable' is set to 'True' else 0. Bit 26 is set if 'McuGeneralConfiguration/McuOscCapacitance2Enable' is set to 'True' else 0. Bit 27 is set if 'McuGeneralConfiguration/McuOscCapacitance3Enable' is set to 'True' else 0. Other bits are always set to 0.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/McuOscAmpRegulationEnable = True McuGeneralConfiguration/McuOscCapacitance0Enable = False McuGeneralConfiguration/McuOscCapacitance1Enable = True McuGeneralConfiguration/McuOscCapacitance2Enable = True McuGeneralConfiguration/McuOscCapacitance3Enable = True	`#define MCU_OSC_CAPACITANCE_EN (0x0f800000U)`
	McuGeneralConfiguration/McuOscAmpRegulationEnable = False	`#define MCU_OSC_CAPACITANCE_EN (0x00000000U)`

1.1.45 Macro: MCU_OSC_MODE

Table 45 MCU_OSC_MODE

Name	MCU_OSC_MODE	
Description	Specifies the mode in which the oscillator is configured	
Verification method	The macro is generated with the suffixed value specified in the option selected in configuration parameter 'McuGeneralConfiguration/McuOscillatorMode' after '_SEL' keyword.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/McuOscillatorMode = EXT_CRYSTAL_CERAMIC_RES_MODE_SEL0	`#define MCU_OSC_MODE (0U)`
	McuGeneralConfiguration/McuOscillatorMode = EXT_INPUT_CLOCK_MODE_SEL2	`#define MCU_OSC_MODE (2U)`

1.1.46 Macro: MCU_SYSTEM_MODE_CORE

Table 46 MCU_SYSTEM_MODE_CORE

Name	MCU_SYSTEM_MODE_CORE	
Description	Specifies the CPU responsible for initiating sleep/standby modes.	
Verification method	The macro is generated with a numeric value only when any one of sleep or standby mode is configured in configuration parameter 'McuModeSettingConf/McuMode' else it is generated as 'MCU_SYSTEM_CORE_NOT_DEFINED'. When sleep/standby mode is configured the macro is generated based on the suffixed numeric value after '_SEL' keyword in configuration parameter 'McuGeneralConfiguration/ McuSystemModeCpuCore' and the value is added with 1.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/McuSystemModeCpuCore = CPU_SYSTEM_CORE1_SEL1 McuModeSettingConf/McuMode = Sleep/Standby	`#define MCU_SYSTEM_MODE_CORE (2U)`
	McuGeneralConfiguration/McuSystemModeCpuCore = CPU_SYSTEM_CORE1_SEL4 McuModeSettingConf/McuMode = Sleep/Standby	`#define MCU_SYSTEM_MODE_CORE (5U)`
	McuModeSettingConf/McuMode = Idle	`#define MCU_SYSTEM_MODE_CORE (MCU_SYSTEM_CORE_NOT_DEFINED)`

1.1.47 Macro: MCU_IDLE_MODE_CORE

Table 47 MCU_IDLE_MODE_CORE

Name	MCU_IDLE_MODE_CORE	
Description	Specifies the CPU responsible for initiating idle mode.	
Verification method	The macro is generated with a value only when idle mode is configured in configuration parameter 'McuModeSettingConf/McuMode' else it is generated as 'MCU_IDLE_CORE_NOT_DEFINED'. When idle mode is configured the macro is generated based on the suffixed numeric value after '_SEL' keyword in configuration parameter 'McuGeneralConfiguration/ McuIdleModeCpuCore'.	
Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/McuIdleModeCpuCore = CPU_IDLE_CORE0_SEL1 McuModeSettingConf/McuMode = Idle	`#define MCU_IDLE_MODE_CORE (1U)`
	McuGeneralConfiguration/McuIdleModeCpuCore = CPU_IDLE_CORE3_SEL4	`#define MCU_SYSTEM_MODE_CORE (4U)`

	McuModeSettingConf/McuMode = Idle	
	McuModeSettingConf/McuMode = Sleep/Standby	`#define MCU_IDLE_MODE_CORE (MCU_IDLE_CORE_NOT_DEFINED)`

1.1.48 Macro: MCU_NO_OF_STDBY_RAM_BLK

Table 48 MCU_NO_OF_STDBY_RAM_BLK

Name	MCU_NO_OF_STDBY_RAM_BLK	
Description	Specifies the number of standby RAM blocks configured.	
Verification method	The macro is generated with a value only when standby mode is configured in configuration parameter 'McuModeSettingConf/McuMode' else it is generated as '0'. When standby mode is configured the macro is generated based on the suffixed numeric value after '_SEL' keyword in configuration parameter 'McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeRamEnable'. If the numeric value is 1, macro is generated with a value of 1. If numeric value is 2, macro is generated with value 2. If numeric value is 4, macro is generated with value 3. If numeric value is 7, macro is generated with value 4. Other values are not configurable by the user.	
Example(s)	**Action**	**Generated output**
	McuStdbyModeSettingConf/McuStdbyModeRamEnable = MCU_STANDBYRAM_CPU0_BLK0_SEL1 McuModeSettingConf/McuMode = Standby	`#define MCU_NO_OF_STDBY_RAM_BLK (1U)`
	McuStdbyModeSettingConf/McuStdbyModeRamEnable = MCU_STANDBYRAM_CPU0_BLK0_BLK1_NONCACHED_SEL2 McuModeSettingConf/McuMode = Standby	`#define MCU_NO_OF_STDBY_RAM_BLK (2U)`
	McuStdbyModeSettingConf/McuStdbyModeRamEnable = MCU_STANDBYRAM_CPU1_BLK0_BLK1_SEL4 McuModeSettingConf/McuMode = Standby	`#define MCU_NO_OF_STDBY_RAM_BLK (3U)`
	McuStdbyModeSettingConf/McuStdbyModeRamEnable = MCU_STANDBYRAM_CPU0_CPU1_BLK0_BLK1_NONCACHED_SEL7 McuModeSettingConf/McuMo	`#define MCU_NO_OF_STDBY_RAM_BLK (4U)`

	de = Standby	
	McuModeSettingConf/McuMode = Idle/Sleep	`#define MCU_NO_OF_STDBY_RAM_BLK (0U)`

1.1.49 Macro: MCU_TRAPDIS0_RESET_VAL

Table 49 MCU_TRAPDIS0_RESET_VAL

Name	MCU_TRAPDIS0_RESET_VAL	
Description	Specifies the reset value of Trap disable configuration register for CPUs 0-3 based on availability	
Verification method	The macro is generated with a fixed value based on number of CPUs available. Bits 0-3 will be 1 for CPU0. Bits 8-11 will be 1 for CPU1 if available else these bits will be 0. Bits 16-19 will be 1 for CPU2 if available else these bits will be 0. Bits 24-27 will be 1 for CPU3 if available else these bits will be 0. Other bits are reserved and written with a value of 1.	
Example(s)	**Action**	**Generated output**
	Number of cores available =1	`#define MCU_TRAPDIS0_RESET_VAL (0xF0F0F0FFU)`
	Number of cores available =2	`#define MCU_TRAPDIS0_RESET_VAL (0xF0F0FFFFU)`
	Number of cores available =3	`#define MCU_TRAPDIS0_RESET_VAL (0xF0FFFFFFU)`
	Number of cores available =4	`#define MCU_TRAPDIS0_RESET_VAL (0xFFFFFFFFU)`

1.1.50 Macro: MCU_TRAPDIS1_RESET_VAL

Table 50 MCU_TRAPDIS1_RESET_VAL

Name	MCU_TRAPDIS1_RESET_VAL	
Description	Specifies the reset value of Trap disable configuration register for CPUs 4-5 based on availability	
Verification method	The macro is generated with a fixed value based on number of CPUs available. Bits 0-3 will be 1 for CPU4 if available else these bits will be 0. Bits 8-11 will be 1 for CPU5 if available else these bits will be 0. Bits 16-31 are reserved and written with a value of 0. Other bits are reserved and written with a value of 1.	
Example(s)	**Action**	**Generated output**
	Number of cores available =5	`#define MCU_TRAPDIS1_RESET_VAL (0x0000F0FFU)`
	Number of cores available =6	`#define MCU_TRAPDIS1_RESET_VAL (0x0000FFFFU)`

1.1.51 Macro: MCU_MCAL_SUPERVISOR

Table 51 MCU_MCAL_SUPERVISOR

Name	MCU_MCAL_SUPERVISOR	
Description	Specifies the mode of operation for MCU driver is Supervisor. *Note: This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_MCAL_SUPERVISOR (0U)`

1.1.52 Macro: MCU_MCAL_USER1

Table 52 MCU_MCAL_USER1

Name	MCU_MCAL_USER1	
Description	Specifies the mode of operation for MCU driver is User1. *Note: This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 1.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_MCAL_USER1 (1U)`

1.1.53 Macro: MCU_SYSTEM_CORE_NOT_DEFINED

Table 53 MCU_SYSTEM_CORE_NOT_DEFINED

Name	MCU_SYSTEM_CORE_NOT_DEFINED	
Description	Specifies that CPU to initiate sleep/standby modes is not defined. *Note: This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_SYSTEM_CORE_NOT_DEFINED (0U)`

1.1.54 Macro: MCU_IDLE_CORE_NOT_DEFINED

Table 54 MCU_IDLE_CORE_NOT_DEFINED

Name	MCU_IDLE_CORE_NOT_DEFINED
Description	Specifies that CPU to initiate idle mode is not defined.

	Note: *This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 7.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_IDLE_CORE_NOT_DEFINED (7U)`

1.1.55 Macro: MCU_IDLE

Table 55 MCU_IDLE

Name	MCU_IDLE	
Description	Specifies the value of idle mode. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_IDLE (0U)`

1.1.56 Macro: MCU_SLEEP

Table 56 MCU_SLEEP

Name	MCU_SLEEP	
Description	Specifies the value of sleep mode. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 1.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_SLEEP (1U)`

1.1.57 Macro: MCU_STANDBY

Table 57 MCU_STANDBY

Name	MCU_STANDBY	
Description	Specifies the value of standby mode. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 2.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_STANDBY (2U)`

1.1.58 Macro: MCU_MAX_NO_MODES

Table 58 MCU_MAX_NO_MODES

Name	MCU_MAX_NO_MODES	
Description	Specifies the maximum number of powerdown modes available (idle, sleep and standby). *Note: This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 3.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_MAX_NO_MODES (3U)`

1.1.59 Macro: MCU_ENABLE_DEM_REPORT

Table 59 MCU_ENABLE_DEM_REPORT

Name	MCU_ENABLE_DEM_REPORT	
Description	Indicates that reporting of Production errors is enabled. *Note: This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 1.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_ENABLE_DEM_REPORT (1U)`

1.1.60 Macro: MCU_DISABLE_DEM_REPORT

Table 60 MCU_DISABLE_DEM_REPORT

Name	MCU_DISABLE_DEM_REPORT	
Description	Indicates that reporting of Production errors is disabled. *Note: This macro is not configurable by the user.*	
Verification method	The macro is always generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define MCU_DISABLE_DEM_REPORT (0U)`

1.1.61 Macro: MCU_E_OSC_FAILURE_DEM_REPORT

Table 61 MCU_E_OSC_FAILURE_DEM_REPORT

Name	MCU_E_OSC_FAILURE_DEM_REPORT
Description	Enables/Disables Production error reporting for Oscillator failure.

Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_OSC_FAILURE' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure a node in McuDemEventParameterRefs Conf/ McuDemEventParameterRefs Conf_0/ MCU_E_OSC_FAILURE	`#define MCU_E_OSC_FAILURE_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefs Conf/ McuDemEventParameterRefs Conf_0/ MCU_E_OSC_FAILURE	`#define MCU_E_OSC_FAILURE_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.62 Macro: MCU_E_OSC_FAILURE

Table 62 MCU_E_OSC_FAILURE

Name	MCU_E_OSC_FAILURE	
Description	Specifies the value configured for DEM for oscillator failure.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_OSC_FAILURE'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefs Conf/ McuDemEventParameterRefs Conf_0/ MCU_E_OSC_FAILURE = DemEventParameter_0	`#define MCU_E_OSC_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefs Conf/ McuDemEventParameterRefs Conf_0/ MCU_E_OSC_FAILURE = DemEventParameter_2	`#define MCU_E_OSC_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.63 Macro: MCU_E_SYSTEM_PLL_TIMEOUT_ERR_DEM_REPORT

Table 63 MCU_E_SYSTEM_PLL_TIMEOUT_ERR_DEM_REPORT

Name	MCU_E_SYSTEM_PLL_TIMEOUT_ERR_DEM_REPORT
Description	Enables/Disables Production error reporting for System pll locking timeout.

Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_TIMEOUT_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure a node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_TIMEOUT_ERR	`#define MCU_E_SYSTEM_PLL_TIMEOUT_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_TIMEOUT_ERR	`#define MCU_E_SYSTEM_PLL_TIMEOUT_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.64 Macro: MCU_E_SYSTEM_PLL_TIMEOUT_ERR

Table 64 MCU_E_SYSTEM_PLL_TIMEOUT_ERR

Name	MCU_E_SYSTEM_PLL_TIMEOUT_ERR	
Description	Specifies the value configured for DEM for system pll locking timeout error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_TIMEOUT_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_TIMEOUT_ERR = DemEventParameter_0	`#define MCU_E_SYSTEM_PLL_TIMEOUT_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_TIMEOUT_ERR = DemEventParameter_2	`#define MCU_E_SYSTEM_PLL_TIMEOUT_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.64.1 Macro: MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR_DEM_REPORT

Table 65 MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR_DEM_REPORT

Name	MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for peripheral pll locking timeout.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure a node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR	`#define MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR	`#define MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.65 Macro: MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR

Table 66 MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR

Name	MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR	
Description	Specifies the value configured for DEM for peripheral pll locking timeout error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR = DemEventParameter_0	`#define MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/	`#define MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR (DemConf_DemEventParameter_DemEventPa`

	McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_TIMEOUT_ERR = DemEventParameter_2	rameter_2)

1.1.66 Macro: MCU_E_SYSTEM_PLL_LOCK_LOSS_DEM_REPORT

Table 67 MCU_E_SYSTEM_PLL_LOCK_LOSS_DEM_REPORT

Name	MCU_E_SYSTEM_PLL_LOCK_LOSS_DEM_REPORT	
Description	Enables/Disables Production error reporting for system pll lock loss.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_LOCK_LOSS' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure a node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_LOCK_LOSS	`#define MCU_E_SYSTEM_PLL_LOCK_LOSS_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure a node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_LOCK_LOSS	`#define MCU_E_SYSTEM_PLL_LOCK_LOSS_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.67 Macro: MCU_E_SYSTEM_PLL_LOCK_LOSS

Table 68 MCU_E_SYSTEM_PLL_LOCK_LOSS

Name	MCU_E_SYSTEM_PLL_LOCK_LOSS	
Description	Specifies the value configured for DEM for system pll lock loss.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_LOCK_LOSS'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/	`#define MCU_E_SYSTEM_PLL_LOCK_LOSS (DemConf_DemEventParameter_DemEventParameter_0)`

	MCU_E_SYSTEM_PLL_LOCK_LOSS= DemEventParameter_0	
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_SYSTEM_PLL_LOCK_LOSS = DemEventParameter_2	`#define MCU_E_SYSTEM_PLL_LOCK_LOSS (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.68 Macro: MCU_E_PERIPHERAL_PLL_LOCK_LOSS_DEM_REPORT

Table 69 MCU_E_PERIPHERAL_PLL_LOCK_LOSS_DEM_REPORT

Name	MCU_E_PERIPHERAL_PLL_LOCK_LOSS_DEM_REPORT	
Description	Enables/Disables Production error reporting for peripheral pll lock loss.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_LOCK_LOSS' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure a node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_LOCK_LOSS	`#define MCU_E_PERIPHERAL_PLL_LOCK_LOSS_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_LOCK_LOSS	`#define MCU_E_PERIPHERAL_PLL_LOCK_LOSS_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.69 Macro: MCU_E_PERIPHERAL_PLL_LOCK_LOSS

Table 70 MCU_E_PERIPHERAL_PLL_LOCK_LOSS

Name	MCU_E_PERIPHERAL_PLL_LOCK_LOSS
Description	Specifies the value configured for DEM for peripheral pll lock loss.
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_LOCK_LOSS'

Example(s)	Action	Generated output
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_LOCK_LOSS= DemEventParameter_0	`#define MCU_E_PERIPHERAL_PLL_LOCK_LOSS (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PERIPHERAL_PLL_LOCK_LOSS = DemEventParameter_2	`#define MCU_E_PERIPHERAL_PLL_LOCK_LOSS (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.70 Macro: MCU_E_GTM_CLC_ENABLE_ERR_DEM_REPORT

Table 71 MCU_E_GTM_CLC_ENABLE_ERR_DEM_REPORT

Name	MCU_E_GTM_CLC_ENABLE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for GTM CLC enabling error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_ENABLE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_ENABLE_ERR	`#define MCU_E_GTM_CLC_ENABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_ENABLE_ERR	`#define MCU_E_GTM_CLC_ENABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.71 Macro: MCU_E_GTM_CLC_ENABLE_ERR

Table 72 MCU_E_GTM_CLC_ENABLE_ERR

Name	MCU_E_GTM_CLC_ENABLE_ERR

Description	Specifies the value configured for DEM for GTM CLC enabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_ENABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_ENABLE_ERR= DemEventParameter_0	`#define MCU_E_GTM_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_ENABLE_ERR = DemEventParameter_2	`#define MCU_E_GTM_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.72 Macro: MCU_E_GTM_CLC_DISABLE_ERR_DEM_REPORT

Table 73 MCU_E_GTM_CLC_DISABLE_ERR_DEM_REPORT

Name	MCU_E_GTM_CLC_DISABLE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for GTM CLC disabling error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_DISABLE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_DISABLE_ERR	`#define MCU_E_GTM_CLC_DISABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_DISABLE_ERR	`#define MCU_E_GTM_CLC_DISABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.73 Macro: MCU_E_GTM_CLC_DISABLE_ERR

Table 74 MCU_E_GTM_CLC_DISABLE_ERR

Name	MCU_E_GTM_CLC_DISABLE_ERR	
Description	Specifies the value configured for DEM for GTM CLC disabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_DISABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_DISABLE_ERR= DemEventParameter_0	`#define MCU_E_GTM_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GTM_CLC_DISABLE_ERR = DemEventParameter_2	`#define MCU_E_GTM_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.74 Macro: MCU_E_CONVCTRL_CLC_ENABLE_ERR_DEM_REPORT

Table 75 MCU_E_CONVCTRL_CLC_ENABLE_ERR_DEM_REPORT

Name	MCU_E_CONVCTRL_CLC_ENABLE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for converter control CLC enabling error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_ENABLE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_ENABLE_ERR	`#define MCU_E_CONVCTRL_CLC_ENABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefs	`#define MCU_E_CONVCTRL_CLC_ENABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

	Conf_0/ MCU_E_CONVCTRL_CLC_ENABLE_ERR	

1.1.75 Macro: MCU_E_CONVCTRL_CLC_ENABLE_ERR

Table 76 MCU_E_CONVCTRL_CLC_ENABLE_ERR

Name	MCU_E_CONVCTRL_CLC_ENABLE_ERR	
Description	Specifies the value configured for DEM for converter control CLC enabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_ENABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_ENABLE_ERR= DemEventParameter_0	`#define MCU_E_CONVCTRL_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_ENABLE_ERR = DemEventParameter_2	`#define MCU_E_CONVCTRL_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.76 Macro: MCU_E_CONVCTRL_CLC_DISABLE_ERR_DEM_REPORT

Table 77 MCU_E_CONVCTRL_CLC_DISABLE_ERR_DEM_REPORT

Name	MCU_E_CONVCTRL_CLC_DISABLE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for converter control CLC disabling error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_DISABLE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_DIS	`#define MCU_E_CONVCTRL_CLC_DISABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`

	ABLE_ERR	
	Don't configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_DISABLE_ERR	`#define MCU_E_CONVCTRL_CLC_DISABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.77 Macro: MCU_E_CONVCTRL_CLC_DISABLE_ERR

Table 78 MCU_E_CONVCTRL_CLC_DISABLE_ERR

Name	MCU_E_CONVCTRL_CLC_DISABLE_ERR	
Description	Specifies the value configured for DEM for converter control CLC disabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_DISABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_DISABLE_ERR= DemEventParameter_0	`#define MCU_E_CONVCTRL_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CONVCTRL_CLC_DISABLE_ERR = DemEventParameter_2	`#define MCU_E_CONVCTRL_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.78 Macro: MCU_E_CCUCON_UPDATE_ERR_DEM_REPORT

Table 79 MCU_E_CCUCON_UPDATE_ERR_DEM_REPORT

Name	MCU_E_CCUCON_UPDATE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for ccucon register update error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCUCON_UPDATE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in	`#define`

	McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCUCON_UPDATE_ERR	`MCU_E_CCUCON_UPDATE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure a node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCUCON_UPDATE_ERR	`#define MCU_E_CCUCON_UPDATE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.79 Macro: MCU_E_CCUCON_UPDATE_ERR

Table 80 MCU_E_CCUCON_UPDATE_ERR

Name	MCU_E_CCUCON_UPDATE_ERR	
Description	Specifies the value configured for DEM for ccucon register update error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCUCON_UPDATE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCUCON_UPDATE_ERR = DemEventParameter_0	`#define MCU_E_CCUCON_UPDATE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCUCON_UPDATE_ERR = DemEventParameter_2	`#define MCU_E_CCUCON_UPDATE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.80 Macro: MCU_E_CCU6_CLC_ENABLE_ERR_DEM_REPORT

Table 81 MCU_E_CCU6_CLC_ENABLE_ERR_DEM_REPORT

Name	MCU_E_CCU6_CLC_ENABLE_ERR_DEM_REPORT
Description	Enables/Disables Production error reporting for CCU6 CLC enabling error.
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_ENABLE_ERR' else it is

	generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_ENABLE_ERR	`#define MCU_E_CCU6_CLC_ENABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_ENABLE_ERR	`#define MCU_E_CCU6_CLC_ENABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.81 Macro: MCU_E_CCU6_CLC_ENABLE_ERR

Table 82 MCU_E_CCU6_CLC_ENABLE_ERR

Name	MCU_E_CCU6_CLC_ENABLE_ERR	
Description	Specifies the value configured for DEM for CCU6 CLC enabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_ENABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_ENABLE_ERR= DemEventParameter_0	`#define MCU_E_CCU6_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_ENABLE_ERR = DemEventParameter_2	`#define MCU_E_CCU6_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.82 Macro: MCU_E_CCU6_CLC_DISABLE_ERR_DEM_REPORT

Table 83 MCU_E_CCU6_CLC_DISABLE_ERR_DEM_REPORT

Name	MCU_E_CCU6_CLC_DISABLE_ERR_DEM_REPORT

Description	Enables/Disables Production error reporting for CCU6 CLC disabling error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_DISABLE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_DISABLE_ERR	`#define MCU_E_CCU6_CLC_DISABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_DISABLE_ERR	`#define MCU_E_CCU6_CLC_DISABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.83 Macro: MCU_E_CCU6_CLC_DISABLE_ERR

Table 84 MCU_E_CCU6_CLC_DISABLE_ERR

Name	MCU_E_CCU6_CLC_DISABLE_ERR	
Description	Specifies the value configured for DEM for CCU6 CLC disabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_DISABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_DISABLE_ERR= DemEventParameter_0	`#define MCU_E_CCU6_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_CCU6_CLC_DISABLE_ERR = DemEventParameter_2	`#define MCU_E_CCU6_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.84 Macro: MCU_E_GPT12_CLC_ENABLE_ERR_DEM_REPORT

Table 85 MCU_E_GPT12_CLC_ENABLE_ERR_DEM_REPORT

Name	MCU_E_GPT12_CLC_ENABLE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for GPT12 CLC enabling error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_ENABLE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_ENABLE_ERR	`#define MCU_E_GPT12_CLC_ENABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_ENABLE_ERR	`#define MCU_E_GPT12_CLC_ENABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.85 Macro: MCU_E_GPT12_CLC_ENABLE_ERR

Table 86 MCU_E_GPT12_CLC_ENABLE_ERR

Name	MCU_E_GPT12_CLC_ENABLE_ERR	
Description	Specifies the value configured for DEM for GPT12 CLC enabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_ENABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_ENABLE_ERR= DemEventParameter_0	`#define MCU_E_GPT12_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefs	`#define MCU_E_GPT12_CLC_ENABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

	Conf_0/ MCU_E_GPT12_CLC_ENABLE_ERR = DemEventParameter_2	

1.1.86 Macro: MCU_E_GPT12_CLC_DISABLE_ERR_DEM_REPORT

Table 87 MCU_E_GPT12_CLC_DISABLE_ERR_DEM_REPORT

Name	MCU_E_GPT12_CLC_DISABLE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for GPT12 CLC disabling error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_DISABLE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_DISABLE_ERR	`#define MCU_E_GPT12_CLC_DISABLE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_DISABLE_ERR	`#define MCU_E_GPT12_CLC_DISABLE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.87 Macro: MCU_E_GPT12_CLC_DISABLE_ERR

Table 88 MCU_E_GPT12_CLC_DISABLE_ERR

Name	MCU_E_GPT12_CLC_DISABLE_ERR	
Description	Specifies the value configured for DEM for GPT12 CLC disabling error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_DISABLE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_DISABLE	`#define MCU_E_GPT12_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_0)`

	_ERR= DemEventParameter_0	
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_GPT12_CLC_DISABLE_ERR = DemEventParameter_2	`#define MCU_E_GPT12_CLC_DISABLE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.88 Macro: MCU_E_PMSWCR_UPDATE_ERR_DEM_REPORT

Table 89 MCU_E_PMSWCR_UPDATE_ERR_DEM_REPORT

Name	MCU_E_PMSWCR_UPDATE_ERR_DEM_REPORT	
Description	Enables/Disables Production error reporting for PMSWCR register update error.	
Verification method	The macro is generated as 'MCU_ENABLE_DEM_REPORT' if a node exists in the configuration 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PMSWCR_UPDATE_ERR' else it is generated as 'MCU_DISABLE_DEM_REPORT'	
Example(s)	**Action**	**Generated output**
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PMSWCR_UPDATE_ERR	`#define MCU_E_PMSWCR_UPDATE_ERR_DEM_REPORT (MCU_ENABLE_DEM_REPORT)`
	Don't configure any node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PMSWCR_UPDATE_ERR	`#define MCU_E_PMSWCR_UPDATE_ERR_DEM_REPORT (MCU_DISABLE_DEM_REPORT)`

1.1.89 Macro: MCU_E_PMSWCR _UPDATE_ERR

Table 90 MCU_E_PMSWCR_UPDATE_ERR

Name	MCU_E_PMSWCR_UPDATE_ERR	
Description	Specifies the value configured for DEM for PMSWCR register update error.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PMSWCR_UPDATE_ERR'	
Example(s)	**Action**	**Generated output**
	Configure the node in	`#define MCU_E_PMSWCR_UPDATE_ERR`

	McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PMSWCR_UPDATE_ERR = DemEventParameter_0	`(DemConf_DemEventParameter_DemEventParameter_0)`
	Configure the node in McuDemEventParameterRefsConf/ McuDemEventParameterRefsConf_0/ MCU_E_PMSWCR_UPDATE_ERR = DemEventParameter_2	`#define MCU_E_PMSWCR_UPDATE_ERR (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.90 Macro: McuConf_McuModeSettingConf_McuModeSettingConf_0

Table 91 McuConf_McuModeSettingConf_McuModeSettingConf_0

Name	McuConf_McuModeSettingConf_McuModeSettingConf_0	
Description	Specifies the the container name of the McuModeSettingConfiguration. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuModeSettingConf_McuModeSettingConf_0 (0U)`

1.1.91 Macro: McuConf_McuModeSettingConf_McuModeSettingConf_1

Table 92 McuConf_McuModeSettingConf_McuModeSettingConf_1

Name	McuConf_McuModeSettingConf_McuModeSettingConf_1	
Description	Specifies the container name of the McuModeSettingConfiguration. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 2.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuModeSettingConf_McuModeSettingConf_1 (2U)`

1.1.92 Macro: McuConf_McuClockSettingConfig_McuClockSettingConfig_0

Table 93 McuConf_McuClockSettingConfig_McuClockSettingConfig_0

Name	McuConf_McuClockSettingConfig_McuClockSettingConfig_0

Description	Specifies the container name of the McuClockSettingConfiguration. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuClockSettingConfig_McuClockSettingConfig_0 (0U)`

1.1.93 Macro: McuConf_McuRamSectorSettingConf_McuRamSectorSettingConf_0

Table 94 McuConf_McuRamSectorSettingConf_McuRamSectorSettingConf_0

Name	McuConf_McuRamSectorSettingConf_McuRamSectorSettingConf_0	
Description	Specifies the container name of the McuRamSectorSettingConfiguration. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuRamSectorSettingConf_McuRamSectorSettingConf_0 (0U)`

1.1.94 Macro: McuConf_McuResetReasonConf_MCU_ESR0_RESET

Table 95 McuConf_McuResetReasonConf_MCU_ESR0_RESET

Name	McuConf_McuResetReasonConf_MCU_ESR0_RESET	
Description	Specifies the value of ESR0 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 0.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_ESR0_RESET (0U)`

1.1.95 Macro: McuConf_McuResetReasonConf_MCU_ESR1_RESET

Table 96 McuConf_McuResetReasonConf_MCU_ESR1_RESET

Name	McuConf_McuResetReasonConf_MCU_ESR1_RESET
Description	Specifies the value of ESR1 reset.

	Note: *This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 1.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_ESR1_RESET (1U)`

1.1.96 Macro: McuConf_McuResetReasonConf_MCU_SMU_RESET

Table 97 McuConf_McuResetReasonConf_MCU_SMU_RESET

Name	McuConf_McuResetReasonConf_MCU_SMU_RESET	
Description	Specifies the value of SMU reset. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 2.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_SMU_RESET (2U)`

1.1.97 Macro: McuConf_McuResetReasonConf_MCU_SW_RESET

Table 98 McuConf_McuResetReasonConf_MCU_SW_RESET

Name	McuConf_McuResetReasonConf_MCU_SW_RESET	
Description	Specifies the value of SW reset. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 3.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_SW_RESET (3U)`

1.1.98 Macro: McuConf_McuResetReasonConf_MCU_STM0_RESET

Table 99 McuConf_McuResetReasonConf_MCU_STM0_RESET

Name	McuConf_McuResetReasonConf_MCU_STM0_RESET
Description	Specifies the value of STM0 reset. *Note:* *This macro is not configurable by the user.*

Verification method	The macro if already not defined is generated with a value of 4.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_STM0_RESET (4U)`

1.1.99 Macro: McuConf_McuResetReasonConf_MCU_STM1_RESET

Table 100 McuConf_McuResetReasonConf_MCU_STM1_RESET

Name	McuConf_McuResetReasonConf_MCU_STM1_RESET	
Description	Specifies the value of STM1 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 5.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_STM1_RESET (5U)`

1.1.100 Macro: McuConf_McuResetReasonConf_MCU_STM2_RESET

Table 101 McuConf_McuResetReasonConf_MCU_STM2_RESET

Name	McuConf_McuResetReasonConf_MCU_STM2_RESET	
Description	Specifies the value of STM2 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 6.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_STM2_RESET (6U)`

1.1.101 Macro: McuConf_McuResetReasonConf_MCU_STM3_RESET

Table 102 McuConf_McuResetReasonConf_MCU_STM3_RESET

Name	McuConf_McuResetReasonConf_MCU_STM3_RESET	
Description	Specifies the value of STM3 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 7.	
Example(s)	**Action**	**Generated output**

	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_STM3_RESET (7U)`

1.1.102 Macro: McuConf_McuResetReasonConf_MCU_STM4_RESET

Table 103 McuConf_McuResetReasonConf_MCU_STM4_RESET

Name	McuConf_McuResetReasonConf_MCU_STM4_RESET	
Description	Specifies the value of STM4 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 8.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_STM4_RESET (8U)`

1.1.103 Macro: McuConf_McuResetReasonConf_MCU_STM5_RESET

Table 104 McuConf_McuResetReasonConf_MCU_STM5_RESET

Name	McuConf_McuResetReasonConf_MCU_STM5_RESET	
Description	Specifies the value of STM5 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 9.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_STM5_RESET (9U)`

1.1.104 Macro: McuConf_McuResetReasonConf_MCU_POWER_ON_RESET

Table 105 McuConf_McuResetReasonConf_MCU_POWER_ON_RESET

Name	McuConf_McuResetReasonConf_MCU_POWER_ON_RESET	
Description	Specifies the value of power on reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 10.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_POWER_ON_RESET (10U)`

1.1.105 Macro: McuConf_McuResetReasonConf_MCU_CB0_RESET

Table 106 McuConf_McuResetReasonConf_MCU_CB0_RESET

Name	McuConf_McuResetReasonConf_MCU_CB0_RESET	
Description	Specifies the value of Cerberus 0 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 11.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_CB0_RESET (11U)`

1.1.106 Macro: McuConf_McuResetReasonConf_MCU_CB1_RESET

Table 107 McuConf_McuResetReasonConf_MCU_CB1_RESET

Name	McuConf_McuResetReasonConf_MCU_CB1_RESET	
Description	Specifies the value of Cerberus 1 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 12.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_CB1_RESET (12U)`

1.1.107 Macro: McuConf_McuResetReasonConf_MCU_CB3_RESET

Table 108 McuConf_McuResetReasonConf_MCU_CB3_RESET

Name	McuConf_McuResetReasonConf_MCU_CB3_RESET	
Description	Specifies the value of Cerberus 3 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 13.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_CB3_RESET (13U)`

1.1.108 Macro: McuConf_McuResetReasonConf_MCU_EVRC_RESET

Table 109 McuConf_McuResetReasonConf_MCU_EVRC_RESET

Name	McuConf_McuResetReasonConf_MCU_EVRC_RESET	
Description	Specifies the value of EVRC reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 14.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_EVRC_RESET (14U)`

1.1.109 Macro: McuConf_McuResetReasonConf_MCU_EVR33_RESET

Table 110 McuConf_McuResetReasonConf_MCU_EVR33_RESET

Name	McuConf_McuResetReasonConf_MCU_EVR33_RESET	
Description	Specifies the value of EVR33 reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 15.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_EVR33_RESET (15U)`

1.1.110 Macro: McuConf_McuResetReasonConf_MCU_SUPPLY_WDOG_RESET

Table 111 McuConf_McuResetReasonConf_MCU_SUPPLY_WDOG_RESET

Name	McuConf_McuResetReasonConf_MCU_SUPPLY_WDOG_RESET	
Description	Specifies the value of supply watchdog reset. *Note: This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 16.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_SUPPLY_WDOG_RESET (16U)`

1.1.111 Macro: McuConf_McuResetReasonConf_MCU_STBYR_RESET

Table 112 McuConf_McuResetReasonConf_MCU_STBYR_RESET

Name	McuConf_McuResetReasonConf_MCU_STBYR_RESET
Description	Specifies the value of STBYR reset.

	Note: *This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 17.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_STBYR_RESET (17U)`

1.1.112 Macro: McuConf_McuResetReasonConf_MCU_LBIST_RESET

Table 113 McuConf_McuResetReasonConf_MCU_LBIST_RESET

Name	McuConf_McuResetReasonConf_MCU_LBIST_RESET	
Description	Specifies the value of LBIST reset. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 18.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_EVR33_RESET (18U)`

1.1.113 Macro: McuConf_McuResetReasonConf_MCU_ RESET_MULTIPLE

Table 114 McuConf_McuResetReasonConf_MCU _RESET_MULTIPLE

Name	McuConf_McuResetReasonConf_MCU_RESET_MULTIPLE	
Description	Specifies the value of multiple resets. *Note:* *This macro is not configurable by the user.*	
Verification method	The macro if already not defined is generated with a value of 254.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_RESET_MULTIPLE (254U)`

1.1.114 Macro: McuConf_McuResetReasonConf_MCU_ RESET_UNDEFINED

Table 115 McuConf_McuResetReasonConf_MCU _RESET_UNDEFINED

Name	McuConf_McuResetReasonConf_MCU_RESET_UNDEFINED
Description	Specifies the value of undefined reset reason. *Note:* *This macro is not configurable by the user.*

Verification method	The macro if already not defined is generated with a value of 255.	
Example(s)	**Action**	**Generated output**
	Generate Mcu_Cfg.h	`#define McuConf_McuResetReasonConf_MCU_RESET_UNDEFINED (255U)`

1.2 File: Mcu_17_TimerIp_Cfg.h

The generated header file contains all pre-compile configuration parameters. Pre-compile time configuration allows decoupling of the static configuration from implementation. The file is generated in 'inc' folder.

1.2.1 Macro: MCU_17_GTM_NO_OF_TIM_MODULES

Table 116 MCU_17_GTM_NO_OF_TIM_MODULES

Name	MCU_17_GTM_NO_OF_TIM_MODULES	
Description	Specifies the number of TIM modules available in GTM. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of TIM modules available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of TIMs available = 8	`#define MCU_17_GTM_NO_OF_TIM_MODULES (8U)`
	Number of TIMs available = 12	`#define MCU_17_GTM_NO_OF_TIM_MODULES (12U)`

1.2.2 Macro: MCU_17_GTM_NO_OF_TIM_CHANNELS

Table 117 MCU_17_GTM_NO_OF_TIM_CHANNELS

Name	MCU_17_GTM_NO_OF_TIM_CHANNELS	
Description	Specifies the number of TIM channels available per TIM module inside GTM. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of TIM channels per TIM module available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of TIM channels available = 8	`#define MCU_17_GTM_NO_OF_TIM_CHANNELS (8U)`
	Number of TIMs available = 4	`#define MCU_17_GTM_NO_OF_TIM_CHANNELS (4U)`

1.2.3 Macro: MCU_17_GTM_NO_OF_TOM_MODULES

Table 118 MCU_17_GTM_NO_OF_TOM_MODULES

Name	MCU_17_GTM_NO_OF_TOM_MODULES
Description	Specifies the number of TOM modules available inside GTM.

	Note: *This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of TOM modules available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of TOM modules available = 5	`#define MCU_17_GTM_NO_OF_TOM_MODULES (5U)`
	Number of TOM modules available = 12	`#define MCU_17_GTM_NO_OF_TOM_MODULES (12U)`

1.2.4 Macro: MCU_17_GTM_NO_OF_TOM_CHANNELS

Table 119 MCU_17_GTM_NO_OF_TOM_CHANNELS

Name	MCU_17_GTM_NO_OF_TOM_CHANNELS	
Description	Specifies the number of TOM channels available per TOM module inside GTM. *Note:* *This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of TOM channels per TOM module available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of TOM channels available = 5	`#define MCU_17_GTM_NO_OF_TOM_CHANNELS (5U)`
	Number of TOM channels available = 8	`#define MCU_17_GTM_NO_OF_TOM_CHANNELS (8U)`

1.2.5 Macro: MCU_17_GTM_NO_OF_TOM_TGC

Table 120 MCU_17_GTM_NO_OF_TOM_TGC

Name	MCU_17_GTM_NO_OF_TOM_TGC	
Description	Specifies the number of TOM global control registers available per TOM module inside GTM. *Note:* *This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of TOM TGCs per TOM module available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of TOM TGCs available = 1	`#define MCU_17_GTM_NO_OF_TOM_TGC (1U)`
	Number of TOM TGCs available = 2	`#define MCU_17_GTM_NO_OF_TOM_TGC (2U)`

1.2.6 Macro: MCU_17_GTM_NO_OF_ATOM_MODULES

Table 121 MCU_17_GTM_NO_OF_ATOM_MODULES

Name	MCU_17_GTM_NO_OF_ATOM_MODULES	
Description	Specifies the number of ATOM modules available inside GTM. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of ATOM modules available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of ATOM modules available = 5	`#define MCU_17_GTM_NO_OF_ATOM_MODULES (5U)`
	Number of ATOM modules available = 12	`#define MCU_17_GTM_NO_OF_ATOM_MODULES (12U)`

1.2.7 Macro: MCU_17_GTM_NO_OF_ATOM_CHANNELS

Table 122 MCU_17_GTM_NO_OF_ATOM_CHANNELS

Name	MCU_17_GTM_NO_OF_ATOM_CHANNELS	
Description	Specifies the number of ATOM channels available per ATOM module inside GTM. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of ATOM channels per ATOM module available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of ATOM channels available = 5	`#define MCU_17_GTM_NO_OF_ATOM_CHANNELS (5U)`
	Number of ATOM channels available = 8	`#define MCU_17_GTM_NO_OF_ATOM_CHANNELS (8U)`

1.2.8 Macro: MCU_17_GTM_NO_OF_ATOM_AGC

Table 123 MCU_17_GTM_NO_OF_ATOM_AGC

Name	MCU_17_GTM_NO_OF_ATOM_AGC	
Description	Specifies the number of ATOM global control registers available per ATOM module inside GTM. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of ATOM AGCs per ATOM module available in GTM.	
Example(s)	**Action**	**Generated output**
	Number of ATOM AGCs	`#define MCU_17_GTM_NO_OF_ATOM_AGC`

available = 1	`(1U)`
Number of ATOM AGCs available = 2	`#define MCU_17_GTM_NO_OF_ATOM_AGC (2U)`

1.2.9 Macro: MCU_17_CCU6_NO_OF_KERNELS

Table 124 MCU_17_CCU6_NO_OF_KERNELS

Name	MCU_17_CCU6_NO_OF_KERNELS	
Description	Specifies the number of CCU6 kernels available in the device. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of CCU6 kernels available in device.	
Example(s)	**Action**	**Generated output**
	Number of CCU6 kernels available = 1	`#define MCU_17_CCU6_NO_OF_KERNELS (1U)`
	Number of CCU6 kernels available = 4	`#define MCU_17_CCU6_NO_OF_KERNELS (4U)`

1.2.10 Macro: MCU_17_CCU6_NO_OF_COMPARATORS

Table 125 MCU_17_CCU6_NO_OF_COMPARATORS

Name	MCU_17_CCU6_NO_OF_COMPARATORS	
Description	Specifies the number of CCU6 comparators available in the device. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of CCU6 comparators available in device.	
Example(s)	**Action**	**Generated output**
	Number of CCU6 comparators available = 1	`#define MCU_17_CCU6_NO_OF_COMPARATORS (1U)`
	Number of CCU6 comparators available = 4	`#define MCU_17_CCU6_NO_OF_COMPARATORS (4U)`

1.2.11 Macro: MCU_17_GPT12_NO_OF_TIMERS

Table 126 MCU_17_GPT12_NO_OF_TIMERS

Name	MCU_17_GPT12_NO_OF_TIMERS
Description	Specifies the number of GPT12 timers available in the device. *Note: This macro is not configurable by the user*
Verification method	The macro is generated based on the number of GPT12 timers available in device.

Example(s)	Action	Generated output
	Number of GPT12 timers available = 3	`#define MCU_17_GPT12_NO_OF_TIMERS (3U)`
	Number of GPT12 timers available = 5	`#define MCU_17_GPT12_NO_OF_TIMERS (5U)`

1.2.12 Macro: MCU_17_ERU_NO_OF_OGU

Table 127 MCU_17_ERU_NO_OF_OGU

Name	MCU_17_ERU_NO_OF_OGU	
Description	Specifies the number of ERU output gating units available in device. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of ERU OGUs available in device.	
Example(s)	**Action**	**Generated output**
	Number of OGUs available = 3	`#define MCU_17_ERU_NO_OF_OGU (3U)`
	Number of OGUs available = 5	`#define MCU_17_ERU_NO_OF_OGU (5U)`

1.2.13 Macro: MCU_17_STM_NO_OF_TIMERS

Table 128 MCU_17_STM_NO_OF_TIMERS

Name	MCU_17_STM_NO_OF_TIMERS	
Description	Specifies the number of STM modules available in device. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on the number of STMs available in device.	
Example(s)	**Action**	**Generated output**
	Number of STMs available = 3	`#define MCU_17_STM_NO_OF_TIMERS (3U)`
	Number of STMs available = 5	`#define MCU_17_STM_NO_OF_TIMERS (5U)`

1.2.14 Macro: MCU_17_TIMERIP_ADC_USER

Table 129 MCU_17_TIMERIP_ADC_USER

Name	MCU_17_TIMERIP_ADC_USER
Description	Indicates if ADC has reserved any resources available in McuHardwareResourceAllocationConf. *Note: This macro is not configurable by the user*

Verification method	The macro is generated as STD_ON when ADC has reserved any one of GTM, ERU or CCU6 else it is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	ADC has reserved GTM, ERU or CCU6 resource	`#define MCU_17_TIMERIP_ADC_USER (STD_ON)`
	ADC doesn't have any resource reserved	`#define MCU_17_TIMERIP_ADC_USER (STD_OFF)`

1.2.15 Macro: MCU_17_TIMERIP_WDG_USER

Table 130 MCU_17_TIMERIP_WDG_USER

Name	MCU_17_TIMERIP_WDG_USER	
Description	Indicates if WDG has reserved any resources available in McuHardwareResourceAllocationConf. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated as STD_ON when WDG has reserved any one of GTM or STM else it is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	WDG has reserved GTM or STM resource	`#define MCU_17_TIMERIP_WDG_USER (STD_ON)`
	WDG doesn't have any resource reserved	`#define MCU_17_TIMERIP_WDG_USER (STD_OFF)`

1.2.16 Macro: MCU_17_TIMERIP_PWM_USER

Table 131 MCU_17_TIMERIP_PWM_USER

Name	MCU_17_TIMERIP_PWM_USER	
Description	Indicates if PWM has reserved any resources available in McuHardwareResourceAllocationConf. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated as STD_ON when PWM has reserved any one of GTM or CCU6 else it is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	PWM has reserved GTM or CCU6 resource	`#define MCU_17_TIMERIP_PWM_USER (STD_ON)`
	PWM doesn't have any resource reserved	`#define MCU_17_TIMERIP_PWM_USER (STD_OFF)`

1.2.17 Macro: MCU_17_TIMERIP_GPT_USER

Table 132 MCU_17_TIMERIP_GPT_USER

Name	MCU_17_TIMERIP_GPT_USER	
Description	Indicates if GPT has reserved any resources available in McuHardwareResourceAllocationConf. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated as STD_ON when GPT has reserved GTM else it is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	GPT has reserved GTM resource	`#define MCU_17_TIMERIP_GPT_USER (STD_ON)`
	GPT doesn't have any resource reserved	`#define MCU_17_TIMERIP_GPT_USER (STD_OFF)`

1.2.18 Macro: MCU_17_TIMERIP_OCU_USER

Table 133 MCU_17_TIMERIP_OCU_USER

Name	MCU_17_TIMERIP_OCU_USER	
Description	Indicates if OCU has reserved any resources available in McuHardwareResourceAllocationConf. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated as STD_ON when OCU has reserved any one of GTM or ERU else it is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	OCU has reserved GTM or ERU resource	`#define MCU_17_TIMERIP_OCU_USER (STD_ON)`
	OCU doesn't have any resource reserved	`#define MCU_17_TIMERIP_OCU_USER (STD_OFF)`

1.2.19 Macro: MCU_17_TIMERIP_ICU_USER

Table 134 MCU_17_TIMERIP_ICU_USER

Name	MCU_17_TIMERIP_ICU_USER
Description	Indicates if ICU has reserved any resources available in McuHardwareResourceAllocationConf. *Note: This macro is not configurable by the user*
Verification method	The macro is generated as STD_ON when ICU has reserved any one of GTM, CCU6,

	GPT12 or ERU else it is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	ICU has reserved GTM, CCU6, GPT12 or ERU resource	`#define MCU_17_TIMERIP_ICU_USER (STD_ON)`
	ICU doesn't have any resource reserved	`#define MCU_17_TIMERIP_ICU_USER (STD_OFF)`

1.2.20 Macro: MCU_17_TIMERIP_STM_USER

Table 135 MCU_17_TIMERIP_STM_USER

Name	MCU_17_TIMERIP_STM_USER	
Description	Indicates if STM driver has reserved any resources available in McuHardwareResourceAllocationConf. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated as STD_ON when STM driver has reserved STM else it is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	STM has reserved STM resource	`#define MCU_17_TIMERIP_STM_USER (STD_ON)`
	STM doesn't have any resource reserved	`#define MCU_17_TIMERIP_STM_USER (STD_OFF)`

1.3 File: Mcu [_<variant>]_PBcfg.c

The generated source file contains all post-build configuration parameters. Post-build time configuration mechanism allows configurable functionality of MCU driver that is deployed as object code. The file is generated in 'src' folder.

1.3.1 Structure: Mcu_Config[_<variant>]

Table 136 Mcu_Config[_<variant>]

Name	Mcu_Config[_<variant>]	
Type	Mcu_ConfigType	
Description	Root configuration structure of MCU driver which will be used during initialization.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**

	Configure MCU (variant unaware)	`const Mcu_ConfigType Mcu_Config =`
		`{`
		`  /*McuModuleConfiguration*/`
		`  /* MCU clock Configuration Pointer*/`
		`  Mcu_kClockConfiguration_Config,`
		`  /* Ram Section configuration Pointer*/`
		`  NULL_PTR,`
		`  #if (MCU_GTM_USED == STD_ON)`
		`  /* GTM Global Configuration Pointer*/`
		`  &Mcu_kGtmConfiguration_Config,`
		`  #endif`
		`  /*Ptr to GPT12 Prescaler config structure */`
		`  #if ((MCU_GPT1_USED == STD_ON) || (MCU_GPT2_USED == STD_ON))`
		`&Mcu_kGpt12PrescalerConfiguration_Config,`
		`  #endif`
		`  /* Ptr to Standby Mode in config structure */`
		`&Mcu_kLowPowerModeConfiguration_Config,`
		`  /* Reset configuration */`

	0x00000000U, /* Application Reset Disable configuration */ 0x00000000U, /* Trap configuration */ 0xffffffffU, 0x0000ffffU, /*Eru global input filter configuration */ 0x00000000U, /* Total number of Clock settings */ ((Mcu_ClockType)1U), /* Total number of RAM Sectors */ ((Mcu_RamSectionType)0U), /*GPT12 sleep mode setting */ #if ((MCU_GPT1_USED == STD_ON) || (MCU_GPT2_USED == STD_ON)) (boolean)FALSE, #endif #if (MCU_CCU60_USED == STD_ON) (boolean)FALSE, #endif #if (MCU_CCU61_USED == STD_ON) (boolean)FALSE #endif };
Configure MCU (variant aware. Variant name is 'Petrol')	const Mcu_ConfigType Mcu_Config_Petrol = { /*McuModuleConfiguration*/ /* MCU clock Configuration Pointer*/ Mcu_kClockConfiguration_Config_Petrol, /* Ram Section configuration Pointer*/ NULL_PTR, #if (MCU_GTM_USED == STD_ON) /* GTM Global Configuration Pointer*/

```
&Mcu_kGtmConfiguration_Config_Petrol,
  #endif
  /*Ptr to GPT12 Prescaler config
structure */
  #if ((MCU_GPT1_USED == STD_ON) ||
(MCU_GPT2_USED == STD_ON))

&Mcu_kGpt12PrescalerConfiguration_Con
fig_Petrol,
  #endif
  /* Ptr to Standby Mode in config
structure */

&Mcu_kLowPowerModeConfiguration_Confi
g_Petrol,
  /* Reset configuration */
  0x00000000U,
  /* Application Reset Disable
configuration */
  0x00000000U,
  /* Trap configuration */
  0xffffffffU,
  0x0000ffffU,
  /*Eru global input filter
configuration */
  0x00000000U,
  /* Total number of Clock settings
*/
  ((Mcu_ClockType)1U),
  /* Total number of RAM Sectors */
  ((Mcu_RamSectionType)0U),
  /*GPT12 sleep mode setting */
  #if ((MCU_GPT1_USED == STD_ON) ||
(MCU_GPT2_USED == STD_ON))
  (boolean)FALSE,
  #endif
  #if (MCU_CCU60_USED == STD_ON)
  (boolean)FALSE,
  #endif
  #if (MCU_CCU61_USED == STD_ON)
  (boolean)FALSE
  #endif
};
```

1.3.1.1 Member: Mcu_kClockConfiguration_Config[_<variant>]

Table 137 Mcu_kClockConfiguration_Config[_<variant>]

Name	Mcu_kClockConfiguration_Config[_<variant>]	
Type	Mcu_ClockConfigType	
Description	Pointer to Mcu clock configuration structure.	
User configurable	No	
Verification method	The generated structure member is present in the Mcu_Config[_<variant>] structure. It is always generated as Mcu_kClockConfiguration_Config[_<variant>]	
Example(s)	**Action**	**Generated output**
	Clock setting is configured (variant unaware)	`/* MCU clock Configuration Pointer*/` `Mcu_kClockConfiguration_Config,`
	Clock setting is configured (variant aware. Variant name is 'Petrol')	`/* MCU clock Configuration Pointer*/` `Mcu_kClockConfiguration_Config_Petrol ,`

1.3.1.2 Member: Mcu_kRamConfiguration_Config[_<variant>]

Table 138 Mcu_kRamConfiguration_Config[_<variant>]

Name	Mcu_kRamConfiguration_Config[_<variant>]	
Type	Mcu_RamConfigType	
Description	Pointer to RAM configuration structure	
Verification method	The generated structure member points to the RAM configuration structure. It is generated as Mcu_kRamConfiguration__Config[_<variant>] if atleast one node is configured in 'McuRamSectorSettingConf' else it is generated as 'NULL_PTR'	
Example(s)	**Action**	**Generated output**
	Atleast one node is configured in McuRamSectorSettingConf (variant unaware)	`/* Ram Section configuration Pointer*/` `Mcu_kRamConfiguration_Config,`
	No node is configured in McuRamSectorSettingConf	`/* Ram Section configuration Pointer*/` `NULL_PTR,`
	Atleast one node is configured in McuRamSectorSettingConf (variant aware, variant name is Petrol)	`/* Ram Section configuration Pointer*/` `Mcu_kRamConfiguration_Config_Petrol,`

1.3.1.3 Member: Mcu_kGtmConfiguration_Config[_<variant>]

Table 139 Mcu_kGtmConfiguration_Config[_<variant>]

Name	Mcu_kGtmConfiguration_Config[_<variant>]	
Type	Mcu_GtmConfigType	
Description	Pointer to GTM global configuration structure	
Verification method	The generated structure member is present in the Mcu_Config[_<variant>] structure. It is always generated as &Mcu_kGtmConfiguration_Config[_<variant>]. The member is only generated when GTM is available in the device.	
Example(s)	**Action**	**Generated output**

	Configure GTM global configuration (variant unaware)	`#if (MCU_GTM_USED == STD_ON)`
		`/* GTM Global Configuration Pointer*/`
		`&Mcu_kGtmConfiguration_Config,`
		`#endif`

	Configure GTM global configuration (variant aware, variant name is Petrol)	`#if (MCU_GTM_USED == STD_ON)`
		`/* GTM Global Configuration Pointer*/`
		`&Mcu_kGtmConfiguration_Config_Petrol,`
		`#endif`

1.3.1.4 Member: Mcu_kGpt12PrescalerConfiguration_Config[_<variant>]

Table 140 Mcu_ kGpt12PrescalerConfiguration _Config[_<variant>]

Name	Mcu_ kGpt12PrescalerConfiguration _Config[_<variant>]	
Type	Mcu_Gpt12ConfigType	
Description	Pointer to GPT12 prescaler configuration structure	
Verification method	The generated structure member is present in the Mcu_Config[_<variant>] structure. It is always generated as & Mcu_kGpt12PrescalerConfiguration _Config[_<variant>]. In case both GPT block 1 and block 2 are not configured, this structure is not generated.	
Example(s)	**Action**	**Generated output**

	Configure GPT12 prescaler configuration (variant unaware)	`/*Ptr to GPT12 Prescaler config structure */`
		`#if ((MCU_GPT1_USED == STD_ON) || (MCU_GPT2_USED == STD_ON))`
		`&Mcu_kGpt12PrescalerConfiguration_Config,`
		`#endif`

	Configure GPT12 prescaler configuration (variant aware, variant name is Petrol)	`/*Ptr to GPT12 Prescaler config structure */` `#if ((MCU_GPT1_USED == STD_ON) ||`

		(MCU_GPT2_USED == STD_ON)) &Mcu_kGpt12PrescalerConfiguration_Config_Petrol, #endif

1.3.1.5 Member: Mcu_kLowPowerModeConfiguration_Config[_<variant>]

Table 141 Mcu_kLowPowerModeConfiguration_Config[_<variant>]

Name	Mcu_kLowPowerModeConfiguration_Config[_<variant>]	
Type	Mcu_LowPowerModeType	
Description	Pointer to low power mode configuration structure	
Verification method	The generated structure member is present in the Mcu_Config[_<variant>] structure. It is always generated as &Mcu_kLowPowerModeConfiguration_Config[_<variant>].	
Example(s)	**Action**	**Generated output**
	Configure low power mode in McuModeSettingConf (variant unaware)	/* Ptr to Standby Mode in config structure */ &Mcu_kLowPowerModeConfiguration_Config,
	Configure low power mode in McuModeSettingConf (variant aware, variant name is Petrol)	/* Ptr to Standby Mode in config structure */ &Mcu_kLowPowerModeConfiguration_Config_Petrol,

1.3.1.6 Member: McuResetCfg

Table 142 McuResetCfg

Name	McuResetCfg
Type	uint32
Description	Indicates the configured value for reset.
Verification method	The value for this member is generated based on option selected in container 'McuModuleConfiguration/ McuResetSettingConf': Bit 0 is updated based on numeric value suffixed after 'SEL' keyword in McuESR0ResetConf Bit 2 is updated based on numeric value suffixed after 'SEL' keyword in McuESR1ResetConf Bit 6 is updated based on numeric value suffixed after 'SEL' keyword in McuSMUResetConf Bit 8 is updated based on numeric value suffixed after 'SEL' keyword in McuSWResetConf Bit 10 is updated based on numeric value suffixed after 'SEL' keyword in McuSTM0ResetConf Bit 12 is updated based on numeric value suffixed after 'SEL' keyword in McuSTM1ResetConf based on availability of STM1 Bit 14 is updated based on numeric value suffixed after 'SEL' keyword in McuSTM2ResetConf based on availability of STM2 Bit 16 is updated based on numeric value suffixed after 'SEL' keyword in McuSTM3ResetConf based on availability of STM3 Bit 18 is updated based on numeric value suffixed after 'SEL' keyword in McuSTM4ResetConf based on availability of STM4 Bit 20 is updated based on numeric value suffixed after 'SEL' keyword in McuSTM5ResetConf

	based on availability of STM5	
Example(s)	**Action**	**Generated output**
	Configure all parameters to default values in container McuModuleConfiguration/ McuResetSettingConf	`/* Reset configuration */ 0x00000000U,`
	• McuModuleConfiguration/ McuResetSettingConf/McuESR0ResetConf = MCU_ESR0_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuESR1ResetConf = MCU_ESR1_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuSWResetConf = MCU_SW_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuSMUResetConf = MCU_SMU_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuSTMxResetConf = MCU_STMx_SYSTEM_RESET_SEL1 for STMs 0-5 based on availability	`/* Reset configuration */ 0x00000145U,`
	• McuModuleConfiguration/ McuResetSettingConf/McuESR0ResetConf = MCU_ESR0_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuESR1ResetConf = MCU_ESR1_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuSWResetConf = MCU_SW_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuSMUResetConf = MCU_SMU_SYSTEM_RESET_SEL1 • McuModuleConfiguration/ McuResetSettingConf/McuSTMxResetConf = MCU_STMx_SYSTEM_RESET_SEL1 for STM 0	`/* Reset configuration */ 0x00000445U,`

1.3.1.7 Member: McuArstDisCfg

Table 143 McuArstDisCfg

Name	McuArstDisCfg	
Type	uint32	
Description	Indicates the configured value for Application reset disable.	
Verification method	The value for this member is generated based on option selected in container 'McuModuleConfiguration/ McuResetSettingConf': Bit 0 is set if McuSTM0ResetOnApplResetEnable is set to 'False' else 0 Bit 1 is set if McuSTM1ResetOnApplResetEnable is set to 'False' else 0 Bit 2 is set if McuSTM2ResetOnApplResetEnable is set to 'False' else 0 Bit 3 is set if McuSTM3ResetOnApplResetEnable is set to 'False' else 0 Bit 4 is set if McuSTM4ResetOnApplResetEnable is set to 'False' else 0 Bit 5 is set if McuSTM5ResetOnApplResetEnable is set to 'False' else 0	
Example(s)	**Action**	**Generated output**
	Set McuSTMxResetOnApplResetEnable for STMx (x:0-5) to True	`/* Application Reset Disable configuration */` `0x00000000U,`
	Set McuSTMxResetOnApplResetEnable for STMx (x:0-5) to False	`/* Application Reset Disable configuration */` `0x0000003fU,`

1.3.1.8 Member: McuTrapSettingConf0

Table 144 McuTrapSettingConf0

Name	McuTrapSettingConf0
Type	uint32
Description	Indicates the trap setting configuration value for CPUs 0-3 based on availability
Verification method	The value for this member is generated based on option selected in container 'McuModuleConfiguration/ McuTrapSettingConf': Bit 0 is set if McuCPU0ESR0TrapEnable is set to 'False' else 0 Bit 1 is set if McuCPU0ESR1TrapEnable is set to 'False' else 0 Bit 2 is set if McuCPU0Trap2Enable is set to 'False' else 0 Bit 3 is set if McuCPU0SMUTrapEnable is set to 'False' else 0 Bit 8 is set if McuCPU1ESR0TrapEnable is set to 'False' (when CPU1 is available) else 0 Bit 9 is set if McuCPU1ESR1TrapEnable is set to 'False' (when CPU1 is available) else 0 Bit 10 is set if McuCPU1Trap2Enable is set to 'False' (when CPU1 is available) else 0 Bit 11 is set if McuCPU1SMUTrapEnable is set to 'False' (when CPU1 is available) else 0 Bit 16 is set if McuCPU2ESR0TrapEnable is set to 'False' (when CPU2 is available) else 0 Bit 17 is set if McuCPU2ESR1TrapEnable is set to 'False' (when CPU2 is available) else 0 Bit 18 is set if McuCPU2Trap2Enable is set to 'False' (when CPU2 is available) else 0 Bit 19 is set if McuCPU2SMUTrapEnable is set to 'False' (when CPU2 is available) else 0 Bit 24 is set if McuCPU3ESR0TrapEnable is set to 'False' (when CPU3 is available) else 0 Bit 25 is set if McuCPU3ESR1TrapEnable is set to 'False' (when CPU3 is available) else 0 Bit 26 is set if McuCPU3Trap2Enable is set to 'False' (when CPU3 is available) else 0 Bit 27 is set if McuCPU3SMUTrapEnable is set to 'False' (when CPU3 is available) else 0

	Bits 4-7, 12-15, 20-23 and 28-31 are reserved and set to 1 always.	
Example(s)	**Action**	**Generated output**
	• Set McuCPUxESR0TrapEnable to True • Set McuCPUxESR1TrapEnable to True • Set McuCPUxSMUTrapEnable to True • Set McuCPUxTrap2Enable to True (x: 0-3)	`/* Trap configuration */` `0xf0f0f0f0U,`
	• Set McuCPUxESR0TrapEnable to False • Set McuCPUxESR1TrapEnable to False • Set McuCPUxSMUTrapEnable to False • Set McuCPUxTrap2Enable to False • (x: 0-3)	`/* Trap configuration */` `0xffffffffU,`

1.3.1.9 Member: McuTrapSettingConf1

Table 145 McuTrapSettingConf1

Name	McuTrapSettingConf1
Type	uint32
Description	Indicates the trap setting configuration value for CPUs 4-5 based on availability
Verification method	The value for this member is generated based on option selected in container 'McuModuleConfiguration/ McuTrapSettingConf': Bit 0 is set if McuCPU4ESR0TrapEnable is set to 'False' (when CPU4 is available) else 0 Bit 1 is set if McuCPU4ESR1TrapEnable is set to 'False' (when CPU4 is available) else 0 Bit 2 is set if McuCPU4Trap2Enable is set to 'False' (when CPU4 is available) else 0 Bit 3 is set if McuCPU4SMUTrapEnable is set to 'False' (when CPU4 is available) else 0 Bit 8 is set if McuCPU5ESR0TrapEnable is set to 'False' (when CPU5 is available) else 0 Bit 9 is set if McuCPU5ESR1TrapEnable is set to 'False' (when CPU5 is available) else 0 Bit 10 is set if McuCPU5Trap2Enable is set to 'False' (when CPU5 is available) else 0 Bit 11 is set if McuCPU5SMUTrapEnable is set to 'False' (when CPU5 is available) else 0

	Bits 4-7, 12-15 are reserved and set to 1 always. Bits 16-31 are reserved and written with 0 always.	
Example(s)	**Action**	**Generated output**
	• Set McuCPUxESR0TrapEnable to True • Set McuCPUxESR1TrapEnable to True • Set McuCPUxSMUTrapEnable to True • Set McuCPUxTrap2Enable to True (x: 4-5)	`/* Trap configuration */` `0x0000f0f0U,`
	• Set McuCPUxESR0TrapEnable to False • Set McuCPUxESR1TrapEnable to False • Set McuCPUxSMUTrapEnable to False • Set McuCPUxTrap2Enable to False • (x: 4-5)	`/* Trap configuration */` `0x0000ffffU,`

1.3.1.10 Member: McuEruEiFiltCfg

Table 146 McuEruEiFiltCfg

Name	McuEruEiFiltCfg	
Type	uint32	
Description	Indicates the configured value for ERU input filter.	
Verification method	The value for this member is generated based on option selected in container 'McuModuleConfiguration/ McuEruInputFilterRegVal'. EruFilterVal = (McuModuleConfiguration/McuEruInputFilterRegVal) & 0xff01ffff	
Example(s)	**Action**	**Generated output**
	McuModuleConfiguration/ McuEruInputFilterRegVal = 1500	`/*Eru global input filter configuration */` `0x000005dcU,`
	McuModuleConfiguration/	`/* ERU input filter`

	McuEruInputFilterRegVal = 0x0fffffff	configuration*/ 0x0f01ffffU,

1.3.1.11 Member: McuNoOfClockCfg

Table 147 McuNoOfClockCfg

Name	McuNoOfClockCfg	
Type	Mcu_ClockType	
Description	Indicates the number of clock settings configured.	
Verification method	The value for this member is generated based on number of nodes configured in container 'McuClockSettingConf'.	
Example(s)	**Action**	**Generated output**
	Number of nodes in McuClockSettingConf = 5	/* Total number of Clock settings */ ((Mcu_ClockType)5U),
	Number of nodes in McuClockSettingConf = 255	/* Total number of Clock settings */ ((Mcu_ClockType)255U),

1.3.1.12 Member: McuNoOfRamCfg

Table 148 McuNoOfRamCfg

Name	McuNoOfRamCfg	
Type	Mcu_RamSectionType	
Description	Indicates the number of RAM sectors configured.	
Verification method	The value for this member is generated based on number of nodes configured in container 'McuRamSectorSettingConf'.	
Example(s)	**Action**	**Generated output**
	Number of nodes in McuRamSectorSettingConf = 5	/* Total number of RAM Sectors */ ((Mcu_RamSectionType)5U),
	Number of nodes in McuRamSectorSettingConf = 255	/* Total number of RAM Sectors */ ((Mcu_RamSectionType)255U),

1.3.1.13 Member: IsGpt12SleepModeEnabled

Table 149 IsGpt12SleepModeEnabled

Name	IsGpt12SleepModeEnabled	
Type	Boolean	
Description	Indicates whether GPT12 sleep mode is enable or disabled.	
Verification method	The value for this member is generated as TRUE if the parameter 'McuGeneralConfiguration/McuGpt12SleepModeEnabled' is set to 'True' else is generated as FALSE	
Example(s)	**Action**	**Generated output**

	McuGeneralConfiguration/ McuGpt12SleepModeEnabled = True	`/*GPT12 sleep mode setting */`
		`#if ((MCU_GPT1_USED == STD_ON) || (MCU_GPT2_USED == STD_ON))`
		`(boolean)TRUE,`
		`#endif`

	McuGeneralConfiguration/ McuGpt12SleepModeEnabled = False	`/*GPT12 sleep mode setting */`
		`#if ((MCU_GPT1_USED == STD_ON) || (MCU_GPT2_USED == STD_ON))`
		`(boolean)FALSE,`
		`#endif`

1.3.1.14 Member: IsCcu60SleepModeEnabled

Table 150 IsCcu60SleepModeEnabled

Name	IsCcu60SleepModeEnabled	
Type	Boolean	
Description	Indicates whether CCU6 kernel 0 sleep mode is enable or disabled.	
Verification method	The value for this member is generated as TRUE if the parameter ‘McuGeneralConfiguration/McuCcu60SleepModeEnabled’ is set to ‘True’ else is generated as FALSE	
Example(s)	**Action**	**Generated output**

	McuGeneralConfiguration/ McuCcu60SleepModeEnabled = True	`#if (MCU_CCU60_USED == STD_ON)`
		`(boolean)TRUE,`
		`#endif`

	McuGeneralConfiguration/ McuCcu60SleepModeEnabled = False	`#if (MCU_CCU60_USED == STD_ON)`
		`(boolean)FALSE,`
		`#endif`

1.3.1.15 Member: IsCcu61SleepModeEnabled

Table 151 IsCcu61SleepModeEnabled

Name	IsCcu61SleepModeEnabled	
Type	Boolean	
Description	Indicates whether CCU6 kernel 1 sleep mode is enable or disabled.	
Verification method	The value for this member is generated as TRUE if the parameter ‘McuGeneralConfiguration/McuCcu61SleepModeEnabled’ is set to ‘True’ else is generated as FALSE	
Example(s)	**Action**	**Generated output**

	McuGeneralConfiguration/ McuCcu61SleepModeEnabled = True	`#if (MCU_CCU61_USED == STD_ON)`
		`(boolean)TRUE,`
		`#endif`

	McuGeneralConfiguration/ McuCcu61SleepModeEnabled = False	`#if (MCU_CCU61_USED == STD_ON)`
		`(boolean)FALSE,`
		`#endif`

1.3.2 Structure: Mcu_kRamConfiguration_Config[_<variant>]

Table 152 Mcu_kRamConfiguration_Config [_<variant>]

Name	Mcu_kRamConfiguration_Config [_<variant>]	
Type	Mcu_RamConfigType	
Description	Configuration structure for RAM configuration.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**

	Configure RAM in McuRamSectorSettingConf (variant unaware)	`static const Mcu_RamConfigType Mcu_kRamConfiguration_Config[1] =`
		`{`
		`/*McuRamSectorSettingConf_0*/`
		`{`
		`/* RAM Section Configuration: McuRamSectorSettingConf_0 */`
		`/* RAM section base address */`
		`(Mcu_RamBaseAdrType)0x70000000U,`
		`/* RAM section size */`
		`(Mcu_RamSizeType)0x00000004U,`
		`/* Default initialization value */`
		`(Mcu_RamPrstDatType)0x00U`
		`}`
		`};`

	Configure RAM in McuRamSectorSettingConf (variant aware. Variant name is ‘Petrol’)	`static const Mcu_RamConfigType Mcu_kRamConfiguration_Config_Petrol[1] =`
		`{`
		`/*McuRamSectorSettingConf_0*/`
		`{`
		`/* RAM Section Configuration: McuRamSectorSettingConf_0 */`
		`/* RAM section base address */`
		`/*`
		`(Mcu_RamBaseAdrType)0x70000000U,`

		/* RAM section size */
		(Mcu_RamSizeType)0x00000004U,
		/* Default initialization value */
		(Mcu_RamPrstDatType)0x00U
		}
		};

1.3.2.1 Member: RamBaseAdrPtr

Table 153 RamBaseAdrPtr

Name	RamBaseAdrPtr	
Type	Mcu_RamBaseAdrType	
Description	Pointer to the RAM base address.	
Verification method	The value for this member is generated based on value in 'McuRamSectorSettingConf/ McuRamSectionBaseAddress'.	
Example(s)	**Action**	**Generated output**
	McuRamSectorSettingConf/ McuRamSectionBaseAddress = 0x70000000	/* RAM section base address */ (Mcu_RamBaseAdrType)0x70000000U,
	McuRamSectorSettingConf/ McuRamSectionBaseAddress = 0x80008000	/* RAM section base address */ (Mcu_RamBaseAdrType) 0x80008000U,

1.3.2.2 Member: RamSize

Table 154 RamSize

Name	RamSize	
Type	Mcu_RamSizeType	
Description	Indicates the size of RAM section.	
Verification method	The value for this member is generated based on size of RAM specified in 'McuRamSectorSettingConf/McuRamSectionSize'.	
Example(s)	**Action**	**Generated output**
	McuRamSectorSettingConf/McuRamSectionSize = 4	/* RAM section size */ (Mcu_RamSizeType)0x00000004U,
	McuRamSectorSettingConf/McuRamSectionSize = 255	/* RAM section size */ (Mcu_RamSizeType)0x000000ffU,

1.3.2.3 Member: RamPrstData

Table 155 RamPrstData

Name	RamPrstData	
Type	Mcu_RamPrstDatType	
Description	Indicates the data value used to initialize the RAM.	
Verification method	The value for this member is generated based on value in 'McuRamSectorSettingConf/ McuRamDefaultValue'.	
Example(s)	**Action**	**Generated output**
	McuRamSectorSettingConf/ McuRamDefaultValue = 123	`/* Default initialization value */ (Mcu_RamPrstDatType)0x7bU`
	McuRamSectorSettingConf/ McuRamDefaultValue = 192	`/* Default initialization value */ (Mcu_RamPrstDatType)0xc0U`

1.3.2.4 Member: RamData

Table 156 RamData

Name	RamData	
Type	uint64	
Description	Indicates the prepared data value to be written in RAM at once.	
Verification method	The value for this member is generated based on value in 'McuRamSectorSettingConf/ McuRamDefaultValue' using following algorithm: RamWriteSize = McuRamSectorSettingConf/ McuRamSectionWriteSize RamDataValue = McuRamSectorSettingConf/ McuRamDefaultValue A loop is run from 0 to RamWriteSize RamData = RamData | (RamDataValue << (8 * LoopIndex))	
Example(s)	**Action**	**Generated output**
	McuRamSectorSettingConf/ McuRamDefaultValue = 123 McuRamSectorSettingConf/ McuSectionWriteSize = 8	`/*Prepared Ram Data to be written at once*/ (uint64)0x7b7b7b7b7b7b7b7bU`
	McuRamSectorSettingConf/ McuRamDefaultValue = 192 McuRamSectorSettingConf/ McuSectionWriteSize = 4	`/*Prepared Ram Data to be written at once*/ (uint64)0x00000000c0c0c0c0U`

1.3.2.5 Member: RamWriteSize

Table 157 RamWriteSize

Name	RamWriteSize

Type	Mcu_RamWriteSizeType	
Description	Indicates the bytes of data to be written in RAM at once.	
Verification method	The value for this member is generated based on value in 'McuRamSectorSettingConf/ McuRamSectionWriteSize'	
Example(s)	**Action**	**Generated output**
	McuRamSectorSettingConf/ McuSectionWriteSize = 4	`/* RAM section write size */ (Mcu_RamWriteSizeType)0x00000004U`
	McuRamSectorSettingConf/ McuSectionWriteSize = 8	`/* RAM section write size */ (Mcu_RamWriteSizeType)0x00000008U`

1.3.3 Structure: Mcu_kPllDistributionConfiguration_Config[_<variant>]

Table 158 Mcu_kPllDistributionConfiguration_Config[_<variant>]

Name	Mcu_kPllDistributionConfiguration_Config[_<variant>]	
Type	Mcu_RamConfigType	
Description	Configuration structure for clock distribution configuration.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**
	Configure clock setting in McuClockSettingConf (variant unaware)	`static const Mcu_PllDistributionConfigType Mcu_kPllDistributionConfiguration_Config_0 = { /* CCUCON0 value */ 0x17230133U, /* CCUCON1 value */ 0x00000280U, /* CCUCON2 value */ 0x00000000U, /* CCUCON5 value */ 0x00000132U, /* CCUCON6...CCUCON11 value */ { 0x00000000U, 0x00000000U, 0x00000000U, 0x00000000U,`

	0x00000000U,
	0x00000000U
	}
	};

Configure clock setting in McuClockSettingConf (variant aware. Variant name is 'Petrol')	static const Mcu_PllDistributionConfigType Mcu_kPllDistributionConfiguration_Config_Petrol_0 =
	{
	/* CCUCON0 value */
	0x17230133U,
	/* CCUCON1 value */
	0x00000280U,
	/* CCUCON2 value */
	0x00000000U,
	/* CCUCON3 value */
	0x00000000U,
	/* CCUCON4 value */
	0x40000000U,
	/* CCUCON5 value */
	0x00000032U,
	/* CCUCON6...CCUCON11 value */
	{
	0x00000000U,
	0x00000000U,
	0x00000000U,
	0x00000000U,
	0x00000000U,
	0x00000000U
	}
	};

1.3.3.1 Member: Ccucon0

Table 159 Ccucon0

Name	Ccucon0
Type	uint32
Description	Indicates the value to be written in CCUCON0 register.
Verification method	The value for this member is generated based McuClockReferencePointFrequency0 frequency divided by values specified in 'McuClockSettingConfig /McuPllDistributionSettingConfig': Bits 0-3 are configured based on parameter McuSTMFrequency.

	Bits 4-7 are configured based on parameter McuGTMFrequency. Bits 8-11 are configured based on parameter McuSRIFrequency. Bits 12-14 are configured based on value suffixed after 'SEL' keyword in parameter McuLowPowerDivValue. Bits 16-19 are configured based on parameter McuSPBFrequency. Bits 20-23 are configured based on parameter McuBBBFrequency. Bits 24-25 are configured based on parameter McuFSIFrequency. Bits 26-27 are configured based on parameter McuFSI2Frequency. Bits 28-29 are configured based on value suffixed after 'SEL' keyword in parameter McuClockDistributionInpClockSel. Bits 4 – 11, 16 – 25 are set to 0 if McuLowPowerDivValue is selected. Other bits are written with 0	
Example(s)	**Action**	**Generated output**
	• McuClockReferencePointFrequency0 = 300 MHz • McuSTMFrequency = 100 MHz • McuGTMFrequency = 200 MHz • McuSRIFrequency = 300 MHz • McuLowPowerDivValue = LOW_POWER_DIVIDER_DISABLE_SEL0 • McuSPBFrequency = 100 MHz • McuBBBFrequency = 150 MHz • McuFSIFrequency = 100 MHz • McuFSI2Frequency = 300 MHz • McuClockDistributionInpClockSel = PLL_INPUT_CLOCK_SRC_SELECT_SEL1	`/* CCUCON0 value */` `0x17230113U,`
	• McuClockReferencePointFrequency0 = 300 MHz • McuSTMFrequency = 100 MHz • McuGTMFrequency = 100 MHz • McuSRIFrequency = 100 MHz • McuLowPowerDivValue =	`/* CCUCON0 value */` `0x0f230313U,`

	LOW_POWER_DIVIDER_DISABLE_SEL0 • McuSPBFrequency = 100 MHz • McuBBBFrequency = 150 MHz • McuFSIFrequency = 100 MHz • McuFSI2Frequency = 100 MHz • McuClockDistributionInpClockSel = BACKUP_INPUT_CLOCK_SRC_SELECT_SEL0	

1.3.3.2 Member: Ccucon1

Table 160 Ccucon1

Name	Ccucon1	
Type	uint32	
Description	Indicates the value to be written in CCUCON1 register.	
Verification method	The value for this member is generated based on McuClockReferencePointFrequency1 or McuClockReferencePointFrequency2 frequency divided by values specified in 'McuClockSettingConfig /McuPllDistributionSettingConfig'. (For McuI2CFrequency McuClockReferencePointFrequency2 is used McuMCanFrequency, McuMscFrequency and McuQspiFrequency the configured clock source frequency is used): Bits 0-3 are configured based on parameter McuMCanFrequency. Bits 4-5 are configured based on on value suffixed after 'SEL' keyword in parameter McuMCanClockSourceSelection. Bit 7 is configured based on value suffixed after 'SEL' keyword in parameter McuPeripheralPllSettingConfig/McuFreqSource1ClockDivSelect. Bits 8-11 are configured based on parameter McuI2CFrequency. Bits 16-19 are configured based on parameter McuMscFrequency. Bits 20-21 are configured based on value suffixed after 'SEL' keyword in parameter McuMscClockSourceSelection. Bits 24-27 are configured based on parameter McuQspiFrequency. Bits 28-29 are configured based on value suffixed after 'SEL' keyword in parameter McuQspiClockSourceSelection. Other bits are written with 0	
Example(s)	**Action**	**Generated output**
	• McuClockReferencePointFrequency1 = 160 MHz • McuMCanFrequency = 80 MHz	`/* CCUCON1 value */` `0x12120192U,`

	• McuMCanClockSourceSelection = MCAN_CLOCK_SOURCE_MCANI_SEL1 • McuPll2DivSelect = MCU_K3_DIV_FACTOR_BYPASSED_SEL1 • McuI2CFrequency = 80 MHz • McuMscFrequency = 80 MHz • McuMscClockSourceSelection = MSC_CLOCK_SOURCE_SOURCE1_SEL1 • McuQspiFrequency = 80 MHz • McuQspiClockSourceSelection = QSPI_CLOCK_SOURCE_SOURCE1_SEL1	
	• McuClockReferencePointFrequency1 = 160 MHz • McuMCanFrequency = 40 MHz • McuMCanClockSourceSelection = MCAN_CLOCK_SOURCE_MCANI_SEL1 • McuPll2DivSelect = MCU_K3_DIV_FACTOR_BYPASSED_SEL1 • McuI2CFrequency = 40 MHz • McuMscFrequency = 40 MHz • McuMscClockSourceSelection = MSC_CLOCK_SOURCE_SOURCE1_SEL1 • McuQspiFrequency = 40 MHz • McuQspiClockSourceSelection = QSPI_CLOCK_SOURCE_SOURCE1_SEL1	`/* CCUCON0 value */` `0x14140594U,`

1.3.3.3 Member: Ccucon2

Table 161 Ccucon2

Name	Ccucon2	
Type	uint32	
Description	Indicates the value to be written in CCUCON2 register.	
Verification method	The value for this member is generated based on McuClockReferencePointFrequency2 (for McuAscLinFastFrequency) or McuClockReferencePointFrequency1 (for McuAscLinSlowFrequency) frequency divided by values specified in 'McuClockSettingConfig /McuPllDistributionSettingConfig': Bits 0-3 are configured based on parameter McuAscLinFastFrequency. Bit 8-11 is configured based on parameter McuAscLinSlowFrequency. Bits 12-13 are configured based on value suffixed after 'SEL' keyword in parameter McuAscLinSlowClockSourceSelection. Bit 24 is set if parameter McuEbuClkEnable is set to True else 0. Bit 25 is set if parameter McuErayClkEnable is set to True else 0. Bit 25 is set if parameter McuHspdmClkEnable is set to True else 0. Other bits are written with 0	
Example(s)	**Action**	**Generated output**
	• McuClockReferencePointFrequency2 = 200 MHz • McuAscLinFastFrequency = 100 MHz • McuAscLinSlowFrequency = 80 MHz • McuAscLinSlowClockSourceSelection = ASCLINS_CLOCK_SOURCE_ASCLINSI_SEL1 • McuEbuClkEnable = False • McuErayClkEnable = False • McuHspdmClkEnable = False	`/* CCUCON2 value */` `0x00001202U,`
	• McuClockReferencePointFrequency2 = 200 MHz • McuAscLinFastFrequency = 100 MHz • McuAscLinSlowFrequency = 80 MHz • McuAscLinSlowClockSourceSelection = ASCLINS_CLOCK_SOUR	`/* CCUCON0 value */` `0x07001202U,`

	CE_ASCLINSI_SEL1 • McuEbuClkEnable = True • McuErayClkEnable = True • McuHspdmClkEnable = True	

1.3.3.4 Member: Ccucon3

Table 162 Ccucon3

Name	Ccucon3	
Type	uint32	
Description	Indicates the value to be written in CCUCON3 register.	
Verification method	This parameter is generated only when macro MCU_SAFETY_ENABLE is STD_ON. Bit 0 is set when parameter McuPll0ClockMonEnable is set to True else 0. Bit 1 is set when parameter McuPll1ClockMonEnable is set to True else 0. Bit 2 is set when parameter McuPll2ClockMonEnable is set to True else 0. Bit 3 is set when parameter McuSpbClockMonEnable is set to True else 0. Bit 4 is set when parameter McuBackupClockMonEnable is set to True else 0. Other bits are written with 0	
Example(s)	**Action**	**Generated output**
	• McuPll0ClockMonEnable = True • McuPll1ClockMonEnable = True • McuPll2ClockMonEnable = True • McuSpbClockMonEnable = True • McuBackupClockMonEnable = True	`/* CCUCON2 value */` `0x0000001fU,`
	• McuPll0ClockMonEnable = False • McuPll1ClockMonEnable = True • McuPll2ClockMonEnable = True • McuSpbClockMonEnable = False • McuBackupClockMonEnable = False	`/* CCUCON0 value */` `0x00000006U,`

1.3.3.5 Member: Ccucon4

Table 163 Ccucon4

Name	Ccucon4	
Type	uint32	
Description	Indicates the value to be written in CCUCON4 register.	
Verification method	This parameter is generated only when macro MCU_SAFETY_ENABLE is STD_ON. BackupLowThresh = (((512/ McuClockReferencePointFrequency0)*0.9)*10^8) & 4095 BackupUpThresh = (((512/ McuClockReferencePointFrequency0)*1.1)*10^8) & 4095 Bits 0-11 are configured with value of BackupLowThresh and McuBackupClockRangeMonEnable is True else 0. Bits 12-23 are configured with value of BackupUpThresh and McuBackupClockRangeMonEnable is True else 0. Bit 24 is set if parameter McuBackupClockRangeMonEnable is True else 0. Bit 30 is set always in order to update the register. Other bits are written with 0	
Example(s)	**Action**	**Generated output**
	• BackupLowThresh = 153 MHz • BackupUpThresh = 187 MHz • McuBackupClockRangeMonEnable = True	`/* CCUCON2 value */` `0x410bb099U,`
	• BackupLowThresh = 153 MHz • BackupUpThresh = 187 MHz • McuBackupClockRangeMonEnable = False	`/* CCUCON0 value */` `0x40000000U,`

1.3.3.6 Member: Ccucon5

Table 164 Ccucon5

Name	Ccucon5
Type	uint32
Description	Indicates the value to be written in CCUCON5 register.
Verification method	The value for this member is generated based McuClockReferencePointFrequency0 frequency divided by values specified in 'McuClockSettingConfig /McuPllDistributionSettingConfig': Bits 0-3 are configured based on parameter McuGEthFrequency. Bits 4-7 are configured based on parameter McuMcanHFrequency. Bits 8-11 are configured based on parameter McuAdasFrequency. Other bits are written with 0

Example(s)	Action	Generated output
	• McuClockReferencePointFrequency0 = 300 MHz • McuGEthFrequency = 150 MHz • McuMcanHFrequency = 100 MHz • McuAdasFrequency = 300 MHz	/* CCUCON2 value */ 0x00000132U,
	• McuClockReferencePointFrequency0 = 300 MHz • McuGEthFrequency = 100 MHz • McuMcanHFrequency = 100 MHz • McuAdasFrequency = 100 MHz	/* CCUCON0 value */ 0x00000333U,

1.3.3.7 Member: CcuconCpu[Core]

Table 165 CcuconCpu[Core]

Name	CcuconCpu[Core]	
Type	uint32	
Description	Indicates the values to be written in CCUCON6- CCUCON11 register for CPU0- CPU5 based on availability of CPUs	
Verification method	The value for this member is generated as: CPUxDIV = 64 - ((f_{CPUx} * 64)/ f_{SRI}) where f_{CPUx}: CPU frequency, f_{SRI}: SRI frequency in MHz Bits 0-5 are configured as per value of CPUxDIV (x:0-5 based on availability) Other bits are written with 0	
Example(s)	**Action**	**Generated output**

	• McuSRIFrequency = 300 MHz
	• McuCPU0Frequency = 150 MHz
	• McuCPU1Frequency = 300 MHz
	• McuCPU2Frequency = 300 MHz
	• McuCPU3Frequency = 300 MHz
	• McuCPU4Frequency = 300 MHz
	• McuCPU5Frequency = 300 MHz
		/* CCUCON6...CCUCON11 value */
		{
		0x00000020U,
		0x00000000U,
		0x00000000U,
		0x00000000U,
		0x00000000U,
		0x00000000U
		}

• McuSRIFrequency = 300 MHz
• McuCPU0Frequency = 150 MHz
• McuCPU1Frequency = 300 MHz
• McuCPU2Frequency = 150 MHz
• McuCPU3Frequency = 300 MHz
• McuCPU4Frequency = 150 MHz
• McuCPU5Frequency = 300 MHz
	`/* CCUCON6...CCUCON11 value */`
	`{`
	`0x00000020U,`
	`0x00000000U,`
	`0x00000020U,`
	`0x00000000U,`
	`0x00000020U,`
	`0x00000000U`
	`}`

1.3.4 Structure: Mcu_kLowPowerModeConfiguration_Config[_<variant>]

Table 166 Mcu_kLowPowerModeConfiguration_Config[_<variant>]

Name	Mcu_kLowPowerModeConfiguration_Config[_<variant>]	
Type	Mcu_LowPowerModeType	
Description	Configuration structure for low power mode (standby) configuration.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**

	Configure standby mode in McuModeSettingConf/ McuMode (variant unaware)	`static const Mcu_LowPowerModeType Mcu_kLowPowerModeConfiguration_Config =`
		`{`
		`/*MaxModeEvrcCtrl value*/`
		`{`
		`0x5U,`
		`0x0U,`
		`0U`
		`},`
		`/* PMSWCR0 Register value */`
		`0x40070000U,`
		`/* PMSWCR3Value */`
		`0x00000000U,`
		`/* PMSWCR4Value */`
		`0x00000000U,`
		`/* PMSWCR5Value */`

	0x00000001U,
	/* EVRUVMONValue */
	0x0075a7b8U,
	/* EVRMONCTRLValue */
	0x00b5a595U,
	/* Standby RAM start address(es) */
	{
	(uint32*)0x90000000U,
	(uint32*)0x90008000U,
	(uint32*)0x90010000U,
	(uint32*)0x90018000U
	}
	};

Configure standby mode in McuModeSettingConf/ McuMode (variant aware. Variant name is 'Petrol')	static const Mcu_LowPowerModeType Mcu_kLowPowerModeConfiguration_Config_Petrol = { /*MaxModeEvrcCtrl value*/ { 0x5U, 0x0U, 0U }, /* PMSWCR0 Register value */ 0x40070000U, /* PMSWCR3Value */ 0x00000000U, /* PMSWCR4Value */ 0x00000000U, /* PMSWCR5Value */ 0x00000001U, /* EVRUVMONValue */ 0x0075a7b8U, /* EVRMONCTRLValue */ 0x00b5a595U, /* Standby RAM start address(es) */ { (uint32*)0x90000000U, (uint32*)0x90008000U, (uint32*)0x90010000U,

		(uint32*)0x90018000U
		}
		};

1.3.4.1 Member: MaxModeEvrcCtrl

Table 167 MaxModeEvrcCtrl

Name	MaxModeEvrcCtrl	
Type	Mcu_ModeEvrcCtrlType	
Description	Structure to hold the value for mode and Evrc configuration.	
Verification method	The structure holds 3 members: • First member is generated based on the value of configuration parameter 'McuModeSettingconf/ McuMode'. When configuring IDLE, SLEEP and STANDBY mode the bits 0, 1 and 2 is respectively set to 1. • Second member is generated based on the value of 'McuModeSettingConf/ McuEvrcLPMOnSleepReqEnable'. TRUE: Entering into Low power mode for EVRC on sleep mode request is enabled. FALSE: Entering into Low power mode for EVRC on sleep mode request is disabled. • Third member of the structure is reserved and always generated as 0.	
Example(s)	**Action**	**Generated output**
	McuModeSettingconf/ McuMode = Idle McuModeSettingConf/ McuEvrcLPMOnSleepReqEnable = False	/*MaxModeEvrcCtrl value*/ { 0x1U, 0x0U, 0U },
	McuModeSettingconf/ McuMode = Sleep McuModeSettingConf/ McuEvrcLPMOnSleepReqEnable = True	/*MaxModeEvrcCtrl value*/ { 0x2U, 0x1U, 0U },

1.3.4.2 Member: Pmswcr0

Table 168 Pmswcr0

Name	Pmswcr0
Type	uint32
Description	Indicates the value to be written in PMSWCR0 register.
Verification method	The value for this member is generated based on: 1. If McuModeSettingConf/ McuStdbyModeESR0Conf/

McuStdbyModeESR0WakeupEnable is True then:

- Bit 4 is set if McuModeSettingConf/ McuStdbyModeESR0Conf/ McuStdbyModeESR0FltEnable is set to True else 0.
- Bit 5 is generated based on the numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeESR0Conf/ McuStdbyModeESR0EdgeDetection.
- Bit 24 is set with the value of McuStdbyModeESR0WakeupEnable.

2. If McuModeSettingConf/ McuStdbyModeESR1Conf/ McuStdbyModeESR1WakeupEnable is True then:

- Bit 7 is set if McuModeSettingConf/ McuStdbyModeESR1Conf/ McuStdbyModeESR1FltEnable is set to True else 0.
- Bit 8 is generated based on the numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeESR1Conf/ McuStdbyModeESR1EdgeDetection.
- Bit 25 is set with the value of McuStdbyModeESR1WakeupEnable.

3. If McuModeSettingConf/ McuStdbyModePinAConf/ McuStdbyModePinAWakeupEnable is True then:

- Bit 10 is set if McuModeSettingConf/ McuStdbyModePinAConf/ McuStdbyModePinAFltEnable is set to True else 0.
- Bit 11 is generated based on the numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModePinAConf/ McuStdbyModePinAEdgeDetection.
- Bit 26 is set with the value of McuStdbyModePinAWakeupEnable.

4. If McuModeSettingConf/ McuStdbyModePinBConf/ McuStdbyModePinBWakeupEnable is True then:

- Bit 13 is set if McuModeSettingConf/ McuStdbyModePinBConf/ McuStdbyModePinBFltEnable is set to True else 0.
- Bit 14 is generated based on the numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModePinBConf/ McuStdbyModePinBEdgeDetection.
- Bit 27 is set with the value of McuStdbyModePinBWakeupEnable.
- Bit 16 is set based on numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeRamEnable
- Bit 20 is set based on numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeBlankingFilterDelay
- Bit 28 is set if McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeWakeupFromEVR is set to True else 0.
- Bit 29 is set if McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeWakeupFromSCR is set to True else 0.
- Bit 30 is set if McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeWakeupFromPORST is set to True else 0.
- Bit 31 is set if McuModeSettingConf/ McuStdbyModeWakeupTimerConf/ McuStdbyModeWakeupTimerEnable is set to True else 0.

5. Bit 2 is set to True if McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeEntryOnVEXTRampDown is set to

	True. 6. Bit 3 is set to True if McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeEntryOnVDDRampDown is set to True. • Other bits are set to 0 always.

Example(s)

Action	Generated output
McuModeSettingConf/ McuStdbyModeESR0Conf/ McuStdbyModeESR0WakeupEnable = True McuModeSettingConf/ McuStdbyModeESR0Conf/ McuStdbyModeESR0FltEnable = True McuModeSettingConf/ McuStdbyModeESR0Conf/ McuStdbyModeESR0EdgeDetection = ESR0_TRIG_RISING_EDGE_SEL 1 McuStdbyModeWakeupFromPORST = True	`/* PMSWCR0 Register value */` `0x41000030U,`
McuModeSettingConf/ McuStdbyModeESR1Conf/ McuStdbyModeESR1WakeupEnable = True McuModeSettingConf/ McuStdbyModeESR1Conf/ McuStdbyModeESR1FltEnable = True McuModeSettingConf/ McuStdbyModeESR1Conf/ McuStdbyModeESR1EdgeDetection = ESR1_TRIG_RISING_EDGE_SEL 1 McuStdbyModeWakeupFromPORST = False	`/* PMSWCR0 Register value */` `0x02000180U,`

1.3.4.3 Member: Pmswcr3

Table 169 Pmswcr3

Name	Pmswcr3

Type	uint32	
Description	Indicates the value to be written in PMSWCR3 register.	
Verification method	The value for this member is generated based on: 1. If McuModeSettingConf/ McuStdbyModeWakeupTimerConf/McuStdbyModeWakeupTimerEnable is True then: • Bits 0-23 are set based on the value of McuModeSettingConf/ McuStdbyModeWakeupTimerConf/McuStdbyModeWakeupTimerValue. • Bit 27 is set if McuModeSettingConf/ McuStdbyModeWakeupTimerConf/McuStdbyModeWakeupTimerEnable is set to True else 0. • Bit 29 is set based on numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeWakeupTimerConf/ McuStdbyModeWakeupTimerClkDiv. • Bit 30 is set based on numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeWakeupTimerConf/ McuStdbyModeWakeupTimerMode is set to True else 0. • Other bits are set to 0 always.	
Example(s)	**Action**	**Generated output**
	McuModeSettingConf/ McuStdbyModeWakeupTimerConf/McuStdbyModeWakeupTimerEnable = True McuModeSettingConf/ McuStdbyModeWakeupTimerConf/McuStdbyModeWakeupTimerValue = 15336 McuModeSettingConf/ McuStdbyModeWakeupTimerConf/ McuStdbyModeWakeupTimerClkDiv = WUT_100KHZ_DIV_CLK_SEL1 McuModeSettingConf/ McuStdbyModeWakeupTimerConf/ McuStdbyModeWakeupTimerMode = WUT_AUTO_STOP_MODE_SEL1	`/* PMSWCR3 Register value */` `0x68003be8U,`
	McuModeSettingConf/ McuStdbyModeWakeupTimerConf/McuStdbyModeWakeupTimerEnable = True	`/* PMSWCR3 Register value */` `0x08ffffffU,`

	McuModeSettingConf/ McuStdbyModeWakeupTimerConf/McuStdbyModeWakeupTimerValue = 16777215 McuModeSettingConf/ McuStdbyModeWakeupTimerConf/ McuStdbyModeWakeupTimerClkDiv = WUT_100KHZ_NO_DIV_CLK_SEL0 McuModeSettingConf/ McuStdbyModeWakeupTimerConf/ McuStdbyModeWakeupTimerMode = WUT_AUTO_RELOAD_MODE_SEL0	

1.3.4.4 Member: Pmswcr4

Table 170 Pmswcr4

Name	Pmswcr4	
Type	uint32	
Description	Indicates the value to be written in PMSWCR4 register.	
Verification method	Bit 6 is set based on numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeClkSelection. Other bits are set to 0 always.	
Example(s)	**Action**	**Generated output**
	McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeClkSelection = OSC_CLOCK_100KHZ_ONLY_SEL0	`/* PMSWCR4 Register value */` `0x00000000U,`
	McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeClkSelection = OSC_CLOCK_100KHZ_100MHZ_SEL1	`/* PMSWCR4 Register value */` `0x00000040U,`

1.3.4.5 Member: Pmswcr5

Table 171 Pmswcr5

Name	Pmswcr5	
Type	uint32	
Description	Indicates the value to be written in PMSWCR5 register.	
Verification method	Bit 0 is always set in order to enable bit protection tristate request bit. Bit 1 is set if McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModePortTriStateEnable is set to True else 0. Bit 2 is set if McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeESR0TriStateEnable is set to True else 0. Bit 4 is set if McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModePORSTFilterEnable is set to True else 0. Other bits are set to 0 always.	
Example(s)	**Action**	**Generated output**
	McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModePortTriStateEnable = False McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeESR0TriStateEnable = True McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModePORSTFilterEnable = False	`/* PMSWCR5 Register value */` `0x00000005U,`
	McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModePortTriStateEnable = True McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeESR0TriStateEnable = False McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModePORSTFilterEnable = True	`/* PMSWCR5 Register value */` `0x00000013U,`

1.3.4.6 Member: Evruvmon

Table 172 Evruvmon

Name	Evruvmon

Type	uint32
Description	Indicates the value to be written in EVRUVMON register.
Verification method	Bits 0-7 are generated with value specified in McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVddUVThres if McuStdbyModeEntryOnVDDRampDown is set to True else reset value is generated. Bits 16-23 are generated with value specified in McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVextUVThres if McuStdbyModeEntryOnVEXTRampDown is set to True else reset value is generated. All other bits are configured with the reset value.

Example(s)

Action	Generated output
McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVddUVThres = 117 McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVextUVThres = 184	`/* EVRUVMONValue */` `0x0075a7b8U,`
McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVddUVThres = 127 McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVextUVThres = 200	`/* EVRUVMONValue */` `0x00c8a77fU,`

1.3.4.7 Member: EvrmonCtrl

Table 173 EvrmonCtrl

Name	EvrmonCtrl
Type	uint32
Description	Indicates the value to be written in EVRMONCTRL register.
Verification method	Bits 4-5 are configured with the numeric value after keyword '_SEL' specified in McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVddUMMonMode if McuStdbyModeEntryOnVDDRampDown is set to True else reset value is generated. Bits 20-21 are configured with the numeric value after keyword '_SEL' specified in McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVextUMMonMode if McuStdbyModeEntryOnVEXTRampDown is set to

	True else reset value is generated. All other bits are configured with the reset value.	
Example(s)	**Action**	**Generated output**
	McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVddUMMonMode = VDD_UV_MON_MODE_SEL1 McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVextUMMonMode = VEXT_UV_MON_MODE_SEL3	`/* EVRMONCTRLValue */` `0x00b5a595U,`
	McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVddUMMonMode = VDD_UV_MON_MODE_SEL0 McuModeSettingConf/ McuStdbyModeSettingconf/ McuStdbyModeVddVextConf/ McuStdbyModeVextUMMonMode = VEXT_UV_MON_MODE_SEL1	`/* EVRMONCTRLValue */` `0x0095a585U,`

1.3.4.8 Member: StdbyRamAdr[MCU_NO_OF_STDBY_RAM_BLK]

Table 174 StdbyRamAdr[MCU_NO_OF_STDBY_RAM_BLK]

Name	StdbyRamAdr[MCU_NO_OF_STDBY_RAM_BLK]
Type	uint32*
Description	Structure of RAM addresses.
Verification method	The structure is generated based on the numeric value suffixed after '_SEL' keyword in McuModeSettingConf/ McuStdbyModeSettingConf/ McuStdbyModeRamEnable. If MCU_STANDBYRAM_CPU0_BLK0_SEL1 is selected, cached start address of CPU0 LMU is generated If MCU_STANDBYRAM_CPU0_BLK0_BLK1_SEL2 is selected, cached start addresses of CPU0 LMU and CPU0 LMU block 1 are generated If MCU_STANDBYRAM_CPU1_BLK0_BLK1_SEL4 is selected, cached start addresses of CPU1 LMU and CPU1 LMU block 1 are generated

<table>
<tr><td></td><td colspan="2">If MCU_STANDBYRAM_CPU0_CPU1_BLK0_BLK1_SEL7 is selected, cached start addresses of CPU0 LMU, CPU0 LMU block 1, CPU1 LMU and CPU1 LMU block 1 are generated

If MCU_STANDBYRAM_CPU0_BLK0_NONCACHED_SEL1 is selected, non-cached start address of CPU0 LMU is generated

If MCU_STANDBYRAM_CPU0_BLK0_BLK1_NONCACHED_SEL2 is selected, non-cached start addresses of CPU0 LMU and CPU0 LMU block 1 are generated

If MCU_STANDBYRAM_CPU1_BLK0_BLK1_NONCACHED_SEL4 is selected, non-cached start addresses of CPU1 LMU and CPU1 LMU block 1 are generated

If MCU_STANDBYRAM_CPU0_CPU1_BLK0_BLK1_NONCACHED_SEL7 is selected, non-cached start addresses of CPU0 LMU, CPU0 LMU block 1, CPU1 LMU and CPU1 LMU block 1 are generated</td></tr>
<tr><td>Example(s)</td><td>Action</td><td>Generated output</td></tr>
<tr><td></td><td>McuModeSettingConf/
McuStdbyModeSettingConf/
McuStdbyModeRamEnable =
MCU_STANDBYRAM_CPU0_BLK0_BLK1_SEL2</td><td><pre>/* Standby RAM start address(es) */
{
  (uint32*)0x90000000U,
  (uint32*)0x90008000U
}</pre></td></tr>
<tr><td></td><td>McuModeSettingConf/
McuStdbyModeSettingConf/
McuStdbyModeRamEnable =
MCU_STANDBYRAM_CPU0_CPU1_BLK0_BLK1_SEL7</td><td><pre>/* Standby RAM start address(es) */
{
  (uint32*)0x90000000U,
  (uint32*)0x90008000U,
  (uint32*)0x90010000U,
  (uint32*)0x90018000U
}</pre></td></tr>
<tr><td></td><td>McuModeSettingConf/
McuStdbyModeSettingConf/
McuStdbyModeRamEnable =
MCU_STANDBYRAM_CPU0_BLK0_BLK1_NONCACHED_SEL2</td><td><pre>/* Standby RAM start address(es) */
{
  (uint32*)0xb0000000U,
  (uint32*)0xb0008000U
}</pre></td></tr>
<tr><td></td><td>McuModeSettingConf/
McuStdbyModeSettingConf/
McuStdbyModeRamEnable =
MCU_STANDBYRAM_CPU0_CPU1_BLK0_BLK1_NONCACHED_SEL7</td><td><pre>/* Standby RAM start address(es) */
{
  (uint32*)0xb0000000U,
  (uint32*)0xb0008000U,
  (uint32*)0xb0010000U,
  (uint32*)0xb0018000U
}</pre></td></tr>
</table>

1.3.5 Structure: Mcu_kClockConfiguration_Config[_<variant>]

Table 175 Mcu_kClockConfiguration_Config[_<variant>]

Name	Mcu_kClockConfiguration_Config[_<variant>]	
Type	Mcu_ClockConfigType	
Description	Configuration structure for PLL initialization.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**
	Configure PLLs in McuClockSettingConf (variant unaware)	`static const Mcu_ClockConfigType Mcu_kClockConfiguration_Config[1] = { /*McuClockSettingConfig_0*/ { /* System PLL configuration value */ { 0U, 4U, 29U, 1U, 0U, 0U }, /* Peripheral PLL configuration value */ { 39U, 4U, 4U, 1U, 1U, 0U }, /* System PLL K2 divider increment step change delay */ 10U, /* System PLL K2 divider decrement step change delay */ 10U,`

		/* Peripheral PLL K2 divider step change increment */ 10U, /* Peripheral PLL K2 divider step change decrement */ 10U, /* Peripheral PLL K3 divider step change increment */ 10U, /* Peripheral PLL K3 divider step change decrement */ 10U, /* PLL clock divider configuration pointer */ &Mcu_kPllDistributionConfiguration_Config_0, /* External Clock configuration */ 0x00000000U, /* Backup frequency K Divider value for both PLLs */ 0x0375U, /* Converter Control Phase Synchronization configuration */ 0x00U, },/*McuClockReferencePointConfig*/ };
	Configure PLLs in McuClockSettingConf (variant aware. Variant name is 'Petrol')	static const Mcu_ClockConfigType Mcu_kClockConfiguration_Config_Petrol[1] = { /*McuClockSettingConfig_0*/ { /* System PLL configuration value */ { 1U, 0U, 29U, 1U, 0U, 0U

```
    },
    /* Peripheral PLL configuration
value */
    {
      39U,
      0U,
      4U,
      1U,
      1U,
      0U
    },
    /* System PLL K2 divider
increment step change delay */
    10U,
    /* System PLL K2 divider
decrement step change delay */
    10U,
    /* Peripheral PLL K2 divider step
change increment */
    10U,
    /* Peripheral PLL K2 divider step
change decrement */
    10U,
    /* Peripheral PLL K3 divider step
change increment */
    10U,
    /* Peripheral PLL K3 divider step
change decrement */
    10U,
    /* PLL clock divider
configuration pointer */

&Mcu_kPllDistributionConfiguration_Co
nfig_Petrol_0,
    /* External Clock configuration
*/
    0x00000000U,
    /* Backup frequency K Divider
value for both PLLs */
    0x0375U,
    /* Converter Control Phase
Synchronization configuration */
    0x00U,
  },/*McuClockReferencePointConfig*/
```

		};

1.3.5.1 Member: SystemPllCfg

Table 176 SystemPllCfg

Name	SystemPllCfg
Type	Mcu_SystemPllConfigType
Description	System PLL configuration structure
Verification method	The structure is generated based on the values configured in McuClockSettingConf/ McuSystemPllSettingConfig. 1. First element is generated based on the numeric value suffixed after '_SEL' keyword in 'McuClockSettingConf/ McuSystemPllSettingConfig/ McuPllInputSrcSelection'. Possible values are 'OSC_CLOCK_SRC_SELECT_SEL1' and 'BACKUP_CLOCK_SRC_SELECT_SEL0' 2. Second element is generated based on 'McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllPDivider' (0-7) 3. Third element is generated based on the value in McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllNDivider (0-127) 4. Fourth element is generated based on the value in McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllK2Divider (0-7) 5. If 'McuClockSettingConf/ McuSystemPllSettingConfig/ McuFmPllEnable' is set to True, Fifth element is generated as 1 and as 0 if McuFmPllEnable is False. 6. Sixth element is generated as 0 if 'McuClockSettingConf/ McuSystemPllSettingConfig/ McuFmPllEnable' is False, else based on the value in McuClockSettingConf/ 'McuSystemPllSettingConfig/ McuFMPllModAmp' using the formula: a. FMPllAmp value = McuFMPllModAmp * ((Input frequency based on McuPllInputSrcSelection)*64*(McuSystemPllNDivider + 1) / (100*3.6* (McuSystemPllPDivider + 1))

Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuSystemPllSettingConfig/ McuPllInputSrcSelection = OSC_CLOCK_SRC_SELECT_SEL1 McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllPDivider = 0 McuClockSettingConf/ McuSystemPllSettingConfig/	/*McuClockSettingConfig_0*/ { /* System PLL configuration value */ { 1U, 0U, 29U, 1U, 0U,

McuSystemPllNDivider = 29 McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllK2Divider = 1 McuClockSettingConf/ McuSystemPllSettingConfig/ McuFmPllEnable = False	0U },
McuClockSettingConf/ McuSystemPllSettingConfig/ McuPllInputSrcSelection = BACKUP_CLOCK_SRC_SELECT_SEL0 McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllPDivider = 0 McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllNDivider = 29 McuClockSettingConf/ McuSystemPllSettingConfig/ McuSystemPllK2Divider = 1 McuClockSettingConf/ McuSystemPllSettingConfig/ McuFmPllEnable = True McuClockSettingConf/ McuSystemPllSettingConfig/McuFMPllModAmp = 1.25	/*McuClockSettingConfig_0*/ { /* System PLL configuration value */ { 1U, 0U, 29U, 1U, 1U, 62597U },

1.3.5.2 Member: PeripheralPllCfg

Table 177 PeripheralCfg

Name	PeripheralPllCfg
Type	Mcu_PeripheralPllConfigType
Description	Peripheral PLL configuration structure
Verification method	The structure is generated based on the values configured in McuClockSettingConf/ McuPeripheralPllSettingConfig. 1. First element is genarted based on the value in McuClockSettingConf/ McuPeripheralPllSettingConfig/McuPeripheralPllNDivider (0-127). 2. Second element is generated based on McuClockSettingConf/ McuPeripheralPllSettingConfig/McuPeripheralPllPDivider (0-7)

	3. Third element is generated based on the value in McuClockSettingConf/ McuPeripheralPllSettingConfig/McuPeripheralPllK2Divider (0-7) 4. Fourth element is generated based on the value in McuClockSettingConf/ McuPeripheralPllSettingConfig/McuPeripheralPllK3Divider (0-7) 5. Fifth element is generated based on the numeric value suffixed after '_SEL' keyword in McuClockSettingConf/ McuPeripheralPllSettingConfig/McuPll2DivSelect (MCU_K3_DIV_FACTOR_BYPASSED_SEL1/ MCU_K3_DIV_FACTOR_NOT_BYPASSED_SEL0). 6. Sixth element is for reserved bits and is always generated as 0.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllNDivider = 39 McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllPDivider = 0 McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllK2Divider = 4 McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllK3Divider = 1 McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPll2DivSelect = MCU_K3_DIV_FACTOR_BYPASSED_SEL1	/* Peripheral PLL configuration value */ { 39U, 0U, 4U, 1U, 1U, 0U },
	McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllNDivider = 39 McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllPDivider = 0 McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllK2Divider = 4 McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPeripheralPllK3Divider = 1	/* Peripheral PLL configuration value */ { 39U, 0U, 4U, 1U, 0U, 0U },

	McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPll2DivSelect = MCU_K3_DIV_FACTOR_NOT_BYPASSED_SEL0	

1.3.5.3 Member: SysPllK2DivStepUpChangeDelay

Table 178 SysPllK2DivStepUpChangeDelay

Name	SysPllK2DivStepUpChangeDelay	
Type	uint32	
Description	Delay for incrementing system PLL K2 divider value	
Verification method	The structure is generated based on the value configured in McuClockSettingConf/ McuSystemPllSettingConfig/McuSysPllK2DivStepUpChangeDelay.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuSystemPllSettingConfig/McuSysPllK2DivStepUpChangeDelay = 10	`/* System PLL K2 divider increment step change delay */ 10U,`
	McuClockSettingConf/ McuSystemPllSettingConfig/McuSysPllK2DivStepUpChangeDelay = 50	`/* System PLL K2 divider increment step change delay */ 50U,`

1.3.5.4 Member: SysPllK2DivStepDownChangeDelay

Table 179 SysPllK2DivStepDownChangeDelay

Name	SysPllK2DivStepDownChangeDelay	
Type	uint32	
Description	Delay for decrementing system PLL K2 divider value	
Verification method	The structure is generated based on the value configured in McuClockSettingConf/ McuSystemPllSettingConfig/McuSysPllK2DivStepDownChangeDelay.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuSystemPllSettingConfig/McuSysPllK2DivStepDownChangeDelay = 10	`/* System PLL K2 divider decrement step change delay */ 10U,`
	McuClockSettingConf/	`/* System PLL K2 divider decrement`

	McuSystemPllSettingConfig/McuSysPllK2DivStepDownChangeDelay = 50	`step change delay */` `50U,`

1.3.5.5 Member: PeripheralPllK2StepUpChangeDelay

Table 180 PeripheralPllK2StepUpChangeDelay

Name	PeripheralPllK2StepUpChangeDelay	
Type	uint32	
Description	Delay for incrementing peripheral PLL K2 divider value	
Verification method	The structure is generated based on the value configured in McuClockSettingConf/McuPeripheralPllSettingConfig/McuPerPllK2DivStepUpChangeDelay.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/McuPeripheralPllSettingConfig/McuPerPllK2DivStepDownChangeDelay = 10	`/* Peripheral PLL K2 divider increment step change delay */` `10U,`
	McuClockSettingConf/McuPeripheralPllSettingConfig/McuPerPllK2DivStepDownChangeDelay = 50	`/* Peripheral PLL K2 divider increment step change delay */` `50U,`

1.3.5.6 Member: PeripheralPllK2StepDownChangeDelay

Table 181 PeripheralPllK2StepDownChangeDelay

Name	PeripheralPllK2StepDownChangeDelay	
Type	uint32	
Description	Delay for decrementing peripheral PLL K2 divider value	
Verification method	The structure is generated based on the value configured in McuClockSettingConf/McuPeripheralPllSettingConfig/McuPerPllK2DivStepDownChangeDelay.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/McuPeripheralPllSettingConfig/McuPerPllK2DivStepDownChangeDelay = 10	`/* Peripheral PLL K2 divider decrement step change delay */` `10U,`
	McuClockSettingConf/McuPeripheralPllSettingConfig/McuPerPllK2DivStepDownCha	`/* Peripheral PLL K2 divider decrement step change delay */`

	ngeDelay = 50	50U,

1.3.5.7 Member: PeripheralPllK3StepUpChangeDelay

Table 182 PeripheralPllK3StepUpChangeDelay

Name	PeripheralPllK3StepUpChangeDelay	
Type	uint32	
Description	Delay for incrementing peripheral PLL K3 divider value	
Verification method	The structure is generated based on the value configured in McuClockSettingConf/ McuPeripheralPllSettingConfig/McuPerPllK3DivStepUpChangeDelay.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPerPllK3DivStepDownChangeDelay = 10	/* Peripheral PLL K3 divider increment step change delay */ 10U,
	McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPerPllK3DivStepDownChangeDelay = 50	/* Peripheral PLL K3 divider increment step change delay */ 50U,

1.3.5.8 Member: PeripheralPllK3StepDownChangeDelay

Table 183 PeripheralPllK3StepDownChangeDelay

Name	PeripheralPllK3StepDownChangeDelay	
Type	uint32	
Description	Delay for decrementing peripheral PLL K3 divider value	
Verification method	The structure is generated based on the value configured in McuClockSettingConf/ McuPeripheralPllSettingConfig/McuPerPllK3DivStepDownChangeDelay.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPerPllK3DivStepDownChangeDelay = 10	/* Peripheral PLL K3 divider decrement step change delay */ 10U,
	McuClockSettingConf/ McuPeripheralPllSettingConfig /McuPerPllK3DivStepDownChangeDelay = 50	/* Peripheral PLL K3 divider decrement step change delay */ 50U,

1.3.5.9 Member: Mcu_kPllDistributionConfiguration_Config[_variant]

Table 184 Mcu_kPllDistributionConfiguration_Config[_variant]

Name	Mcu_kPllDistributionConfiguration_Config[_variant]	
Type	Mcu_PllDistributionConfigType	
Description	Pointer to the clock distribution configuration structure	
Verification method	The element is always generated as Mcu_kPllDistributionConfiguration_Config[_variant]	
Example(s)	**Action**	**Generated output**
	For variant unaware	`/* PLL clock divider configuration pointer */` `&Mcu_kPllDistributionConfiguration_Config_0,`
	For variant aware, variant name is Petrol	`/* PLL clock divider configuration pointer */` `&Mcu_kPllDistributionConfiguration_Config_Petrol_0,`

1.3.5.10 Member: ExternalClockCfg

Table 185 ExternalClockCfg

Name	ExternalClockCfg
Type	Mcu_ExternalClockConfigType
Description	External clock configuration value
Verification method	The element is always generated based on: Bit 0 is set if McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock0Enable is set to True else 0. Bits 2-5 are set based on the numeric value suffixed after '_SEL' keyword in McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClockOutSel0. Bit 16 is set if McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Enable is set to True else 0. Bit 17 is set if McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Inverted is set to False else 0. Bits 18-21 are set based on the numeric value suffixed after '_SEL' keyword in McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClockOutSel1. Bits 24-31 are set based on the value in McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Div substracted with 1 if SPB is selected as output.

	Other bits are set as 0 always.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock0Enable = True McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClockOutSel0 = OSC0_EXT_CLOCK0_SEL3 McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Enable = False McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Inverted = False	`/* External Clock configuration */ 0x0000000dU,`
	McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock0Enable = False McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClockOutSel1 = SPB_EXT_CLOCK1_SEL9 McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Enable = True McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Inverted = True McuClockSettingConf/ McuExternalClockOutputConfig/ McuExtClock1Div = 24	`/* External Clock configuration */ 0x17270000U,`

1.3.5.11 Member: BackupFreqKDiv

Table 186 BackupFreqKDiv

Name	BackupFreqKDiv
Type	uint16
Description	K divider value to reach backup clock frequency
Verification method	The element is generated based on following formula:

	1. NDiv = McuClockSettingConf/ McuPeripheralPllSettingConfig/ McuPeripheralPllNDivider+1 2. PDiv = McuClockSettingConf/ McuPeripheralPllSettingConfig/ McuPeripheralPllPDivider+1 3. KDivVal = (INSELFREQ*NDiv)/(1.6*PDiv*F_{BACK}) a. Where INSELFREQ = Oscillator frequency or Backup clock frequency based on which is selected in McuClockSettingConf/ McuSystemPllSettingConfig/ McuPllInputSrcSelection. b. F_{BACK}: Backup clock frequency 4. BackupFreqKDiv = KDivVal(systemPll) – 1 5. BackupFreqKDiv = BackupFreqKDiv | ((KDivVal(McuPeripheralPllK2Divider)-1)<<4) 6. BackupFreqKDiv = BackupFreqKDiv | ((KDivVal(McuPeripheralPllK3Divider)-1)<<8)	
Example(s)	**Action**	**Generated output**
	INSELFREQ = 20 MHz System Pll NDiv = 30 System Pll PDiv = 1 Peripheral Pll NDiv = 40 Peripheral Pll PDiv = 1	`/* Backup frequency K Divider value for both PLLs */` `0x0375U,`

1.3.5.12 Member: ConvCtrlBlockConf

Table 187 ConvCtrlBlockConf

Name	ConvCtrlBlockConf	
Type	uint8	
Description	Converter control block configuration value	
Verification method	The element is generated based on the numeric value suffixed after '_SEL' keyword in McuClockSettingConf/ McuPllDistributionSettingConfig/ McuConvCtrlPhaseSynchConf.	
Example(s)	**Action**	**Generated output**
	McuClockSettingConf/ McuPllDistributionSettingConfig/ McuConvCtrlPhaseSynchConf = PHASE_SYNCH_PER_FREQ_BY_3_SEL2	`/* Converter Control Phase Synchronization configuration */` `0x02U,`
	McuClockSettingConf/ McuPllDistributionSettingConfig/ McuConvCtrlPhaseSynchConf =	`/* Converter Control Phase Synchronization configuration */` `0x08U,`

	PHASE_SYNCH_PER_FREQ_BY_9_SEL8	

1.3.6 Structure: Mcu_kGtmClusterConfigPtr_Config[_<variant>]

Table 188 Mcu_kGtmClusterConfigPtr_Config[_<variant>]

Name	Mcu_kGtmClusterConfigPtr_Config[_<variant>]	
Type	Mcu_GtmClusterConfigType	
Description	Configuration structure for GTM clusters initialization.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored. The memebers are generated based on number of clusters available in the device.	
Example(s)	**Action**	**Generated output**

	Configure Clusters in GtmGlobalConfiguration/ GtmClusterConf (variant unaware)	`static const Mcu_GtmClusterConfigType Mcu_kGtmClusterConfigPtr_Config[12] =`
		`{`
		
		`  /*GTM Cluster_0 configuration*/`
		`  {`
		`  /*GTM cluster TIM/TOM/ATOM enable settings*/`
		`  0x00000000U,`
		`  /*GTM cluster config clock settings*/`
		`  0x00000000U,`
		`  /*GTM cluster fixed clock settings*/`
		`  0x00000000U`
		`  },`
		`  /*GTM Cluster_1 configuration*/`
		`  {`
		`  /*GTM cluster TIM/TOM/ATOM enable settings*/`
		`  0x00000000U,`
		`  /*GTM cluster config clock settings*/`
		`  0x00000000U,`
		`  /*GTM cluster fixed clock settings*/`
		`  0x00000000U`
		` },

```
  /*GTM Cluster_2 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_3 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_4 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_5 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
```

```
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_6 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_7 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_8 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_9 configuration*/
  {
```

	/*GTM cluster TIM/TOM/ATOM enable settings*/
	0x00000000U,
	/*GTM cluster config clock settings*/
	0x00000000U,
	/*GTM cluster fixed clock settings*/
	0x00000000U
	},
	/*GTM Cluster_10 configuration*/
	{
	/*GTM cluster TIM/TOM/ATOM enable settings*/
	0x00000000U,
	/*GTM cluster config clock settings*/
	0x00000000U,
	/*GTM cluster fixed clock settings*/
	0x00000000U
	},
	/*GTM Cluster_11 configuration*/
	{
	/*GTM cluster TIM/TOM/ATOM enable settings*/
	0x00000000U,
	/*GTM cluster config clock settings*/
	0x00000000U,
	/*GTM cluster fixed clock settings*/
	0x00000000U
	}
	};

Configure Clusters in GtmGlobalConfiguration/ GtmClusterConf (variant aware. Variant name is 'Petrol')	static const Mcu_GtmClusterConfigType Mcu_kGtmClusterConfigPtr_Config_Petrol[12] = { /*GTM Cluster_0 configuration*/ { /*GTM cluster TIM/TOM/ATOM enable

```
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_1 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_2 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_3 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
```

```
    0x00000000U
  },
  /*GTM Cluster_4 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_5 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_6 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_7 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
```

```
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_8 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_9 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
  /*GTM Cluster_10 configuration*/
  {
    /*GTM cluster TIM/TOM/ATOM enable
settings*/
    0x00000000U,
    /*GTM cluster config clock
settings*/
    0x00000000U,
    /*GTM cluster fixed clock
settings*/
    0x00000000U
  },
```

		/*GTM Cluster_11 configuration*/
		{
		/*GTM cluster TIM/TOM/ATOM enable settings*/
		0x00000000U,
		/*GTM cluster config clock settings*/
		0x00000000U,
		/*GTM cluster fixed clock settings*/
		0x00000000U
		}
		};

1.3.6.1 Member: Gtm_Cluster[ClusterIndex]

Table 189 Gtm_Cluster[ClusterIndex]

Name	Gtm_Cluster[ClusterIndex]
Type	Mcu_GtmClusterConfigType
Description	Structure to store individual cluster configuration
Verification method	The member is generated based on the number of GTM clusters available in the device. First element is generated based on: • Bit 0 is set if any TIM channel is used in McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuTimAllocationConf. • Bit 1 is set if any TOM channel is used in McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuTomAllocationConf. • Bit 2 is set if any ATOM channel is used in McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuAtomAllocationConf. Second element is generated with a value if any TIM, TOM or ATOM channel is used else 0. • Bits 0-1 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock0Src. • Bits 4-5 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock1Src. • Bits 8-9 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock2Src. • Bits 12-13 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/

GtmClusterConfClock3Src.

- Bits 16-17 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock4Src.
- Bits 20-21 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock5Src.
- Bits 24-25 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock6Src.
- Bits 28-29 are set based on numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock0Src.
- Other bits are set to 0 always.

Third element is generated based on the numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterFixedClockSetting/ GtmClusterFixedClockSrc.

Example(s)

Action	Generated output
McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuTimAllocationConf = GTM_TIM_CHANNEL_NOT_USED McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuTomAllocationConf = GTM_TOM_CHANNEL_USED_BY_PWM McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuAtomAllocationConf = GTM_ATOM_CHANNEL_NOT_USED GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock0Src = CMU_CONF_CLOCK0_SEL0 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock1Src =	/*GTM Cluster_0 configuration*/ { /*GTM cluster TIM/TOM/ATOM enable settings*/ 0x00000002U, /*GTM cluster config clock settings*/ 0x00021000U, /*GTM cluster fixed clock settings*/ 0x00000000U },

CMU_CONF_CLOCK0_SEL0 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock2Src = CMU_CONF_CLOCK0_SEL0 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock3Src = CMU_CONF_CLOCK8_SEL1 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock4Src = EXT_CAPTURE_SEL2 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock5Src = CMU_CONF_CLOCK0_SEL0 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock6Src = CMU_CONF_CLOCK0_SEL0 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock7Src = CMU_CONF_CLOCK0_SEL0 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterFixedClockSetting/ GtmClusterFixedClockSrc = CMU_FIXED_CLOCK0_SEL0	
McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuTimAllocationConf = GTM_TIM_CHANNEL_USED_BY_ICU McuHardwareResourceAllocationConf/ McuGtmAllocationConf/	/*GTM Cluster_0 configuration*/ { /*GTM cluster TIM/TOM/ATOM enable settings*/ 0x00000003U, /*GTM cluster config clock settings*/ 0x00021000U,

McuTomAllocationConf = GTM_TOM_CHANNEL_USED_BY_PWM
 	
McuHardwareResourceAllocationConf/ McuGtmAllocationConf/ McuAtomAllocationConf = GTM_ATOM_CHANNEL_NOT_USED
 	
GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock0Src = CMU_CONF_CLOCK0_SEL0
 	
GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock1Src = CMU_CONF_CLOCK0_SEL0
 	
GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock2Src = CMU_CONF_CLOCK0_SEL0
 	
GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock3Src = CMU_CONF_CLOCK8_SEL1
 	
GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock4Src = EXT_CAPTURE_SEL2
 	
GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock5Src = CMU_CONF_CLOCK0_SEL0
 	
GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock6Src = CMU_CONF_CLOCK0_SEL0
 	
GtmGlobalConfiguration/
	`/*GTM cluster fixed clock settings*/`
	`0x00000001U`
	`},`

	GtmClusterConf/ GtmClusterConfClockSetting/ GtmClusterConfClock7Src = CMU_CONF_CLOCK0_SEL0 GtmGlobalConfiguration/ GtmClusterConf/ GtmClusterFixedClockSetting/ GtmClusterFixedClockSrc = CMU_CONF_CLOCK8_SEL1	

1.3.7 Structure: Mcu_kGtmConfiguration_Config[_<variant>]

Table 190 Mcu_kGtmConfiguration_Config[_<variant>]

Name	Mcu_kGtmConfiguration_Config[_<variant>]	
Type	Mcu_GtmConfigType	
Description	Configuration structure for GTM global initialization.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**

	Configure GTM in GtmGlobalConfiguration (variant unaware)	`static const Mcu_GtmConfigType Mcu_kGtmConfiguration_Config =`
		`{`
		`  /* Ptr to GTM clock configuration - GtmClockCfgPtr */`
		`  &Mcu_kGtmClockConfigPtr_Config,`
		`  /* Ptr to GTM cluster configuration -  GtmClusterCfgPtr */`
		`  Mcu_kGtmClusterConfigPtr_Config,`
		`  /*Configuration for TOM global settings*/`
		`  {`
		`  /*Configuration for Tom global settings -  GtmTomCfg*/`
		
		`  /*GtmTomGlobalConf_0*/`
		`  {`
		`  /*TomTgcIntTrigRstCn0 value for group0*/`
		`  0x55555555U,`
		`  /*TomTgcActTb value for group0*/`
		`  0x00000001U`

```
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_1*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_2*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
```

```
    /*GtmTomGlobalConf_3*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_4*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_5*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
```

```
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    }
  },
  /*Configuration for ATOM global
settings*/
  {
    /*Configuration for Atom global
settings -  GtmAtomCfg*/

    /*GtmAtomGlobalConf_0*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_1*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_2*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_3*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
```

```
    0x00000001U
  },
  /*GtmAtomGlobalConf_4*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_5*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_6*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_7*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_8*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_9*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
```

```
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_10*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_11*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    }
  },
  /*Configuration for Gtm to Adc
trigger settings*/
  {
    /*Configuration of Gtm Adc
trigger 0*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 1*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 2*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
```

```
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 3*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 4*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    }
  },
  /*Configuration for Gtm to Dsadc
trigger settings*/
  {
    /*Configuration of Gtm Dsadc
trigger 0*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
      /*GtmDsadcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm
Dsadc trigger 1*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
      /*GtmDsadcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm
Dsadc trigger 2*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
      /*GtmDsadcOut1 value */
      0x00000000U
```

```
    },    /*Configuration of Gtm
Dsadc trigger 3*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
      /*GtmDsadcOut1 value */
      0x00000000U
    }
  },
  /*Configuration for Timer to Port
connections*/
  {
    /*Configuration of ToutSel*/
    /*Toutsel 0 value*/
    0x00000004U,
    /*Toutsel 1 value*/
    0x00006400U,
    /*Toutsel 2 value*/
    0x00000000U,
    /*Toutsel 3 value*/
    0x00000000U,
    /*Toutsel 4 value*/
    0x00000000U,
    /*Toutsel 5 value*/
    0x00000000U,
    /*Toutsel 6 value*/
    0x00000000U,
    /*Toutsel 7 value*/
    0x00000000U,
    /*Toutsel 8 value*/
    0x00000000U,
    /*Toutsel 9 value*/
    0x00000000U,
    /*Toutsel 10 value*/
    0x00000000U,
    /*Toutsel 11 value*/
    0x00000000U,
    /*Toutsel 12 value*/
    0x00000000U,
    /*Toutsel 13 value*/
```

	0x00000000U, /*Toutsel 14 value*/ 0x00a00080U, /*Toutsel 15 value*/ 0x00050000U, /*Toutsel 16 value*/ 0x00000000U, /*Toutsel 17 value*/ 0x00000000U, /*Toutsel 18 value*/ 0x00080000U, /*Toutsel 19 value*/ 0x00000000U, /*Toutsel 20 value*/ 0x00000000U, /*Toutsel 21 value*/ 0x00000000U, /*Toutsel 22 value*/ 0x00000000U, /*Toutsel 23 value*/ 0x00000000U, /*Toutsel 24 value*/ 0x00000000U, /*Toutsel 25 value*/ 0x00000000U, /*Toutsel 26 value*/ 0x00000000U, /*Toutsel 27 value*/ 0x00000000U, /*Toutsel 28 value*/ 0x00000000U, /*Toutsel 29 value*/ 0x00000000U, /*Toutsel 30 value*/ 0x00000000U, /*Toutsel 31 value*/ 0x00000000U, /*Toutsel 32 value*/ 0x00000000U, /*Toutsel 33 value*/

```
    0x00000000U
  },
  /*Mask for TOUTSEL configuration.
SELx used will be generated as 0xF*/
  {
    /*Toutsel 0 mask value*/
    0x0000000fU,
    /*Toutsel 1 mask value*/
    0x0000ff00U,
    /*Toutsel 2 mask value*/
    0x00000000U,
    /*Toutsel 3 mask value*/
    0x00000000U,
    /*Toutsel 4 mask value*/
    0x00000000U,
    /*Toutsel 5 mask value*/
    0x00000000U,
    /*Toutsel 6 mask value*/
    0x00000000U,
    /*Toutsel 7 mask value*/
    0x00000000U,
    /*Toutsel 8 mask value*/
    0x00000000U,
    /*Toutsel 9 mask value*/
    0x00000000U,
    /*Toutsel 10 mask value*/
    0x00000000U,
    /*Toutsel 11 mask value*/
    0x00000000U,
    /*Toutsel 12 mask value*/
    0x00000000U,
    /*Toutsel 13 mask value*/
    0x00000000U,
    /*Toutsel 14 mask value*/
    0x00f000f0U,
    /*Toutsel 15 mask value*/
    0x000f0000U,
    /*Toutsel 16 mask value*/
    0x00000000U,
    /*Toutsel 17 mask value*/
```

```
    0x00000000U,
    /*Toutsel 18 mask value*/
    0x000f0000U,
    /*Toutsel 19 mask value*/
    0x00000000U,
    /*Toutsel 20 mask value*/
    0x00000000U,
    /*Toutsel 21 mask value*/
    0x00000000U,
    /*Toutsel 22 mask value*/
    0x00000000U,
    /*Toutsel 23 mask value*/
    0x00000000U,
    /*Toutsel 24 mask value*/
    0x00000000U,
    /*Toutsel 25 mask value*/
    0x00000000U,
    /*Toutsel 26 mask value*/
    0x00000000U,
    /*Toutsel 27 mask value*/
    0x00000000U,
    /*Toutsel 28 mask value*/
    0x00000000U,
    /*Toutsel 29 mask value*/
    0x00000000U,
    /*Toutsel 30 mask value*/
    0x00000000U,
    /*Toutsel 31 mask value*/
    0x00000000U,
    /*Toutsel 32 mask value*/
    0x00000000U,
    /*Toutsel 33 mask value*/
    0x00000000U
  },
  /*Configuration for Port to Timer
Connections*/
  {
    /*Configuration of TimInSel*/
    /*TimInsel 0 value*/
    0x00060402U,
```

	/*TimInsel 1 value*/ 0x00000000U, /*TimInsel 2 value*/ 0x00907000U, /*TimInsel 3 value*/ 0x00000000U, /*TimInsel 4 value*/ 0x00000000U, /*TimInsel 5 value*/ 0x00c0000aU, /*TimInsel 6 value*/ 0x00000000U, /*TimInsel 7 value*/ 0x00000000U }, /*Configuration for TBU channel - GtmTbuCfg (GtmTbuChannelConf)*/ 0x00000000U, /* TOM modules used configuration */ 0x0000U, /* ATOM modules used configuration */ 0x0000U, /* Gtm sleep mode configuration */ (boolean)FALSE };
Configure GTM in GtmGlobalConfiguration (variant aware. Variant name is 'Petrol')	static const Mcu_GtmConfigType Mcu_kGtmConfiguration_Config_Petrol = { /* Ptr to GTM clock configuration - GtmClockCfgPtr */ &Mcu_kGtmClockConfigPtr_Config_Petrol, /* Ptr to GTM cluster configuration - GtmClusterCfgPtr */ Mcu_kGtmClusterConfigPtr_Config_Petrol, /*Configuration for TOM global settings*/ {

```
    /*Configuration for Tom global
settings -  GtmTomCfg*/


    /*GtmTomGlobalConf_0*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_1*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_2*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
```

```
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_3*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    },
    /*GtmTomGlobalConf_4*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
```

```
      0x00000001U
    },
    /*GtmTomGlobalConf_5*/
    {
      /*TomTgcIntTrigRstCn0 value for
group0*/
      0x55555555U,
      /*TomTgcActTb value for
group0*/
      0x00000001U
    },
    {
      /*TomTgcIntTrigRstCn1 value for
group1*/
      0x55555555U,
      /*TomTgcActTb value for
group1*/
      0x00000001U
    }
  },
  /*Configuration for ATOM global
settings*/
  {
    /*Configuration for Atom global
settings -  GtmAtomCfg*/

    /*GtmAtomGlobalConf_0*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_1*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_2*/
    {
```

```
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_3*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_4*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_5*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_6*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
  /*GtmAtomGlobalConf_7*/
  {
    /*AtomTgcIntTrigRstCn0 value*/
    0x55555555U,
    /*AtomTgcActTb value */
    0x00000001U
  },
```

```
    /*GtmAtomGlobalConf_8*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_9*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_10*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    },
    /*GtmAtomGlobalConf_11*/
    {
      /*AtomTgcIntTrigRstCn0 value*/
      0x55555555U,
      /*AtomTgcActTb value */
      0x00000001U
    }
  },

  /*Configuration for Gtm to Adc
trigger settings*/
  {
    /*Configuration of Gtm Adc
trigger 0*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
```

```
trigger 1*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 2*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 3*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 4*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    }
  },
  /*Configuration for Gtm to Dsadc
trigger settings*/
  {
    /*Configuration of Gtm Adc
trigger 0*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 1*/
    {
```

```
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 2*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm Adc
trigger 3*/
    {
      /*GtmAdcOut0 value*/
      0x00000000U,
      /*GtmAdcOut1 value */
      0x00000000U
    },
  },
  /*Configuration for Gtm to Dsadc
trigger settings*/
  {
    /*Configuration of Gtm Dsadc
trigger 0*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
      /*GtmDsadcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm
Dsadc trigger 1*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
      /*GtmDsadcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm
Dsadc trigger 2*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
```

```
      /*GtmDsadcOut1 value */
      0x00000000U
    },    /*Configuration of Gtm
Dsadc trigger 3*/
    {
      /*GtmDsadcOut0 value*/
      0x00000000U,
      /*GtmDsadcOut1 value */
      0x00000000U
    }
  },
  /*Configuration for Timer to Port
connections*/
  {
    /*Configuration of ToutSel*/
    /*Toutsel 0 value*/
    0x00000004U,
    /*Toutsel 1 value*/
    0x00006400U,
    /*Toutsel 2 value*/
    0x00000000U,
    /*Toutsel 3 value*/
    0x00000000U,
    /*Toutsel 4 value*/
    0x00000000U,
    /*Toutsel 5 value*/
    0x00000000U,
    /*Toutsel 6 value*/
    0x00000000U,
    /*Toutsel 7 value*/
    0x00000000U,
    /*Toutsel 8 value*/
    0x00000000U,
    /*Toutsel 9 value*/
    0x00000000U,
    /*Toutsel 10 value*/
    0x00000000U,
    /*Toutsel 11 value*/
    0x00000000U,
    /*Toutsel 12 value*/
```

```
0x00000000U,
/*Toutsel 13 value*/
0x00000000U,
/*Toutsel 14 value*/
0x00a00080U,
/*Toutsel 15 value*/
0x00050000U,
/*Toutsel 16 value*/
0x00000000U,
/*Toutsel 17 value*/
0x00000000U,
/*Toutsel 18 value*/
0x00080000U,
/*Toutsel 19 value*/
0x00000000U,
/*Toutsel 20 value*/
0x00000000U,
/*Toutsel 21 value*/
0x00000000U,
/*Toutsel 22 value*/
0x00000000U,
/*Toutsel 23 value*/
0x00000000U,
/*Toutsel 24 value*/
0x00000000U,
/*Toutsel 25 value*/
0x00000000U,
/*Toutsel 26 value*/
0x00000000U,
/*Toutsel 27 value*/
0x00000000U,
/*Toutsel 28 value*/
0x00000000U,
/*Toutsel 29 value*/
0x00000000U,
/*Toutsel 30 value*/
0x00000000U,
/*Toutsel 31 value*/
0x00000000U,
/*Toutsel 32 value*/
```

```
    0x00000000U,
    /*Toutsel 33 value*/
    0x00000000U
  },
  /*Mask for TOUTSEL configuration.
SELx used will be generated as 0xF*/
  {
    /*Toutsel 0 mask value*/
    0x0000000fU,
    /*Toutsel 1 mask value*/
    0x0000ff00U,
    /*Toutsel 2 mask value*/
    0x00000000U,
    /*Toutsel 3 mask value*/
    0x00000000U,
    /*Toutsel 4 mask value*/
    0x00000000U,
    /*Toutsel 5 mask value*/
    0x00000000U,
    /*Toutsel 6 mask value*/
    0x00000000U,
    /*Toutsel 7 mask value*/
    0x00000000U,
    /*Toutsel 8 mask value*/
    0x00000000U,
    /*Toutsel 9 mask value*/
    0x00000000U,
    /*Toutsel 10 mask value*/
    0x00000000U,
    /*Toutsel 11 mask value*/
    0x00000000U,
    /*Toutsel 12 mask value*/
    0x00000000U,
    /*Toutsel 13 mask value*/
    0x00000000U,
    /*Toutsel 14 mask value*/
    0x00f000f0U,
    /*Toutsel 15 mask value*/
    0x000f0000U,
    /*Toutsel 16 mask value*/
```

```
    0x00000000U,
    /*Toutsel 17 mask value*/
    0x00000000U,
    /*Toutsel 18 mask value*/
    0x000f0000U,
    /*Toutsel 19 mask value*/
    0x00000000U,
    /*Toutsel 20 mask value*/
    0x00000000U,
    /*Toutsel 21 mask value*/
    0x00000000U,
    /*Toutsel 22 mask value*/
    0x00000000U,
    /*Toutsel 23 mask value*/
    0x00000000U,
    /*Toutsel 24 mask value*/
    0x00000000U,
    /*Toutsel 25 mask value*/
    0x00000000U,
    /*Toutsel 26 mask value*/
    0x00000000U,
    /*Toutsel 27 mask value*/
    0x00000000U,
    /*Toutsel 28 mask value*/
    0x00000000U,
    /*Toutsel 29 mask value*/
    0x00000000U,
    /*Toutsel 30 mask value*/
    0x00000000U,
    /*Toutsel 31 mask value*/
    0x00000000U,
    /*Toutsel 32 mask value*/
    0x00000000U,
    /*Toutsel 33 mask value*/
    0x00000000U
  },
/*Configuration for Port to Timer
Connections*/
  {
    /*Configuration of TimInSel*/
```

		/*TimInsel 0 value*/ 0x00060402U, /*TimInsel 1 value*/ 0x00000000U, /*TimInsel 2 value*/ 0x00907000U, /*TimInsel 3 value*/ 0x00000000U, /*TimInsel 4 value*/ 0x00000000U, /*TimInsel 5 value*/ 0x00c0000aU, /*TimInsel 6 value*/ 0x00000000U, /*TimInsel 7 value*/ 0x00000000U }, /*Configuration for TBU channel - GtmTbuCfg (GtmTbuChannelConf)*/ 0x011400aaU, /* TOM modules used configuration */ 0x0000U, /* ATOM modules used configuration */ 0x0000U, /* Gtm sleep mode configuration */ (boolean)FALSE };

1.3.7.1 Member: Mcu_kGtmClockConfigPtr_ Config[_<variant>]

Table 191 Mcu_kGtmClockConfigPtr_ Config[_<variant>]

Name	Mcu_kGtmClockConfigPtr_ Config[_<variant>]	
Type	Mcu_GtmClockSettingType	
Description	Pointer to GTM clock configuration structure	
Verification method	The generated structure member is present in the Mcu_kGtmConfiguration_Config[_<variant>] structure. It is always generated as pointer to Mcu_GtmClockSettingType structure (&Mcu_kGtmClockConfigPtr_ Config[_<variant>])	
Example(s)	**Action**	**Generated output**

	Generate Mcu_PBcfg.c (variant unaware)	/* Ptr to GTM clock configuration - GtmClockCfgPtr */ &Mcu_kGtmClockConfigPtr_Config,
	Generate Mcu_PBcfg.c (variant aware, variant name is Petrol)	/* Ptr to GTM clock configuration - GtmClockCfgPtr */ &Mcu_kGtmClockConfigPtr_Config_Petrol,

1.3.7.2 Member: Mcu_kGtmClusterConfigPtr_ Config[_<variant>]

Table 192 Mcu_kGtmClusterConfigPtr_ Config[_<variant>]

Name	Mcu_kGtmClusterConfigPtr_Config[_<variant>]	
Type	Mcu_GtmClusterConfigType	
Description	Pointer to GTM cluster configuration structure	
Verification method	The generated structure member is present in the Mcu_kGtmConfiguration_Config[_<variant>] structure. It is always generated as pointer to Mcu_GtmClusterConfigType (&Mcu_GtmClusterConfigType_ Config[_<variant>])	
Example(s)	**Action**	**Generated output**
	Generate Mcu_PBcfg.c (variant unaware)	/* Ptr to GTM cluster configuration - GtmClusterCfgPtr */ Mcu_kGtmClusterConfigPtr_Config,
	Generate Mcu_PBcfg.c (variant aware, variant name is Petrol)	/* Ptr to GTM cluster configuration - GtmClusterCfgPtr */ Mcu_kGtmClusterConfigPtr_Config_Petrol,

1.3.7.3 Member: GtmTomCfg[MCU_GTM_NO_OF_TOM_AVAILABLE * 2]

Table 193 GtmTomCfg[MCU_GTM_NO_OF_TOM_AVAILABLE * 2]

Name	GtmTomCfg[MCU_GTM_NO_OF_TOM_AVAILABLE * 2]
Type	Mcu_GtmTomConfigType
Description	Array to store GTM Tom global Configurations
Verification method	The array is generated with a size of number of TGCs available. First element is generated based on: • If GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChInternalTriggerEnable is set to True If TomChannelId < 8 Group0: ConfigVal = ConfigVal | (0x2 << (TomChannelId*2)) if TomChannelId > 8 Group1: ConfigVal = ConfigVal | (0x2 << ((TomChannelId – 8)*2)) • Else

If TomChannelId < 8 Group0: ConfigVal = ConfigVal | (0x1 << (TomChannelId*2))

if TomChannelId > 8 Group1: ConfigVal = ConfigVal | (0x1 << ((TomChannelId – 8)*2))

- If GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChResetCn0OnTriggerEnable is set to True.

If TomChannelId < 8 Group0: ConfigVal = ConfigVal | (0x2 << (TomChannelId*2 + 16))

if TomChannelId > 8 Group1: ConfigVal = ConfigVal | (0x2 << ((TomChannelId – 8)*2 + 16))

- Else

If TomChannelId < 8 Group0: ConfigVal = ConfigVal | (0x1 << (TomChannelId*2 + 16))

if TomChannelId > 8 Group1: ConfigVal = ConfigVal | (0x1 << ((TomChannelId – 8)*2 + 16))

Second element is generated based on:

- Bits 0-24 are configured based on GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseValue
- Bits 25-26 are configured based on numeric value suffixed after '_TS' keyword in GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseSelection
- Others bits are set to 0 always.

Example(s)

Action	Generated output
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChInternalTriggerEnable for channels 0-5 = True GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChInternalTriggerEnable for channels 6-7 = False GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChInternalTriggerEnable for channels 8-15 = True GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChResetCn0OnTriggerEnable for channels 0-5 = False GtmGlobalConfiguration/ GtmTomGlobalConf/	/*Configuration for Tom global settings - GtmTomCfg*/ /*GtmTomGlobalConf_0*/ { /*TomTgcIntTrigRstCn0 value for group0*/ 0xa5555aaaU, /*TomTgcActTb value for group0*/ 0x0400ffffU }, { /*TomTgcIntTrigRstCn1 value for group1*/ 0x5555aaaaU, /*TomTgcActTb value for group1*/ 0x00003039U },

GtmTomChannelConf/ GtmTomChResetCn0OnTrigger Enable for channels 6-7 = True GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChResetCn0OnTrigger Enable for channels 8-15 = False GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseValue for Group0 = 65535 GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseSelection for Group 0= TOM_ACT_TB_TBU_TS2 GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseValue for Group1 = 12345 GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseSelection for Group 1= TOM_ACT_TB_TBU_TS0	

GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChInternalTriggerEnable for channels 0-5 = False
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChInternalTriggerEnable for channels 6-7 = True
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChInternalTriggerEna
	`/*Configuration for Tom global settings - GtmTomCfg*/`
	
	`/*GtmTomGlobalConf_0*/`
	`{`
	`/*TomTgcIntTrigRstCn0 value for group0*/`
	`0x5aaaaa555U,`
	`/*TomTgcActTb value for group0*/`
	`0x0400ffffU`
	`},`
	`{`

ble for channels 8-15 = False
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChResetCn0OnTrigger Enable for channels 0-5 = True
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChResetCn0OnTrigger Enable for channels 6-7 = False
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomChannelConf/ GtmTomChResetCn0OnTrigger Enable for channels 8-15 = True
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseValue for Group 0 = 65535
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseSelection for Group 0 = TOM_ACT_TB_TBU_TS2
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseValue for Group 1 = 1
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseSelection for Group 1 = TOM_ACT_TB_TBU_TS0
	`/*TomTgcIntTrigRstCn1 value for group1*/`
	`0xaaaa5555U,`
	`/*TomTgcActTb value for group1*/`
	`0x00000001U`
	`},`

1.3.7.4 Member: GtmAtomCfg[MCU_GTM_NO_OF_ATOM_AVAILABLE]

Table 194 GtmAtomCfg[MCU_GTM_NO_OF_ATOM_AVAILABLE]

Name	GtmAtomCfg[MCU_GTM_NO_OF_ATOM_AVAILABLE]
Type	Mcu_GtmAtomConfigType

Description	Array to store GTM Atom global Configurations
Verification method	First element is generated based on: • If GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChInternalTriggerEnable is set to True. ConfigVal = ConfigVal | (0x2 << (AtomChannelId*2)) • Else ConfigVal = ConfigVal | (0x1 << (AtomChannelId*2)) • If GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChResetCn0OnTriggerEnable is set to True. ConfigVal = ConfigVal | (0x2 << (AtomChannelId *2 + 16)) • Else ConfigVal = ConfigVal | (0x1 << (AtomChannelId *2 + 16)) Second element is generated based on: • Bits 0-24 are configured based on GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomGroupConf/ GtmAtomActionTimeBaseValue • Bits 25-26 are configured based on numeric value suffixed after '_TS' keyword in GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomGroupConf/ GtmAtomActionTimeBaseSelection • Others bits are set to 0 always.

Example(s)	**Action**	**Generated output**

	GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChInternalTriggerEnable for channels 0-5 = True
	GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChInternalTriggerEnable for channels 6-7 = False
	GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChResetCn0OnTriggerEnable for channels 0-5 = False
	GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChResetCn0OnTriggerEnable for channels 6-7 = True
	GtmGlobalConfiguration/ GtmTomGlobalConf/
		`/*Configuration for Atom global settings - GtmAtomCfg*/`
		`/*GtmAtomGlobalConf_0*/`
		`{`
		`/*AtomTgcIntTrigRstCn0 value*/`
		`0xa5555aaaU,`
		`/*AtomTgcActTb value */`
		`0x00003039U`
		`},`

GtmTomGroupConf/ GtmTomActionTimeBaseValue = 12345 GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseSelection = ATOM_ACT_TB_TBU_TS0	

GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChInternalTriggerEnable for channels 0-5 = False
 	
GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChInternalTriggerEnable for channels 6-7 = True
 	
GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChResetCn0OnTriggerEnable for channels 0-5 = True
 	
GtmGlobalConfiguration/ GtmAtomGlobalConf/ GtmAtomChannelConf/ GtmAtomChResetCn0OnTriggerEnable for channels 6-7 = False
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseValue = 65535
 	
GtmGlobalConfiguration/ GtmTomGlobalConf/ GtmTomGroupConf/ GtmTomActionTimeBaseSelection = ATOM_ACT_TB_TBU_TS2
	`/*Configuration for Atom global settings - GtmAtomCfg*/`
	
	`/*GtmAtomGlobalConf_0*/`
	`{`
	`/*AtomTgcIntTrigRstCn0 value*/`
	`0x5aaaa555U,`
	`/*AtomTgcActTb value */`
	`0x0400ffffU`
	`},`

1.3.7.5 Member: GtmAdcTrigCfg[MCU_NO_OF_GTM_ADC_TRIGGER]

Table 195 GtmAdcTrigCfg[MCU_NO_OF_GTM_ADC_TRIGGER]

Name	GtmAdcTrigCfg[MCU_NO_OF_GTM_ADC_TRIGGER]	
Type	Mcu_GtmAdcTrigType	
Description	Array to store GTM to ADC trigger configurations	
Verification method	The member is generated based on following algorithm: AdcNo = numeric value suffixed after 'Adc_' keyword in GtmGlobalConfiguration/ GtmTrrigerForAdc Value = numeric value suffixed after '_' keyword in GtmGlobalConfiguration/ GtmTrrigerForAdc_(AdcNo)/ GtmAdcTriggerSelect First element is generated as (Value << (4*AdcNo)). First element holds the value of SEL0 – SEL7 of Trigger0 Second element is generated as (Value << (4*AdcNo) - 32). Second element holds the value of SEL0 – SEL3 of Trigger1	
Example(s)	**Action**	**Generated output**

	GtmGlobalConfiguration/ GtmTrrigerForAdc_0/ GtmAdcTrigger0Select = TRIG_3
	GtmGlobalConfiguration/ GtmTrrigerForAdc_10/ GtmAdcTrigger0Select = TRIG_3
		`/*Configuration of Gtm Adc trigger 0*/`
		`{`
		`/*GtmAdcOut0 value*/`
		`0x00000003U,`
		`/*GtmAdcOut1 value */`
		`0x00000300U`
		`},`

	GtmGlobalConfiguration/ GtmTrrigerForAdc_6/ GtmAdcTrigger0Select = TRIG_10
	GtmGlobalConfiguration/ GtmTrrigerForAdc_11/ GtmAdcTrigger0Select = TRIG_5
		`/*Configuration for Gtm to Adc trigger settings*/`
		`{`
		`/*Configuration of Gtm Adc trigger 0*/`
		`{`
		`/*GtmAdcOut0 value*/`
		`0x0a000000U,`
		`/*GtmAdcOut1 value */`
		`0x00005000U`
		`},`

1.3.7.6 Member: GtmDsadcTrigCfg[MCU_NO_OF_GTM_DSADC_TRIGGER]

Table 196 GtmDsadcTrigCfg[MCU_NO_OF_GTM_DSADC_TRIGGER]

Name	GtmDsadcTrigCfg[MCU_NO_OF_GTM_DSADC_TRIGGER]
Type	Mcu_GtmDsadcTrigType
Description	Array to store GTM to DSADC trigger configurations

Verification method	The member is generated based on following algorithm: DsadcNo = numeric value suffixed after 'Dsadc_' keyword in GtmGlobalConfiguration/ GtmTrrigerForDsadc Value = numeric value suffixed after 'TRIG_' keyword in GtmGlobalConfiguration/ GtmTrrigerForDsadc_(DsadcNo)/ GtmDsadcTrigger[x]Select (where x:0-3) First element is generated as (Value << (4*DsadcNo)). First element holds the value of SEL0 – SEL7 Second element is generated as (Value << (4*DsadcNo) - 32). Second element holds the value of SEL8 – SEL13

Example(s)

Action	Generated output

GtmGlobalConfiguration/ GtmTriggerForDsadc_0/ GtmDsadcTrigger0Select = TRIG_2_TOM_0_13
GtmGlobalConfiguration/ GtmTriggerForDsadc_10/ GtmDsadcTrigger0Select = TRIG_2_TOM_0_13
	`/*Configuration of Gtm to Dsadc trigger 0*/`
	`{`
	`/*GtmDsadcOut0 value*/`
	`0x00000002U,`
	`/*GtmDsadcOut1 value */`
	`0x00000200U`
	`},`

GtmGlobalConfiguration/ GtmTriggerForDsadc_6/ GtmDsadcTrigger0Select = TRIG_10_ATOM_1_6
GtmGlobalConfiguration/ GtmTriggerForDsadc_11/ GtmDsadcTrigger0Select = TRIG_5_ATOM_0_5
	`/*Configuration for Gtm to Dsadc trigger settings*/`
	`{`
	`/*Configuration of Gtm Dsadc trigger 0*/`
	`{`
	`/*GtmDsadcOut0 value*/`
	`0x0a000000U,`
	`/*GtmDsadcOut1 value */`
	`0x00005000U`
	`},`

1.3.7.7 Member: GtmToutSelCfg[MCU_GTM_NO_OF_TOUTSEL_AVAILABLE]

Table 197 GtmToutSelCfg[MCU_GTM_NO_OF_TOUTSEL_AVAILABLE]

Name	GtmToutSelCfg[MCU_GTM_NO_OF_TOUTSEL_AVAILABLE]
Type	uint32
Description	Array to store Gtm to port pin connection configuration
Verification method	The member is generated based on following algorithm: A loop is run for number of TOUTSEL registers available in the hardware. A second loop is run for the number of Gtm to port pins configured.

	The values are generated based on the selections in "GtmGlobalConfiguration/GtmTomGlobalConf_x/GtmTomChannelConf_y/ GtmTimerPortPinSelect" and "GtmGlobalConfiguration/GtmAtomGlobalConf_x/GtmAtomChannelConf_y/ GtmTimerPortPinSelect". The value of 'x' specifies the TOM/ATOM module number and the value of 'y' specifies the channel number. TOUTSEL register number and SEL number is extracted from the selection based on following algorithm: TOUTSEL register index value = (Numeric value after "TOUT" and before "_SEL") / 8 SEL number = (Numeric value after "TOUT" and before "_SEL") % 8 If the TOUTSEL register index value is equal to first loop's count value is generated based on following algorithm: SelValue = Numeric value after "SEL[x]_" and before "_PORT" where [x]: A-L SelIndex = SEL number * 4 TOUTSEL register value = TOUTSEL register value | (SelValue << SelIndex) The value of 'NONE' is programmed as 0.

Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_0/ GtmTimerPortPinSelect = TOUT0_SELE_4_PORT02_PIN0 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_1/ GtmTimerPortPinSelect = TOUT10_SELE_4_PORT00_PIN1 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_4/ GtmTimerPortPinSelect = TOUT124_SELF_5_PORT11_PIN8 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_5/ GtmTimerPortPinSelect = TOUT11_SELG_6_PORT00_PIN2 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomCh	/*Configuration for Timer to Port connections*/ { /*Configuration of ToutSel*/ /*Toutsel 0 value*/ 0x00000004U, /*Toutsel 1 value*/ 0x00006400U, /*Toutsel 2 value*/ 0x00000000U, /*Toutsel 3 value*/ 0x00000000U, /*Toutsel 4 value*/ 0x00000000U, /*Toutsel 5 value*/ 0x00000000U, /*Toutsel 6 value*/ 0x00000000U, /*Toutsel 7 value*/ 0x00000000U, /*Toutsel 8 value*/ 0x00000000U,

annelConf_1/ GtmTimerPortPinSelect = TOUT113_SELI_8_PORT01_PIN5 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_2/ GtmTimerPortPinSelect = TOUT117_SELK_10_PORT02_PIN10 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_5/ GtmTimerPortPinSelect = TOUT148_SELI_8_PORT34_PIN3 Rest all are set to NONE	/*Toutsel 9 value*/ 0x00000000U, /*Toutsel 10 value*/ 0x00000000U, /*Toutsel 11 value*/ 0x00000000U, /*Toutsel 12 value*/ 0x00000000U, /*Toutsel 13 value*/ 0x00000000U, /*Toutsel 14 value*/ 0x00a00080U, /*Toutsel 15 value*/ 0x00050000U, /*Toutsel 16 value*/ 0x00000000U, /*Toutsel 17 value*/ 0x00000000U, /*Toutsel 18 value*/ 0x00080000U, /*Toutsel 19 value*/ 0x00000000U, /*Toutsel 20 value*/ 0x00000000U, /*Toutsel 21 value*/ 0x00000000U, /*Toutsel 22 value*/ 0x00000000U, /*Toutsel 23 value*/ 0x00000000U, /*Toutsel 24 value*/ 0x00000000U, /*Toutsel 25 value*/ 0x00000000U, /*Toutsel 26 value*/ 0x00000000U, /*Toutsel 27 value*/ 0x00000000U, /*Toutsel 28 value*/ 0x00000000U,

	/*Toutsel 29 value*/ 0x00000000U, /*Toutsel 30 value*/ 0x00000000U, /*Toutsel 31 value*/ 0x00000000U, /*Toutsel 32 value*/ 0x00000000U, /*Toutsel 33 value*/ 0x00000000U }
GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_0/ GtmTimerPortPinSelect = TOUT0_SELE_4_PORT02_PIN0 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_1/ GtmTimerPortPinSelect = TOUT10_SELE_4_PORT00_PIN1 GtmGlobalConfiguration/GtmTomGlobalConf_2/GtmTomChannelConf_15/ GtmTimerPortPinSelect = TOUT37_SELB_1_PORT32_PIN1 GtmGlobalConfiguration/GtmTomGlobalConf_5/GtmTomChannelConf_7/ GtmTimerPortPinSelect = TOUT213_SELB_1_PORT25_PIN7 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_1/ GtmTimerPortPinSelect = TOUT113_SELI_8_PORT01_PIN5 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_2/ GtmTimerPortPinSelect =	/*Configuration for Timer to Port connections*/ { /*Configuration of ToutSel*/ /*Toutsel 0 value*/ 0x00000004U, /*Toutsel 1 value*/ 0x00000400U, /*Toutsel 2 value*/ 0x00000000U, /*Toutsel 3 value*/ 0x00000000U, /*Toutsel 4 value*/ 0x00100000U, /*Toutsel 5 value*/ 0x00000000U, /*Toutsel 6 value*/ 0x00000000U, /*Toutsel 7 value*/ 0x00000000U, /*Toutsel 8 value*/ 0x00000000U, /*Toutsel 9 value*/ 0x00000000U, /*Toutsel 10 value*/ 0x00000000U, /*Toutsel 11 value*/ 0x00000000U, /*Toutsel 12 value*/

TOUT117_SELK_10_PORT02_PIN10 GtmGlobalConfiguration/GtmAtomGlobalConf_2/GtmAtomChannelConf_2/ GtmTimerPortPinSelect = TOUT106_SELK_10_PORT10_PIN4 Rest all are set to NONE	0x00000000U, /*Toutsel 13 value*/ 0x00000a00U, /*Toutsel 14 value*/ 0x00a00080U, /*Toutsel 15 value*/ 0x00000000U, /*Toutsel 16 value*/ 0x00000000U, /*Toutsel 17 value*/ 0x00000000U, /*Toutsel 18 value*/ 0x00000000U, /*Toutsel 19 value*/ 0x00000000U, /*Toutsel 20 value*/ 0x00000000U, /*Toutsel 21 value*/ 0x00000000U, /*Toutsel 22 value*/ 0x00000000U, /*Toutsel 23 value*/ 0x00000000U, /*Toutsel 24 value*/ 0x00000000U, /*Toutsel 25 value*/ 0x00000000U, /*Toutsel 26 value*/ 0x00100000U, /*Toutsel 27 value*/ 0x00000000U, /*Toutsel 28 value*/ 0x00000000U, /*Toutsel 29 value*/ 0x00000000U, /*Toutsel 30 value*/ 0x00000000U, /*Toutsel 31 value*/ 0x00000000U, /*Toutsel 32 value*/

	0x00000000U, /*Toutsel 33 value*/ 0x00000000U }

1.3.7.8 Member: GtmToutSelCfgMsk[MCU_GTM_NO_OF_TOUTSEL_AVAILABLE]

Table 198 GtmToutSelCfgMsk[MCU_GTM_NO_OF_TOUTSEL_AVAILABLE]

Name	GtmToutSelCfgMsk[MCU_GTM_NO_OF_TOUTSEL_AVAILABLE]	
Type	uint32	
Description	Array to store mask for used Gtm to port pin connection configuration	
Verification method	The member is generated based on following algorithm: A loop is run for number of TOUTSEL registers available in the hardware. A second loop is run for the number of Gtm to port pins configured. The values are generated based on the selections in "GtmGlobalConfiguration/GtmTomGlobalConf_x/GtmTomChannelConf_y/ GtmTimerPortPinSelect" and "GtmGlobalConfiguration/GtmAtomGlobalConf_x/GtmAtomChannelConf_y/ GtmTimerPortPinSelect". The value of 'x' specifies the TOM/ATOM module number and the value of 'y' specifies the channel number. TOUTSEL register number and SEL number is extracted from the selection based on following algorithm: TOUTSEL register index value = (Numeric value after "TOUT" and before "_SEL") / 8 SEL number = (Numeric value after "TOUT" and before "_SEL") % 8 If the TOUTSEL register index value is equal to first loop's count value is generated based on following algorithm: SelValue = 0xf SelIndex = SEL number * 4 TOUTSEL register value = TOUTSEL register value | (SelValue << SelIndex) The value of 'NONE' is programmed as 0.	
Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_0/ GtmTimerPortPinSelect = TOUT0_SELE_4_PORT02_PIN0 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_1/ GtmTimerPortPinSelect =	/*Mask for TOUTSEL configuration. SELx used will be generated as 0xF*/ { /*Toutsel 0 mask value*/ 0x0000000fU, /*Toutsel 1 mask value*/ 0x0000ff00U, /*Toutsel 2 mask value*/

TOUT10_SELE_4_PORT00_PIN1 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_4/ GtmTimerPortPinSelect = TOUT124_SELF_5_PORT11_PIN8 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_5/ GtmTimerPortPinSelect = TOUT11_SELG_6_PORT00_PIN2 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_1/ GtmTimerPortPinSelect = TOUT113_SELI_8_PORT01_PIN5 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_2/ GtmTimerPortPinSelect = TOUT117_SELK_10_PORT02_PIN10 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_5/ GtmTimerPortPinSelect = TOUT148_SELI_8_PORT34_PIN3 Rest all are set to NONE	0x00000000U, /*Toutsel 3 mask value*/ 0x00000000U, /*Toutsel 4 mask value*/ 0x00000000U, /*Toutsel 5 mask value*/ 0x00000000U, /*Toutsel 6 mask value*/ 0x00000000U, /*Toutsel 7 mask value*/ 0x00000000U, /*Toutsel 8 mask value*/ 0x00000000U, /*Toutsel 9 mask value*/ 0x00000000U, /*Toutsel 10 mask value*/ 0x00000000U, /*Toutsel 11 mask value*/ 0x00000000U, /*Toutsel 12 mask value*/ 0x00000000U, /*Toutsel 13 mask value*/ 0x00000000U, /*Toutsel 14 mask value*/ 0x00f000f0U, /*Toutsel 15 mask value*/ 0x000f0000U, /*Toutsel 16 mask value*/ 0x00000000U, /*Toutsel 17 mask value*/ 0x00000000U, /*Toutsel 18 mask value*/ 0x000f0000U, /*Toutsel 19 mask value*/ 0x00000000U, /*Toutsel 20 mask value*/ 0x00000000U, /*Toutsel 21 mask value*/ 0x00000000U, /*Toutsel 22 mask value*/

	0x00000000U, /*Toutsel 23 mask value*/ 0x00000000U, /*Toutsel 24 mask value*/ 0x00000000U, /*Toutsel 25 mask value*/ 0x00000000U, /*Toutsel 26 mask value*/ 0x00000000U, /*Toutsel 27 mask value*/ 0x00000000U, /*Toutsel 28 mask value*/ 0x00000000U, /*Toutsel 29 mask value*/ 0x00000000U, /*Toutsel 30 mask value*/ 0x00000000U, /*Toutsel 31 mask value*/ 0x00000000U, /*Toutsel 32 mask value*/ 0x00000000U, /*Toutsel 33 mask value*/ 0x00000000U }
GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_0/ GtmTimerPortPinSelect = TOUT0_SELE_4_PORT02_PIN0 GtmGlobalConfiguration/GtmTomGlobalConf_0/GtmTomChannelConf_1/ GtmTimerPortPinSelect = TOUT10_SELE_4_PORT00_PIN1 GtmGlobalConfiguration/GtmTomGlobalConf_2/GtmTomChannelConf_15/ GtmTimerPortPinSelect = TOUT37_SELB_1_PORT32_PIN1 GtmGlobalConfiguration/GtmT	/*Mask for TOUTSEL configuration. SELx used will be generated as 0xF*/ { /*Toutsel 0 mask value*/ 0x0000000fU, /*Toutsel 1 mask value*/ 0x00000f00U, /*Toutsel 2 mask value*/ 0x00000000U, /*Toutsel 3 mask value*/ 0x00000000U, /*Toutsel 4 mask value*/ 0x00f00000U, /*Toutsel 5 mask value*/ 0x00000000U, /*Toutsel 6 mask value*/

omGlobalConf_5/GtmTomChannelConf_7/ GtmTimerPortPinSelect = TOUT213_SELB_1_PORT25_PIN7 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_1/ GtmTimerPortPinSelect = TOUT113_SELI_8_PORT01_PIN5 GtmGlobalConfiguration/GtmAtomGlobalConf_0/GtmAtomChannelConf_2/ GtmTimerPortPinSelect = TOUT117_SELK_10_PORT02_PIN10 GtmGlobalConfiguration/GtmAtomGlobalConf_2/GtmAtomChannelConf_2/ GtmTimerPortPinSelect = TOUT106_SELK_10_PORT10_PIN4 Rest all are set to NONE	0x00000000U, /*Toutsel 7 mask value*/ 0x00000000U, /*Toutsel 8 mask value*/ 0x00000000U, /*Toutsel 9 mask value*/ 0x00000000U, /*Toutsel 10 mask value*/ 0x00000000U, /*Toutsel 11 mask value*/ 0x00000000U, /*Toutsel 12 mask value*/ 0x00000000U, /*Toutsel 13 mask value*/ 0x00000f00U, /*Toutsel 14 mask value*/ 0x00f000f0U, /*Toutsel 15 mask value*/ 0x00000000U, /*Toutsel 16 mask value*/ 0x00000000U, /*Toutsel 17 mask value*/ 0x00000000U, /*Toutsel 18 mask value*/ 0x00000000U, /*Toutsel 19 mask value*/ 0x00000000U, /*Toutsel 20 mask value*/ 0x00000000U, /*Toutsel 21 mask value*/ 0x00000000U, /*Toutsel 22 mask value*/ 0x00000000U, /*Toutsel 23 mask value*/ 0x00000000U, /*Toutsel 24 mask value*/ 0x00000000U, /*Toutsel 25 mask value*/ 0x00000000U, /*Toutsel 26 mask value*/

	0x00f00000U, /*Toutsel 27 mask value*/ 0x00000000U, /*Toutsel 28 mask value*/ 0x00000000U, /*Toutsel 29 mask value*/ 0x00000000U, /*Toutsel 30 mask value*/ 0x00000000U, /*Toutsel 31 mask value*/ 0x00000000U, /*Toutsel 32 mask value*/ 0x00000000U, /*Toutsel 33 mask value*/ 0x00000000U }

1.3.7.9 Member: GtmTimInSelCfg [MCU_GTM_NO_OF_TIM_AVAILABLE]

Table 199 GtmTimInSelCfg [MCU_GTM_NO_OF_TIM_AVAILABLE]

Name	GtmTimInSelCfg [MCU_GTM_NO_OF_TIM_AVAILABLE]	
Type	uint32	
Description	Array to store port pin to GtmTim connection configuration	
Verification method	The member is generated based on following algorithm: A loop is run for number of TIM modules available in the hardware. A second loop is run for the number of TIM channels. The values are generated based on the selections in “GtmGlobalConfiguration/GtmTimGlobalConf_x/GtmTimChannelConf_y/ GtmTimInpPortPinSel”. The value of ‘x’ specifies the TIM module number and the value of ‘y’ specifies the channel number. TIMINSEL register number and SEL number is extracted from the selection based on following algorithm: TIMINSEL register index value = First LoopIndex (TIM module number) TimChIdx = Second LoopIndex(TIM channel number) SEL number = (Numeric value after “SEL” and before “_PORT”) TIMINSEL register value = TIMINSEL register value | (SEL number << (TimChIdx * 4)) The value of ‘SEL0_NONE’ is programmed as 0.	
Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/GtmT	/*Configuration for Port to Timer

imGlobalConf_0/GtmTimChannelConf_0/ GtmTimInpPortPinSel = SEL2_PORT02_PIN0
 	
GtmGlobalConfiguration/GtmTimGlobalConf_0/GtmTimChannelConf_2/ GtmTimInpPortPinSel = SEL4_PORT10_PIN5
 	
GtmGlobalConfiguration/GtmTimGlobalConf_0/GtmTimChannelConf_4/ GtmTimInpPortPinSel = SEL6_PORT33_PIN0
 	
GtmGlobalConfiguration/GtmTimGlobalConf_2/GtmTimChannelConf_3/ GtmTimInpPortPinSel = SEL7_PORT01_PIN5
 	
GtmGlobalConfiguration/GtmTimGlobalConf_2/GtmTimChannelConf_5/ GtmTimInpPortPinSel = SEL9_PORT34_PIN3
 	
GtmGlobalConfiguration/GtmTimGlobalConf_5/GtmTimChannelConf_0/ GtmTimInpPortPinSel = SEL10_PORT01_PIN8
 	
GtmGlobalConfiguration/GtmTimGlobalConf_5/GtmTimChannelConf_5/ GtmTimInpPortPinSel = SEL12_PORT21_PIN3
 	
Rest all are set to SEL0_NONE
	Connections*/
	{
	/*Configuration of TimInSel*/
	/*TimInsel 0 value*/
	0x00060402U,
	/*TimInsel 1 value*/
	0x00000000U,
	/*TimInsel 2 value*/
	0x00907000U,
	/*TimInsel 3 value*/
	0x00000000U,
	/*TimInsel 4 value*/
	0x00000000U,
	/*TimInsel 5 value*/
	0x00c0000aU,
	/*TimInsel 6 value*/
	0x00000000U,
	/*TimInsel 7 value*/
	0x00000000U
	}

GtmGlobalConfiguration/GtmTimGlobalConf_0/GtmTimChannelConf_1/ GtmTimInpPortPinSel = SEL10_PORT10_PIN9 GtmGlobalConfiguration/GtmTimGlobalConf_0/GtmTimChannelConf_5/ GtmTimInpPortPinSel =	/*Configuration for Port to Timer Connections*/ { /*Configuration of TimInSel*/ /*TimInsel 0 value*/ 0x507000a0U, /*TimInsel 1 value*/ 0x00000000U,

SEL7_PORT21_PIN7
 	
GtmGlobalConfiguration/GtmTimGlobalConf_0/GtmTimChannelConf_7/ GtmTimInpPortPinSel = SEL5_PORT23_PIN4
 	
GtmGlobalConfiguration/GtmTimGlobalConf_2/GtmTimChannelConf_0/ GtmTimInpPortPinSel = SEL14_PORT01_PIN3
 	
GtmGlobalConfiguration/GtmTimGlobalConf_2/GtmTimChannelConf_3/ GtmTimInpPortPinSel = SEL9_PORT34_PIN1
 	
GtmGlobalConfiguration/GtmTimGlobalConf_6/GtmTimChannelConf_0/ GtmTimInpPortPinSel = SEL9_PORT31_PIN8
 	
GtmGlobalConfiguration/GtmTimGlobalConf_6/GtmTimChannelConf_5/ GtmTimInpPortPinSel = SEL6_PORT25_PIN5
 	
Rest all are set to SEL0_NONE
	`/*TimInsel 2 value*/`
	`0x0000900eU,`
	`/*TimInsel 3 value*/`
	`0x00000000U,`
	`/*TimInsel 4 value*/`
	`0x00000000U,`
	`/*TimInsel 5 value*/`
	`0x00000000U,`
	`/*TimInsel 6 value*/`
	`0x00600009U,`
	`/*TimInsel 7 value*/`
	`0x00000000U`
	`}`

1.3.7.10 Member: GtmTbuCfg

Table 200 GtmTbuCfg

Name	GtmTbuCfg
Type	uint32
Description	Variable to store GTM TBU configuration
Verification method	The member is generated based on following algorithm: A loop is run for number of nodes configured in GtmGlobalConfiguration/ GtmTbuChannelConf. If GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChannelEnable is set to True • Loopcntr = 0 • GtmTbuCfg = GtmTbuCfg | (0x2 << 2*Loopcntr) • If Loopcntr = 0, GtmTbuCfg = GtmTbuCfg | (numeric value suffixed after 'SEL' keyword in GtmGlobalConfiguration/ GtmTbuChannelConf/

	GtmTbuChResolutionSel << (8 + 4*Loopcntr)) • If Loopcntr = 3, GtmTbuCfg = GtmTbuCfg | (1 << (8 + 4*Loopcntr)) • If Loopcntr = 1 or 2, GtmTbuCfg = GtmTbuCfg | (numeric value suffixed after 'SEL' keyword in GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChMode << (8 + 4*Loopcntr)) • If Loopcntr < 3, GtmTbuCfg = GtmTbuCfg | (numeric value suffixed after 'SEL' keyword in GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChClockSourceSelection << (9 + 4*Loopcntr)) • If Loopcntr = 3, GtmTbuCfg = GtmTbuCfg | (numeric value suffixed after 'SEL' keyword in GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChModuloCntrSel << (12 + 4*Loopcntr))	
Example(s)	**Action**	**Generated output**
	4 nodes are configured in GtmGlobalConfiguration/ GtmTbuChannelConf GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChannelEnable = True GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChResolutionSel = TBU_CH_LOWER_COUNT_BITS_SEL0 GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChMode = FREE_RUNNING_COUNTER_SEL0 GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChClockSourceSelection = CMU_CLOCK1_SEL1 GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChModuloCntrSel = TBU_CH2_SEL1	`/*Configuration for TBU channel - GtmTbuCfg (GtmTbuChannelConf)*/` `0x011422aaU,`
	4 nodes are configured in GtmGlobalConfiguration/ GtmTbuChannelConf GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChannelEnable = False GtmGlobalConfiguration/	`/*Configuration for TBU channel - GtmTbuCfg (GtmTbuChannelConf)*/` `0x01042200U,`

	GtmTbuChannelConf/ GtmTbuChResolutionSel = TBU_CH_LOWER_COUNT_BITS_SEL0 GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChMode = FREE_RUNNING_COUNTER_SEL0 GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChClockSourceSelection = CMU_CLOCK1_SEL1 GtmGlobalConfiguration/ GtmTbuChannelConf/ GtmTbuChModuloCntrSel = TBU_CH2_SEL1	

1.3.7.11 Member: GtmTomModuleUsage

Table 201 GtmTomModuleUsage

Name	GtmTomModuleUsage	
Type	uint16	
Description	Indicates which TOM module has been reserved	
Verification method	The member is generated based on: Instead, the member is generated as bit-wise representation of TOM modules in use. Bit x corresponds to TOMx If TOM<x> in use, the bit is set else 0, where x = 0 to number of TOM available in the device.	
Example(s)	**Action**	**Generated output**
	TOMs 0, 1 and 2 are reserved by PWM in McuHardwareResourceAllocationConf.	`/* TOM modules used configuration */` `0x0007U,`
	TOMs 0-5 are reserved by PWM in McuHardwareResourceAllocationConf.	`/* TOM modules used configuration */` `0x001fU,`

1.3.7.12 Member: GtmAtomModuleUsage

Table 202 GtmAtomModuleUsage

Name	GtmAtomModuleUsage
Type	uint16
Description	Indicates which ATOM module has been reserved
Verification method	The member is generated based on: Instead, the member is generated as bit-wise representation of ATOM modules in use. Bit x corresponds to ATOMx If ATOM<x> in use, the bit is set else 0, where x = 0 to number of ATOM available in the device.

Example(s)	**Action**	**Generated output**
	ATOMs 0, 1 and 2 are reserved by PWM in McuHardwareResourceAllocationConf.	`/* ATOM modules used configuration */` `0x0007U,`
	ATOMs 0-11 are reserved by PWM in McuHardwareResourceAllocationConf.	`/* ATOM modules used configuration */` `0x07ffU,`

1.3.7.13 Member: IsGtmSleepModeEnabled

Table 203 IsGtmSleepModeEnabled

Name	IsGtmSleepModeEnabled
Type	boolean
Description	Indicates whether GTM sleep mode is enabled or disabled
Verification method	The member is generated as True if McuGeneralConfiguration/ McuGtmSleepModeEnabled is set to True else is generated as False

Example(s)	**Action**	**Generated output**
	McuGeneralConfiguration/ McuGtmSleepModeEnabled = True	`/* Gtm sleep mode configuration */` `(boolean)TRUE`
	McuGeneralConfiguration/ McuGtmSleepModeEnabled = False	`/* Gtm sleep mode configuration */` `(boolean)FALSE`

1.3.8 Structure: Mcu_kGtmClockConfigPtr_Config[_<variant>]

Table 204 Mcu_kGtmClockConfigPtr_Config[_<variant>]

Name	Mcu_kGtmClockConfigPtr_Config[_<variant>]
Type	Mcu_GtmClockSettingType

Description	Configuration structure for GTM clock initialization.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**

	Configure GTM in GtmGlobalConfiguration (variant unaware)	`static const Mcu_GtmClockSettingType Mcu_kGtmClockConfigPtr_Config =`
		`{`
		`  /*CMU config clock, external and fixed clock enable - GtmCmuClockEnable*/`
		`  0x0080aaaaU,`
		`  /*CMU global clock numerator - GtmCmuGlobalNumerator*/`
		`  0x00000001U,`
		`  /*CMU global clock denominator - GtmCmuGlobalDenominator*/`
		`  0x00000001U,`
		
		`  /*CMU config clock_0...7 Numerator and Denominator - GtmCmuConfClkCtrl*/`
		`  {`
		`  0x00000000U,`
		`  0x00000000U,`
		`  0x00000000U,`
		`  0x00000000U,`
		`  0x00000000U,`
		`  0x00000000U,`
		`  0x00000000U,`
		`  0x00000000U`
		`  },`
		
		`  /*CMU fixed clock Divider selection - GtmCmuFixedClkCtrl*/`
		`  0x00000000U,`
		
		`  /*GTM cluster input clock divider configuration - GtmCmuClusterInputClockDividerEnable*/`
		`  0x00aaaaaaU,`

	/* External clock settings - GtmEclkCtrl*/
	{
	/*External Clock_0 disabled - reset value of numerator and denominator*/
	{1U, 1U},
	/*External Clock_1 disabled - reset value of numerator and denominator*/
	{1U, 1U},
	/*External Clock_2 disabled - reset value of numerator and denominator*/
	{1U, 1U}
	}
	};

Configure GTM in GtmGlobalConfiguration (variant aware. Variant name is 'Petrol')	static const Mcu_GtmClockSettingType Mcu_kGtmClockConfigPtr_Config_Petrol = { /*CMU config clock, external and fixed clock enable - GtmCmuClockEnable*/ 0x0080aaaaU, /*CMU global clock numerator - GtmCmuGlobalNumerator*/ 0x00000001U, /*CMU global clock denominator - GtmCmuGlobalDenominator*/ 0x00000001U, /*CMU config clock_0...7 Numerator and Denominator - GtmCmuConfClkCtrl*/ { 0x00000000U, 0x00000000U, 0x00000000U, 0x00000000U, 0x00000000U, 0x00000000U, 0x00000000U, 0x00000000U },

```
  /*CMU fixed clock Divider selection
- GtmCmuFixedClkCtrl*/
    0x00000000U,

  /*GTM cluster input clock divider
configuration -
GtmCmuClusterInputClockDividerEnable
*/
  0x00aaaaa2U,

  /* External clock settings -
GtmEclkCtrl*/
  {
    /*External Clock_0 disabled -
reset value of numerator and
denominator*/
    {1U, 1U},
    /*External Clock_1 disabled -
reset value of numerator and
denominator*/
    {1U, 1U},
    /*External Clock_2 disabled -
reset value of numerator and
denominator*/
    {1U, 1U}
  }
};
```

1.3.8.1 Member: GtmCmuClockEnable

Table 205 GtmCmuClockEnable

Name	GtmCmuClockEnable
Type	uint32
Description	Indicates the value of configurable clocks to be enabled
Verification method	The member is generated based on: Bits 0-1 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock0Enable is set to True else 0. Bits 2-3 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock1Enable is set to True else 0. Bits 4-5 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/

	GtmCmuConfigClock2Enable is set to True else 0. Bits 6-7 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock3Enable is set to True else 0. Bits 8-9 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock4Enable is set to True else 0. Bits 10-11 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock5Enable is set to True else 0. Bits 12-13 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock6Enable is set to True else 0. Bits 14-15 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock7Enable is set to True else 0. Bits 16-17 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock0Enable is set to True else 0. Bits 18-19 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock1Enable is set to True else 0. Bits 20-21 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock2Enable is set to True else 0. Bits 22-23 are written with 0x2 if GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuFixedClockEnable is set to True else 0. Other bits are always set to 0.

Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock0Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock1Enable = False GtmGlobalConfiguration/ McuGtmClockManagementCon	`/*CMU config clock, external and fixed clock enable - GtmCmuClockEnable*/` `0x00a22222U,`

f/ GtmConfigClockSetting/ GtmCmuConfigClock2Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock3Enable = False GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock4Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock5Enable = False GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock6Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock7Enable = False GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock0Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock1Enable = False GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock2Enable = True GtmGlobalConfiguration/ McuGtmClockManagementCon	

f/ GtmConfigClockSetting/ GtmCmuFixedClockEnable = True	
GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock0Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock1Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock2Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock3Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock4Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock5Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock6Enable = True GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuConfigClock7Enable = True GtmGlobalConfiguration/ McuGtmClockManagementCon	`/*CMU config clock, external and fixed clock enable - GtmCmuClockEnable*/` `0x0082aaaaU,`

	f/ GtmConfigClockSetting/ GtmCmuExtClock0Enable = False GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock1Enable = False GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuExtClock2Enable = False GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmConfigClockSetting/ GtmCmuFixedClockEnable = True	

1.3.8.2 Member: GtmCmuGlobalNumerator

Table 206 GtmCmuGlobalNumerator

Name	GtmCmuGlobalNumerator	
Type	uint32	
Description	Indicates the value of global numerator for CMU	
Verification method	The member is generated based on value in GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmCmuGlobalClockNumerator.	
Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmCmuGlobalClockNumerator = 65536	`/*CMU global clock numerator - GtmCmuGlobalNumerator*/` `0x00010000U,`
	GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmCmuGlobalClockNumerator = 1234568	`/*CMU global clock numerator - GtmCmuGlobalNumerator*/` `0x0012d688U,`

1.3.8.3 Member: GtmCmuGlobalDenominator

Table 207 GtmCmuGlobalDenominator

Name	GtmCmuGlobalDenominator

Type	uint32	
Description	Indicates the value of global denominator for CMU	
Verification method	The member is generated based on value in GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmCmuGlobalClockDenominator.	
Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmCmuGlobalClockDenominator = 65536	`/*CMU global clock numerator - GtmCmuGlobalDenominator*/` `0x00010000U,`
	GtmGlobalConfiguration/ McuGtmClockManagementConf/ GtmCmuGlobalClockDenominator = 1234568	`/*CMU global clock numerator - GtmCmuGlobalDenominator*/` `0x0012d688U,`

1.3.8.4 Member: GtmCmuConfClkCtrl[MCU_GTM_NO_OF_CFGCLK]

Table 208 GtmCmuConfClkCtrl[MCU_GTM_NO_OF_CFGCLK]

Name	GtmCmuConfClkCtrl[MCU_GTM_NO_OF_CFGCLK]	
Type	uint32	
Description	Indicates the value of GTM configurable clocks 0-7 divider values	
Verification method	Each element is generated based on value in GtmGlobalConfiguration/ McuGtmClockMangementConf/GtmConfigClockSetting/GtmCmuConfigClock0Div if GtmGlobalConfiguration/ McuGtmClockMangementConf/GtmConfigClockSetting/ GtmCmuConfigClock<x>Enable is set to True else as 0, where <x> = 0 to 7.	
Example(s)	**Action**	**Generated output**

	GtmGlobalConfiguration/ McuGtmClockMangementConf/GtmConfigClockSetting/ GtmCmuConfigClock0Enable = True
	GtmGlobalConfiguration/ McuGtmClockMangementConf/GtmConfigClockSetting/GtmCmuConfigClock0Div = 12345
	GtmGlobalConfiguration/ McuGtmClockMangementConf/GtmConfigClockSetting/ GtmCmuConfigClock1Enable =
		`/*CMU config clock_0...7 Numerator and Denominator - GtmCmuConfClkCtrl*/`
		`{`
		`0x00003039U,`
		`0x00003039U,`
		`0x00003039U,`
		`0x00003039U,`
		`0x00003039U,`
		`0x00003039U,`
		`0x00003039U,`
		`0x00003039U`
		`},`

True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock1Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock2Enable = True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock2Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock3Enable = True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock3Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock4Enable = True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock4Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock5Enable = True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock5Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock6Enable =	

True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock6Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock7Enable = True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock7Div = 12345	

GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock0Enable = True
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock0Div = 12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock1Enable = False
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock1Div = 12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock2Enable = True
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock2Div = 12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/
	`/*CMU config clock_0...7 Numerator and Denominator - GtmCmuConfClkCtrl*/`
	`{`
	`0x00003039U,`
	`0U,`
	`0x00003039U,`
	`0U,`
	`0x00003039U,`
	`0U,`
	`0x00003039U,`
	`0U`
	`},`

	GtmCmuConfigClock3Enable = False GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock3Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock4Enable = True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock4Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock5Enable = False GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock5Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock6Enable = True GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock6Div = 12345 GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/ GtmCmuConfigClock7Enable = False GtmGlobalConfiguration/ McuGtmClockMangementConf /GtmConfigClockSetting/GtmC muConfigClock7Div = 12345	

1.3.8.5 Member: GtmCmuFixedClkCtrl

Table 209 GtmCmuFixedClkCtrl

Name	GtmCmuFixedClkCtrl	
Type	uint32	
Description	Indicates the value of fixed clock divider value	
Verification method	The member is generated based on the numeric value suffixed after 'SEL' keyword in GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockSel if GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockEnable is set to True else as 0.	
Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockEnable = True GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockSel = CMU_CLOCK1_SEL2	`/*CMU fixed clock Divider selection - GtmCmuFixedClkCtrl*/` `0x00000002U,`
	GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockEnable = True GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockSel = CMU_CLOCK5_SEL6	`/*CMU fixed clock Divider selection - GtmCmuFixedClkCtrl*/` `0x00000006U,`

	GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockEnable = False
	GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmFixedClockSetting/ GtmCmuFixedClockSel = CMU_CLOCK5_SEL6
		`/*CMU fixed clock Divider selection - GtmCmuFixedClkCtrl*/`
		`/*Fixed Clock disabled - reset value*/`
		`0U,`

1.3.8.6 Member: GtmCmuClsInDiv

Table 210 GtmCmuClsInDiv

Name	GtmCmuClsInDiv	
Type	uint32	
Description	Indicates the value of input cluster clock divider	
Verification method	The member is generated based on: DivVal = numeric value suffixed after '_SEL' keyword in GtmGlobalConfiguration/ GtmClusterConf/ GtmCmuClusterInputClockDividerEnable. Loop for number of clusters present in the device ClusterClkDivVal = ClusterClkDivVal | (DivVal << 2*ClusterIndex) Where ClusterIndex = 0 to (Number of clusters available in device-1).	
Example(s)	**Action**	**Generated output**
	GtmGlobalConfiguration/ GtmClusterConf/ GtmCmuClusterInputClockDividerEnable = CLS_CLK_CFG_ENABLED_WITH_DIV_SEL2 for all clusters	/*GTM cluster input clock divider configuration - GtmCmuClusterInputClockDividerEnable */ 0x00aaaaaaU,
	GtmGlobalConfiguration/ GtmClusterConf/ GtmCmuClusterInputClockDividerEnable = CLS_CLK_CFG_ENABLED_WITH_DIV_SEL2 for cluster 0 and disabled for clusters 1-11	/*GTM cluster input clock divider configuration - GtmCmuClusterInputClockDividerEnable */ 0x00000002U,

1.3.8.7 Member: GtmEclkCtrl[MCU_GTM_NO_OF_EXTCLK]

Table 211 GtmEclkCtrl[MCU_GTM_NO_OF_EXTCLK]

Name	GtmEclkCtrl[MCU_GTM_NO_OF_EXTCLK]
Type	Mcu_GtmExtClkType
Description	Indicates the value of numerator and denominator for external clocks 0-2
Verification method	Each member <x> is an array of 2 elements generated based on value in • GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmExtClockSetting / GtmCmuExtClock[x]Numerator • And GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmExtClockSetting / GtmCmuExtClock[x]Denominator if GtmGlobalConfiguration/ McuGtmClockMangementConf/ GtmExtClockSetting / GtmCmuExtClock[x]Enable is set to True else is 1.

	Where [x]: 0-2	
Example(s)	**Action**	**Generated output**

	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock0Enable = True
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock0Numerator = 12345
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock0Denominator =12345
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock1Enable = True
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock1Numerator = 12345
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock1Denominator =12345
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock2Enable = True
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock2Numerator = 12345
	
	GtmGlobalConfiguration/ McuGtmClockMangementConf
		`/* External clock settings - GtmEclkCtrl*/`
		`{`
		`/*External clock_0 Numerator and Denominator*/`
		`{12345U, 12345U},`
		`/*External clock_1 Numerator and Denominator*/`
		`{12345U, 12345U},`
		`/*External clock_2 Numerator and Denominator*/`
		`{12345U, 12345U}`
		`}`

/ GtmExtClockSetting / GtmCmuExtClock2Denominator =12345	

GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock0Enable = False
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock0Numerator = 12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock0Denominator =12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock1Enable = True
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock1Numerator = 12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock1Denominator =12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock2Enable = False
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf / GtmExtClockSetting / GtmCmuExtClock2Numerator = 12345
 	
GtmGlobalConfiguration/ McuGtmClockMangementConf
	`/* External clock settings - GtmEclkCtrl*/`
	`{`
	`/*External Clock_0 disabled - reset value of numerator and denominator*/`
	`{1U, 1U},`
	`/*External clock_1 Numerator and Denominator*/`
	`{12345U, 12345U},`
	`/*External Clock_2 disabled - reset value of numerator and denominator*/`
	`{1U, 1U}`
	`}`

	/ GtmExtClockSetting / GtmCmuExtClock2Denominator =12345	

1.3.9 Structure: Mcu_kGpt12PrescalerConfiguration_Config[_<variant>]

Table 212 Mcu_kGpt12PrescalerConfiguration_Config[_<variant>]

Name	Mcu_kGpt12PrescalerConfiguration_Config[_<variant>]	
Type	Mcu_Gpt12ConfigType	
Description	Configuration structure for GPT12 prescaler configuration.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant aware configuration the structure name is appended with the variant name. For variant unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**

	Configure GPT12 in McuGpt12PrescalerConf (variant unaware)	`#if (MCU_GPT1_USED == STD_ON) || (MCU_GPT2_USED == STD_ON)`
		`/* GPT12 Prescalar configuration structure */`
		`static const Mcu_Gpt12ConfigType Mcu_kGpt12PrescalerConfiguration_Config =`
		`{`
		`  /*GPT Block 1 Prescalar */`
		`  0x00U,`
		`  /*GPT Block 2 Prescalar */`
		`  0x00U,`
		`};`
		`#endif`

	Configure GPT12 in McuGpt12PrescalerConf (variant aware. Variant name is ‘Petrol’)	`#if (MCU_GPT1_USED == STD_ON) || (MCU_GPT2_USED == STD_ON)`
		`/* GPT12 Prescalar configuration structure */`
		`static const Mcu_Gpt12ConfigType Mcu_kGpt12PrescalerConfiguration_Config_Petrol =`
		`{`
		`  /*GPT Block 1 Prescalar */`
		`  0x00U,`
		`  /*GPT Block 2 Prescalar */`
		`  0x00U,`
		`};`

		#endif

1.3.9.1 Member: Gpt1PrescalarDiv

Table 213 Gpt1PrescalarDiv

Name	Gpt1PrescalarDiv	
Type	unsigned_int	
Description	2 bit value indicating the block prescaler selected for GPT block 1	
Verification method	The member is generated based on: A value of 0 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt1BlockPrescalerSel' is set to 'GPT1_PRESCALING_FACTOR_8' A value of 1 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt1BlockPrescalerSel' is set to 'GPT1_PRESCALING_FACTOR_4' A value of 2 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt1BlockPrescalerSel' is set to 'GPT1_PRESCALING_FACTOR_32' A value of 3 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt1BlockPrescalerSel' is set to 'GPT1_PRESCALING_FACTOR_16'	
Example(s)	**Action**	**Generated output**
	McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt1BlockPrescalerSel = GPT1_PRESCALING_FACTOR_8	`/*GPT Block 1 Prescalar */` `0x00U,`
	McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt1BlockPrescalerSel = GPT1_PRESCALING_FACTOR_16	`/*GPT Block 1 Prescalar */` `0x03U,`

1.3.9.2 Member: Gpt2PrescalarDiv

Table 214 Gpt2PrescalarDiv

Name	Gpt2PrescalarDiv
Type	unsigned_int
Description	2 bit value indicating the block prescaler selected for GPT block 2
Verification method	The member is generated based on: A value of 0 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt2BlockPrescalerSel' is set to 'GPT2_PRESCALING_FACTOR_4' A value of 1 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt2BlockPrescalerSel' is set to 'GPT2_PRESCALING_FACTOR_2' A value of 2 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/

	Gpt2BlockPrescalerSel' is set to 'GPT2_PRESCALING_FACTOR_16' A value of 3 is generated if 'McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt2BlockPrescalerSel' is set to 'GPT2_PRESCALING_FACTOR_8'	
Example(s)	**Action**	**Generated output**
	McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt2BlockPrescalerSel = GPT2_PRESCALING_FACTOR_2	`/*GPT Block 2 Prescalar */` `0x01U,`
	McuModuleConfiguration/ McuGpt12PrescalerConf/ Gpt2BlockPrescalerSel = GPT2_PRESCALING_FACTOR_16	`/*GPT Block 2 Prescalar */` `0x02U,`

1.4 File: Mcu_17_TimerIp_Cfg.c

The generated source file contains all pre-compile configuration parameters for userdata of TimerIp. The file is generated in 'src' folder.

1.4.1 Info: Mcu_17_Timer_DrivFuncCallbackLst

The array is defined in Mcu_17_TimerIp.c file. The array stores the unique-id of MCU users and the array index defines the User id for that module. Below are the users and ids of MCU:

- Adc id = 1
- Wdg id = 2
- Pwm id = 3
- Gpt id = 4
- Icu id = 5
- Ocu id = 6
- Dsadc id = 7
- Stm id = 8.

1.4.2 Structure: Mcu_17_Eru_ChUserData[MCU_17_ERU_NO_OF_OGU]

Table 215 Mcu_17_Eru_ChUserData[MCU_17_ERU_NO_OF_OGU]

Name	Mcu_17_Eru_ChUserData[MCU_17_ERU_NO_OF_OGU]
Type	uint32
Description	User data structure indicating the users of ERU.

Verification method	The generated structure is present in Mcu_17_TimerIp_Cfg.c file. The number of members in the structure depends on number of OGUs available in the ERU.
	The initial value of each element in Userdata[] array is calculated as below
	• Bits 16-19 store the user id of the user of that ERU.
	• Bits 8-11 store the logical channel id of the respective user. Refer section 1.4.1 for list of User ids.
	• Others bits are always 0.
	Final values are calculated based on following steps:
	A loop is run for number of OGUs available / 2
	• Erudata1 = Userdata[Loopindex] & 0xFF00FF
	• Erudata2 = Userdata[Loopindex + 4] & 0xFF00FF
	• If Erudata1 and Erudata2 are equal to 0,
	EruMask = 0
	• Else
	if((Erudata1&0xFF0000)>0) EruMask1=1 else EruMask1=0
	if((Erudata2&0xFF0000)>0) EruMask2=1 else EruMask2=0
	• EruMask = (EruMask1 << (EruData1 & 0xFF)) | (EruMask2 << (EruData2 & 0xFF))
	• If Erudata1 = 0 and EruMask !=0
	Erudata1 = ((EruMask << 24) | 0xFF) | Userdata[Loopindex]
	• Else

	Erudata1 = (EruMask << 24) | Userdata[Loopindex] • If Erudata2 = 0 and EruMask !=0 Erudata2 = ((EruMask << 24) | 0xFF) | Userdata[Loopindex + 4] • Else Erudata2 = (EruMask << 0x24) | Userdata[Loopindex + 4] • Userdata[Loopindex] = Erudata1 • Userdata[Loopindex + 4] = Erudata2	
Example(s)	**Action**	**Generated output**
	Reserve OGU 0 for ICU Userid for ICU is 5 Logical channel id for Icu is 0 ERS reference is 0	`const uint32 Mcu_17_Eru_ChUserData[MCU_17_ERU_NO_OF_OGU] = { 0x01050000, 0x00000000, 0x00000000, 0x00000000, 0x010000ff, 0x00000000, 0x00000000, 0x00000000 };`
	Reserve OGU 0, OGU1 and OGU3 for ICU Userid for ICU is 5 Logical channel id for ICU is 0, 1 and 2 respectively ERS reference is 0, 1 and 2 respectively	`const uint32 Mcu_17_Eru_ChUserData[MCU_17_ERU_NO_OF_OGU] = { 0x01050000, 0x02050101, 0x00000000, 0x04050202, 0x010000ff, 0x020000ff, 0x00000000, 0x040000ff };`

1.4.3 Structure: Mcu_17_Ccu6_ChUserData[MCU_17_CCU6_NO_OF_KERNELS][MCU_17_CCU6_NO_OF_COMPARATORS]

Table 216 Mcu_17_Ccu6_ChUserData[MCU_17_CCU6_NO_OF_KERNELS][MCU_17_CCU6_NO_OF_COMPARATORS]

Name	Mcu_17_Ccu6_ChUserData[MCU_17_CCU6_NO_OF_KERNELS][MCU_17_CCU6_NO_OF_COMPARATORS]	
Type	uint16	
Description	Array to store user information for CCU6 kernels and comparators.	
Verification method	The generated structure is present in Mcu_17_TimerIp_Cfg.c file. The number of members in the structure depends on number of kernels and comparators available in the CCU6 module. Bits 0-4 store the userid of the user of that comparator. Refer section 1.4.1 for list of User ids. Bits 8-11 store the logical channel id of the respective user. Others bits are always 0.	
Example(s)	**Action**	**Generated output**

	Reserve CCU6 kernel0 comparator 2 for ICU. The logical channel Id for Icu is 1.
	
	User id for Icu is 5
		`const uint16 Mcu_17_Ccu6_ChUserData[MCU_17_CCU6_NO_OF_KERNELS][MCU_17_CCU6_NO_OF_COMPARATORS] =`
		`{`
		`  {`
		`  0x0000,`
		`  0x0105,`
		`  0x0000,`
		`  0x0000`
		`  },`
		`  {`
		`  0x0000,`
		`  0x0000,`
		`  0x0000,`
		`  0x0000`
		`  }`
		`};`

	Reserve CCU6 kernel1 comparator 3 for ICU. The logical channel Id for Icu is 1.
	
	User id for Icu is 5
		`const uint16 Mcu_17_Ccu6_ChUserData[MCU_17_CCU6_NO_OF_KERNELS][MCU_17_CCU6_NO_OF_COMPARATORS] =`
		`{`
		`  {`
		`  0x0000,`
		`  0x0000,`
		`  0x0000,`
		`  0x0000`
		`  },`
		` {

		0x0000,
		0x0000,
		0x0105,
		0x0000
		}
		};

1.4.4 Structure: Mcu_17_Gpt12_ChUserData[MCU_17_GPT12_NO_OF_TIMERS]

Table 217 Mcu_17_Gpt12_ChUserData[MCU_17_GPT12_NO_OF_TIMERS]

Name	Mcu_17_Gpt12_ChUserData[MCU_17_GPT12_NO_OF_TIMERS]	
Type	uint16	
Description	Array to store user information for GPT12 timers.	
Verification method	The generated structure is present in Mcu_17_TimerIp_Cfg.c file. The number of members in the structure depends on number of timers available in the GPT12 module. Bits 0-4 store the userid of the user of that comparator. Refer section 1.4.1 for list of User ids. Bits 8-11 store the logical channel id of the respective user. Others bits are always 0.	
Example(s)	**Action**	**Generated output**

	Reserve GPT timer 2 for Icu. The logical channel Id for Icu is 0.
	
	User id for Icu is 5
		const uint16 Mcu_17_Gpt12_ChUserData[MCU_17_GPT12_NO_OF_TIMERS] =
		{
		0x0005,
		0x0000,
		0x0000,
		0x0000,
		0x0000,
		};

	Reserve GPT Timer 2 and Timer4 for Icu. The logical channel Id for Icu is 0 and 1. User id for Icu is 5	const uint16 Mcu_17_Gpt12_ChUserData[MCU_17_GPT1_NO_OF_TIMERS + MCU_17_GPT2_NO_OF_TIMERS] = { 0x0005, 0x0000, 0x0105, 0x0000, 0x0000,

		};

1.4.5 Structure: Mcu_17_Stm_ChUserData[MCU_17_STM_NO_OF_TIMERS]

Table 218 Mcu_17_Stm_ChUserData[MCU_17_STM_NO_OF_TIMERS]

Name	Mcu_17_Stm_ChUserData[MCU_17_STM_NO_OF_TIMERS]	
Type	uint32	
Description	Array to store user information for STM timers.	
Verification method	The generated structure is present in Mcu_17_TimerIp_Cfg.c file. The number of members in the structure depends on number of timers available in the STM module. 1. If Comparator 0 is reserved • Bits 0-4 of the selected timer userdata value stores 0x1. • Bits 8-11 store the user id of the respective user. Refer section 1.4.1 for list of User ids. 2. If Comparator 1 is reserved • Bits 16-19 of the selected timer userdata value stores 0x2. • Bits 24-27 store the user id of the respective user. Refer section 1.4.1 for list of User ids. Others bits are always 0.	
Example(s)	**Action**	**Generated output**
	Reserve STM 2 comparator 0 for WDG in resource manager User Id for watchdog is 2	`const uint32 Mcu_17_Stm_ChUserData[MCU_17_STM_NO_OF_TIMERS] = { 0x00000000, 0x00000000, 0x00000201, 0x00000000, 0x00000000, 0x00000000 };`
	Reserve STM 5 comparator 1 for WDG in resource manager User Id for watchdog is 2	`const uint32 Mcu_17_Stm_ChUserData[MCU_17_STM_NO_OF_TIMERS] = { 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x02020000`

		};

1.4.6 Structure: Mcu_17_Gtm_TomChUserData [MCU_17_GTM_NO_OF_TOM_MODULES] [MCU_17_GTM_NO_OF_TOM_CHANNELS]

Table 219 Mcu_17_Gtm_TomChUserData[MCU_17_GTM_NO_OF_TOM_MODULES][MCU_17_GTM_NO_OF_TOM_CHANNELS]

Name	Mcu_17_Gtm_TomChUserData[MCU_17_GTM_NO_OF_TOM_MODULES][MCU_17_GTM_NO_OF_TOM_CHANNELS]	
Type	uint32	
Description	Array to store user information for TOM.	
Verification method	The generated structure is present in Mcu_17_TimerIp_Cfg.c file. The number of members in the structure depends on number of TOM modules and channels available in the GTM module. Bits 0-4 store the userid of the user of that module and channel. Refer section 1.4.1 for list of User ids. Bits 8-11 store the logical channel id of the respective user. Others bits are always 0. In case the channel is reserved by PWM and if the configuration parameter 'McuGtmTomAllocationConf/ McuTomChannelEventHandledByDsadc' is set to 'True', the user is set as DSADC. Refer section 1.4.1 for User id of DSADC.	
Example(s)	**Action**	**Generated output**
	Reserve TOM module 0 channel 5 (PWM logical channel id 0), TOM module 3 channel 13 (PWM logical channel id 1) for PWM User Id for PWM is 3	`const uint32 Mcu_17_Gtm_TomChUserData[MCU_17_GTM_NO_OF_TOM_MODULES][MCU_17_GTM_NO_OF_TOM_CHANNELS] = { /* TOM Module 0 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0003, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,`

```
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* TOM Module 1 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* TOM Module 2 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
```

```
    0x0000,
    0x0000
  },
  /* TOM Module 3 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0103,
    0x0000,
    0x0000
  },
  /* TOM Module 4 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
```

	},
	/* TOM Module 5 */
	{
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000
	}
	};

Reserve TOM module 0 channel 5 (PWM logical channel id 0), TOM module 3 channel 13 (PWM logical channel id 1), TOM module 2 channel 10 (PWM logical channel id 2), TOM module 4 channel 7 (PWM logical channel id 3) for PWM Set McuTomChannelEventHandledByDsadc = True for TOM module 0 channel 5 User Id for PWM is 3	const uint16 Mcu_17_Gtm_TomChUserData[MCU_17_GTM_NO_OF_TOM_MODULES][MCU_17_GTM_NO_OF_TOM_CHANNELS] = { /* TOM Module 0 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0007, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,

```
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* TOM Module 1 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* TOM Module 2 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0203,
    0x0000,
    0x0000,
    0x0000,
```

```
    0x0000,
    0x0000
  },
  /* TOM Module 3 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0103,
    0x0000,
    0x0000
  },
  /* TOM Module 4 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0303,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
```

```
  },
  /* TOM Module 5 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  }
};
```

1.4.7 Structure: Mcu_17_Gtm_TimChUserData [MCU_17_GTM_NO_OF_TIM_MODULES] [MCU_17_GTM_NO_OF_TIM_CHANNELS]

Table 220 Mcu_17_Gtm_TimChUserData[MCU_17_GTM_NO_OF_TIM_MODULES][MCU_17_GTM_NO_OF_TIM_CHANNELS]

Name	Mcu_17_Gtm_TimChUserData[MCU_17_GTM_NO_OF_TIM_MODULES][MCU_17_GTM_NO_OF_TIM_CHANNELS]	
Type	uint16	
Description	Array to store user information for TIM.	
Verification method	The generated structure is present in Mcu_17_TimerIp_Cfg.c file. The number of members in the structure depends on number of TIM modules and channels available in the GTM module. Bits 0-4 store the userid of the user of that module and channel. Refer section 1.4.1 for list of User ids. Bits 8-11 store the logical channel id of the respective user. Others bits are always 0.	
Example(s)	**Action**	**Generated output**
	Reserve TIM module 0 channel 5 (ICU logical channel id 0), TIM	`const uint16 Mcu_17_Gtm_TimChUserData[MCU_17_GTM_N`

module 3 channel 7 (ICU logical channel id 1) for ICU User Id for ICU is 5	O_OF_TIM_MODULES][MCU_17_GTM_NO_OF_TIM_CHANNELS] = { /* TIM Module 0 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0005, 0x0000, 0x0000 }, /* TIM Module 1 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, /* TIM Module 2 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, /* TIM Module 3 */ { 0x0000, 0x0000,

```
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0105
  },
  /* TIM Module 4 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* TIM Module 5 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* TIM Module 6 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
```

	/* TIM Module 7 */
	{
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000
	}
	};

Reserve TIM module 0 channel 5 (ICU logical channel id 0), TIM module 3 channel 7 (ICU logical channel id 1), TIM module 2 channel 3 (ICU logical channel id 2), TIM module 7 channel 7 (ICU logical channel id 3) for ICU User Id for PWM is 3	const uint16 Mcu_17_Gtm_TimChUserData[MCU_17_GTM_NO_OF_TIM_MODULES][MCU_17_GTM_NO_OF_TIM_CHANNELS] = { /* TIM Module 0 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0005, 0x0000, 0x0000 }, /* TIM Module 1 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, /* TIM Module 2 */

```
{
  0x0000,
  0x0000,
  0x0000,
  0x0205,
  0x0000,
  0x0000,
  0x0000,
  0x0000
},
/* TIM Module 3 */
{
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0105
},
/* TIM Module 4 */
{
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000
},
/* TIM Module 5 */
{
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
```

```
    0x0000,
    0x0000
  },
  /* TIM Module 6 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* TIM Module 7 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0305
  }
};
```

1.4.8 Structure: Mcu_17_Gtm_AtomChUserData [MCU_17_GTM_NO_OF_ATOM_MODULES] [MCU_17_GTM_NO_OF_ATOM_CHANNELS]

Table 221 Mcu_17_Gtm_AtomChUserData[MCU_17_GTM_NO_OF_ATOM_MODULES][MCU_17_GTM_NO_OF_ATOM_CHANNELS]

Name	Mcu_17_Gtm_AtomChUserData[MCU_17_GTM_NO_OF_ATOM_MODULES][MCU_17_GTM_NO_OF_ATOM_CHANNELS]
Type	uint32
Description	Array to store user information for ATOM.
Verification method	The generated structure is present in Mcu_17_TimerIp_Cfg.c file. The number of members in the structure depends on number of ATOM modules and channels available in the GTM module. Bits 0-4 store the userid of the user of that module and channel. Refer section 1.4.1 for list of User ids.

	Bits 8-11 store the logical channel id of the respective user. Others bits are always 0. In case the channel is reserved by PWM and if the configuration parameter 'McuGtmAtomAllocationConf/ McuAtomChannelEventHandledByDsadc' is set to 'True', the user is set as DSADC. Refer section 1.4.1 for User id of DSADC.	
Example(s)	**Action**	**Generated output**
	Reserve ATOM module 0 channel 5 (PWM logical channel id 0), ATOM module 3 channel 6 (PWM logical channel id 1), ATOM module 4 channel 4 (PWM logical channel id 2), ATOM module 5 channel 6 (PWM logical channel id 3), ATOM module 8 channel 3 (PWM logical channel id 4) for PWM User Id for PWM is 3	const uint32 Mcu_17_Gtm_AtomChUserData[MCU_17_GTM_NO_OF_ATOM_MODULES][MCU_17_GTM_NO_OF_ATOM_CHANNELS] = { /* ATOM Module 0 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0003, 0x0000, 0x0000 }, /* ATOM Module 1 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, /* ATOM Module 2 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,

```
    0x0000,
    0x0000,
    0x0000
  },
  /* ATOM Module 3 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0103,
    0x0000
  },
  /* ATOM Module 4 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0203,
    0x0000,
    0x0000,
    0x0000
  },
  /* ATOM Module 5 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0303,
    0x0000
  },
  /* ATOM Module 6 */
  {
    0x0000,
```

```
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* ATOM Module 7 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* ATOM Module 8 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0403,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* ATOM Module 9 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
```

	},
	/* ATOM Module 10 */
	{
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000
	},
	/* ATOM Module 11 */
	{
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000,
	0x0000
	}
	};

Reserve ATOM module 0 channel 5 (PWM logical channel id 0), ATOM module 3 channel 6 (PWM logical channel id 1), ATOM module 4 channel 4 (PWM logical channel id 2), ATOM module 5 channel 6 (PWM logical channel id 3), ATOM module 8 channel 3 (PWM logical channel id 4), ATOM module 9 channel 5 (PWM logical channel id 5), ATOM module 11 channel 7 (PWM logical channel id 6) for PWM Set McuAtomChannelEventHandledByDsadc = True for ATOM	const uint16 Mcu_17_Gtm_AtomChUserData[MCU_17_GTM_NO_OF_ATOM_MODULES][MCU_17_GTM_NO_OF_ATOM_CHANNELS] = { /* ATOM Module 0 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0003, 0x0000, 0x0000 },

module 3 channel 6 User Id for PWM is 3	/* ATOM Module 1 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, /* ATOM Module 2 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 }, /* ATOM Module 3 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0107, 0x0000 }, /* ATOM Module 4 */ { 0x0000, 0x0000, 0x0000, 0x0000, 0x0203,

```
  0x0000,
  0x0000,
  0x0000
},
/* ATOM Module 5 */
{
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0303,
  0x0000
},
/* ATOM Module 6 */
{
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000
},
/* ATOM Module 7 */
{
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000,
  0x0000
},
/* ATOM Module 8 */
{
  0x0000,
```

```
    0x0000,
    0x0000,
    0x0403,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* ATOM Module 9 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0503,
    0x0000,
    0x0000
  },
  /* ATOM Module 10 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000
  },
  /* ATOM Module 11 */
  {
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
    0x0603
```

		} };

1.5 File: Mcu[_<variant>]_PBcfg.h

The generated header file contains the declaration of the root configuration structure. Post-build time configuration mechanism allows configurable functionality of MCU driver that is deployed as object code. The file is generated in 'inc' folder.

1.5.1 Structure: Mcu_Config[_<variant>]

Table 222 Icu_17_TimerIp_Config_[_<varaint>]

Name	Mcu_Config[_<variant>]	
Type	Mcu_ConfigType	
Description	Declaration of root configuration structure of MCU driver which will be used during initialization.	
Verification method	The generated structure is present in Mcu[_<variant>]_PBcfg.h file. The <variant> indicates the name of the post-build variant. For a variant-aware configuration the structure name is appended with the variant name. For variant-unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**
	Configure MCU (variant-unaware)	`extern const Mcu_ConfigType Mcu_Config;`
	Configure MCU (variant-aware. Variant name is 'Petrol')	`extern const Mcu_ConfigType Mcu_Config_Petrol;`

Revision history

Major changes since the last revision

Date	Version	Description
2023-05-26	8.0	Document Released
2023-05-23	7.1	In section 1.1, description of production error macros is updated to change DEM to production error. The following parameter description is updated to correct inconsistencies between implementation and documentation. - MaxModeEvrcCtrl - CCUCON1 - CCUCON2 - CCUCON3 - CCUCON4
2022-08-11	7.0	Document released
2022-08-10	6.2	Incorporated review comments of gerrit 106185
2022-07-29	6.1	• Updated Verification method for following parameters, - Mcu_17_Gpt12_ChUserData - Mcu_kPllDistributionConfiguration_Config.Ccucon0 - Mcu_kPllDistributionConfiguration_Config.CcuconCpu - Mcu_kGtmConfiguration_Config.GtmTomCfg - Mcu_kGtmConfiguration_Config.GtmAtomCfg - Mcu_kGtmConfiguration_Config.GtmTimInSelCfg - Mcu_17_Eru_ChUserData[MCU_17_ERU_NO_OF_OGU] structure. • Type is updated for structures Mcu_17_Gtm_TomChUserData, Mcu_17_Gtm_TomChUserData
2021-10-20	6.0	Document Released
2021-10-12	5.1	• Added non-cached address for McuStdbyModeRamEnable (Section 1.1.48, 1.3.4.8) • Updated value of TomTgcIntTrigRstCn0, TomTgcIntTrigRstCn1 and AtomTgcIntTrigRstCn0 of Mcu_kGtmConfiguration structure (Section 1.3.7)
2020-10-13	5.0	Document Released
2020-10-13	4.1	• Mcu driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_Basic.pdf to this document • Added structure member GtmTimInSelCfg (Section 1.3.7.9). • Added structure member Evruvmon (Section 1.3.4.6). • Added structure member EvrmonCtrl (Section 1.3.4.7).

Date	Version	Description
2020-06-22	4.0	Document Released
2020-06-22	3.1	• Added macro MCU_GTM_NO_OF_TOUTSEL_AVAILABLE (Section 1.1.30) • Structure member RamData and RamWriteSize added in Mcu_kRamConfiguration_Config (Section 1.3.2) • Updated structure Mcu_kGtmConfiguration (Section 1.3.7) • Added members GtmToutSelCfg (Section 1.3.7.7) and GtmToutSelCfgMsk (Section 1.3.7.8) • Added macro MCU_TBU_CH_EN_MSK (Section 1.1.34) • Added macro MCU_GTM_TO_DSADC_TRIG_AVAILABLE (Section 1.1.32) • Added new macros MCU_SYSCLK_FREQ (section 1.1.36) and MCU_SYSCLK_OSCVAL (section 1.1.43)
2019-06-25	3.0	Document Released
2019-06-21	2.2	Incorporated review comments of Gerrit 61373
2019-06-20	2.1	Updated document for • Added new macros (section 1.1.36, 1.1.37, 1.1.39 • Updated generation logic for Mcu_17_Gtm_TomChUserData and Mcu_17_Gtm_AtomChUserData with examples
2019-02-26	1.10.0_2.0	Added Pbcfg.h file
2019-02-25	1.10.0_1.0	Initial Release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-05-26
Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
Doc_Number

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.