MCAL Configuration Verification Manual for Eth_17_GEthMacV2

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 Eth_17_GEthMacV2

Table of contents

Table of contents

Table of contents

1 Eth_17_GEthMacV2 driver

This chapter describes the details of the configuration data generated from the Eth_17_GEthMacV2 driver.

1.1 File: Eth_17_GEthMacV2_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: ETH_17_GETHMACV2_AR_RELEASE_MAJOR_VERSION

Table 1 ETH_17_GETHMACV2_AR_RELEASE_MAJOR_VERSION

Name	ETH_17_GETHMACV2_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the Eth_17_GEthMacV2 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 Eth_17_GEthMacV2_Cfg.h file with ArMajorVersion 4	#define ETH_17_GETHMACV2_AR_RELEASE_MAJOR_VERSION (4U)

1.1.2 Macro: ETH_17_GETHMACV2_AR_RELEASE_MINOR_VERSION

Table 2 ETH_17_GETHMACV2_AR_RELEASE_MINOR_VERSION

Name	ETH_17_GETHMACV2_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the Eth_17_GEthMacV2 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 Eth_17_GEthMacV2_Cfg.h file with ArMinorVersion 4	#define ETH_17_GETHMACV2_AR_RELEASE_MINOR_VERSION (4U)

1.1.3 Macro: ETH_17_GETHMACV2_AR_RELEASE_REVISION_VERSION

Table 3 ETH_17_GETHMACV2_AR_RELEASE_REVISION_VERSION

Name	ETH_17_GETHMACV2_AR_RELEASE_REVISION_VERSION	
Description	Revision version number of AUTOSAR release on which the Eth_17_GEthMacV2 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 Eth_17_GEthMacV2_Cfg.h file with ArPatchVersion 0	#define ETH_17_GETHMACV2_AR_RELEASE_REVISION_VERSION (0U)

1.1.4 Macro: ETH_17_GETHMACV2_SW_MAJOR_VERSION

Table 4 ETH_17_GETHMACV2_SW_MAJOR_VERSION

Name	ETH_17_GETHMACV2_SW_MAJOR_VERSION	
Description	Major version number of the Eth_17_GEthMacV2 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 Eth_17_GEthMacV2_Cfg.h file with SwMajorVersion 20	#define ETH_17_GETHMACV2_SW_MAJOR_VERSION (20U)

1.1.5 Macro: ETH_17_GETHMACV2_SW_MINOR_VERSION

Table 5 ETH_17_GETHMACV2_SW_MINOR_VERSION

Name	ETH_17_GETHMACV2_SW_MINOR_VERSION	
Description	Minor version number of the Eth_17_GEthMacV2 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 Eth_17_GEthMacV2_Cfg.h file with SwMinorVersion 0	#define ETH_17_GETHMACV2_SW_MINOR_VERSION (0U)

1.1.6 Macro: ETH_17_GETHMACV2_SW_PATCH_VERSION

Table 6 ETH_17_GETHMACV2_SW_PATCH_VERSION

Name	ETH_17_GETHMACV2_SW_PATCH_VERSION	
Description	Patch version number of the Eth_17_GEthMacV2 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 Eth_17_GEthMacV2_Cfg.h file with SwPatchVersion 1	#define ETH_17_GETHMACV2_SW_PATCH_VERSION (1U)

1.1.7 Macro: ETH_17_GETHMACV2_DEV_ERROR_DETECT

Table 7 ETH_17_GETHMACV2_DEV_ERROR_DETECT

Name	ETH_17_GETHMACV2_DEV_ERROR_DETECT	
Description	Enables/disables the Development Error Detection.	
Verification method	The macro is generated as STD_ON if EthDevErrorDetect configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthDevErrorDetect as True	#define ETH_17_GETHMACV2_DEV_ERROR_DETECT (STD_ON)
	Set EthDevErrorDetect as False	#define ETH_17_GETHMACV2_DEV_ERROR_DETECT (STD_OFF)

1.1.8 Macro: ETH_17_GETHMACV2_VERSION_INFO_API

Table 8 ETH_17_GETHMACV2_VERSION_INFO_API

Name	ETH_17_GETHMACV2_VERSION_INFO_API	
Description	Enables/ disables the Eth_17_GEthMacV2_GetVersionInfo API	
Verification method	The macro is generated as STD_ON if EthVersionInfoApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthVersionInfoApi as True	#define ETH_17_GETHMACV2_VERSION_INFO_API (STD_ON)
	Set EthVersionInfoApi as False	#define ETH_17_GETHMACV2_VERSION_INFO_API (STD_OFF)

1.1.9 Macro: Eth_17_GEthMacV2Conf_EthCtrlConfig_<Container Name>

Table 9 Eth_17_GEthMacV2Conf_EthCtrlConfig_<Container Name>

Name	Eth_17_GEthMacV2Conf_EthCtrlConfig_<Container Name>	
Description	The macro is the symbolic name generated for the configuration parameter EthCtrlIdx	
Verification method	The macro is generated with a numeric value that is configured in EthCtrlIdx. < Container Name > is the name of the Ethernet controller container.	
Example(s)	**Action**	**Generated output**
	If EthCtrlIdx is 1 and Ethernet controller container name is EthCtrlConfig_0	#define Eth_17_GEthMacV2Conf_EthCtrlConfig_EthCtrlConfig_0 (1U)

1.1.10 Macro: ETH_17_GETHMACV2_INDEX

Table 10 ETH_17_GETHMACV2_INDEX

Name	ETH_17_GETHMACV2_INDEX	
Description	Ethernet driver instance ID.	
Verification method	The macro is generated as a numeric value that is set in the configuration parameter 'EthGeneral/EthIndex'	
Example(s)	**Action**	**Generated output**
	Set EthIndex as 0	#define ETH_17_GETHMACV2_INDEX (0U)
	Set EthIndex as 5	#define ETH_17_GETHMACV2_INDEX (5U)

1.1.11 Macro: ETH_17_GETHMACV2_ENA_MII_API

Table 11 ETH_17_GETHMACV2_ENA_MII_API

Name	ETH_17_GETHMACV2_ENA_MII_API	
Description	Enables/ disables Eth_17_GEthMacV2_WriteMii and Eth_17_GEthMacV2_ReadMii APIs	
Verification method	The macro is generated as STD_ON if EthCtrlEnableMii configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlEnableMii as True	#define ETH_17_GETHMACV2_ENA_MII_API (STD_ON)
	Set EthCtrlEnableMii as False	#define ETH_17_GETHMACV2_ENA_MII_API (STD_OFF)

1.1.12 Macro: ETH_17_GETHMACV2_RX_IRQHDLR

Table 12 ETH_17_GETHMACV2_RX_IRQHDLR

Name	ETH_17_GETHMACV2_RX_IRQHDLR	
Description	Enables/ disables the receive interrupt handler Eth_17_GEthMacV2_RxDmaIrqHdlr	
Verification method	The macro is generated as STD_ON if EthCtrlEnableRxInterrupt configuration parameter is set to 'True' for any of the configured Ethernet controllers else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlEnableRxInterrupt as True for any controller configured within EthCtrlConfig container	#define ETH_17_GETHMACV2_RX_IRQHDLR (STD_ON)
	Set EthCtrlEnableRxInterrupt as False for all controllers configured within EthCtrlConfig container	#define ETH_17_GETHMACV2_RX_IRQHDLR (STD_OFF)

1.1.13 Macro: ETH_17_GETHMACV2_TX_IRQHDLR

Table 13 ETH_17_GETHMACV2_TX_IRQHDLR

Name	ETH_17_GETHMACV2_TX_IRQHDLR	
Description	Enables/ disables the transmit interrupt handler Eth_17_GEthMacV2_TxDmaIrqHdlr	
Verification method	The macro is generated as STD_ON if EthCtrlEnableTxInterrupt configuration parameter is set to 'True' for any of the configured Ethernet controllers else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlEnableTxInterrupt as True for any controller configured within EthCtrlConfig container	#define ETH_17_GETHMACV2_TX_IRQHDLR (STD_ON)
	Set EthCtrlEnableTxInterrupt as False for all controllers configured within EthCtrlConfig container	#define ETH_17_GETHMACV2_TX_IRQHDLR (STD_OFF)

1.1.14 Macro: ETH_17_GETHMACV2_MAXTIMEOUT_COUNT

Table 14 ETH_17_GETHMACV2_MAXTIMEOUT_COUNT

Name	ETH_17_GETHMACV2_MAXTIMEOUT_COUNT
Description	Specifies maximum timeout count in nanoseconds to wait for hardware timeout
Verification method	The macro is generated as the value set in the configuration parameter 'EthGeneral/EthTimeoutCount'.

Example(s)	Action	Generated output
	Set EthTimeoutCount as 100	#define ETH_17_GETHMACV2_MAXTIMEOUT_COUNT (100U)
	Set EthTimeoutCount as 4294967295	#define ETH_17_GETHMACV2_MAXTIMEOUT_COUNT (4294967295U)

1.1.15 Macro: ETH_17_GETHMACV2_SWT_MANAGEMENT_SUPPORT

Table 15 ETH_17_GETHMACV2_SWT_MANAGEMENT_SUPPORT

Name	ETH_17_GETHMACV2_SWT_MANAGEMENT_SUPPORT	
Description	Enables/ disables the Ethernet Switch Management support functionality	
Verification method	The macro is generated as STD_ON if the configuration parameter 'EthSwtGeneral/EthSwtManagementSupportApi' within the EthSwt module is set to 'True' else the macro is generated as STD_OFF. If the EthSwt module is not added to the configuration project, the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthSwtManagementSupportApi (within the EthSwt module) as True	#define ETH_17_GETHMACV2_SWT_MANAGEMENT_SUPPORT (STD_ON)
	Set EthSwtManagementSupportApi (within the EthSwt module) as False	#define ETH_17_GETHMACV2_SWT_MANAGEMENT_SUPPORT (STD_OFF)
	Do not include the EthSwt module in the project configuration.	#define ETH_17_GETHMACV2_SWT_MANAGEMENT_SUPPORT (STD_OFF)

1.1.16 Macro: ETH_17_GETHMACV2_FIFO<IngressFifoIndex>_CTRL<CtrlIndex>_RXBUF_COUNT

Table 16 ETH_17_GETHMACV2_FIFO<IngressFifoIndex>_CTRL<CtrlIndex>_RXBUF_COUNT

Name	ETH_17_GETHMACV2_FIFO<IngressFifoIndex>_CTRL<CtrlIndex>_RXBUF_COUNT	
Description	Indicates the total number of buffers configured within the ingress FIFO with <IngressFifoIndex> for the Eth controller with <CtrlIndex>.	
Verification method	The macro is generated as a numeric value which is configured in the configuration parameter 'EthCtrlConfigIngressFifo/EthCtrlConfigIngressFifoBufTotal' for an ingress FIFO.	
Example(s)	**Action**	**Generated output**

Set EthCtrlConfigIngressFifoBufTotal as 4 for ingress FIFO with EthCtrlConfigIngressFifoIdx 0 for EthCtrlIdx 0.	#define ETH_17_GETHMACV2_FIFO0_CTRL0_RXBUF_COUNT (4U)
Set EthCtrlConfigIngressFifoBufTotal as 255 for ingress FIFO with EthCtrlConfigIngressFifoIdx 3 for EthCtrlIdx 1.	#define ETH_17_GETHMACV2_FIFO3_CTRL1_RXBUF_COUNT (255U)

1.1.17 Macro: ETH_17_GETHMACV2_FIFO<IngressFifoIndex>_CTRL<CtrlIndex>_RXBUF_SIZE

Table 17 ETH_17_GETHMACV2_FIFO<IngressFifoIndex>_CTRL<CtrlIndex>_RXBUF_SIZE

Name	ETH_17_GETHMACV2_FIFO<IngressFifoIndex>_CTRL<CtrlIndex>_RXBUF_SIZE	
Description	Total size of RAM allocated for the configured buffers of ingress FIFO with <IngressFifoIndex> for the Eth controller with <CtrlIndex>.	
Verification method	The macro is generated as the total receive buffer size with 8 byte alignment required for an ingress FIFO (value of the macro is a product of the values provided in EthCtrlConfigIngressFifoBufTotal and EthCtrlConfigIngressFifoBufLenByte configuration parameters for the ingress FIFO). *Note: If EthCtrlConfigIngressFifoBufLenByte is not 8-byte aligned, it is changed to next 8-byte aligned value. This is done to align the Rx buffers on 8-byte aligned addresses for maximum hardware performance.*	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigIngressFifoBufTotal to 255 and EthCtrlConfigIngressFifoBufLenByte to 1522 for ingress FIFO with EthCtrlConfigIngressFifoIdx 0 for EthCtrlIdx 0 *Note: 8 byte alignment for EthCtrlConfigIngressFifoBufLenByte: if (EthCtrlConfigIngressFifoBufLenByte mod 8 ! =0) then ((1522/8)+1)*8 = 191 * 8 = 1528*	#define ETH_17_GETHMACV2_FIFO0_CTRL0_RXBUF_SIZE (389640U)
	Set EthCtrlConfigIngressFifoBufTotal to 1 and EthCtrlConfigIngressFifoBufLenByte to 1522 for ingress FIFO with EthCtrlConfigIngressFifoIdx 2 for EthCtrlIdx 1	#define ETH_17_GETHMACV2_FIFO2_CTRL1_RXBUF_SIZE (1528U)

*Note: 8 byte alignment for EthCtrlConfigIngressFifoBufLenByte : if (EthCtrlConfigIngressFifoBufLenByte mod 8 ! =0) then ((1522/8)+1)*8 = 191 * 8 = 1528*	
Set EthCtrlConfigIngressFifoBufTotal to 4 and EthCtrlConfigIngressFifoBufLenByte to 64 for ingress FIFO with EthCtrlConfigIngressFifoIdx 1 for EthCtrlIdx 0 *Note: EthCtrlConfigIngressFifoBufLenByte is 8 byte aligned*	#define ETH_17_GETHMACV2_FIFO2_CTRL1_RXBUF_SIZE (256U)

1.1.18 Macro: ETH_17_GETHMACV2_CTRL<CtrlIndex>_RXFIFO_CFGD

Table 18 ETH_17_GETHMACV2_CTRL<CtrlIndex>_RXFIFO_CFGD

Name	ETH_17_GETHMACV2_CTRL<CtrlIndex>_RXFIFO_CFGD	
Description	Indicates the total number of ingress FIFOs configured for the Eth controller with <CtrlIndex>.	
Verification method	The macro is generated as a numeric value that corresponds to the number of ingress FIFO containers ‘EthCtrlConfigIngressFifo’ configured.	
Example(s)	**Action**	**Generated output**
	Configure one ingress FIFO container EthCtrlConfigIngressFifo for EthCtrlIdx 0.	#define ETH_17_GETHMACV2_CTRL0_RXFIFO_CFGD (1U)
	Configure 4 ingress FIFO containers EthCtrlConfigIngressFifo for EthCtrlIdx 1.	#define ETH_17_GETHMACV2_CTRL1_RXFIFO_CFGD (4U)

1.1.19 Macro: ETH_17_GETHMACV2_FIFO<EgressFifoIndex>_CTRL<CtrlIndex>_TXBUF_COUNT

Table 19 ETH_17_GETHMACV2_FIFO<EgressFifoIndex>_CTRL<CtrlIndex>_TXBUF_COUNT

Name	ETH_17_GETHMACV2_FIFO<EgressFifoIndex>_CTRL<CtrlIndex>_TXBUF_COUNT
Description	Indicates the total number of buffers configured within the egress FIFO with <EgressFifoIndex> for the Eth controller with <CtrlIndex>.

Verification method	The macro is generated as a numeric value which is configured in the configuration parameter 'EthCtrlConfigEgressFifo/EthCtrlConfigEgressFifoBufTotal' for an egress FIFO.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigEgressFifoBufTotal as 4 for egress FIFO with EthCtrlConfigEgressFifoIdx 0 for EthCtrlIdx 0.	#define ETH_17_GETHMACV2_FIFO0_CTRL0_TXBUF_COUNT (4U)
	Set EthCtrlConfigEgressFifoBufTotal as 255 for egress FIFO with EthCtrlConfigEgressFifoIdx 3 for EthCtrlIdx 1.	#define ETH_17_GETHMACV2_FIFO3_CTRL1_TXBUF_COUNT (255U)

1.1.20 Macro: ETH_17_GETHMACV2_FIFO<EgressFifoIndex>_CTRL<CtrlIndex>_TXBUF_SIZE

Table 20 ETH_17_GETHMACV2_FIFO<EgressFifoIndex>_CTRL<CtrlIndex>_TXBUF_SIZE

Name	ETH_17_GETHMACV2_FIFO<EgressFifoIndex>_CTRL<CtrlIndex>_TXBUF_SIZE	
Description	Total size of RAM allocated for the configured buffers of egress FIFO with <EgressFifoIndex> for the Eth controller with <CtrlIndex>.	
Verification method	The macro is generated as the total transmit buffer size with 8 byte alignment required for an egress FIFO (value of the macro is a product of the values provided in EthCtrlConfigEgressFifoBufTotal and EthCtrlConfigEgressFifoBufLenByte configuration parameters for the egress FIFO). *Note: If EthCtrlConfigEgressFifoBufLenByte is not 8-byte aligned, it is changed to next 8-byte aligned value. This is done to align the Tx buffers on 8-byte aligned addresses for maximum hardware performance.*	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigEgressFifoBufTotal to 255 and EthCtrlConfigEgressFifoBufLenByte to 1522 for egress FIFO with EthCtrlConfigEgressFifoIdx 0 for EthCtrlIdx 0 *Note: 8 byte alignment for EthCtrlConfigEgressFifoBufLenByte: if (EthCtrlConfigEgressFifoBufLenByte mod 8 ! =0) then ((1522/8)+1)*8 = 191 * 8 = 1528*	#define ETH_17_GETHMACV2_FIFO0_CTRL0_TXBUF_SIZE (389640U)
	Set EthCtrlConfigEgressFifoBufTotal to 1 and EthCtrlConfigEgressFifoBufLenByte to 1522 for egress FIFO with EthCtrlConfigEgressFifoIdx 2 for EthCtrlIdx 1	#define ETH_17_GETHMACV2_FIFO2_CTRL1_TXBUF_SIZE (1528U)

*Note: 8 byte alignment for EthCtrlConfigEgressFifoBufLenByte : if (EthCtrlConfigEgressFifoBufLenByte mod 8 ! =0) then ((1522/8)+1)*8 = 191 * 8 = 1528*	
Set EthCtrlConfigEgressFifoBufTotal to 4 and EthCtrlConfigEgressFifoBufLenByte to 64 for egress FIFO with EthCtrlConfigEgressFifoIdx 1 for EthCtrlIdx 0 *Note: EthCtrlConfigEgressFifoBufLenByte is 8 byte aligned*	#define ETH_17_GETHMACV2_FIFO2_CTRL1_TXBUF_SIZE (256U)

1.1.21 Macro: ETH_17_GETHMACV2_CTRL<CtrlIndex>_TXFIFO_CFGD

Table 21 ETH_17_GETHMACV2_CTRL<CtrlIndex>_TXFIFO_CFGD

Name	ETH_17_GETHMACV2_CTRL<CtrlIndex>_TXFIFO_CFGD	
Description	Indicates the total number of egress FIFOs configured for the Eth controller with <CtrlIndex>.	
Verification method	The macro is generated as a numeric value that corresponds to the number of egress FIFO containers 'EthCtrlConfigEgressFifo' configured.	
Example(s)	**Action**	**Generated output**
	Configure one egress FIFO container EthCtrlConfigEgressFifo for EthCtrlIdx 0.	#define ETH_17_GETHMACV2_CTRL0_TXFIFO_CFGD (1U)
	Configure 4 egress FIFO containers EthCtrlConfigEgressFifo for EthCtrlIdx 1.	#define ETH_17_GETHMACV2_CTRL1_TXFIFO_CFGD (4U)

1.1.22 Macro: ETH_17_GETHMACV2_FSPB_PERIOD_IN_NANOSEC

Table 22 ETH_17_GETHMACV2_FSPB_PERIOD_IN_NANOSEC

Name	ETH_17_GETHMACV2_FSPB_PERIOD_IN_NANOSEC
Description	SPB frequency in nanoseconds.
Verification method	The macro is generated as the time period value in nanoseconds of the frequency value configured by the user in EthPeripheralBusFrequency configuration parameter which in turn refers McuSPBFrequency configuration parameter in the MCU module.

Example(s)	Action	Generated output
	EthPeripheralBusFrequency = McuSPBFrequency = 100000000 EthSpbPeriodInNanoSeconds =1000000000/ EthPeripheralBusFrequency =10	#define ETH_17_GETHMACV2_FSPB_PERIOD_IN_NANOSEC (10U)
	EthPeripheralBusFrequency = McuSPBFrequency = 50000000 EthSpbPeriodInNanoSeconds =1000000000/ EthPeripheralBusFrequency =20	#define ETH_17_GETHMACV2_FSPB_PERIOD_IN_NANOSEC (20U)

1.1.23 Macro: ETH_17_GETHMACV2_TIMESTAMP_ADDEND_VAL

Table 23 ETH_17_GETHMACV2_TIMESTAMP_ADDEND_VAL

Name	ETH_17_GETHMACV2_TIMESTAMP_ADDEND_VAL	
Description	Timestamp Addend register value required for fine update to achieve 20 naneseconds resolution	
Verification method	The macro is generated as the value computed using the below equation: value = (4294967295)/ (EthOperateFrequency / 50 MHz) , where, EthOperateFrequency is the frequency value configured by the user in EthOperationFrequency configuration parameter which in turn refers McuGEthFrequency configuration parameter in the MCU module.	
Example(s)	Action	Generated output
	EthOperateFrequency = McuGEthFrequency = 150000000 EthAddendValue = 4294967295/ (150000000/ 50000000) = 1431655765	#define ETH_17_GETHMACV2_TIMESTAMP_ADDEND_VAL (1431655765U)
	EthOperateFrequency = McuGEthFrequency = 100000000 EthAddendValue = 4294967295/ (100000000/ 50000000) = 2147483647	#define ETH_17_GETHMACV2_TIMESTAMP_ADDEND_VAL (2147483647U)

1.1.24 Macro: ETH_17_GETHMACV2_KRNLRST_RGMII_WAITCNT

Table 24 ETH_17_GETHMACV2_KRNLRST_RGMII_WAITCNT

Name	ETH_17_GETHMACV2_KRNLRST_RGMII_WAITCNT	
Description	Wait time in nanoseconds after a kernel reset in RGMII mode	
Verification method	The macro is generated as 35 times the time period value in nanoseconds of the frequency value configured by the user in EthPeripheralBusFrequency configuration parameter which in turn refers McuSPBFrequency configuration parameter in the MCU module, that is, generated value = 35 fSPB cycles.	
Example(s)	**Action**	**Generated output**
	EthPeripheralBusFrequency = McuSPBFrequency = 100000000 WaitCnt = 35 * EthSpbPeriodInNanoSeconds = 35 * (1000000000/ EthPeripheralBusFrequency) =350	#define ETH_17_GETHMACV2_KRNLRST_RGMII_WAITCNT (350U)
	EthPeripheralBusFrequency = McuSPBFrequency = 50000000 WaitCnt = 35 * EthSpbPeriodInNanoSeconds = 35 * (1000000000/ EthPeripheralBusFrequency) =700	#define ETH_17_GETHMACV2_KRNLRST_RGMII_WAITCNT (700U)

1.1.25 Macro: ETH_17_GETHMACV2_KRNLRST_MII_WAITCNT

Table 25 ETH_17_GETHMACV2_KRNLRST_MII_WAITCNT

Name	ETH_17_GETHMACV2_KRNLRST_MII_WAITCNT	
Description	Wait time in nanoseconds after a kernel reset in MII/ RMII mode	
Verification method	The macro is generated as 70 times the time period value in nanoseconds of the frequency value configured by the user in EthPeripheralBusFrequency configuration parameter which in turn refers McuSPBFrequency configuration parameter in the MCU module, that is, generated value = 70 fSPB cycles.	
Example(s)	**Action**	**Generated output**
	EthPeripheralBusFrequency = McuSPBFrequency = 100000000 WaitCnt = 70 * EthSpbPeriodInNanoSeconds =	#define ETH_17_GETHMACV2_KRNLRST_MII_WAITCNT (700U)

	70 * (1000000000/ EthPeripheralBusFrequency) =700	
	EthPeripheralBusFrequency = McuSPBFrequency = 50000000 WaitCnt = 70 * EthSpbPeriodInNanoSeconds = 70 * (1000000000/ EthPeripheralBusFrequency) =1400	#define ETH_17_GETHMACV2_KRNLRST_MII_WAITCNT (1400U)

1.1.26 Macro: ETH_17_GETHMACV2_GETCNTRVALUES_API

Table 26 ETH_17_GETHMACV2_GETCNTRVALUES_API

Name	ETH_17_GETHMACV2_GETCNTRVALUES_API	
Description	Enables/ disables Eth_17_GEthMacV2_GetCounterValues API	
Verification method	The macro is generated as STD_ON if EthGetDropCountApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthGetDropCountApi as True	#define ETH_17_GETHMACV2_GETCNTRVALUES_API (STD_ON)
	Set EthGetDropCountApi as False	#define ETH_17_GETHMACV2_GETCNTRVALUES_API (STD_OFF)

1.1.27 Macro: ETH_17_GETHMACV2_GETRXSTATS_API

Table 27 ETH_17_GETHMACV2_GETRXSTATS_API

Name	ETH_17_GETHMACV2_GETRXSTATS_API	
Description	Enables/ disables Eth_17_GEthMacV2_GetRxStats API	
Verification method	The macro is generated as STD_ON if EthGetEtherStatsApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthGetEtherStatsApi as True	#define ETH_17_GETHMACV2_GETRXSTATS_API (STD_ON)
	Set EthGetEtherStatsApi as False	#define ETH_17_GETHMACV2_GETRXSTATS_API (STD_OFF)

1.1.28 Macro: ETH_17_GETHMACV2_GETTXSTATS_API

Table 28 ETH_17_GETHMACV2_GETTXSTATS_API

Name	ETH_17_GETHMACV2_GETTXSTATS_API	
Description	Enables/ disables Eth_17_GEthMacV2_GetTxStats API	
Verification method	The macro is generated as STD_ON if EthGetTxStatsApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthGetTxStatsApi as True	#define ETH_17_GETHMACV2_GETTXSTATS_API (STD_ON)
	Set EthGetTxStatsApi as False	#define ETH_17_GETHMACV2_GETTXSTATS_API (STD_OFF)

1.1.29 Macro: ETH_17_GETHMACV2_GETTXERRCNTRVAL_API

Table 29 ETH_17_GETHMACV2_GETTXERRCNTRVAL_API

Name	ETH_17_GETHMACV2_GETTXERRCNTRVAL_API	
Description	Enables/ disables Eth_17_GEthMacV2_GetTxErrorCounterValues API	
Verification method	The macro is generated as STD_ON if EthGetTxErrorCounterValuesApi configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthGetTxErrorCounterValuesApi as True	#define ETH_17_GETHMACV2_GETTXERRCNTRVAL_API (STD_ON)
	Set EthGetTxErrorCounterValuesApi as False	#define ETH_17_GETHMACV2_GETTXERRCNTRVAL_API (STD_OFF)

1.1.30 Macro: ETH_17_GETHMACV2_GLOBALTIMESUPPORT

Table 30 ETH_17_GETHMACV2_GLOBALTIMESUPPORT

Name	ETH_17_GETHMACV2_GLOBALTIMESUPPORT	
Description	Enables/disables Eth_17_GEthMacV2_GetCurrentTime, Eth_17_GEthMacV2_EnableEgressTimeStamp, Eth_17_GEthMacV2_GetEgressTimeStamp, Eth_17_GEthMacV2_GetIngressTimeStamp APIs	
Verification method	The macro is generated as STD_ON if EthGlobalTimeSupport configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthGlobalTimeSupport as True	#define ETH_17_GETHMACV2_GLOBALTIMESUPPORT (STD_ON)

	Set EthGlobalTimeSupport as False	#define ETH_17_GETHMACV2_GLOBALTIMESUPPORT (STD_OFF)

1.1.31 Macro: ETH_17_GETHMACV2_MULTICORE_ERROR_DETECT

Table 31 ETH_17_GETHMACV2_MULTICORE_ERROR_DETECT

Name	ETH_17_GETHMACV2_MULTICORE_ERROR_DETECT	
Description	Enables/disables multi core error detection and reporting from the core.	
Verification method	The macro is generated as STD_ON if EthMultiCoreErrorDetect configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthMultiCoreErrorDetect as True	#define ETH_17_GETHMACV2_MULTICORE_ERROR_DETECT (STD_ON)
	Set EthMultiCoreErrorDetect as False	#define ETH_17_GETHMACV2_MULTICORE_ERROR_DETECT (STD_OFF)

1.1.32 Macro: ETH_17_GETHMACV2_ICMP_CHECKSUMOFFLOAD_ENABLE

Table 32 ETH_17_GETHMACV2_ICMP_CHECKSUMOFFLOAD_ENABLE

Name	ETH_17_GETHMACV2_ICMP_CHECKSUMOFFLOAD_ENABLE	
Description	Enables/ disables the checksum offloading of ICMP frames for both transmission and reception.	
Verification method	The macro is generated as STD_ON if EthCtrlEnableOffloadChecksumICMP configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlEnableOffloadChecksumICMP as True	#define ETH_17_GETHMACV2_ICMP_CHECKSUMOFFLOAD_ENABLE (STD_ON)
	Set EthCtrlEnableOffloadChecksumICMP as False	#define ETH_17_GETHMACV2_ICMP_CHECKSUMOFFLOAD_ENABLE (STD_OFF)

1.1.33 Macro: ETH_17_GETHMACV2_IPV4_CHECKSUMOFFLOAD_ENABLE

Table 33 ETH_17_GETHMACV2_IPV4_CHECKSUMOFFLOAD_ENABLE

Name	ETH_17_GETHMACV2_IPV4_CHECKSUMOFFLOAD_ENABLE	
Description	Enables/ disables the checksum offloading of IPv4 frames for both transmission and reception.	
Verification method	The macro is generated as STD_ON if EthCtrlEnableOffloadChecksumIPv4 configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**

	Set EthCtrlEnableOffloadChecksumIPv4 as True	#define ETH_17_GETHMACV2_IPV4_CHECKSUMOFFLOAD_ENABLE (STD_ON)
	Set EthCtrlEnableOffloadChecksumIPv4 as False	#define ETH_17_GETHMACV2_IPV4_CHECKSUMOFFLOAD_ENABLE (STD_OFF)

1.1.34 Macro: ETH_17_GETHMACV2_TCP_CHECKSUMOFFLOAD_ENABLE

Table 34 ETH_17_GETHMACV2_TCP_CHECKSUMOFFLOAD_ENABLE

Name	ETH_17_GETHMACV2_TCP_CHECKSUMOFFLOAD_ENABLE	
Description	Enables/ disables the checksum offloading of TCP frames for both transmission and reception.	
Verification method	The macro is generated as STD_ON if EthCtrlEnableOffloadChecksumTCP configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlEnableOffloadChecksumTCP as True	#define ETH_17_GETHMACV2_TCP_CHECKSUMOFFLOAD_ENABLE (STD_ON)
	Set EthCtrlEnableOffloadChecksumTCP as False	#define ETH_17_GETHMACV2_TCP_CHECKSUMOFFLOAD_ENABLE (STD_OFF)

1.1.35 Macro: ETH_17_GETHMACV2_UDP_CHECKSUMOFFLOAD_ENABLE

Table 35 ETH_17_GETHMACV2_UDP_CHECKSUMOFFLOAD_ENABLE

Name	ETH_17_GETHMACV2_UDP_CHECKSUMOFFLOAD_ENABLE	
Description	Enables/ disables the checksum offloading of UDP frames for both transmission and reception.	
Verification method	The macro is generated as STD_ON if EthCtrlEnableOffloadChecksumUDP configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlEnableOffloadChecksumUDP as True	#define ETH_17_GETHMACV2_UDP_CHECKSUMOFFLOAD_ENABLE (STD_ON)
	Set EthCtrlEnableOffloadChecksumUDP as False	#define ETH_17_GETHMACV2_UDP_CHECKSUMOFFLOAD_ENABLE (STD_OFF)

1.1.36 Macro: ETH_17_GETHMACV2_INIT_API_MODE

Table 36 ETH_17_GETHMACV2_INIT_API_MODE

Name	ETH_17_GETHMACV2_INIT_API_MODE	
Description	Decides the mode of execution of the Init API.	
Verification method	The macro is generated as ETH_17_GETHMACV2_MCAL_SUPERVISOR if EthInitApiMode configuration parameter is set to 'ETH_MCAL_SUPERVISOR' else the macro is generated as ETH_17_GETHMACV2_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	Set EthInitApiMode as ETH_MCAL_USER1	#define ETH_17_GETHMACV2_INIT_API_MODE (ETH_17_GETHMACV2_MCAL_USER1)
	Set EthInitApiMode as ETH_MCAL_SUPERVISOR	#define ETH_17_GETHMACV2_INIT_API_MODE (ETH_17_GETHMACV2_MCAL_SUPERVISOR)

1.1.37 Macro: ETH_17_GETHMACV2_RUNTIME_API_MODE

Table 37 ETH_17_GETHMACV2_RUNTIME_API_MODE

Name	ETH_17_GETHMACV2_RUNTIME_API_MODE	
Description	Decides the mode of execution of Run Time APIs	
Verification method	The macro is generated as ETH_17_GETHMACV2_MCAL_SUPERVISOR if EthRuntimeApiMode configuration parameter is set to 'ETH_MCAL_SUPERVISOR' else the macro is generated as ETH_17_GETHMACV2_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	Set EthRuntimeApiMode as ETH_MCAL_USER1	#define ETH_17_GETHMACV2_RUNTIME_API_MODE (ETH_17_GETHMACV2_MCAL_USER1)
	Set EthRuntimeApiMode as ETH_MCAL_SUPERVISOR	#define ETH_17_GETHMACV2_RUNTIME_API_MODE (ETH_17_GETHMACV2_MCAL_SUPERVISOR)

1.1.38 Macro: ETH_17_GETHMACV2_CTRL<CtrlIndex>_CORE<CoreId>

Table 38 ETH_17_GETHMACV2_CTRL<CtrlIndex>_CORE<CoreId>

Name	ETH_17_GETHMACV2_CTRL<CtrlIndex>_CORE<CoreId>	
Description	Indicates the core Id to which the ethernet controller with <CtrlIndex> is assigned. Controller<CtrlIndex> configured to <Core Id>.	
Verification method	The macro is generated as STD_ON if available controller is configured to any core else the macro will not be generated.	
Example(s)	**Action**	**Generated output**
	Configure controller 0 to Core 1	#define ETH_17_GETHMACV2_CTRL0_CORE1 (STD_ON)

	Configure controller 1 to Core 2	#define ETH_17_GETHMACV2_CTRL1_CORE2 (STD_ON)

1.1.39 Macro: ETH_17_GETHMACV2_CTRL<CtrlIndex>_CONFIGURED

Table 39 ETH_17_GETHMACV2_CTRL<CtrlIndex>_CONFIGURED

Name	ETH_17_GETHMACV2_CTRL<CtrlIndex>_CONFIGURED	
Description	Controllers configured in the project.	
Verification method	The macro is generated as STD_ON if available controller is configured else the macro will not be generated.	
Example(s)	**Action**	**Generated output**
	Configure controller 0.	#define ETH_17_GETHMACV2_CTRL0_CONFIGURED (STD_ON)
	Configure controller 1.	#define ETH_17_GETHMACV2_CTRL1_CONFIGURED (STD_ON)

1.1.40 Macro: ETH_17_GETHMACV2_MAX_CTRL_CORE<CoreId>

Table 40 ETH_17_GETHMACV2_MAX_CTRL_CORE<CoreId>

Name	ETH_17_GETHMACV2_MAX_CTRL_CORE<CoreId>	
Description	Maximum Controllers allocated to core with <CoreId> Value 255 - represents core is not available in current device.	
Verification method	The macro is generated as a numeric value that corresponds to the max controllers configured in the particular core.	
Example(s)	**Action**	**Generated output**
	Configure controller 0 and 1 to Core 0.	#define ETH_17_GETHMACV2_MAX_CTRL_CORE0 (2U)
	No controller configured to Core 1.	#define ETH_17_GETHMACV2_MAX_CTRL_CORE1 (0U)

1.1.41 Macro: ETH_17_GETHMACV2_MAX_CORES

Table 41 ETH_17_GETHMACV2_MAX_CORES

Name	ETH_17_GETHMACV2_MAX_CORES	
Description	Maximum available cores in the device. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on maximum number of cores available.	
Example(s)	**Action**	**Generated output**
	Device has 6 cores.	#define ETH_17_GETHMACV2_MAX_CORES (6U)

	Device has 4 cores.	#define ETH_17_GETHMACV2_MAX_CORES (4U)

1.1.42 Macro: ETH_17_GETHMACV2_MAX_CONTROLLERS

Table 42 ETH_17_GETHMACV2_MAX_CONTROLLERS

Name	ETH_17_GETHMACV2_MAX_CONTROLLERS	
Description	Maximum available controllers in the device. *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated based on maximum number of controllers available in the device.	
Example(s)	**Action**	**Generated output**
	Device has 1 controller.	#define ETH_17_GETHMACV2_MAX_CONTROLLERS (1U)
	Device has 2 controllers.	#define ETH_17_GETHMACV2_MAX_CONTROLLERS (2U)

1.2 File: Eth_17_GEthMacV2[_<variant>]_PBcfg.c

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

1.2.1 Structure: Eth_17_GEthMacV2_TxFifoCfgCtrl<CtrlIdx>[_Variant][Egress FIFO count]

Table 43 Eth_17_GEthMacV2_TxFifoCfgCtrl<CtrlIdx>[_Variant][Egress FIFO count]

Name	Eth_17_GEthMacV2_TxFifoCfgCtrl<CtrlIdx>[_Variant][Egress FIFO count]	
Type	Eth_17_GEthMacV2_TxFifoCfgType	
Description	Structure to store the configured parameters of egress FIFOs.	
Verification method	The generated structure is present in Eth_17_GEthMacV2[_<variant>]_PBcfg.c 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 controller 0 with 4 egress FIFOs and schedule these 4 egress FIFOs for transmission.	/* Egress FIFO configuration */ static const Eth_17_GEthMacV2_TxFifoCfgType Eth_17_GEthMacV2_TxFifoCfgCtrl0[4] = { {

```
  /* HiCredit Value */
  (uint32)0x0U,
  /* LoCredit Value */
  (uint32)0x0U,
  /* Egress Fifo buffer total */
  (uint16)10U,
  /* Egress Fifo length byte */
  (uint16)1522U,
  /* Egress Fifo length byte aligned */
  (uint16)1528U,
  /* Idle Slope Credit Value or
  Configured weight for WRR algorithm */
  (uint16)0x0U,
  /* Send Slope Credit Value */
  (uint16)0x0U,
  /* Egress Fifo Index*/
  (uint8)0U,
  /* DMA Channel Weight */
  (uint8)0U,
  /* Egress Queue Mode */
  (uint8)0x2U,
  /* Egress Queue Size */
  (uint8)0x7U
 },
 {
  /* HiCredit Value */
  (uint32)0x0U,
  /* LoCredit Value */
  (uint32)0x0U,
  /* Egress Fifo buffer total */
  (uint16)4U,
  /* Egress Fifo length byte */
  (uint16)1000U,
  /* Egress Fifo length byte aligned */
  (uint16)1000U,
  /* Idle Slope Credit Value or
  Configured weight for WRR algorithm */
```

```
  (uint16)0x0U,
  /* Send Slope Credit Value */
  (uint16)0x0U,
  /* Egress Fifo Index*/
  (uint8)1U,
  /* DMA Channel Weight */
  (uint8)0U,
  /* Egress Queue Mode */
  (uint8)0x2U,
  /* Egress Queue Size */
  (uint8)0x5U
 },
 {
  /* HiCredit Value */
  (uint32)0x0U,
  /* LoCredit Value */
  (uint32)0x0U,
  /* Egress Fifo buffer total */
  (uint16)4U,
  /* Egress Fifo length byte */
  (uint16)1000U,
  /* Egress Fifo length byte aligned */
  (uint16)1000U,
  /* Idle Slope Credit Value or
  Configured weight for WRR algorithm */
  (uint16)0x0U,
  /* Send Slope Credit Value */
  (uint16)0x0U,
  /* Egress Fifo Index*/
  (uint8)2U,
  /* DMA Channel Weight */
  (uint8)0U,
  /* Egress Queue Mode */
  (uint8)0x2U,
  /* Egress Queue Size */
  (uint8)0x5U
 },
```

	{
	/* HiCredit Value */
	(uint32)0x0U,
	/* LoCredit Value */
	(uint32)0x0U,
	/* Egress Fifo buffer total */
	(uint16)4U,
	/* Egress Fifo length byte */
	(uint16)500U,
	/* Egress Fifo length byte aligned */
	(uint16)504U,
	/* Idle Slope Credit Value or
	Configured weight for WRR algorithm */
	(uint16)0x0U,
	/* Send Slope Credit Value */
	(uint16)0x0U,
	/* Egress Fifo Index*/
	(uint8)3U,
	/* DMA Channel Weight */
	(uint8)0U,
	/* Egress Queue Mode */
	(uint8)0x2U,
	/* Egress Queue Size */
	(uint8)0x3U
	}
	};

Configure controller 1 with 2 egress FIFOs and schedule these 2 egress FIFOs for transmission (variant-aware, variant name is ‘Petrol’).	/* Egress FIFO configuration */ static const Eth_17_GEthMacV2_TxFifoCfgType Eth_17_GEthMacV2_TxFifoCfgCtrl1_Petrol[2] = { { /* HiCredit Value */ (uint32)0x0U, /* LoCredit Value */ (uint32)0x0U, /* Egress Fifo buffer total */ (uint16)10U,

```
  /* Egress Fifo length byte */
  (uint16)1522U,
  /* Egress Fifo length byte aligned */
  (uint16)1528U,
  /* Idle Slope Credit Value or
  Configured weight for WRR algorithm */
  (uint16)0x0U,
  /* Send Slope Credit Value */
  (uint16)0x0U,
  /* Egress Fifo Index*/
  (uint8)0U,
  /* DMA Channel Weight */
  (uint8)0U,
  /* Egress Queue Mode */
  (uint8)0x2U,
  /* Egress Queue Size */
  (uint8)0x7U
 },
 {
  /* HiCredit Value */
  (uint32)0x0U,
  /* LoCredit Value */
  (uint32)0x0U,
  /* Egress Fifo buffer total */
  (uint16)4U,
  /* Egress Fifo length byte */
  (uint16)1000U,
  /* Egress Fifo length byte aligned */
  (uint16)1000U,
  /* Idle Slope Credit Value or
  Configured weight for WRR algorithm */
  (uint16)0x0U,
  /* Send Slope Credit Value */
  (uint16)0x0U,
  /* Egress Fifo Index*/
  (uint8)1U,
  /* DMA Channel Weight */
```

		(uint8)0U, /* Egress Queue Mode */ (uint8)0x2U, /* Egress Queue Size */ (uint8)0x5U } }

1.2.1.1 Member: HiCredit

Table 44 HiCredit

Name	HiCredit	
Type	uint32	
Description	The maximum value that can be accumulated in the credit value for the credit-based shaper algorithm.	
Verification method	The structure member is generated from the configuration parameter EthCtrlConfigShaperHiCredit value scaled by 1024. The generated value is 0 if EthCtrlConfigShaper container is not configured. *Note: EthCtrlConfigShaperHiCredit should be configured only when credit-based shaper feature is required.*	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigShaperHiCredit to 8192 within the EthCtrlConfigShaper container. HiCredit = 8192 * 1024 = 8388608	(uint32)8388608,
	Do not configure EthCtrlConfigShaper container.	(uint32)0,

1.2.1.2 Member: LoCredit

Table 45 LoCredit

Name	LoCredit
Type	uint32
Description	The minimum value that can be accumulated in the credit value for the credit-based shaper algorithm.
Verification method	The structure member is generated as the two's complement value of the configuration parameter EthCtrlConfigShaperLoCredit value scaled by 1024. The generated value is 0 if EthCtrlConfigShaper container is not configured.

	Note: EthCtrlConfigShaperLoCredit should be configured only when credit-based shaper feature is required.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigShaperLoCredit to 4096 within the EthCtrlConfigShaper container. HiCredit = two's complement of (4096 * 1024) = 532676608	(uint32)532676608,
	Do not configure EthCtrlConfigShaper container.	(uint32)0,

1.2.1.3 Member: NumOfTxBuffers

Table 46 NumOfTxBuffers

Name	NumOfTxBuffers	
Type	uint16	
Description	Number of transmit buffers configured within an egress FIFO of the Ethernet controller	
Verification method	The structure member is generated with the value configured in EthCtrlConfigEgressFifoBufTotal parameter within the EthCtrlConfigEgressFifo container.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigEgressFifoBufTotal = 10 within the corresponding EthCtrlConfigEgressFifo container.	(uint16)10U,
	Set EthCtrlConfigEgressFifoBufTotal = 4 within the corresponding EthCtrlConfigEgressFifo container.	(uint16)4U,

1.2.1.4 Member: TxBufferSize

Table 47 TxBufferSize

Name	TxBufferSize	
Type	uint16	
Description	Configured length of each buffer in bytes within the egress FIFO.	
Verification method	The structure member is generated with the value configured in EthCtrlConfigEgressFifoBufLenByte parameter within the EthCtrlConfigEgressFifo container.	
Example(s)	**Action**	**Generated output**

	Set EthCtrlConfigEgressFifoBufLenByte = 120 within the corresponding EthCtrlConfigEgressFifo container.	(uint16)120U,
	Set EthCtrlConfigEgressFifoBufLenByte = 1522 within the corresponding EthCtrlConfigEgressFifo container.	(uint16)1522U,

1.2.1.5 Member: TxBufferAlignSize

Table 48 TxBufferAlignSize

Name	TxBufferAlignSize	
Type	uint16	
Description	The length of each buffer in bytes aligned to 8-byte boundary within the egress FIFO.	
Verification method	The structure member is generated with the 8-byte aligned value configured in EthCtrlConfigEgressFifoBufLenByte parameter within the EthCtrlConfigEgressFifo container.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigEgressFifoBufLenByte = 60 within the corresponding EthCtrlConfigEgressFifo container.	(uint16)64U,
	Set EthCtrlConfigEgressFifoBufLenByte = 1522 within the corresponding EthCtrlConfigEgressFifo container.	(uint16)1528U,

1.2.1.6 Member: IdleSlopeCredit

Table 49 IdleSlopeCredit

Name	IdleSlopeCredit
Type	uint16
Description	Element to store the idleSlopeCredit value required for the credit-based shaper algorithm. If credit-based shaper is not used and if the Tx scheduling algorithm is chosen as Weighted Round Robin (WRR), then this element stores the weight required for the WRR algorithm).
Verification method	The structure member is generated as the computed idleSlope credit in bits per cycle scaled by 1024. The configuration parameter EthCtrlConfigShaperIdleSlope is divided by the PortTransmitRate (MAC speed) to get the idle bandwidth value which is then multiplied by creditValue and scaled by 1024. The structure member is generated as the value configured in EthCtrlConfigSchedulerPredecessorOrder parameter if the Tx scheduling algorithm is chosen as Weighted Round Robin. The generated value is 0 if EthCtrlConfigShaper container is not configured and the Tx scheduling algorithm is chosen as Weighted Round Robin.

Example(s)	Action	Generated output
	Configure EthCtrlConfigShaper container with EthCtrlConfigShaperIdleSlope = 70000000. Set EthCtrlMacLayerSpeed = ETH_MAC_LAYER_SPEED_100M, EthCtrlMacLayerType = ETH_MAC_LAYER_TYPE_XGMII, EthCtrlMacLayerSubType= RGMII. IdleSlopeCredit = (70000000/ 100000000) * 4 * 1024 = 2867	(uint16)2867U,
	Configure EthCtrlConfigSchedulerAlgorithm = ETH_SCHEDULER_WEIGHTED_ROUND_ROBIN and EthCtrlConfigSchedulerPredecessorOrder = 2.	(uint16)2U,
	Set EthCtrlConfigSchedulerAlgorithm = ETH_SCHEDULER_STRICT_PRIORITY and do not configure EthCtrlConfigShaper container.	(uint16)0U,

1.2.1.7 Member: SendSlopeCredit

Table 50 SendSlopeCredit

Name	SendSlopeCredit	
Type	uint16	
Description	Element to store the sendSlope Credit value required for the credit-based shaper algorithm.	
Verification method	The structure member is generated as the computed sendSlope credit in bits per cycle scaled by 1024. The value of sendSlope is derived from the configuration parameter EthCtrlConfigShaperIdleSlope value. The generated value is (1 - the idle bandwidth value derived earlier), which is then multiplied by creditValue and scaled by 1024. The generated value is 0 if EthCtrlConfigShaper container is not configured.	
Example(s)	**Action**	**Generated output**
	Configure EthCtrlConfigShaper container with EthCtrlConfigShaperIdleSlope = 70000000. Set EthCtrlMacLayerSpeed = ETH_MAC_LAYER_SPEED_100M, EthCtrlMacLayerType = ETH_MAC_LAYER_TYPE_XGMII, EthCtrlMacLayerSubType= RGMII. IdleSlopeCredit = (1 - (70000000/ 100000000)) * 4 * 1024 = 1229	(uint16)1229U,
	Do not configure EthCtrlConfigShaper container.	(uint16)0U,

1.2.1.8 Member: TxFifoIdx

Table 51 TxFifoIdx

Name	TxFifoIdx	
Type	uint8	
Description	Index of the configured egress FIFO	
Verification method	The structure member is generated with the value configured in EthCtrlConfigEgressFifoIdx parameter within the EthCtrlConfigEgressFifo container.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigEgressFifoIdx = 0 within the corresponding EthCtrlConfigEgressFifo container.	(uint8)0U,
	Set EthCtrlConfigEgressFifoIdx = 3 within the corresponding EthCtrlConfigEgressFifo container.	(uint8)3U,

1.2.1.9 Member: TxDmaChnlWght

Table 52 TxDmaChnlWght

Name	TxDmaChnlWght	
Type	uint8	
Description	Configured weight of the Transmit DMA channel in case of Weighted Round Robin or Weighted Strict Priority DMA scheduling mechanism.	
Verification method	The structure member is generated with the value configured in EthCtrlConfigDMAArbitrationWeight parameter of EthCtrlConfigDMAWeightAssignment container for the egress FIFO which inturn is within the EthCtrlConfigDMAArbitration container. The generated value is 0, if the Tx DMA scheduling mechanism is chosen as Fixed Priority.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigDMAArbitrationAlgorithm = ETH_DMA_ARBITRATION_WEIGHTED_ROUND_ROBIN and EthCtrlConfigDMAArbitrationWeight = 4 for the corresponding egress FIFO.	(uint8)4U,
	Set EthCtrlConfigDMAArbitrationAlgorithm = ETH_DMA_ARBITRATION_WEIGHTED_STRICT_PRIORITY and EthCtrlConfigDMAArbitrationWeight = 8 for the corresponding egress FIFO.	(uint8)8U,
	Set EthCtrlConfigDMAArbitrationAlgorithm = ETH_DMA_ARBITRATION_FIXED_PRIORITY.	(uint8)0U,

1.2.1.10 Member: TxQueueMode

Table 53 TxQueueMode

Name	TxQueueMode
Type	uint8
Description	Indicates the mode of the transmit queue, either Normal mode or CBS mode.
Verification method	The structure member is generated with the value 2 if the EthCtrlConfigShaper container is not configured for the corresponding egress FIFO, that is, normal mode Tx queue operation.

	The generated value is 1 (CBS mode of Tx queue operation), if the EthCtrlConfigShaper container is configured for the corresponding egress FIFO referenced by EthCtrlConfigShaperPredecessorFifoRef parameter.	
Example(s)	**Action**	**Generated output**
	Configure egress FIFOs with valid parameters, but do not configure EthCtrlConfigShaper container for any of the egress FIFOs.	(uint8)2U,
	Configure egress FIFOs with valid parameters, and configure EthCtrlConfigShaper container for the egress FIFO referenced by EthCtrlConfigShaperPredecessorFifoRef parameter. Check the generated output for this referenced egress FIFO.	(uint8)1U,

1.2.1.11 Member: TxQueueSize

Table 54 TxQueueSize

Name	TxQueueSize	
Type	uint8	
Description	Indicates the size of the allocated Transmit queues in blocks of 256 bytes, where the value 0 indicates size of 256 bytes. The allocated queue size = (TxQueueSize + 1) * 256 bytes.	
Verification method	The structure member is generated based on the values configured in EthCtrlConfigEgressFifoBufLenByte and EthCtrlConfigEgressFifoBufTotal parameters. The MTL Tx FIFO size of 4 kB is allocated among the Tx queues based on the configured egress FIFO buffer length (EthCtrlConfigEgressFifoBufLenByte) and also on the egress FIFO buffer count (EthCtrlConfigEgressFifoBufTotal), and the computed queue size (multiple of 256) is divided by 256 and then reduced by 1 to get TxQueueSize value.	
Example(s)	**Action**	**Generated output**
	Configure 4 egress FIFOs and set EthCtrlConfigEgressFifoBufTotal = 4 and EthCtrlConfigEgressFifoBufLenByte = 64 for all egress FIFOs. Check the generated output for the element TxQueueSize of all egress FIFOs. MTL Tx FIFO size of 4 kB is equally distributed amoung the 4 egress FIFOs, that is, 1 Kb for each FIFO. TxQueueSize = (1024/ 256) – 1 = 3	(uint8)3U,
	Configure 4 egress FIFOs and set EthCtrlConfigEgressFifoBufTotal = 4 and EthCtrlConfigEgressFifoBufLenByte = 64 for first two egress FIFOs, EthCtrlConfigEgressFifoBufTotal = 4 and EthCtrlConfigEgressFifoBufLenByte =	{First FIFO:} (uint8)1U, {Second FIFO:} (uint8)1U, {Third FIFO:} (uint8)7U, {Fourth FIFO:} (uint8)3U,

	1522 for third egress FIFO, and EthCtrlConfigEgressFifoBufTotal = 4 and EthCtrlConfigEgressFifoBufLenByte = 512 for the fourth egress FIFO. Check the generated output for the element TxQueueSize of the individual egress FIFOs. The allocation is done such that atleast one complete frame of EthCtrlConfigEgressFifoBufLenByte size can be placed within the corresponding queue mapped to the egress FIFO, and the remaining space is distributed between them. MTL Tx FIFO size of 4 kB is distributed among the 4 egress FIFOs as below: First FIFO: TxQueueSize = (512/ 256) – 1 = 1 Second FIFO: TxQueueSize = (512/ 256) – 1 = 1 Third FIFO: TxQueueSize = (2048/ 256) – 1 = 7 Fourth FIFO: TxQueueSize = (1024/ 256) – 1 = 3	

1.2.2 Array: Eth_17_GEthMacV2_TxFifoChnlMapCtrl<CtrlIdx>[_Variant][Egress FIFO count]

Table 55 Eth_17_GEthMacV2_TxFifoChnlMapCtrl<CtrlIdx>[_Variant][Egress FIFO count]

Name	Eth_17_GEthMacV2_TxFifoChnlMapCtrl<CtrlIdx>[_Variant][Egress FIFO count]	
Type	uint8	
Description	Array to store mapping between the configured egress FIFO indices to Tx channels	
Verification method	Eth_17_GEthMacV2_TxFifoChnlMapCtrl[<x>][<y>], where <x> = Index of controller configured and <y> is the number of egress FIFOs configured and scheduled for transmission within the controller<x>. The egress FIFO indices are mapped to the Tx channels based on their priorities that are determined by the EthCtrlConfigSchedulerPredecessorOrder parameter. The egress FIFO that is referenced (via Shaper or directly) by the scheduler with the highest value of EthCtrlConfigSchedulerPredecessorOrder is the FIFO with the highest priority and this FIFO is assigned to Tx channel 3. The FIFO with the next higher priority is assigned to Tx channel 2 and so on. <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 4 egress FIFOs within controller 0 and set the EthCtrlConfigSchedulerPredecessorOrder	static const uint8 Eth_17_GEthMacV2_TxFifoChnlMapCtrl0[4] =

values to the referenced egress FIFOs as below:
Order 0 – EthCtrlConfigEgressFifoIdx 0
Order 1 - EthCtrlConfigEgressFifoIdx 3
Order 2 – EthCtrlConfigEgressFifoIdx 1
Order 3 – EthCtrlConfigEgressFifoIdx 2
	{
	0U,
	2U,
	3U,
	1U
	};

Configure 4 egress FIFOs within controller 1 and set the EthCtrlConfigSchedulerPredecessorOrder values to the referenced egress FIFOs as below:
Order 0 – EthCtrlConfigEgressFifoIdx 0
Order 1 - EthCtrlConfigEgressFifoIdx 1
Order 2 – EthCtrlConfigEgressFifoIdx 2
Order 3 – EthCtrlConfigEgressFifoIdx 3
(variant-aware, variant name is 'Petrol').
	static const uint8 Eth_17_GEthMacV2_TxFifoChnlMapCtrl1_Petrol[4] =
	{
	0U,
	1U,
	2U,
	3U
	};

1.2.3 Array: Eth_17_GEthMacV2_TxChnlFifoMapCtrl<CtrlIdx>[_Variant][Max supported DMA channels]

Table 56 Eth_17_GEthMacV2_TxChnlFifoMapCtrl<CtrlIdx>[_Variant][Max supported DMA channels]

Name	Eth_17_GEthMacV2_TxChnlFifoMapCtrl<CtrlIdx>[_Variant][Max supported DMA channels]	
Type	uint8	
Description	Array to store mapping between the Tx channels to the configured egress FIFO indices	
Verification method	Eth_17_GEthMacV2_TxChnlFifoMapCtrl[<x>][<y>], where <x> = Index of controller configured and <y> is the maximum supported Tx DMA channels by the controller<x>. The values are generated by performing the reverse mapping of the array Eth_17_GEthMacV2_TxFifoChnlMapCtrl<CtrlIdx>[_Variant][Egress FIFO count] and if the Tx channel is not used, the generated value is 255. <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 4 egress FIFOs within controller 0 and set the EthCtrlConfigSchedulerPredecessorOrder values to the referenced egress FIFOs as below:
	Order 0 – EthCtrlConfigEgressFifoIdx 0
	Order 1 - EthCtrlConfigEgressFifoIdx 3
	Order 2 – EthCtrlConfigEgressFifoIdx 1
	Order 3 – EthCtrlConfigEgressFifoIdx 2
		static const uint8 Eth_17_GEthMacV2_TxChnlFifoMapCtrl0[4] =
		{
		0U,
		3U,
		1U,
		2U
		};

	Configure 3 egress FIFOs within controller 1 and set the EthCtrlConfigSchedulerPredecessorOrder values to the referenced egress FIFOs as below:
	Order 0 – EthCtrlConfigEgressFifoIdx 0
	Order 1 - EthCtrlConfigEgressFifoIdx 1
	Order 2 – EthCtrlConfigEgressFifoIdx 2
	(variant-aware, variant name is 'Petrol').
		static const uint8 Eth_17_GEthMacV2_TxChnlFifoMapCtrl1_Petrol[4] =
		{
		0U,
		1U,
		2U,
		255U
		};

1.2.4 Array: Eth_17_GEthMacV2_TxPrioFifoMapCtrl<CtrlIdx>[_Variant][Max supported priorities]

Table 57 Eth_17_GEthMacV2_TxPrioFifoMapCtrl<CtrlIdx>[_Variant][Max supported priorities]

Name	Eth_17_GEthMacV2_TxPrioFifoMapCtrl<CtrlIdx>[_Variant][Max supported priorities]	
Type	uint8	
Description	Array to store mapping between the configured priorities to the egress FIFO indices	
Verification method	Eth_17_GEthMacV2_TxPrioFifoMapCtrl[<x>][<y>], where <x> = Index of controller configured and <y> is the maximum number of priorities supported (value 8). The priority values 0 – 7 are mapped to the egress FIFO indices based on the values configured within the EthCtrlConfigEgressFifoPriorityAssignment container. If a priority value is not configured, the value is generated as 0xFF against the corresponding index. <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 4 egress FIFOs within controller 0 and set the priority values within EthCtrlConfigEgressFifoPriorityAssignment container of each egress FIFO as below:
	EthCtrlConfigEgressFifoIdx 0 – Priority values 0, 1, 2
	EthCtrlConfigEgressFifoIdx 1 – Priority values 3
	EthCtrlConfigEgressFifoIdx 2 – Priority values 4, 5
	EthCtrlConfigEgressFifoIdx 3 – Priority values 6, 7
		static const uint8 Eth_17_GEthMacV2_TxPrioFifoMapCtrl0[8] =
		{
		0x0U,
		0x0U,
		0x0U,
		0x1U,
		0x2U,
		0x2U,
		0x3U,
		0x3U
		};

	Configure 2 egress FIFOs within controller 1 and set the priority values within EthCtrlConfigEgressFifoPriorityAssignment container of each egress FIFO as below:
	EthCtrlConfigEgressFifoIdx 0 – Priority values 0, 1, 2
	EthCtrlConfigEgressFifoIdx 1 – Priority values 4, 6, 7
	(variant-aware, variant name is 'Petrol').
		static const uint8 Eth_17_GEthMacV2_TxPrioFifoMapCtrl1_Petrol[4] =
		{
		0x0U,
		0x0U,
		0x0U,
		0xFFU,
		0x1U,
		0xFFU,
		0x1U,
		0x1U
		};

1.2.5 Structure: Eth_17_GEthMacV2_RxFifoCfgCtrl<CtrlIdx>[_Variant][Ingress FIFO count]

Table 58 Eth_17_GEthMacV2_RxFifoCfgCtrl<CtrlIdx>[_Variant][Ingress FIFO count]

Name	Eth_17_GEthMacV2_RxFifoCfgCtrl<CtrlIdx>[_Variant][Ingress FIFO count]	
Type	Eth_17_GEthMacV2_RxFifoCfgType	
Description	Structure to store the configured parameters of ingress FIFOs.	
Verification method	The generated structure is present in Eth_17_GEthMacV2[_<variant>]_PBcfg.c 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 controller 0 with 4 ingress FIFOs.	/* Ingress FIFO configuration */ static const Eth_17_GEthMacV2_RxFifoCfgType Eth_17_GEthMacV2_RxFifoCfgCtrl0[4] = { { /*Ingress Fifo buffer total */ (uint16)4U, /* Ingress Fifo length byte */ (uint16)1522U, /* Ingress Fifo length byte aligned */ (uint16)1528U, /* Ingress Fifo Index*/ (uint8)0U,

```
  /* Priority Configured for FIFO
  (the set bits are the configured priorities) */
  (uint8)0x0fU,
  /* Ingress queue size */
  (uint8)7U
 },
 {
  /*Ingress Fifo buffer total */
  (uint16)4U,
  /* Ingress Fifo length byte */
  (uint16)1522U,
  /* Ingress Fifo length byte aligned */
  (uint16)1528U,
  /* Ingress Fifo Index*/
  (uint8)1U,
  /* Priority Configured for FIFO
  (the set bits are the configured priorities) */
  (uint8)0x30U,
  /* Ingress queue size */
  (uint8)7U
 },
 {
  /*Ingress Fifo buffer total */
  (uint16)4U,
  /* Ingress Fifo length byte */
  (uint16)1522U,
  /* Ingress Fifo length byte aligned */
  (uint16)1528U,
  /* Ingress Fifo Index*/
  (uint8)2U,
  /* Priority Configured for FIFO
  (the set bits are the configured priorities) */
  (uint8)0x40U,
  /* Ingress queue size */
  (uint8)7U
 },
 {
```

	/*Ingress Fifo buffer total */
	(uint16)4U,
	/* Ingress Fifo length byte */
	(uint16)1522U,
	/* Ingress Fifo length byte aligned */
	(uint16)1528U,
	/* Ingress Fifo Index*/
	(uint8)3U,
	/* Priority Configured for FIFO
	(the set bits are the configured priorities) */
	(uint8)0x80U,
	/* Ingress queue size */
	(uint8)7U
	}
	};

Configure controller 1 with 1 ingress FIFO.
(variant-aware, variant name is ‘Petrol’).
	/* Ingress FIFO configuration */
	static const Eth_17_GEthMacV2_RxFifoCfgType Eth_17_GEthMacV2_RxFifoCfgCtrl1_Petrol[1] =
	{
	{
	/*Ingress Fifo buffer total */
	(uint16)4U,
	/* Ingress Fifo length byte */
	(uint16)1522U,
	/* Ingress Fifo length byte aligned */
	(uint16)1528U,
	/* Ingress Fifo Index*/
	(uint8)0U,
	/* Priority Configured for FIFO
	(the set bits are the configured priorities) */
	(uint8)0xffU,
	/* Ingress queue size */
	(uint8)31U
	}
	}

1.2.5.1 Member: NumOfRxBuffers

Table 59 NumOfRxBuffers

Name	NumOfRxBuffers	
Type	uint16	
Description	Number of receive buffers configured within an ingress FIFO of the Ethernet controller	
Verification method	The structure member is generated with the value configured in EthCtrlConfigIngressFifoBufTotal parameter within the EthCtrlConfigIngressFifo container.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigIngressFifoBufTotal = 4 within the corresponding EthCtrlConfigIngressFifo container.	(uint16)4U,
	Set EthCtrlConfigIngressFifoBufTotal = 10 within the corresponding EthCtrlConfigIngressFifo container.	(uint16)10U,

1.2.5.2 Member: RxBufferSize

Table 60 RxBufferSize

Name	RxBufferSize	
Type	uint16	
Description	Configured length of each buffer in bytes within the ingress FIFO.	
Verification method	The structure member is generated with the value configured in EthCtrlConfigIngressFifoBufLenByte parameter within the EthCtrlConfigIngressFifo container.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigIngressFifoBufLenByte = 250 within the corresponding EthCtrlConfigIngressFifo container.	(uint16)250U,
	Set EthCtrlConfigIngressFifoBufLenByte = 1522 within the corresponding EthCtrlConfigIngressFifo container.	(uint16)1522U,

1.2.5.3 Member: RxBufferAlignSize

Table 61 RxBufferAlignSize

Name	RxBufferAlignSize
Type	uint16

Description	The length of each buffer in bytes aligned to 8-byte boundary within the ingress FIFO.	
Verification method	The structure member is generated with the 8-byte aligned value configured in EthCtrlConfigIngressFifoBufLenByte parameter within the EthCtrlConfigIngressFifo container.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigIngressFifoBufLenByte = 60 within the corresponding EthCtrlConfigIngressFifo container.	(uint16)64U,
	Set EthCtrlConfigIngressFifoBufLenByte = 1522 within the corresponding EthCtrlConfigIngressFifo container.	(uint16)1528U,

1.2.5.4 Member: RxFifoIdx

Table 62 RxFifoIdx

Name	RxFifoIdx	
Type	uint8	
Description	Index of the configured ingress FIFO	
Verification method	The structure member is generated with the value configured in EthCtrlConfigIngressFifoIdx parameter within the EthCtrlConfigIngressFifo container.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlConfigIngressFifoIdx = 0 within the corresponding EthCtrlConfigIngressFifo container.	(uint8)0U,
	Set EthCtrlConfigIngressFifoIdx = 3 within the corresponding EthCtrlConfigIngressFifo container.	(uint8)3U,

1.2.5.5 Member: RxFifoPriority

Table 63 RxFifoPriority

Name	RxFifoPriority	
Type	uint8	
Description	The priorities configured for the ingress FIFO; the set bits within the value indicate the configured priorities.	
Verification method	The structure member is generated with a value that has the bits set for the corresponding positions of priority values configured under EthCtrlConfigIngressFifoPriorityAssignment container within the EthCtrlConfigIngressFifo container.	
Example(s)	**Action**	**Generated output**
	Configure one ingress FIFO with EthCtrlConfigIngressFifoPriorityAssignment container having all the possible priorties 0	(uint8)0xffU,

	to 7 within the corresponding EthCtrlConfigIngressFifo container.	
	Configure four ingress FIFOs with EthCtrlConfigIngressFifoPriorityAssignment container configured with the below mentioned values within the corresponding EthCtrlConfigIngressFifo container. First FIFO: Priority values 0-2 Second FIFO: Priority values 3-4 Third FIFO: Priority value 5 Fourth FIFO: Priority values 6-7 Check the generated output for the element RxFifoPriority of the individual ingress FIFOs.	{First FIFO:} (uint8)0x07U, {Second FIFO:} (uint8)0x18U, {Third FIFO:} (uint8)0x20U, {Fourth FIFO:} (uint8)0xC0U

1.2.5.6 Member: RxQueueSize

Table 64 RxQueueSize

Name	RxQueueSize	
Type	uint8	
Description	Indicates the size of the allocated Receive queues in blocks of 256 bytes, where the value 0 indicates size of 256 bytes. The allocated queue size = (RxQueueSize + 1) * 256 bytes.	
Verification method	The structure member is generated based on the values configured in EthCtrlConfigIngressFifoBufLenByte and EthCtrlConfigIngressFifoBufTotal parameters. The MTL Rx FIFO size of 8 kB is allocated among the Rx queues based on the configured ingress FIFO buffer length (EthCtrlConfigIngressFifoBufLenByte) and also on the ingress FIFO buffer count (EthCtrlConfigIngressFifoBufTotal), and the computed queue size (multiple of 256) is divided by 256 and then reduced by 1 to get RxQueueSize value.	
Example(s)	**Action**	**Generated output**
	Configure 4 ingress FIFOs and set EthCtrlConfigIngressFifoBufTotal = 4 and EthCtrlConfigIngressFifoBufLenByte = 1522 for all ingress FIFOs. Check the generated output for the element RxQueueSize of all ingress FIFOs. MTL Rx FIFO size of 8 kB is equally distributed amoung the 4 ingress FIFOs, that is, 2 Kb for each FIFO. RxQueueSize = (2048/ 256) – 1 = 7	(uint8)7U,
	Configure 4 ingress FIFOs and set EthCtrlConfigIngressFifoBufTotal = 2 and EthCtrlConfigIngressFifoBufLenByte =	{First FIFO:} (uint8)11U, {Second FIFO:} (uint8)11U,

	1522 for the first ingress FIFO, EthCtrlConfigIngressFifoBufTotal = 6 and EthCtrlConfigIngressFifoBufLenByte = 1000 for the second ingress FIFO, EthCtrlConfigIngressFifoBufTotal = 10 and EthCtrlConfigIngressFifoBufLenByte = 250 for the third ingress FIFO, and EthCtrlConfigIngressFifoBufTotal = 10 and EthCtrlConfigIngressFifoBufLenByte = 128 for the fourth ingress FIFO. Check the generated output for the element RxQueueSize of the individual ingress FIFOs. The allocation is done such that atleast one complete frame of EthCtrlConfigIngressFifoBufLenByte size can be received within the corresponding queue mapped to the ingress FIFO, and the remaining space is distributed between them based on configured parameters. MTL Rx FIFO size of 8 kB is distributed among the 4 ingress FIFOs as below: First FIFO: RxQueueSize = (3072/ 256) – 1 = 11 Second FIFO: RxQueueSize = (3072/ 256) – 1 = 11 Third FIFO: RxQueueSize = (1024/ 256) – 1 = 3 Fourth FIFO: TxQueueSize = (1024/ 256) – 1 = 3	{Third FIFO:} (uint8)3U, {Fourth FIFO:} (uint8)3U,

1.2.6 Array: Eth_17_GEthMacV2_RxFifoChnlMapCtrl<CtrlIdx>[_Variant][Ingress FIFO count]

Table 65 Eth_17_GEthMacV2_RxFifoChnlMapCtrl<CtrlIdx>[_Variant][Ingress FIFO count]

Name	Eth_17_GEthMacV2_RxFifoChnlMapCtrl<CtrlIdx>[_Variant][Ingress FIFO count]
Type	uint8
Description	Array to store mapping between the configured ingress FIFO indices to Rx channels
Verification method	Eth_17_GEthMacV2_RxFifoChnlMapCtrl[<x>][<y>], where <x> = Index of controller configured and <y> is the number of ingress FIFOs configured within the controller<x>. The ingress FIFO indices are mapped to the Rx channels based on their priorities that are assigned within the EthCtrlConfigIngressFifoPriorityAssignment container. The ingress FIFO with the highest priority value configured in EthCtrlConfigIngressFifoPriorityAssignment is considered as the FIFO with the highest priority and this FIFO is assigned to Rx channel 3. The FIFO with the next higher priority value is assigned to Rx channel 2 and so on.

	<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 4 ingress FIFOs within controller 0 and set the priority values within EthCtrlConfigIngressFifoPriorityAssignment as below:
	EthCtrlConfigIngressFifoIdx 0 – Priority values 0, 1
	EthCtrlConfigIngressFifoIdx 1 – Priority values 2, 3
	EthCtrlConfigIngressFifoIdx 2 – Priority values 4, 5
	EthCtrlConfigIngressFifoIdx 3 – Priority values 6, 7
		static const uint8 Eth_17_GEthMacV2_RxFifoChnlMapCtrl0[4] =
		{
		0U,
		1U,
		2U,
		3U
		};

	Configure 4 ingress FIFOs within controller 0 and set the priority values within EthCtrlConfigIngressFifoPriorityAssignment as below:
	EthCtrlConfigIngressFifoIdx 0 – Priority values 1, 6
	EthCtrlConfigIngressFifoIdx 1 – Priority values 0, 2
	EthCtrlConfigIngressFifoIdx 2 – Priority values 4, 7
	EthCtrlConfigIngressFifoIdx 3 – Priority values 3, 5
	(variant-aware, variant name is 'Petrol').
		static const uint8 Eth_17_GEthMacV2_RxFifoChnlMapCtrl1_Petrol[4] =
		{
		2U,
		0U,
		3U,
		1U
		};

1.2.7 Array: Eth_17_GEthMacV2_RxChnlFifoMapCtrl<CtrlIdx>[_Variant][Max supported DMA channels]

Table 66 Eth_17_GEthMacV2_RxChnlFifoMapCtrl<CtrlIdx>[_Variant][Max supported DMA channels]

Name	Eth_17_GEthMacV2_RxChnlFifoMapCtrl<CtrlIdx>[_Variant][Max supported DMA channels]	
Type	uint8	
Description	Array to store mapping between the Rx channels to the configured ingress FIFO indices	
Verification method	Eth_17_GEthMacV2_RxChnlFifoMapCtrl[<x>][<y>], where <x> = Index of controller configured and <y> is the maximum supported Rx DMA channels by the controller<x>. The values are generated by performing the reverse mapping of the array Eth_17_GEthMacV2_RxFifoChnlMapCtrl<CtrlIdx>[_Variant][Ingress FIFO count] and if the Rx channel is not used, the generated value is 255. <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 4 ingress FIFOs within controller 0 and set the priority values within EthCtrlConfigIngressFifoPriorityAssignment as below:
EthCtrlConfigIngressFifoIdx 0 – Priority values 1, 3
EthCtrlConfigIngressFifoIdx 1 – Priority values 0, 2
EthCtrlConfigIngressFifoIdx 2 – Priority values 4, 7
EthCtrlConfigIngressFifoIdx 3 – Priority values 5, 6
	static const uint8 Eth_17_GEthMacV2_RxFifoChnlMapCtrl0[4] =
	{
	1U,
	0U,
	3U,
	2U
	};

Configure 3 ingress FIFOs within controller 0 and set the priority values within EthCtrlConfigIngressFifoPriorityAssignment as below:
EthCtrlConfigEgressFifoIdx 0 – Priority values 1, 3, 4
EthCtrlConfigEgressFifoIdx 1 – Priority values 5, 7
EthCtrlConfigEgressFifoIdx 2 – Priority values 0, 2, 6
(variant-aware, variant name is 'Petrol').
	static const uint8 Eth_17_GEthMacV2_RxFifoChnlMapCtrl1_Petrol[4] =
	{
	0U,
	2U,
	1U,
	255U
	};

1.2.8 Structure: Eth_17_GEthMacV2_CoreCtrlCfgCore<Core Id>[_Variant][Max controllers configured to the core]

Table 67 Eth_17_GEthMacV2_CoreCtrlCfgCore<Core Id>[_Variant][Max controllers configured]

Name	Eth_17_GEthMacV2_CoreCtrlCfgCore<Core Id>[_Variant][Max controllers configured to the core]	
Type	Eth_17_GEthMacV2_CoreCtrlConfigType	
Description	Array of structures to store controller configuration data for a core.	
Verification method	The generated structure is present in Eth_17_GEthMacV2[_<variant>]_PBcfg.c 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 all the parameters available for the Ethernet controller 0(variant-unaware) and allocate the controller 0 to Core0.	static const Eth_17_GEthMacV2_CoreCtrlConfigType Eth_17_GEthMacV2_CoreCtrlCfgCore0[1]= { { /*Specifies the Tx[0:3]/Rx[4:7] clock delay in RGMII mode for transmit skew timing*/ (uint32)0,

	/* Element to store GETH_GPCTL register value for current controller */ (uint32)2, /* Recommended time(in ns) to wait for back to back register write */ (uint32)60, /* Address of Egress fifo Configuration */ Eth_17_GEthMacV2_TxFifoCfgCtrl0, /* Address of Egress Queue Mapping */ Eth_17_GEthMacV2_TxFifoChnlMapCtrl0, /* Channel to Egress FIFO Map */ Eth_17_GEthMacV2_TxChnlFifoMapCtrl0, /* Address of fifo Priority Mapping */ Eth_17_GEthMacV2_TxPrioFifoMapCtrl0, /* Address of Ingress fifo Configuration */ Eth_17_GEthMacV2_RxFifoCfgCtrl0, /* Address of Ingress Queue Mapping */ Eth_17_GEthMacV2_RxFifoChnlMapCtrl0, /* Channel to Ingress FIFO Map */ Eth_17_GEthMacV2_RxChnlFifoMapCtrl0, /* Properties of Ethernet Controller Bit[0] - Port Select(PS) 0 for 1000Mbps 1 for 10 or 100 Mbps Bit[1] - Speed(FES) 0 for 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0 1 for 100 Mbps when PS bit is 1 Bit[2:4] - PhyInterface (000-MII, 100-RMII,001-RGMII) Bit[5] - Mode of the Controller [0 - HALFDUPLEX, 1-FULLDUPLEX] Bit[6] - Tx Interrupt Enable/Disable [0 - Disabled, 1- Enabled] Bit[7] - Rx Interrupt Enable/Disable [0 - Disabled, 1- Enabled] Bit[8] - CRC Stripping Enable/Disable [0 - Disabled, 1-Enabled] */ (uint16)228, /*DEM Id for Ethernet controller hardware test failure*/ DemConf_DemEventParameter_ETH_E_ACCESS,

	/*DEM Id for Ethernet controller Frames Lost Error*/ DemConf_DemEventParameter_ETH_E_RX_FRAMESLOST, /*DEM Id for Ethernet controller Frames Alignment Error*/ DemConf_DemEventParameter_ETH_E_ALIGNMENT, /*DEM Id for Ethernet controller Frames CRC Error*/ DemConf_DemEventParameter_ETH_E_CRC, /*DEM Id for Ethernet controller Undersize frame Error*/ DemConf_DemEventParameter_ETH_E_UNDERSIZEFRAME, /*DEM Id for Ethernet controller Oversize frame Error*/ DemConf_DemEventParameter_ETH_E_OVERSIZEFRAME, /*DEM Id for Ethernet controller Single collision Error*/ DemConf_DemEventParameter_ETH_E_SINGLECOLLISION, /*DEM Id for Ethernet controller Multiple collision Error*/ DemConf_DemEventParameter_ETH_E_MULTIPLECOLLISION, /*DEM Id for Ethernet controller Late collision Error*/ DemConf_DemEventParameter_ETH_E_LATECOLLISION, /* MAC address of the controller in network byte order */ { (uint8)0x00U, (uint8)0x03U, (uint8)0x19U, (uint8)0x00U, (uint8)0x00U, (uint8)0x01U }, /* Eth Controller Index */ (uint8)0U, /* Total Egress Queue */ (uint8)4U, /* DMA transmit arbitration algorithm */ (uint8)0U, /* MTL transmit scheduling algorithm */ (uint8)3U, /* Total Ingress Queue */ (uint8)4U, /* Queue where the untagged Rx frames are routed */

	(uint8)0U, /* Clock configuration for MDC */ (uint8)0x0U }, };
Configure all the parameters available for the Ethernet controller 0 and allocate the controller 0 to Core0 (variant-aware, variant name is 'Petrol').	static const Eth_17_GEthMacV2_CoreCtrlConfigType Eth_17_GEthMacV2_CoreCtrlCfgCore0_Petrol[1]= { { /*Specifies the Tx[0:3]/Rx[4:7] clock delay in RGMII mode for transmit skew timing*/ (uint32)0, /* Element to store GETH_GPCTL register value for current controller */ (uint32)2, /* Recommended time(in ns) to wait for back to back register write */ (uint32)60, /* Address of Egress fifo Configuration */ Eth_17_GEthMacV2_TxFifoCfgCtrl0_Petrol, /* Address of Egress Queue Mapping */ Eth_17_GEthMacV2_TxFifoChnlMapCtrl0_Petrol, /* Channel to Egress FIFO Map */ Eth_17_GEthMacV2_TxChnlFifoMapCtrl0_Petrol, /* Address of fifo Priority Mapping */ Eth_17_GEthMacV2_TxPrioFifoMapCtrl0_Petrol, /* Address of Ingress fifo Configuration */ Eth_17_GEthMacV2_RxFifoCfgCtrl0_Petrol, /* Address of Ingress Queue Mapping */ Eth_17_GEthMacV2_RxFifoChnlMapCtrl0_Petrol, /* Channel to Ingress FIFO Map */ Eth_17_GEthMacV2_RxChnlFifoMapCtrl0_Petrol, /* Properties of Ethernet Controller Bit[0] - Port Select(PS) 0 for 1000Mbps 1 for 10 or 100 Mbps Bit[1] - Speed(FES) 0 for 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0

		1 for 100 Mbps when PS bit is 1 Bit[2:4] - PhyInterface (000-MII, 100-RMII,001-RGMII) Bit[5] - Mode of the Controller [0 - HALFDUPLEX, 1-FULLDUPLEX] Bit[6] - Tx Interrupt Enable/Disable [0 - Disabled, 1- Enabled] Bit[7] - Rx Interrupt Enable/Disable [0 - Disabled, 1- Enabled] Bit[8] - CRC Stripping Enable/Disable [0 - Disabled, 1-Enabled] */ (uint16)228, /*DEM Id for Ethernet controller hardware test failure*/ DemConf_DemEventParameter_ETH_E_ACCESS, /*DEM Id for Ethernet controller Frames Lost Error*/ DemConf_DemEventParameter_ETH_E_RX_FRAMESLOST, /*DEM Id for Ethernet controller Frames Alignment Error*/ DemConf_DemEventParameter_ETH_E_ALIGNMENT, /*DEM Id for Ethernet controller Frames CRC Error*/ DemConf_DemEventParameter_ETH_E_CRC, /*DEM Id for Ethernet controller Undersize frame Error*/ DemConf_DemEventParameter_ETH_E_UNDERSIZEFRAME, /*DEM Id for Ethernet controller Oversize frame Error*/ DemConf_DemEventParameter_ETH_E_OVERSIZEFRAME, /*DEM Id for Ethernet controller Single collision Error*/ DemConf_DemEventParameter_ETH_E_SINGLECOLLISION, /*DEM Id for Ethernet controller Multiple collision Error*/ DemConf_DemEventParameter_ETH_E_MULTIPLECOLLISION, /*DEM Id for Ethernet controller Late collision Error*/ DemConf_DemEventParameter_ETH_E_LATECOLLISION, /* MAC address of the controller in network byte order */ { (uint8)0x00U, (uint8)0x03U, (uint8)0x19U, (uint8)0x00U, (uint8)0x00U, (uint8)0x01U },

		/* Eth Controller Index */ (uint8)0U, /* Total Egress Queue */ (uint8)4U, /* DMA transmit arbitration algorithm */ (uint8)0U, /* MTL transmit scheduling algorithm */ (uint8)3U, /* Total Ingress Queue */ (uint8)4U, /* Queue where the untagged Rx frames are routed */ (uint8)0U, /* Clock configuration for MDC */ (uint8)0x0U }, };

1.2.8.1 Member: EthSkewDelay

Table 68 EthSkewDelay

Name	EthSkewDelay	
Type	uint32	
Description	Tx/Rx clock delay in RGMII mode for skew timing	
Verification method	The structure member is generated from the configuration parameter EthSkewTxClockDelay and EthSkewRxClockDelay. The bits [0:3] correspond to the value of EthSkewTxClockDelay and bits [4:7] correspond to the value of EthSkewRxClockDelay. *Note: The EthSkewTxClockDelay and EthSkewRxClockDelay skew delay is allowed to be configured only in RGMII mode.*	
Example(s)	**Action**	**Generated output**
	Set EthSkewTxClockDelay to 8 And EthSkewRxClockDelay to 10	(uint32)168
	Set EthSkewTxClockDelay to 1 And EthSkewRxClockDelay to 0	(uint32)1

1.2.8.2 Member: EthGptclRegVal

Table 69 EthGptclRegVal

Name	EthGptclRegVal	
Type	uint32	
Description	Element to store general purpose control register value for the controller.	
Verification method	The structure member is generated from the values configured in EthMdioAlternateInput, EthRxClkInput, EthRxErrMIIInput, EthCarrierSenseMIIInput, EthRecDataValidMIIInput, EthTxClockMIIInput, EthCollisionMII, EthRefClkRMIIInput, EthCRSDVRMIIInput, EthReceiveData0Input, EthReceiveData1Input, EthReceiveData2Input and EthReceiveData3Input configuration parameters where port selection is done. *Note: The configuration parameters will be available for selection based on the selected mode (MII/RMII and RGMI).*	
Example(s)	**Action**	**Generated output**
	Set EthPhyInterface = RGMII EthMdioAlternateInput = ALT3_SELECT_P21_3 EthRxclkInput = ALT0_SELECT_P11_12 EthReceiveData0Input = ALT3_SELECT_NONE EthReceiveData1Input = ALT3_SELECT_NONE EthReceiveData2Input = ALT3_SELECT_NONE EthReceiveData3Input = ALT3_SELECT_NONE	(uint32)3
	Set EthPhyInterface = RMII EthMdioAlternateInput = ALT3_SELECT_P21_3 EthRefClkRMIIInput = ALT3_SELECT_NONE EthCRSDVRMIIInput = ALT3_SELECT_NONE EthReceiveData0Input = ALT0_SELECT_P11_10 EthReceiveData1Input = ALT0_SELECT_P11_9	(uint32)15

1.2.8.3 Member: EthBkRegWrWaitTime

Table 70 EthSkewDelay

Name	EthBkRegWrWaitTime
Type	uint32

Description	Recommended time in nanoseconds to wait between back to back write operations to the same register	
Verification method	The structure member is generated as 6 times the time period value in nanoseconds of the frequency value configured by the user in EthPeripheralBusClock configuration parameter which in turn refers McuSPBFrequency configuration parameter in the MCU module, when the configured speed (in the EthCtrlMacLayerSpeed parameter) is 1 Gbps. For 10 Mbps speed, the structure member is generated with the value 2400, and for 100 Mbps speed, the generated vaue is 240.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlMacLayerSpeed to ETH_MAC_LAYER_SPEED_1G. Configure EthPeripheralBus = McuSPBFrequency = 100000000 EthBkRegWrWaitTime = 6 * EthSpbPeriodInNanoSeconds = 6 * (1000000000/ EthPeripheralBusFrequency) =60	(uint32)60
	Set EthCtrlMacLayerSpeed to ETH_MAC_LAYER_SPEED_10M.	(uint32)2400
	Set EthCtrlMacLayerSpeed to ETH_MAC_LAYER_SPEED_100M.	(uint32)240

1.2.8.4 Member: EthTxFifoCfgPtr

Table 71 EthTxFifoCfgPtr

Name	EthTxFifoCfgPtr	
Type	Eth_17_GEthMacV2_TxFifoCfgType *	
Description	Pointer to the configuration of egress FIFO related parameters	
Verification method	The structure member is generated with the address that stores the egress FIFO configuration.	
Example(s)	**Action**	**Generated output**
	Configure at least one egress FIFO with valid parameters for the Ethernet controller 0.	Eth_17_GEthMacV2_TxFifoCfgCtrl0,
	Configure at least one egress FIFO with valid parameters for the Ethernet controller 1 (variant-aware, variant name is ‘Petrol’).	Eth_17_GEthMacV2_TxFifoCfgCtrl1_Petrol,
	Do not configure any egress FIFO for the Ethernet controller 0.	NULL_PTR,

1.2.8.5 Member: EthTxFifoChnlMapPtr

Table 72 EthTxFifoChnlMapPtr

Name	EthTxFifoChnlMapPtr	
Type	uint8 *	
Description	Pointer to the array that maps the configured egress FIFO indices to Tx channels	
Verification method	The structure member is generated with the address (array name) that maps the configured egress FIFO indices to Tx channels.	
Example(s)	**Action**	**Generated output**
	Configure at least one egress FIFO with valid parameters for the Ethernet controller 0.	Eth_17_GEthMacV2_TxFifoChnlMapCtrl0,
	Configure at least one egress FIFO with valid parameters for the Ethernet controller 1 (variant-aware, variant name is 'Petrol').	Eth_17_GEthMacV2_TxFifoChnlMapCtrl1_Petrol,

1.2.8.6 Member: EthTxChnlFifoMapPtr

Table 73 EthTxChnlFifoMapPtr

Name	EthTxChnlFifoMapPtr	
Type	uint8 *	
Description	Pointer to the array that maps the Tx channels to the configured egress FIFO indices	
Verification method	The structure member is generated with the address (array name) that maps the Tx channels to the configured egress FIFO indices.	
Example(s)	**Action**	**Generated output**
	Configure at least one egress FIFO with valid parameters for the Ethernet controller 0.	Eth_17_GEthMacV2_TxChnlFifoMapCtrl0,
	Configure at least one egress FIFO with valid parameters for the Ethernet controller 1 (variant-aware, variant name is 'Petrol').	Eth_17_GEthMacV2_TxChnlFifoMapCtrl1_Petrol,

1.2.8.7 Member: EthTxPrioFifoMapPtr

Table 74 EthTxPrioFifoMapPtr

Name	EthTxPrioFifoMapPtr	
Type	uint8 *	
Description	Pointer to the array that maps the configured priorities to the egress FIFO indices	
Verification method	The structure member is generated with the address (array name) that maps the configured priorities to the egress FIFO indices.	
Example(s)	**Action**	**Generated output**

	Configure at least one egress FIFO with valid parameters for the Ethernet controller 0.	Eth_17_GEthMacV2_TxPrioFifoMapCtrl0,
	Configure at least one egress FIFO with valid parameters for the Ethernet controller 1 (variant-aware, variant name is 'Petrol').	Eth_17_GEthMacV2_TxPrioFifoMapCtrl1_Petrol,

1.2.8.8 Member: EthRxFifoCfgPtr

Table 75 EthRxFifoCfgPtr

Name	EthRxFifoCfgPtr	
Type	Eth_17_GEthMacV2_RxFifoCfgType *	
Description	Pointer to the configuration of ingress FIFO related parameters	
Verification method	The structure member is generated with the address that stores the ingress FIFO configuration.	
Example(s)	**Action**	**Generated output**
	Configure at least one ingress FIFO with valid parameters for the Ethernet controller 0.	Eth_17_GEthMacV2_RxFifoCfgCtrl0,
	Configure at least one ingress FIFO with valid parameters for the Ethernet controller 1 (variant-aware, variant name is 'Petrol').	Eth_17_GEthMacV2_RxFifoCfgCtrl1_Petrol,
	Do not configure any ingress FIFO for the Ethernet controller 0.	NULL_PTR,

1.2.8.9 Member: EthRxFifoChnlMapPtr

Table 76 EthRxFifoChnlMapPtr

Name	EthRxFifoChnlMapPtr	
Type	uint8 *	
Description	Pointer to the array that maps the configured ingress FIFO indices to Rx channels	
Verification method	The structure member is generated with the address (array name) that maps the configured ingress FIFO indices to Rx channels.	
Example(s)	**Action**	**Generated output**
	Configure at least one ingress FIFO with valid parameters for the Ethernet controller 0.	Eth_17_GEthMacV2_RxFifoChnlMapCtrl0,
	Configure at least one ingress FIFO with valid parameters for the Ethernet controller 1 (variant-aware, variant name is 'Petrol').	Eth_17_GEthMacV2_RxFifoChnlMapCtrl1_Petrol,

1.2.8.10 Member: EthRxChnlFifoMapPtr

Table 77 EthRxChnlFifoMapPtr

Name	EthRxChnlFifoMapPtr	
Type	uint8 *	
Description	Pointer to the array that maps the Rx channels to the configured ingress FIFO indices	
Verification method	The structure member is generated with the address (array name) that maps the Rx channels to the configured ingress FIFO indices.	
Example(s)	**Action**	**Generated output**
	Configure at least one ingress FIFO with valid parameters for the Ethernet controller 0.	Eth_17_GEthMacV2_RxChnlFifoMapCtrl0,
	Configure at least one ingress FIFO with valid parameters for the Ethernet controller 1 (variant-aware, variant name is ‘Petrol’).	Eth_17_GEthMacV2_RxChnlFifoMapCtrl1_Petrol,

1.2.8.11 Member: EthCtrlProperties

Table 78 EthCtrlProperties

Name	EthCtrlProperties	
Type	uint16	
Description	Properties of Ethernet Controller.	
Verification method	The structure member is generated based on the values configured in EthCtrlMacLayerSpeed, EthCtrlMacLayerType, EthCtrlMacLayerSubType, EthOpMode, EthCtrlEnableTxInterrupt, EthCtrlEnableRxInterrupt and EthCtrlEnableCrcStripping.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlMacLayerSpeed = ETH_MAC_LAYER_SPEED_100M EthCtrlMacLayerType = ETH_MAC_LAYER_TYPE_XMII EthCtrlMacLayerSubType = REDUCED EthOpMode = FULLDUPLEX EthCtrlEnableRxInterrupt = True EthCtrlEnableTxInterrupt =True EthCtrlEnableCrcStripping = True	(uint16)499,
	Set Set EthCtrlMacLayerSpeed = ETH_MAC_LAYER_SPEED_1G EthCtrlMacLayerType = ETH_MAC_LAYER_TYPE_XGMII	(uint16)36,

	EthCtrlMacLayerSubType = REDUCED EthOpMode = FULLDUPLEX EthCtrlEnableRxInterrupt = False EthCtrlEnableTxInterrupt = False EthCtrlEnableCrcStripping = False	

1.2.8.12 Member: EthDemAccess

Table 79 EthDemAccess

Name	EthDemAccess	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_ACCESS Failure	
Verification method	DEM Id is generated for ETH_E_ ACCESS as DemConf_DemEventParameter_<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter container in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs container and configure ETH_E_ACCESS.	DemConf_DemEventParameter_ETH_E_ACCESS
	Do not configure ETH_E_ACCESS within the EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.13 Member: EthDemFramesLost

Table 80 EthDemFramesLost

Name	EthDemFramesLost	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_RX_FRAMES_LOST Failure.	
Verification method	DEM Id is generated for ETH_E_RX_FRAMES_LOST as DemConf_DemEventParameter _<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter container in DEM module.*	
Example(s)	**Action**	**Generated output**

	Add EthDemEventParameterRef container and configure ETH_E_RX_FRAMES_LOST.	DemConf_DemEventParameter_ETH_E_RX_FRAMES_LOST
	Do not configure ETH_E_RX_FRAMES_LOST within the EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.14 Member: EthDemAlignment

Table 81 EthDemAlignment

Name	EthDemAlignment	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_ALIGNMENT Failure	
Verification method	DEM Id is generated for ETH_E_ALIGNMENT as DemConf_DemEventParameter_<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <containser name> is the DemEventParameter container in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs container and configure ETH_E_ALIGNMENT.	DemConf_DemEventParameter_ETH_E_ALIGNMENT
	Do not configure ETH_E_ALIGNMENT within EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.15 Member: EthDemCRC

Table 82 EthDemCRC

Name	EthDemCRC	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_CRC Failure	
Verification method	DEM Id is generated for ETH_E_CRC as DemConf_DemEventParameter_<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter container in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs	DemConf_DemEventParameter_ETH_E_CRC

	container and configure ETH_E_CRC.	
	Do not configure ETH_E_CRC within EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.16 Member: EthDemUndersize

Table 83 EthDemUndersize

Name	EthDemUndersize	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_UNDERSIZEFRAME Failure	
Verification method	DEM Id is generated for ETH_E_UNDERSIZEFRAME as DemConf_DemEventParameter_<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter container in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs container and configure ETH_E_UNDERSIZEFRAME.	DemConf_DemEventParameter_ETH_E_UNDERSIZEFRAME
	Do not configure ETH_E_UNDERSIZEFRAME within EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.17 Member: EthDemOversize

Table 84 EthDemOversize

Name	EthDemOversize	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_OVERSIZEFRAME Failure	
Verification method	DEM Id is generated for ETH_E_OVERSIZEFRAME as DemConf_DemEventParameter _<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter contained in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs container and configure ETH_E_OVERSIZEFRAME.	DemConf_DemEventParameter_ETH_E_OVERSIZEFRAME

	Do not configure ETH_E_OVERSIZEFRAME within EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.18 Member: EthDemSingleCollision

Table 85 EthDemSingleCollision

Name	EthDemSingleCollision	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_SINGLECOLLISION Failure	
Verification method	DEM Id is generated for ETH_E_SINGLECOLLISION as DemConf_DemEventParameter_<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter contained in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs container and configure ETH_E_SINGLECOLLISION.	DemConf_DemEventParameter_ETH_E_SINGLECOLLISION
	Do not configure ETH_E_SINGLECOLLISION within EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.19 Member: EthDemMultipleCollision

Table 86 EthDemMultipleCollision

Name	EthDemMultipleCollision	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_MULTIPLECOLLISION Failure	
Verification method	DEM Id is generated for ETH_E_MULTIPLECOLLISION as DemConf_DemEventParameter_<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter contained in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs container and configure ETH_E_MULTIPLECOLLISION.	DemConf_DemEventParameter_ETH_E_MULTIPLECOLLISION

	Do not configure ETH_E_MULTIPLECOLLISION within EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.20 Member: EthDemLateCollision

Table 87 EthDemLateCollision

Name	EthDemLateCollision	
Type	Dem_EventIdType	
Description	DEM Id for ETH_E_LATECOLLISION Failure	
Verification method	DEM Id is generated for ETH_E_LATECOLLISION as DemConf_DemEventParameter _<container name>. If DEM is not configured, then DEM Id is generated as ETH_17_GETHMACV2_DISABLE_DEM_REPORT. *Note: <container name> is the DemEventParameter contained in DEM module.*	
Example(s)	**Action**	**Generated output**
	Add EthDemEventParameterRefs container and configure ETH_E_LATECOLLISION.	DemConf_DemEventParameter_ETH_E_LATECOLLISION
	Do not configure ETH_E_LATECOLLISION within EthDemEventParameterRefs container.	ETH_17_GETHMACV2_DISABLE_DEM_REPORT

1.2.8.21 Member: EthMacAddress [6]

Table 88 EthMacAddress [6]

Name	EthMacAddress [6]	
Type	uint8	
Description	MAC address of the Ethernet controller in Network Byte order	
Verification method	The generated structure member contains an array entry for user-configured PHY MAC address in Network Byte order as configured in the configuration parameter EthCtrlPhyAddress.	
Example(s)	**Action**	**Generated output**
	Set EthCtrlPhyAddress to 00:03:19:00:00:01	{ (uint8)0x00U, (uint8)0x03U, (uint8)0x19U, (uint8)0x00U,

		(uint8)0x00U, (uint8)0x01U },
	Set EthCtrlPhyAddress to 12:34:56:78:00:09	{ (uint8)0x12U, (uint8)0x34U, (uint8)0x56U, (uint8)0x78U, (uint8)0x00U, (uint8)0x09U },

1.2.8.22 Member: EthCtrlIdx

Table 89 EthCtrlIdx

Name	EthCtrlIdx	
Type	uint8	
Description	Index of Ethernet controller	
Verification method	The structure member is generated with the value configured in EthCtrlIdx parameter	
Example(s)	**Action**	**Generated output**
	Set EthCtrlIdx = 0	(uint8)0U,
	Set EthCtrlIdx = 1	(uint8)1U,

1.2.8.23 Member: EthNumTxChnls

Table 90 EthNumTxChnls

Name	EthNumTxChnls	
Type	uint8	
Description	Number of transmit channels internally used within the Ethernet controller	
Verification method	The structure member is generated with the number of egress FIFOs configured for transmit scheduling under the EthCtrlConfigSchedulerPredecessor configuration container.	
Example(s)	**Action**	**Generated output**
	Configure four egress FIFOs with valid parameters and add all the four egress FIFOs for transmit scheduling under EthCtrlConfigSchedulerPredecessor container.	(uint8)4U,
	Configure one egress FIFO with valid parameters and add this	(uint8)1U,

	egress FIFO for transmit scheduling under EthCtrlConfigSchedulerPredecessor container.	

1.2.8.24 Member: EthDmaTxArbAlg

Table 91 EthDmaTxArbAlg

Name	EthDmaTxArbAlg	
Type	uint8	
Description	Indicates the selected arbitration algorithm among the transmit DMA channels	
Verification method	The structure member is generated with a value that corresponds to a selected Tx DMA arbitration algorithm configured in EthCtrlConfigDMAArbitrationAlgorithm parameter. The value generated is: 0 - Fixed Priority 1 – Weighted Strict Priority 2 – Weighted Round Robin	
Example(s)	**Action**	**Generated output**
	Configure EthCtrlConfigDMAArbitrationAlgorithm to ETH_DMA_ARBITRATION_FIXED_PRIORITY within EthCtrlConfigDMAArbitration container.	(uint8)0U,
	Configure EthCtrlConfigDMAArbitrationAlgorithm to ETH_DMA_ARBITRATION_WEIGHTED_STRICT_PRIORITY within EthCtrlConfigDMAArbitration container.	(uint8)1U,
	Configure EthCtrlConfigDMAArbitrationAlgorithm to ETH_DMA_ARBITRATION_WEIGHTED_ROUND_ROBIN within EthCtrlConfigDMAArbitration container.	(uint8)2U,

1.2.8.25 Member: EthMtlTxSchAlg

Table 92 EthMtlTxSchAlg

Name	EthMtlTxSchAlg	
Type	uint8	
Description	Indicates the selected transmit scheduling algorithm at the MTL layer.	
Verification method	The structure member is generated with a value that corresponds to a selected Tx scheduler algorithm configured in EthCtrlConfigSchedulerAlgorithm parameter. The value generated is: 0 - ETH_SCHEDULER_WEIGHTED_ROUND_ROBIN 3 – ETH_SCHEDULER_STRICT_PRIORITY	
Example(s)	**Action**	**Generated output**
	Configure EthCtrlConfigSchedulerAlgorithm to ETH_SCHEDULER_WEIGHTED_ROUND_ROBIN within EthCtrlConfigScheduler container.	(uint8)0U,
	Configure EthCtrlConfigSchedulerAlgorithm to ETH_SCHEDULER_STRICT_PRIORITY within EthCtrlConfigScheduler container.	(uint8)3U,

1.2.8.26 Member: EthNumRxChnls

Table 93 EthNumRxChnls

Name	EthNumRxChnls	
Type	uint8	
Description	Maximum transmit buffer length (frame length) aligned to 8 bytes.	
Verification method	The structure member is generated with the number of ingress FIFOs configured under EthCtrlConfigIngressFifo container.	
Example(s)	**Action**	**Generated output**
	Configure four ingress FIFOs with valid parameters under EthCtrlConfigIngressFifo container.	(uint8)4U,
	Configure one ingress FIFO with valid parameters under EthCtrlConfigIngressFifo container.	(uint8)1U,

1.2.8.27 Member: EthUntagRxQueue

Table 94 EthUntagRxQueue

Name	EthUntagRxQueue	
Type	uint8	
Description	Queue index where the received untagged frames are routed	
Verification method	The structure member is generated with the value of Ingress FIFO index (EthCtrlConfigIngressFifoIdx parameter) that is referenced by the EthCtrlConfigIngressUntaggedPktsFifoRef parameter.	
Example(s)	**Action**	**Generated output**
	Configure four ingress FIFOs with valid parameters and select the ingress FIFO having EthCtrlConfigIngressFifoIdx = 3 in EthCtrlConfigIngressUntaggedPktsFifoRef parameter.	(uint8)3U,
	Configure one ingress FIFO with valid parameters and select the ingress FIFO having EthCtrlConfigIngressFifoIdx = 0 in EthCtrlConfigIngressUntaggedPktsFifoRef parameter.	(uint8)0U,

1.2.8.28 Member: EthMdcRegVal

Table 95 EthMdcRegVal

Name	EthMdcRegVal
Type	uint8
Description	CSR clock divider register field value required for the generation of MDC clock

Verification method	The structure member is generated based on the value configured in EthMDCClockFrequency parameter. To derive the CSR clock divider register field value, the frequency value configured by the user in EthPeripheralBusClock configuration parameter (which in turn refers McuSPBFrequency configuration parameter in the MCU module) is divided by the value of EthMDCClockFrequency parameter to obtain a divider value. If divider value = 4, then EthMdcRegVal = 8 If divider value > 4 and <= 6, then EthMdcRegVal = 9 If divider value > 6 and <= 8, then EthMdcRegVal = 0xA If divider value > 8 and <= 10, then EthMdcRegVal = 0xB If divider value > 10 and <= 12, then EthMdcRegVal = 0xC If divider value > 12 and <= 14, then EthMdcRegVal = 0xD If divider value > 14 and <= 16, then EthMdcRegVal = 0xE If divider value > 16 and <= 18, then EthMdcRegVal = 0xF If divider value > 18 and <= 26, then EthMdcRegVal = 3 If divider value > 26 and <= 42, then EthMdcRegVal = 0	
Example(s)	**Action**	**Generated output**
	Set EthMDCClockFrequency = 2500000. Configure EthPeripheralBus = McuSPBFrequency = 100000000. Clock divider value = McuSPBFrequency/ EthMDCClockFrequency = 40	(uint8)0x0U
	Set EthMDCClockFrequency = 12500000. Configure EthPeripheralBus = McuSPBFrequency = 100000000. Clock divider value = McuSPBFrequency/ EthMDCClockFrequency = 8	(uint8)0xAU

1.2.9 Structure: Eth_17_GEthMacV2_ConfigCore<Core Id>[_Variant]

Table 96 Eth_17_GEthMacV2_ConfigCore<Core Id>[_Variant]

Name	Eth_17_GEthMacV2_ConfigCore<Core Id>[_Variant]	
Type	Eth_17_GEthMacV2_CoreConfigType	
Description	Structure to store core specific configuration data.	
Verification method	The generated structure is present in Eth_17_GEthMacV2[_<variant>]_PBcfg.c 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 controller 0 and controller 1 to core 1.	static const Eth_17_GEthMacV2_CoreConfigType Eth_17_GEthMacV2_ConfigCore1 =
		{
		(Eth_17_GEthMacV2_CoreCtrlConfigType*)
		Eth_17_GEthMacV2_CoreCtrlCfgCore1,
		2U /* Maximum controllers allocated to core1 */
		};

	Configure controller 0 to core 0 (variant-aware, variant name is 'Petrol').	static const Eth_17_GEthMacV2_CoreConfigType Eth_17_GEthMacV2_ConfigCore0_Petrol =
		{
		(Eth_17_GEthMacV2_CoreCtrlConfigType*)
		Eth_17_GEthMacV2_CoreCtrlCfgCore0_
		Petrol,
		1U /* Maximum controllers allocated to core0 */
		};

1.2.9.1 Member: EthCoreCtrlPtr

Table 97 EthCoreCtrlPtr

Name	EthCoreCtrlPtr	
Type	Eth_17_GEthMacV2_CoreCtrlConfigType	
Description	Pointer to the configuration of controller allocated to that core.	
Verification method	The structure member is generated with address which stores controller configuration allocated to that core.	
Example(s)	**Action**	**Generated output**
	Configure controller 0 to Core 1	(Eth_17_GEthMacV2_CoreCtrlConfigType*) Eth_17_GEthMacV2_CoreCtrlCfgCore1,
	Configure controller 0 to core 0 (variant-aware, variant name is 'Petrol').	(Eth_17_GEthMacV2_CoreCtrlConfigType*) Eth_17_GEthMacV2_CoreCtrlCfgCore0_Petrol,

1.2.9.2 Member: EthMaxControllers

Table 98 EthMaxControllers

Name	EthMaxControllers	
Type	uint8	
Description	Maximum controllers allocated to the core.	
Verification method	The structure member is generated as the maximum controllers configured to the core.	
Example(s)	**Action**	**Generated output**

	Configure controller 0 and controller 1 to Core 1	2U
	Configure controller 0 to core 0.	1U

1.2.10 Structure: Eth_17_GEthMacV2_Config[_Variant]

Table 99 Eth_17_GEthMacV2_Config[_Variant]

Name	Eth_17_GEthMacV2_Config[_Variant]	
Type	Eth_17_GEthMacV2_ConfigType	
Description	Ethernet driver configuration root structure.	
Verification method	The generated structure is present in Eth_17_GEthMacV2[_<variant>]_PBcfg.c 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 controller 0 to Core0 and controller 1 to core 1.	const Eth_17_GEthMacV2_ConfigType
		Eth_17_GEthMacV2_Config =
		{
		/* starting address of Core<x>
		Configuration data */
		{
		(Eth_17_GEthMacV2_CoreConfigType*)
		&Eth_17_GEthMacV2_ConfigCore0,
		(Eth_17_GEthMacV2_CoreConfigType*)
		&Eth_17_GEthMacV2_ConfigCore1,
		NULL_PTR
		},
		/* Address of index mapping array */
		(uint8*)
		Eth_17_GEthMacV2_ControllerIndexMap
		};

	Configure controller 0 and controller 1 to core 0.(variant-aware, variant name is ‘Petrol’).	const Eth_17_GEthMacV2_ConfigType Eth_17_GEthMacV2_Config_Petrol = { /* starting address of Core<x> Configuration data */ { (Eth_17_GEthMacV2_CoreConfigType*)

		&Eth_17_GEthMacV2_ConfigCore0_ Petrol, NULL_PTR, NULL_PTR }, /* Address of index mapping array */ (uint8*) Eth_17_GEthMacV2_ControllerIndexMap _Petrol };

1.2.10.1 Member: EthCoreAdd[ETH_17_GETHMACV2_MAX_CORES]

Table 100 EthCoreAdd[ETH_17_GETHMACV2_MAX_CORES]

Name	EthCoreAdd[ETH_17_GETHMACV2_MAX_CORES]	
Type	Eth_17_GEthMacV2_CoreConfigType *	
Description	Array to store starting address of core configuration data.	
Verification method	The generated structure member is present in the Eth_17_GEthMacV2_Config[_<variant>] structure. If at least one controller <y> is allocated to one Core<x>, then the element shall be generated as '&Eth_17_GEthMacV2_ConfigCore<x>' else 'NULL_PTR' is generated.(x in range 0 to 5 and y in range 0 or 1 depending on the derivative).	
Example(s)	**Action**	**Generated output**
	Configure controller 0 to Core0 and controller 1 to core 1.	/* starting address of Core<x> Configuration data */ { (Eth_17_GEthMacV2_CoreConfigType*) &Eth_17_GEthMacV2_ConfigCore0, (Eth_17_GEthMacV2_CoreConfigType*) &Eth_17_GEthMacV2_ConfigCore1, NULL_PTR },
	Configure controller 0 and controller 1 to core 0 (variant-aware, variant name is 'Petrol').	/* starting address of Core<x> Configuration data */ { (Eth_17_GEthMacV2_CoreConfigType*) &Eth_17_GEthMacV2_ConfigCore0_Petrol, NULL_PTR, NULL_PTR },

1.2.10.2 Member: EthNodeIdxMapPtr

Table 101 EthNodeIdxMapPtr

Name	EthNodeIdxMapPtr	
Type	uint8 *	
Description	Pointer to the array index of the controller in the current core.	
Verification method	The structure member is generated as pointer to the array index of the controller.	
Example(s)	**Action**	**Generated output**
	Configure controller 0 and controller 1 to core 0.	(uint8*)Eth_17_GEthMacV2_ControllerIndexMap
	Configure controller 0 and controller 1 to core 0 (variant-aware, variant name is 'Petrol').	(uint8*) Eth_17_GEthMacV2_ControllerIndexMap_Petrol

1.2.11 Array: Eth_17_GEthMacV2_ControllerIndexMap[_Variant][Max Controllers]

Table 102 Eth_17_GEthMacV2_ControllerIndexMap[_Variant][Max Controllers]

Name	Eth_17_GEthMacV2_ControllerIndexMap[_Variant][Max Controllers]	
Type	uint8	
Description	Array to store index of the controller in the allocated core.	
Verification method	Eth_17_GEthMacV2_ControllerIndexMap [<x>] = Index of (Controller = <x>) in the allocated core. <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 controller 0 to core 0 and controller 1 to core 1.	static const uint8 Eth_17_GEthMacV2_ControllerIndexMap [2] =
		{
		0x0U,
		0x0U,
		};

	Configure controller 0 and controller 1 to core 0 (variant-aware, variant name is 'Petrol').	static const uint8 Eth_17_GEthMacV2_ControllerIndexMap_Petrol [2] = { 0x0U, 0x1U,

		};

1.3 File: Eth_17_GEthMacV2[_<variant>]_PBcfg.h

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

1.3.1 Structure: Eth_17_GEthMacV2_Config[_<variant>]

Table 103 Eth_17_GEthMacV2_Config[_<varaint>]

Name	Eth_17_GEthMacV2_Config[_<variant>]	
Type	Eth_17_GEthMacV2_ConfigType	
Description	Declaration of root configuration structure of Ethernet driver which will be used during initialization.	
Verification method	The generated structure is present in Eth_17_GEthMacV2[_<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 all the parameter available for the Ethernet controller (variant-unaware).	extern const Eth_17_GEthMacV2_ConfigType Eth_17_GEthMacV2_Config;
	Configure all the parameter available for the Ethernet controller (variant-aware, variant name is 'Petrol').	extern const Eth_17_GEthMacV2_ConfigType Eth_17_GEthMacV2_Config_Petrol;

Revision history

Major changes since the last revision

Date	Version	Description
2020-12-04	v1.0	Review comments fixed. Released version.
2020-12-02	v0.1	Initial Version

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

Edition 2020-12-04
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.