49. Controller Area Network (MCAN)

49.1 Description

The Controller Area Network (MCAN) performs communication according to ISO 11898-1:2015 and to Bosch CAN-FD specification. Additional transceiver hardware is required for connection to the physical layer.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN core to the Message RAM, as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN core, as well as providing transmit status information.

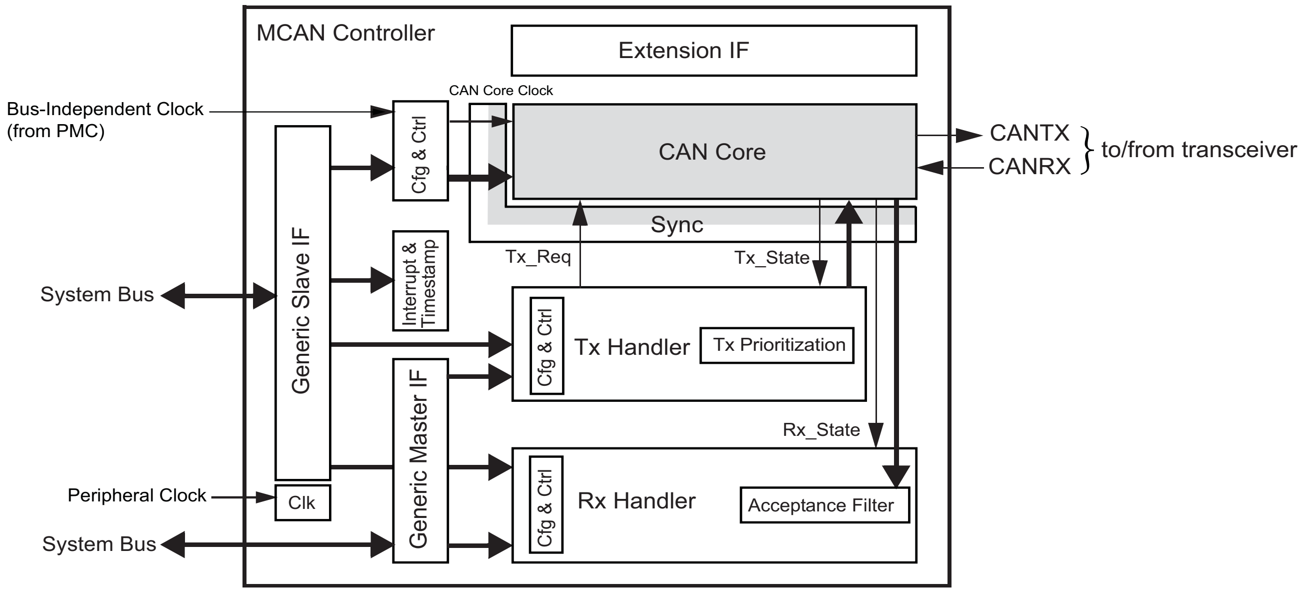
Acceptance filtering is implemented by a combination of up to 128 filter elements, where each element can be configured as a range, as a bit mask, or as a dedicated ID filter.

49.2 Embedded Characteristics

- Compliant with CAN Protocol Version 2.0 Part A, B and ISO 11898-1
- CAN FD with up to 64 Data Bytes Supported
- CAN Error Logging
- AUTOSAR Optimized
- SAE J1939 Optimized
- Improved Acceptance Filtering
- Two Configurable Receive FIFOs
- Separate Signalling on Reception of High Priority Messages
- Up to 64 Dedicated Receive Buffers
- Up to 32 Dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM Access for Processor
- Multiple MCANs May Share the Same Message RAM
- Programmable Loop-back Test Mode
- Maskable Module Interrupts
- Support for Asynchronous CAN and System Bus Clocks
- Power-down Support
- Debug on CAN Support

49.3 Block Diagram

Figure 49-1. MCAN Block Diagram



▨ CAN Clock Domain (Bus-independent Clock)
☐ Peripheral Clock Domain

Note: Refer to section "Power Management Controller (PMC)" for details about the bus-independent clock (PCK5).

49.4 Product Dependencies

49.4.1 I/O Lines

The pins used to interface to the compliant external devices can be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the CAN pins to their peripheral functions.

Table 49-1. I/O Lines

Instance	Signal	I/O Line	Peripheral
MCAN0	CANRX0	PB3	A
MCAN0	CANTX0	PB2	A
MCAN1	CANRX1	PC12	C
MCAN1	CANRX1	PD28	B
MCAN1	CANTX1	PC14	C
MCAN1	CANTX1	PD12	B

49.4.2 Power Management

The MCAN can be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the MCAN clock.

In order to achieve a stable function of the MCAN, the system bus clock must always be faster than or equal to the CAN clock.

It is recommended to use the CAN clock at frequencies of 20, 40 or 80 MHz. To achieve these frequencies, PMC PCK5 must select the UPLLCK (480 MHz) as source clock and divide by 24,12, or 6. PCK5 allows the system bus and processor clock to be modified without affecting the bit rate communication.

49.4.3 Interrupt Sources

The two MCAN interrupt lines (MCAN_INT0, MCAN_INT1) are connected on internal sources of the Interrupt Controller.

Using the MCAN interrupts requires the Interrupt Controller to be programmed first.

Interrupt sources can be routed either to MCAN_INT0 or to MCAN_INT1. By default, all interrupt sources are routed to interrupt line MCAN_INT0/1. By programming MCAN_ILE.EINT0 and MCAN_ILE.EINT1, the interrupt sources can be enabled or disabled separately.

Table 49-2. Peripheral IDs

Instance	ID
MCAN0	35
MCAN0 INT1	36
MCAN1	37
MCAN1 INT1	38

49.4.4 Address Configuration

The LSBs [bits 15:2] for each section of the CAN Message RAM are configured in the respective buffer configuration registers as detailed in Section 49.5.7 “Message RAM”.

The MSBs [bits 31:16] of the CAN Message RAM for CAN0 and CAN1 are configured in CCFG_CAN0 and CCFG_SYSIO registers.

49.4.5 Timestamping

Timestamping uses the value of CV in the TC Counter Value 0 register (TC_CV0) at address 0x4000C010. TC0 can use the programmable clocks PCK6 or PCK7 as input. Refer to the section “Timer Counter (TC)” for more details.

The selection between PCK6 and PCK7 is done in the Matrix Peripheral Clock Configuration Register (CCFG_PCCR), using the bit TC0CC. Refer to this register in the section “Bus Matrix (MATRIX)” for more details.

These clocks can be programmed in the registers PMC Programmable Clock Registers PMC_PCK6 and PMC_PCK7, respectively. Refer to these registers in the section “Power Management Controller (PMC)” for more details.

49.5 Functional Description

49.5.1 Operating Modes

49.5.1.1 Software Initialization

Software initialization is started by setting bit MCAN_CCCR.INIT, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off. While MCAN_CCCR.INIT is set, message transfer from and to the CAN bus is stopped and the status of the CAN bus output CANTX is recessive (HIGH). The counters of the Error Management Logic EML are unchanged. Setting MCAN_CCCR.INIT does not change any configuration register. Resetting MCAN_CCCR.INIT finishes the software initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (≡ Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the MCAN configuration registers is only enabled when both bits MCAN_CCCR.INIT and MCAN_CCCR.CCE are set (protected write).

MCAN_CCCR.CCE can only be configured when MCAN_CCCR.INIT = ‘1’. MCAN_CCCR.CCE is automatically cleared when MCAN_CCCR.INIT = ‘0’.

The following registers are cleared when MCAN_CCCR.CCE = ‘1’:

- High Priority Message Status (MCAN_HPMS)
- Receive FIFO 0 Status (MCAN_RXF0S)
- Receive FIFO 1 Status (MCAN_RXF1S)
- Transmit FIFO/Queue Status (MCAN_TXFQS)
- Transmit Buffer Request Pending (MCAN_TXBRP)
- Transmit Buffer Transmission Occurred (MCAN_TXBTO)
- Transmit Buffer Cancellation Finished (MCAN_TXBCF)
- Transmit Event FIFO Status (MCAN_TXEFS)

The Timeout Counter value MCAN_TOCV.TOC is loaded with the value configured by MCAN_TOCC.TOP when MCAN_CCCR.CCE = ‘1’.

In addition, the state machines of the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR.CCE = ‘1’.

The following registers are only writeable while MCAN_CCCR.CCE = ‘0’

- Transmit Buffer Add Request (MCAN_TXBAR)
- Transmit Buffer Cancellation Request (MCAN_TXBCR)

MCAN_CCCR.TEST and MCAN_CCCR.MON can only be set when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’. Both bits may be cleared at any time. MCAN_CCCR.DAR can only be configured when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’.

49.5.1.2 Normal Operation

Once the MCAN is initialized and MCAN_CCCR.INIT is cleared, the MCAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted, dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

49.5.1.3 CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the MCAN receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit MCAN_PSR.PXE. When Protocol Exception Handling is enabled (MCAN_CCCR.PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR.ACT = 2) to Integrating (MCAN_PSR.ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR.PXHD = 1), the MCAN will treat a recessive res bit as an form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR.FDOE. In case CCCR.FDOE = ‘1’, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With CCCR.FDOE = ‘0’, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. CCCR.FDOE and CCCR.BRSE can only be changed while CCCR.INIT and CCCR.CCE are both set.

With MCAN_CCCR.FDOE = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 0, only bit FDF of a Tx Buffer element is evaluated. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 1, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting according to ISO11898-1 until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD-capable. Non-CAN FD nodes are held in Silent mode until programming has completed. Then all nodes revert to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to Table 49-3 below.

Table 49-3. Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing and Prescaler register (MCAN_NBTP). In the following CAN FD data phase, the data phase CAN bit timing is used as defined by the Data Bit Timing and Prescaler register (MCAN_DBTP). The bit timing reverts back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN core clock frequency. Example: with a CAN clock frequency of 20 MHz and the shortest configurable bit time of 4 t_q, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

49.5.1.4 Transmitter Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CANTX the protocol controller receives the transmitted data from its local CAN transceiver via pin CANRX. The received data is delayed by the transmitter delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transmitter delay, the delay compensation is introduced. Without delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the delay.

Description

The MCAN protocol unit has implemented a delay compensation mechanism to compensate the delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase, the delayed transmit data is compared against the received data at the secondary sample point. If a bit error is detected, the transmitter will react to this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting bit MCAN_DBTP.TDC.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN's transmit output CANTX through the transceiver to the receive input CANRX plus the transmitter delay compensation offset as configured by MCAN_TDCR.TDCO. The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of CAN core clock periods.

MCAN_PSR.TDCV shows the actual transmitter delay compensation value. MCAN_PSR.TDCV is cleared when MCAN_CCCR.INIT is set and is updated at each transmission of an FD frame while MCAN_DBTP.TDC is set.

The following boundary conditions have to be considered for the delay compensation implemented in the MCAN:

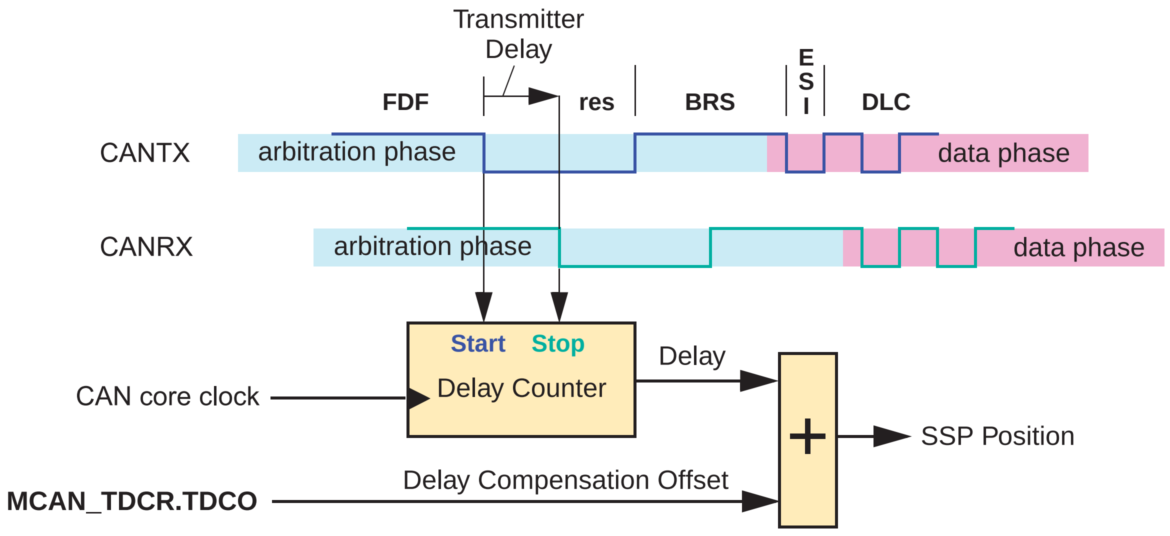
- The sum of the measured delay from CANTX to CANRX and the configured delay compensation offset MCAN_TDCR.TDCO has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANTX to CANRX and the configured delay compensation offset MCAN_TDCR.TDCO has to be less or equal 127 CAN core clock periods. In case this sum exceeds 127 CAN core clock periods, the maximum value of 127 CAN core clock periods is used for delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

Transmitter Delay Measurement

If transmitter delay compensation is enabled by programming MCAN_DBTP.TDC = '1', the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANRX of the transmitter.

The resolution of this measurement is one mtq.

Figure 49-2. Transmitter Delay Measurement



To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a to early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming MCAN_TDCR.TDCF.

This defines a minimum value for the SSP position. Dominant edges on CANRX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR.TDCF AND CANRX is low.

49.5.1.5 Restricted Operation Mode

In Restricted Operation mode, the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters are not incremented. The processor can set the MCAN into Restricted Operation mode by setting bit MCAN_CCCR.ASM. The bit can only be set by the processor when both MCAN_CCCR.CCE and MCAN_CCCR.INIT are set to ‘1’. The bit can be reset by the processor at any time.

Restricted Operation mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

The Restricted Operation mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation mode after it has received a valid frame.

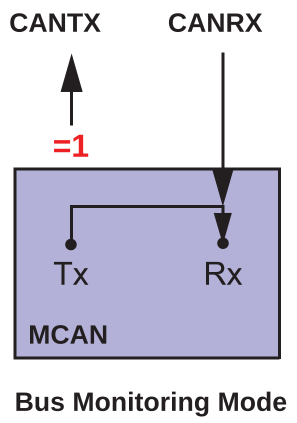
Note: The Restricted Operation Mode must not be combined with the Loop Back mode (internal or external).

49.5.1.6 Bus Monitoring Mode

The MCAN is set in Bus Monitoring mode by setting MCAN_CCCR.MON. In Bus Monitoring mode (see ISO11898-1, 10.12 Bus monitoring), the MCAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the MCAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring mode, the Tx Buffer Request Pending register (MCAN_TXBRP) is held in reset state.

The Bus Monitoring mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 49-4 shows the connection of signals CANTX and CANRX to the MCAN in Bus Monitoring mode.

Figure 49-3. Pin Control in Bus Monitoring Mode



49.5.1.7 Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the MCAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via MCAN_CCCR.DAR.

Frame Transmission in DAR Mode

In DAR mode, all transmissions are automatically cancelled after they start on the CAN bus. A Tx Buffer's Tx Request Pending bit TXBRP.TRPx is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set
  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx not set
- Successful transmission in spite of cancellation:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set
  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
  Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx not set
  Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = “10” (transmission in spite of cancellation).

49.5.1.8 Power-down (Sleep Mode)

The MCAN can be set into Power-down mode via bit MCAN_CCCR.CSR.

When all pending transmission requests have completed, the MCAN waits until bus idle state is detected. Then the MCAN sets MCAN_CCCR.INIT to prevent any further CAN transfers. Now the MCAN acknowledges that it is ready for power down by setting to one the bit MCAN_CCCR.CSA. In this state, before the clocks are switched off, further register accesses can be made. A write access to MCAN_CCCR.INIT will have no effect. Now the bus clock (peripheral clock) and the CAN core clock may be switched off.

To leave Power-down mode, the application has to turn on the MCAN clocks before clearing CC Control Register flag MCAN_CCCR.CSR. The MCAN will acknowledge this by clearing MCAN_CCCR.CSA. The application can then restart CAN communication by clearing the bit CCCR.INIT.

49.5.1.9 Test Modes

To enable write access to the MCAN Test register (MCAN_TEST) (see Section 49.6.5), bit MCAN_CCCR.TEST must be set. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CANTX by programming MCAN_TEST.TX. Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the MCAN's bit timing and it can drive constant dominant or recessive values. The actual value at pin CANRX can be read from MCAN_TEST.RX. Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between CAN clock and system bus clock domain, there may be a delay of several system bus clock periods between writing to MCAN_TEST.TX until the new configuration is visible at output pin CANTX. This applies also when reading input pin CANRX via MCAN_TEST.RX.

Note: Test modes should be used for production tests or self-test only. The software control for pin CANTX interferes with all CAN protocol functions. It is not recommended to use test modes for application.

External Loop Back Mode

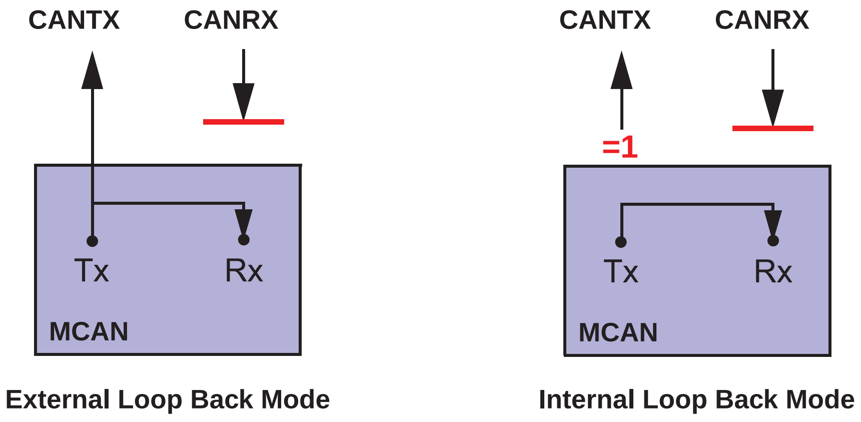
The MCAN can be set in External Loop Back mode by setting the bit MCAN_TEST.LBCK. In Loop Back mode, the MCAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. Figure 49-4 shows the connection of signals CANTX and CANRX to the MCAN in External Loop Back mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the MCAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back mode. In this mode, the MCAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CANRX input pin is disregarded by the MCAN. The transmitted messages can be monitored at the CANTX pin.

Internal Loop Back Mode

Internal Loop Back mode is entered by setting bits MCAN_TEST.LBCK and MCAN_CCCR.MON. This mode can be used for a "Hot Selftest", meaning the MCAN can be tested without affecting a running CAN system connected to the pins CANTX and CANRX. In this mode, pin CANRX is disconnected from the MCAN, and pin CANTX is held recessive. Figure 49-4 shows the connection of CANTX and CANRX to the MCAN when Internal Loop Back mode is enabled.

Figure 49-4. Pin Control in Loop Back Modes



49.5.2 Timestamp Generation

For timestamp generation the MCAN supplies a 16-bit wrap-around counter. A prescaler TSCC.TCP can be configured to clock the counter in multiples of CAN bit times (1…16). The counter is readable via MCAN_TSCV.TSC. A write access to the Timestamp Counter Value register (MCAN_TSCV) resets the counter to zero. When the timestamp counter wraps around, interrupt flag MCAN_IR.TSW is set.

On start of frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

By programming bit MCAN_TSCC.TSS an external 16-bit timestamp can be used. Refer to Section 49.4.5 "Timestamping" for more details.

49.5.3 Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO, the MCAN supplies a 16-bit Timeout Counter. It operates as down-counter and uses the same prescaler controlled by TSCC.TCP as the Timestamp Counter. The Timeout Counter is configured via the Timeout Counter Configuration register (MCAN_TOCC). The actual counter value can be read from MCAN_TOCV.TOC. The Timeout Counter can only be started while MCAN_CCCR.INIT = '0'. It is stopped when MCAN_CCCR.INIT = '1', e.g. when the MCAN enters Bus_Off state.

The operating mode is selected by MCAN_TOCC.TOS. When operating in Continuous mode, the counter starts when MCAN_CCCR.INIT is reset. A write to MCAN_TOCV presets the counter to the value configured by MCAN_TOCC.TOP and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by MCAN_TOCC.TOP. Down-counting is started when the first FIFO element is stored. Writing to MCAN_TOCV has no effect.

When the counter reaches zero, interrupt flag MCAN_IR.TOO is set. In Continuous mode, the counter is immediately restarted at MCAN_TOCC.TOP.

Note: The clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore the point in time where the Timeout Counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the bit rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

49.5.4 Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

49.5.4.1 Acceptance Filtering

The MCAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC)
- Standard ID Filter Configuration (MCAN_SIDFC)
- Extended ID Filter Configuration (MCAN_XIDFC)
- Extended ID and Mask (MCAN_XIDAM)

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag (MCAN_IR.HPM)
- Set High Priority Message interrupt flag (MCAN_IR.HPM) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the effected Rx Buffer or Rx FIFO:

- Rx Buffer
  New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC.
- Rx FIFO
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC. In case the matching Rx FIFO is operated in Overwrite mode, the boundary conditions described in Rx FIFO Overwrite Mode have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

Range Filter

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID resp. EF1ID/EF2ID.

There are two possibilities when range filtering is used together with extended frames:

EFT = “00”: The Message ID of received frames is ANDed with MCAN_XIDAM before the range filter is applied.

EFT = “11”: MCAN_XIDAM is not used for range filtering.

Filter for Specific IDs

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

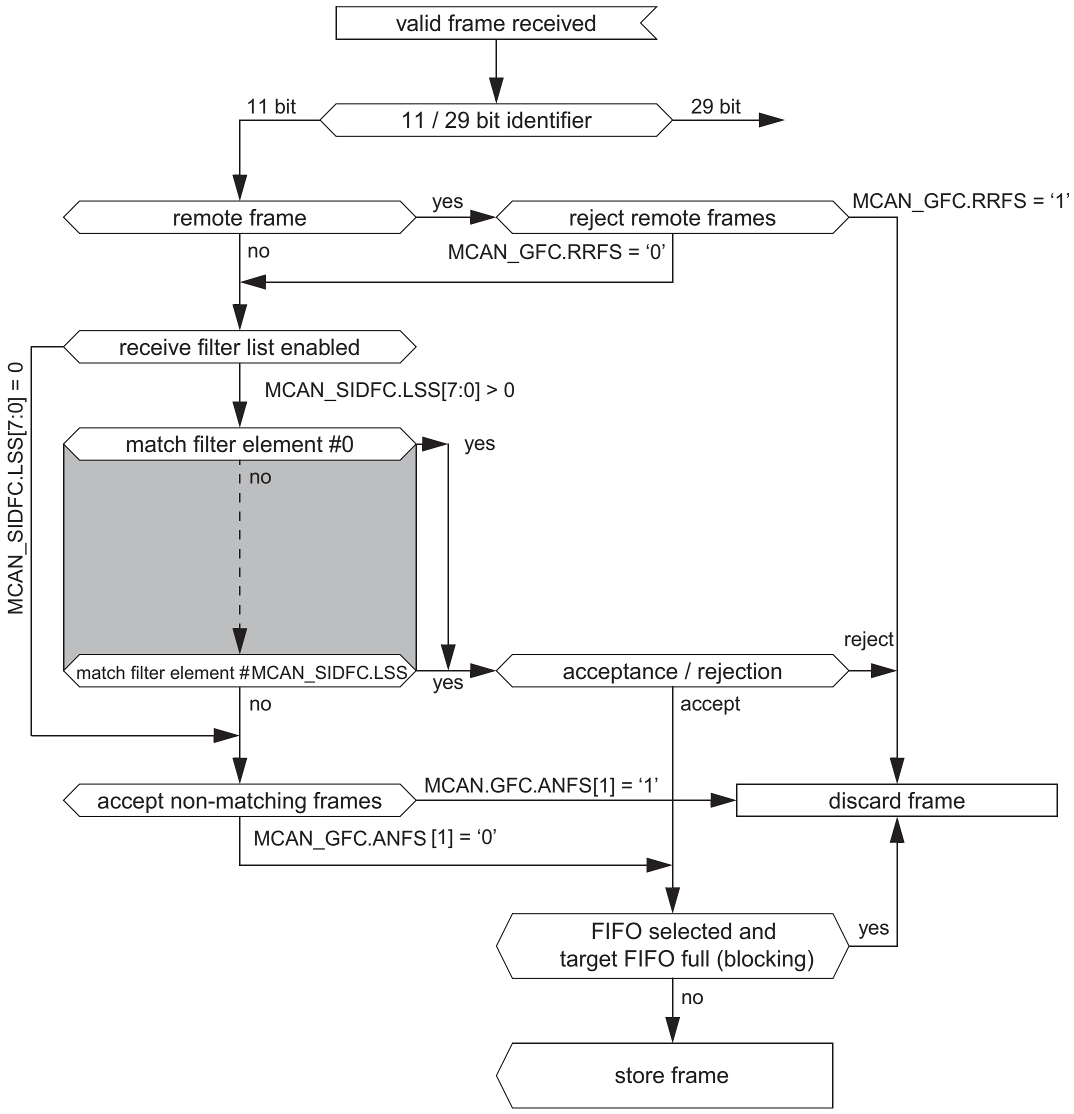
In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

Standard Message ID Filtering

Figure 49-5 below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in Section 49.5.7.5.

Controlled by MCAN_GFC and MCAN_SIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 49-5. Standard Message ID Filter Path



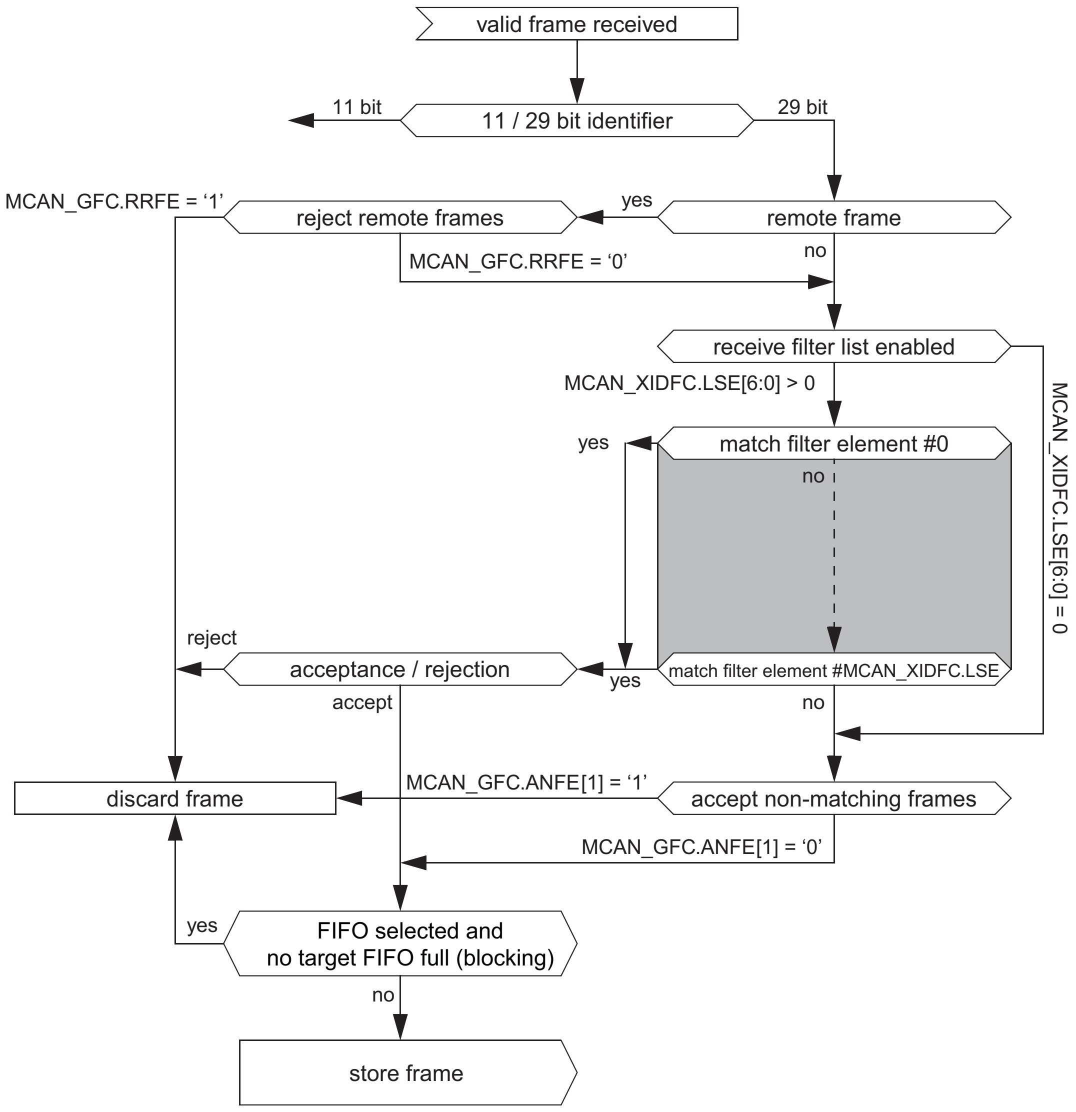
Extended Message ID Filtering

Figure 49-6 below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in Section 49.5.7.6.

Controlled by MCAN_GFC and MCAN_XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

MCAN_XIDAM is ANDed with the received identifier before the filter list is executed.

Figure 49-6. Extended Message ID Filter Path



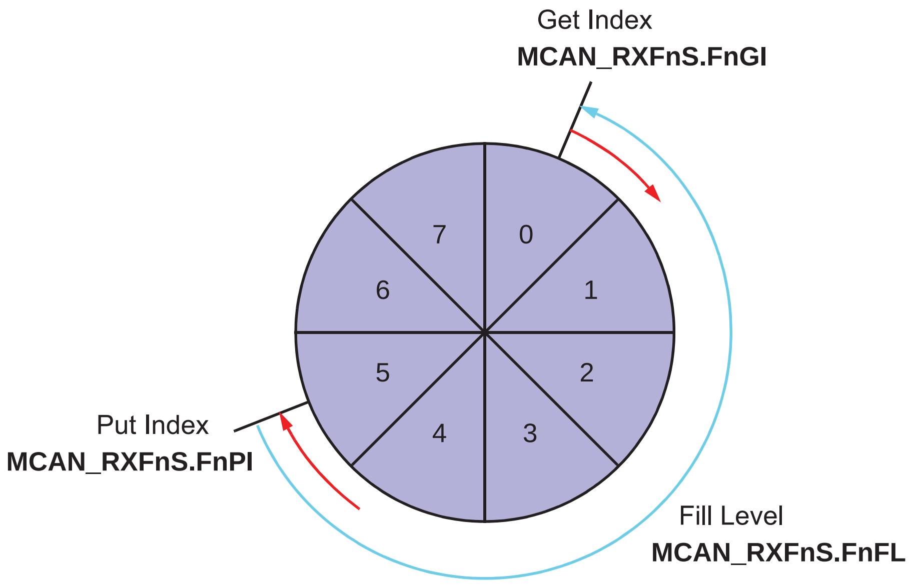
49.5.4.2 Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via the Rx FIFO 0 Configuration register (MCAN_RXF0C) and the Rx FIFO 1 Configuration register (MCAN_RXF1C).

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1, see Section 49.5.4.1. The Rx FIFO element is described in Section 49.5.7.2.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by MCAN_RXFnC.FnWM, interrupt flag MCAN_IR.RFnW is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index, an Rx FIFO Full condition is signalled by MCAN_RXFnS.FnF. In addition, the interrupt flag MCAN_IR.RFnF is set.

Figure 49-7. Rx FIFO Status



When reading from an Rx FIFO, Rx FIFO Get Index MCAN_RXFnS.FnGI × FIFO Element Size has to be added to the corresponding Rx FIFO start address MCAN_RXFnC.FnSA.

Table 49-4. Rx Buffer / FIFO Element Size

MCAN_RXESC.RBDS[2:0] MCAN_RXESC.FnDS[2:0]	Data Field [bytes]	FIFO Element Size [RAM words]
0	8	4
1	12	5
2	16	6
3	20	7
4	24	8
5	32	10
6	48	14
7	64	18

Rx FIFO Blocking Mode

The Rx FIFO Blocking mode is configured by MCAN_RXFnC.FnOM = ‘0’. This is the default operating mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (MCAN_RXFnS.FnPI = MCAN_RXFnS.FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by MCAN_RXFnS.FnF = ‘1’. In addition, the interrupt flag MCAN_IR.RFnF is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by MCAN_RXFnS.RFnL = ‘1’. In addition, the interrupt flag MCAN_IR.RFnL is set.

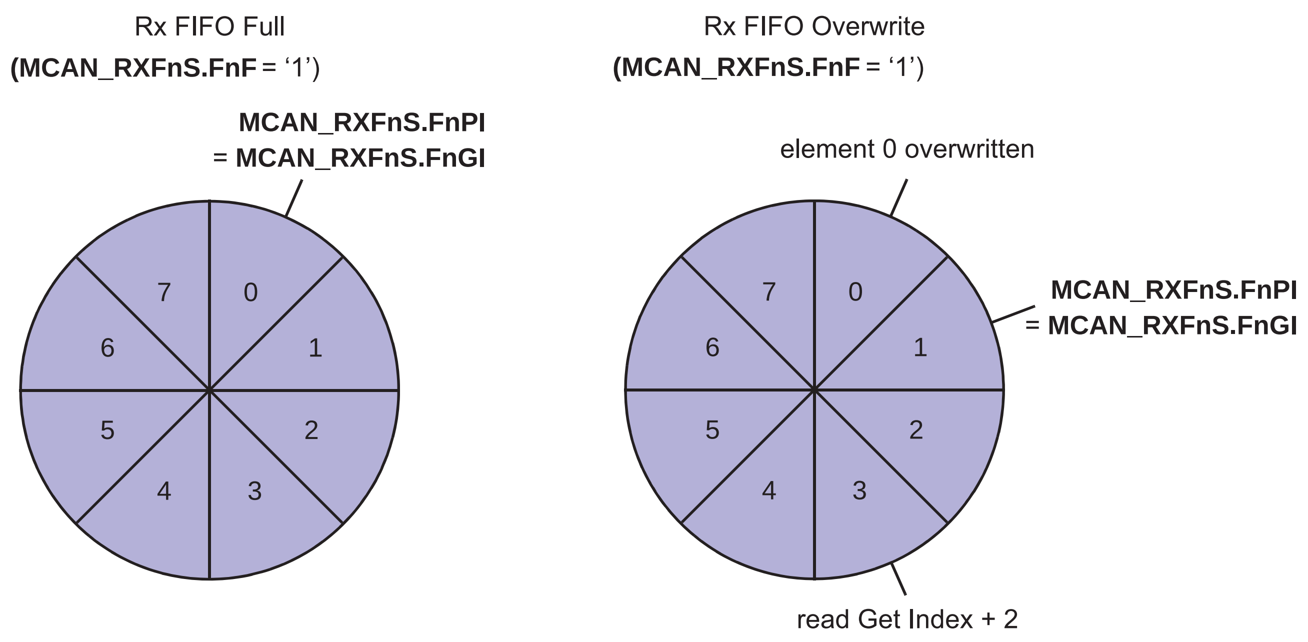
Rx FIFO Overwrite Mode

The Rx FIFO Overwrite mode is configured by MCAN_RXFnC.FnOM = ‘1’.

When an Rx FIFO full condition (MCAN_RXFnS.FnPI = MCAN_RXFnS.FnGI) is signalled by MCAN_RXFnS.FnF = ‘1’, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in Overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the processor is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the processor accesses the Rx FIFO. Figure 49-8 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 49-8. Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA.FnA. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS.FnF = ‘0’).

49.5.4.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via MCAN_RXBC.RBSA.

For each Rx Buffer, a Standard or Extended Message ID Filter Element with SFEC / EFEC = 7 and SFID2 / EFID2[10:9] = 0 has to be configured (see Section 49.5.7.5 and Section 49.5.7.6).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition, the flag MCAN_IR.DRX (Message stored in dedicated Rx Buffer) in MCAN_IR is set.

Table 49-5. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	0	0
1	ID message 2	0	1
2	ID message 3	0	2

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in the New Data 1 register (MCAN_NDAT1) and New Data 2 register (MCAN_NDAT2) is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the processor by writing a ‘1’ to the respective bit position.

While an Rx Buffer’s New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

Rx Buffer Handling

- Reset interrupt flag IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

49.5.4.4 Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element (see Section 49.5.7.2 “Rx Buffer and FIFO Element”).

Advantage: Fixed start address for the DMA transfers (relative to MCAN_RXBC.RBSA), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = ‘111’ have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output m_can_dma_req is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the MCAN while m_can_dma_req is activated. The behavior is similar to that of an Rx Buffer with its New Data flag set.

After the DMA has completed, the MCAN is prepared to receive the next set of debug messages.

Filtering for Debug Messages

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be programmed to “111”. In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning (see Section 49.5.7.5 and Section 49.5.7.6). While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor MCAN_IR.DRX are set. The reception of debug messages can be monitored via RXF1S.DMS.

Table 49-6. Example Filter Configuration for Debug Messages

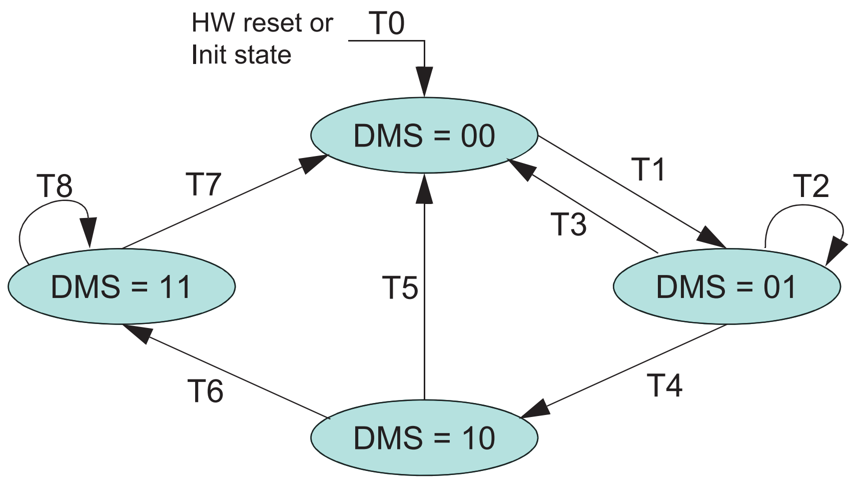
Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID debug message A	1	11 1101
1	ID debug message B	2	11 1110
2	ID debug message C	3	11 1111

Debug Message Handling

The debug message handling state machine ensures that debug messages are stored to three consecutive Rx Buffers in the correct order. If some messages are missing, the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in the correct order.

The status of the debug message handling state machine is signalled via MCAN_RXF1S.DMS.

Figure 49-9. Debug Message Handling State Machine



T0: reset m_can_dma_req output, enable reception of debug messages A, B, and C

T1: reception of debug message A

T2: reception of debug message A

T3: reception of debug message C

T4: reception of debug message B

T5: reception of debug messages A, B

T6: reception of debug message C

T7: DMA transfer completed

T8: reception of debug message A,B,C (message rejected)

49.5.5 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in Section 49.5.7.3. Table 49-7 describes the possible configurations for frame transmission.

Table 49-7. Possible Configurations for Frame Transmission

MCAN_CCCR		Tx Buffer Element		
BRSE	**FDOE**	**FDF**	**BRS**	**Frame Transmission**
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when MCAN_TXBRP is updated, or when a transmission has been started.

49.5.5.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit MCAN_CCCR.TXP. If the bit is set, the MCAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (MCAN_CCCR.TXP = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

49.5.5.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the processor. Each dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via MCAN_TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (see Table 49-8). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0…31) × Element Size to the Tx Buffer Start Address TXBC.TBSA.

Table 49-8. Tx Buffer / FIFO / Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
0	8	4
1	12	5
2	16	6
3	20	7
4	24	8
5	32	10
6	48	14
7	64	18

49.5.5.3 Tx FIFO

Tx FIFO operation is configured by programming MCAN_TXBC.TFQM to ‘0’. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index MCAN_TXFQS.TFGI. After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The MCAN calculates the Tx FIFO Free Level MCAN_TXFQS.TFFL as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQPI. An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (MCAN_TXFQS.TFQF = ‘1’) is signalled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a ‘1’ to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO’s Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via MCAN_TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 49-8). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQPI (0…31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA.

49.5.5.4 Tx Queue

Tx Queue operation is configured by programming MCAN_TXBC.TFQM to ‘1’. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQPI. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (MCAN_TXFQS.TFQF = ‘1’), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

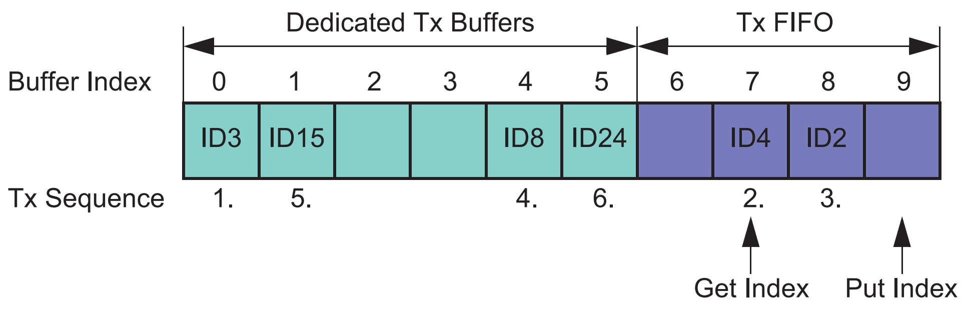
The application may use register MCAN_TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 49-8). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQPI (0…31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA.

49.5.5.5 Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx FIFO. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Buffers assigned to the Tx FIFO is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Figure 49-10. Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO



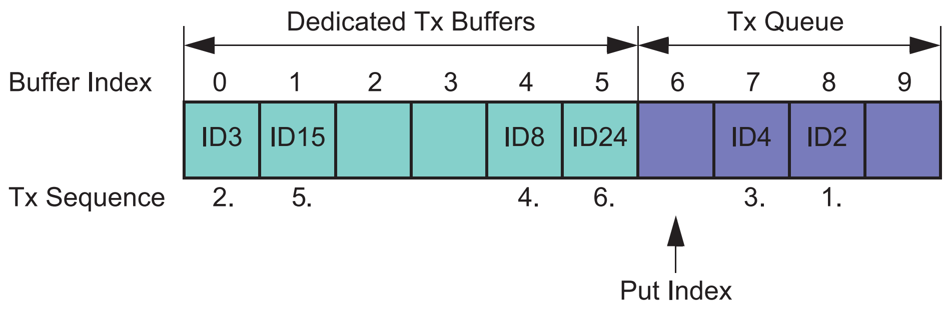
Tx prioritization:

- Scan dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by MCAN_TXFS.TFGI)
- Buffer with lowest Message ID gets highest priority and is transmitted next

49.5.5.6 Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx Queue. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Queue Buffers is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Figure 49-11. Example of Mixed Configuration Dedicated Tx Buffers / Tx Queue



Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

49.5.5.7 Transmit Cancellation

The MCAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR-based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer, the processor has to write a '1' to the corresponding bit position (=number of Tx Buffer) of register MCAN_TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of register MCAN_TXBCF to '1'.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO and MCAN_TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding MCAN_TXBCF bit is set.

Note: In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

49.5.5.8 Tx Event Handling

To support Tx event handling the MCAN has implemented a Tx Event FIFO. After the MCAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in Section 49.5.4.4.

When a Tx Event FIFO full condition is signalled by IR.TEFF, no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag MCAN_IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by MCAN_TXEFC.EFWM, interrupt flag MCAN_IR.TEFW is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS.EFGI has to be added to the Tx Event FIFO start address MCAN_TXEFC.EFSA.

49.5.6 FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see Section 49.6.29, Section 49.6.33, and Section 49.6.47). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the processor has free access to the MCAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN does not check for erroneous values.

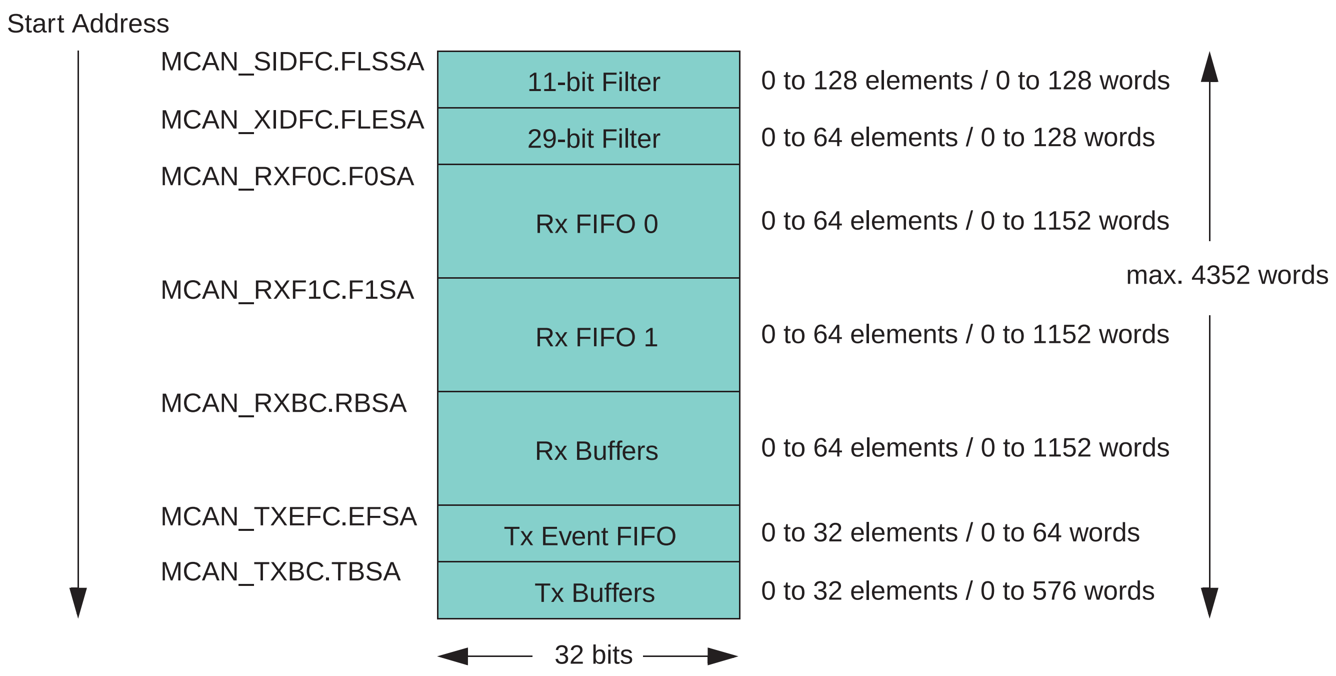
49.5.7 Message RAM

49.5.7.1 Message RAM Configuration

The Message RAM has a width of 32 bits. The MCAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in Figure 49-12, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode, the required Message RAM size depends on the element size configured for Rx FIFO0, Rx FIFO1, Rx Buffers, and Tx Buffers via MCAN_RXESC.F0DS, MCAN_RXESC.F1DS, MCAN_RXESC.RBDS, and MCAN_TXESC.TBDS.

Figure 49-12. Message RAM Configuration


Start Address
MCAN_SIDFC.FLSSA
MCAN_XIDFC.FLESA
MCAN_RXF0C.F0SA
MCAN_RXF1C.F1SA
MCAN_RXBC.RBSA
MCAN_TXEFC.EFSA
MCAN_TXBC.TBSA
11-bit Filter
29-bit Filter
Rx FIFO 0
Rx FIFO 1
Rx Buffers
Tx Event FIFO
Tx Buffers
0 to 128 elements / 0 to 128 words
0 to 64 elements / 0 to 128 words
0 to 64 elements / 0 to 1152 words
0 to 64 elements / 0 to 1152 words
0 to 64 elements / 0 to 1152 words
0 to 32 elements / 0 to 64 words
0 to 32 elements / 0 to 576 words
max. 4352 words
32 bits

When the MCAN addresses the Message RAM, it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses; i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Note: The MCAN does not check for erroneous configuration of the Message RAM. The configuration of the start addresses of the different sections and the number of elements of each section must be checked carefully to avoid falsification or loss of data.

49.5.7.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in Table 49-9 below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register MCAN_RXESC.

Table 49-9. Rx Buffer and FIFO Element

	31			24	23			16	15 8	7 0
R0	ESI	XTD	RTR	ID[28:0]						
R1	ANMF	FIDX[6:0]			–	FDF	BRS	DLC[3:0]	RXTS[15:0]	
R2	DB3[7:0]				DB2[7:0]				DB1[7:0]	DB0[7:0]
R3	DB7[7:0]				DB6[7:0]				DB5[7:0]	DB4[7:0]
...	...				...				...	...
Rn	DBm[7:0]				DBm-1[7:0]				DBm-2[7:0]	DBm-3[7:0]

- **R0 Bit 31 ESI: Error State Indicator**

0: Transmitting node is error active.

1: Transmitting node is error passive.

- **R0 Bit 30 XTD: Extended Identifier**

Signals to the processor whether the received frame has a standard or extended identifier.

0: 11-bit standard identifier.

1: 29-bit extended identifier.

- **R0 Bit 29 RTR: Remote Transmission Request**

Signals to the processor whether the received frame is a data frame or a remote frame.

0: Received frame is a data frame.

1: Received frame is a remote frame.

Note: There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1), bit RTR reflects the state of the reserved bit r1.

- **R0 Bits 28:0 ID[28:0]: Identifier**

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

- **R1 Bit 31 ANMF: Accepted Non-matching Frame**

Acceptance of non-matching frames may be enabled via MCAN_GFC.ANFS and MCAN_GFC.ANFE.

0: Received frame matching filter index FIDX.

1: Received frame did not match any Rx filter element.

- **R1 Bits 30:24 FIDX[6:0]: Filter Index**

0-127: Index of matching Rx acceptance filter element (invalid if ANMF = ‘1’).
Range is 0 to MCAN_SIDFC.LSS - 1 resp. MCAN_XIDFC.LSE - 1.

- **R1 Bit 21 FDF: FD Format**

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

- **R1 Bit 20 BRS: Bit Rate Switch**

0: Frame received without bit rate switching.

1: Frame received with bit rate switching.

Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled (MCAN_CCCR.FDOE = 1). Bit BRS is only evaluated when in addition MCAN_CCCR.BRSE = 1.

- **R1 Bits 19:16 DLC[3:0]: Data Length Code**

0-8: CAN + CAN FD: received frame has 0-8 data bytes.

9-15: CAN: received frame has 8 data bytes.

9-15: CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

- **R1 Bits 15:0 RXTS[15:0]: Rx Timestamp**

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP.

- **R2 Bits 31:24 DB3[7:0]: Data Byte 3**
- **R2 Bits 23:16 DB2[7:0]: Data Byte 2**
- **R2 Bits 15:8 DB1[7:0]: Data Byte 1**
- **R2 Bits 7:0 DB0[7:0]: Data Byte 0**
- **R3 Bits 31:24 DB7[7:0]: Data Byte 7**
- **R3 Bits 23:16 DB6[7:0]: Data Byte 6**
- **R3 Bits 15:8 DB5[7:0]: Data Byte 5**
- **R3 Bits 7:0 DB4[7:0]: Data Byte 4**

... ...

- **Rn Bits 31:24 DBm[7:0]: Data Byte m**
- **Rn Bits 23:16 DBm-1[7:0]: Data Byte m-1**
- **Rn Bits 15:8 DBm-2[7:0]: Data Byte m-2**
- **Rn Bits 7:0 DBm-3[7:0]: Data Byte m-3**

Note: Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3 ..17) are used for storage of a CAN message’s data field.

49.5.7.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 49-10. Tx Buffer Element

	31			24	23				16	15 8	7 0
T0	ESI	XTD	RTR	ID[28:0]							
T1	MM[7:0]				EFC	reserved	FDF	BRS	DLC[3:0]	reserved	
T2	DB3[7:0]				DB2[7:0]					DB1[7:0]	DB0[7:0]
T3	DB7[7:0]				DB6[7:0]					DB5[7:0]	DB4[7:0]
...	...				...					...	...
Tn	DBm[7:0]				DBm-1[7:0]					DBm-2[7:0]	DBm-3[7:0]

- **T0 Bit 30 ESI: Error State Indicator**

T0 Bit 31 ESI: Error State Indicator

0: ESI bit in CAN FD format depends only on error passive flag

1: ESI bit in CAN FD format transmitted recessive

Note: The ESI bit of the transmit buffer is or'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive. This feature can be used in gateway applications when a message from an error passive node is routed to another CAN network.

- **T0 Bit 30 XTD: Extended Identifier**

0: 11-bit standard identifier.

1: 29-bit extended identifier.

- **T0 Bit 29 RTR: Remote Transmission Request**

0: Transmit data frame.

1: Transmit remote frame.

Note: When RTR = 1, the MCAN transmits a remote frame according to ISO11898-1, even if MCAN_CCCR.FDOE enables the transmission in CAN FD format.

- **T0 Bits 28:0 ID[28:0]: Identifier**

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

- **T1 Bits 31:24 MM[7:0]: Message Marker**

Written by processor during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

- **T1 Bit 23 EFC: Event FIFO Control**

0: Do not store Tx events.

1: Store Tx events.

- **T1 Bit 21 FDF: FD Format**

0: Frame transmitted in Classic CAN format

1: Frame transmitted in CAN FD format

- **T1 Bit 20 BRS: Bit Rate Switching**

0: CAN FD frames transmitted without bit rate switching

1: CAN FD frames transmitted with bit rate switching

Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled (MCAN_CCCR.FDOE = 1). Bit BRS is only evaluated when in addition MCAN_CCCR.BRSE = 1.

- **T1 Bits 19:16 DLC[3:0]: Data Length Code**

0-8: CAN + CAN FD: transmit frame has 0-8 data bytes.

9-15: CAN: transmit frame has 8 data bytes.

9-15: CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes.

- **T2 Bits 31:24 DB3[7:0]: Data Byte 3**
- **T2 Bits 23:16 DB2[7:0]: Data Byte 2**
- **T2 Bits 15:8 DB1[7:0]: Data Byte 1**
- **T2 Bits 7:0 DB0[7:0]: Data Byte 0**
- **T3 Bits 31:24 DB7[7:0]: Data Byte 7**
- **T3 Bits 23:16 DB6[7:0]: Data Byte 6**
- **T3 Bits 15:8 DB5[7:0]: Data Byte 5**
- **T3 Bits 7:0 DB4[7:0]: Data Byte 4**

... ...

- **Tn Bits 31:24 DBm[7:0]: Data Byte m**
- **Tn Bits 23:16 DBm-1[7:0]: Data Byte m-1**
- **Tn Bits 15:8 DBm-2[7:0]: Data Byte m-2**
- **Tn Bits 7:0 DBm-3[7:0]: Data Byte m-3**

Note: Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3 ..17) are used for storage of a CAN message’s data field.

49.5.7.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the processor gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

Table 49-11. Tx Event FIFO Element

	31			24	23			16	15	8	7	0
E0	ESI	XTD	RTR	ID[28:0]								
E1	MM[7:0]				ET [1:0]	FDF	BRS	DLC[3:0]	TXTS[15:0]			

- **E0 Bit 31 ESI: Error State Indicator**

0: Transmitting node is error active.

1: Transmitting node is error passive.

- **E0 Bit 30 XTD: Extended Identifier**

0: 11-bit standard identifier.

1: 29-bit extended identifier.

- **E0 Bit 29 RTR: Remote Transmission Request**

0: Data frame transmitted.

1: Remote frame transmitted.

- **E0 Bits 28:0 ID[28:0]: Identifier**

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

- **E1 Bits 31:24 MM[7:0]: Message Marker**

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

- **E1 Bit 23:22 ET[1:0]: Event Type**

Value	Description
0	Reserved
1	Tx event
2	Transmission in spite of cancellation (always set for transmissions in DAR mode)
3	Reserved

- **E1 Bit 21 FDF: FD Format**

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

- **E1 Bit 20 BRS: Bit Rate Switch**

0: Frame transmitted without bit rate switching.

1: Frame transmitted with bit rate switching.

- **E1 Bits 19:16 DLC[3:0]: Data Length Code**

0-8: CAN + CAN FD: frame with 0-8 data bytes transmitted.

9-15: CAN: frame with 8 data bytes transmitted.

9-15: CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted

- **E1 Bits 15:0 TXTS[15:0]: Tx Timestamp**

Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Time-stamp Counter Prescaler MCAN_TSCC.TCP.

49.5.7.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address MCAN_SIDFC.FLSSA plus the index of the filter element (0…127).

Table 49-12. Standard Message ID Filter Element

	31		24 23	16 15	8 7 … 0
S0	SFT[1:0]	SFEC [2:0]	SFID1[10:0]	–	SFID2[10:0]

- **Bits 31:30 SFT[1:0]: Standard Filter Type**

Value	Description
0	Range filter from SF1ID to SF2ID (SF2ID ≥ SF1ID)
1	Dual ID filter for SF1ID or SF2ID
2	Classic filter: SF1ID = filter, SF2ID = mask
3	Reserved

- **Bit 29:27 SFEC[2:0]: Standard Filter Element Configuration**

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101”, or “110” a match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored

- **Bits 26:16 SFID1[10:0]: Standard Filter ID 1**

First ID of standard ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.

- **Bits 10:0 SFID2[10:0]: Standard Filter ID 2**

This field has a different meaning depending on the configuration of SFEC:

- SFEC = “001”...“110”–Second ID of standard ID filter element
- SFEC = “111”–Filter for Rx Buffers or for debug messages

SFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

Value	Description
0	Store message in a Rx buffer
1	Debug Message A
2	Debug Message B
3	Debug Message C

SFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

49.5.7.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC.FLESA plus two times the index of the filter element (0…63).

Table 49-13. Extended Message ID Filter Element

	31 … 24	23 … 16	15 … 8	7 … 0
F0	EFEC [2:0] / EFID1[28:0]			
F1	EFT[1:0] / – / EFID2[28:0]			

- **F0 Bit 31:29 EFEC[2:0]: Extended Filter Element Configuration**

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110”, a match sets the interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, register MCAN_HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored

- **F0 Bits 28:0 EFID1[28:0]: Extended Filter ID 1**

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only MCAN_XIDAM masking mechanism (see Extended Message ID Filtering) is used.

- **F1 Bits 31:30 EFT[1:0]: Extended Filter Type**

Value	Description
0	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID)
1	Dual ID filter for EF1ID or EF2ID
2	Classic filter: EF1ID = filter, EF2ID = mask
3	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID), MCAN_XIDAM mask not applied

- **F1 Bits 28:0 EFID2[28:0]: Extended Filter ID 2**

This field has a different meaning depending on the configuration of EFEC:

- EFEC = “001”...“110”–Second ID of extended ID filter element
- EFEC = “111”–Filter for Rx Buffers or for debug messages

EFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

Value	Description
0	Store message in a Rx buffer
1	Debug Message A
2	Debug Message B
3	Debug Message C

EFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

49.5.8 Hardware Reset Description

After hardware reset, the registers of the MCAN hold the reset values listed in Table 49-14. Additionally the Bus_Off state is reset and the output CANTX is set to recessive (HIGH). The value 0x0001 (MCAN_CCCR.INIT = ‘1’) in the CC Control register enables software initialization. The MCAN does not influence the CAN bus until the processor resets MCAN_CCCR.INIT to ‘0’.

49.5.9 Access to Reserved Register Addresses

In case the application software accesses one of the reserved addresses in the MCAN register map (read or write access), interrupt flag MCAN_IR.ARA is set and, if enabled, the selected interrupt line is risen.

49.6 Controller Area Network (MCAN) User Interface

Table 49-14. Register Mapping

Offset	Register	Name	Access	Reset
0x00	Core Release Register	MCAN_CREL	Read-only	0xrrrddddd[1]
0x04	Endian Register	MCAN_ENDN	Read-only	0x87654321
0x08	Customer Register	MCAN_CUST	Read/Write	0
0x0C	Data Bit Timing and Prescaler Register	MCAN_DBTP	Read/Write	0x00000A33
0x10	Test Register	MCAN_TEST	Read/Write	0x000000x0[2]
0x14	RAM Watchdog Register	MCAN_RWD	Read/Write	0x00000000
0x18	CC Control Register	MCAN_CCCR	Read/Write	0x00000001
0x1C	Nominal Bit Timing and Prescaler Register	MCAN_NBTP	Read/Write	0x06000A03
0x20	Timestamp Counter Configuration Register	MCAN_TSCC	Read/Write	0x00000000
0x24	Timestamp Counter Value Register	MCAN_TSCV	Read/Write	0x00000000
0x28	Timeout Counter Configuration Register	MCAN_TOCC	Read/Write	0xFFFF0000
0x2C	Timeout Counter Value Register	MCAN_TOCV	Read/Write	0x0000FFFF
0x30–0x3C	Reserved	–	–	–
0x40	Error Counter Register	MCAN_ECR	Read-only	0x00000000
0x44	Protocol Status Register	MCAN_PSR	Read-only	0x00000707
0x48	Transmit Delay Compensation Register	MCAN_TDCR	Read/Write	0x00000000
0x4C	Reserved	–	–	–
0x50	Interrupt Register	MCAN_IR	Read/Write	0x00000000
0x54	Interrupt Enable Register	MCAN_IE	Read/Write	0x00000000
0x58	Interrupt Line Select Register	MCAN_ILS	Read/Write	0x00000000
0x5C	Interrupt Line Enable Register	MCAN_ILE	Read/Write	0x00000000
0x60–0x7C	Reserved	–	–	–
0x80	Global Filter Configuration Register	MCAN_GFC	Read/Write	0x00000000
0x84	Standard ID Filter Configuration Register	MCAN_SIDFC	Read/Write	0x00000000
0x88	Extended ID Filter Configuration Register	MCAN_XIDFC	Read/Write	0x00000000
0x8C	Reserved	–	–	–
0x90	Extended ID AND Mask Register	MCAN_XIDAM	Read/Write	0x1FFFFFFF
0x94	High Priority Message Status Register	MCAN_HPMS	Read-only	0x00000000
0x98	New Data 1 Register	MCAN_NDAT1	Read/Write	0x00000000
0x9C	New Data 2 Register	MCAN_NDAT2	Read/Write	0x00000000
0xA0	Receive FIFO 0 Configuration Register	MCAN_RXF0C	Read/Write	0x00000000
0xA4	Receive FIFO 0 Status Register	MCAN_RXF0S	Read-only	0x00000000
0xA8	Receive FIFO 0 Acknowledge Register	MCAN_RXF0A	Read/Write	0x00000000
0xAC	Receive Rx Buffer Configuration Register	MCAN_RXBC	Read/Write	0x00000000
0xB0	Receive FIFO 1 Configuration Register	MCAN_RXF1C	Read/Write	0x00000000

Table 49-14. Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0xB4	Receive FIFO 1 Status Register	MCAN_RXF1S	Read-only	0x00000000
0xB8	Receive FIFO 1 Acknowledge Register	MCAN_RXF1A	Read/Write	0x00000000
0xBC	Receive Buffer / FIFO Element Size Configuration Register	MCAN_RXESC	Read/Write	0x00000000
0xC0	Transmit Buffer Configuration Register	MCAN_TXBC	Read/Write	0x00000000
0xC4	Transmit FIFO/Queue Status Register	MCAN_TXFQS	Read-only	0x00000000
0xC8	Transmit Buffer Element Size Configuration Register	MCAN_TXESC	Read/Write	0x00000000
0xCC	Transmit Buffer Request Pending Register	MCAN_TXBRP	Read-only	0x00000000
0xD0	Transmit Buffer Add Request Register	MCAN_TXBAR	Read/Write	0x00000000
0xD4	Transmit Buffer Cancellation Request Register	MCAN_TXBCR	Read/Write	0x00000000
0xD8	Transmit Buffer Transmission Occurred Register	MCAN_TXBTO	Read-only	0x00000000
0xDC	Transmit Buffer Cancellation Finished Register	MCAN_TXBCF	Read-only	0x00000000
0xE0	Transmit Buffer Transmission Interrupt Enable Register	MCAN_TXBTIE	Read/Write	0x00000000
0xE4	Transmit Buffer Cancellation Finished Interrupt Enable Register	MCAN_TXBCIE	Read/Write	0x00000000
0xE8–0xEC	Reserved	–	–	–
0xF0	Transmit Event FIFO Configuration Register	MCAN_TXEFC	Read/Write	0x00000000
0xF4	Transmit Event FIFO Status Register	MCAN_TXEFS	Read-only	0x00000000
0xF8	Transmit Event FIFO Acknowledge Register	MCAN_TXEFA	Read/Write	0x00000000
0xFC	Reserved	–	–	–

Notes:
1. Due to clock domain crossing, there is a delay between when a register bit or field is written and when the related status register bits are updated.
2. The reset value for bit 7, MCAN_TEST.RX, is undefined.

49.6.1 MCAN Core Release Register

Name: MCAN_CREL

Address: 0x40030000 (0), 0x40034000 (1)

Access: Read-only

31	30	29	28	27	26	25	24
REL				STEP			
23	22	21	20	19	18	17	16
SUBSTEP				YEAR			
15	14	13	12	11	10	9	8
MON							
7	6	5	4	3	2	1	0
DAY							

- **DAY: Timestamp Day**

Two digits, BCD-coded. This field is set by generic parameter on MCAN synthesis.

- **MON: Timestamp Month**

Two digits, BCD-coded. This field is set by generic parameter on MCAN synthesis.

- **YEAR: Timestamp Year**

One digit, BCD-coded. This field is set by generic parameter on MCAN synthesis.

- **SUBSTEP: Sub-step of Core Release**

One digit, BCD-coded.

- **STEP: Step of Core Release**

One digit, BCD-coded.

- **REL: Core Release**

One digit, BCD-coded.

49.6.2 MCAN Endian Register

Name: MCAN_ENDN

Address: 0x40030004 (0), 0x40034004 (1)

Access: Read-only

31	30	29	28	27	26	25	24
ETV							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ETV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ETV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETV							

- **ETV: Endianness Test Value**

The endianness test value is 0x87654321.

49.6.3 MCAN Customer Register

Name: MCAN_CUST

Address: 0x40030008 (0), 0x40034008 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
CSV							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CSV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CSV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CSV							

- **CSV: Customer-specific Value**

Customer-specific value.

49.6.4 MCAN Data Bit Timing and Prescaler Register

Name: MCAN_DBTP

Address: 0x4003000C (0), 0x4003400C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDC	–	–	DBRP				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	DTSEG1				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTSEG2				–	DSJW		

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

The CAN bit time may be programmed in the range of 4 to 25 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 CAN core clock periods. t_q = (DBRP + 1) CAN core clock periods.

DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [DTSEG1 + DTSEG2 + 3] t_q
or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

- **DSJW: Data (Re) Synchronization Jump Width**

The duration of a synchronization jump is t_q x (DSJW + 1).

- **DTSEG2: Data Time Segment After Sample Point**

The duration of time segment is t_q x (DTSEG2 + 1).

- **DTSEG1: Data Time Segment Before Sample Point**

0: Forbidden.

1 to 31: The duration of time segment is t_q x (DTSEG1 + 1).

- **DBRP: Data Bit Rate Prescaler**

The value by which the peripheral clock is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

- **TDC: Transmitter Delay Compensation**

0 (DISABLED): Transmitter Delay Compensation disabled.

1 (ENABLED): Transmitter Delay Compensation enabled.

Notes: 1. With a CAN core clock frequency of 8 MHz, the reset value of 0x00000A33 configures the MCAN for a fast bit rate of 500 kbit/s.
2. The bit rate configured for the CAN FD data phase via MCAN_DBTP must be higher than or equal to the bit rate configured for the arbitration phase via MCAN_NBTP.

49.6.5 MCAN Test Register

Name: MCAN_TEST

Address: 0x40030010 (0), 0x40034010 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX	TX		LBCK	–	–	–	–

Write access to the Test Register has to be enabled by setting bit MCAN_CCCR.TEST to '1'.

All MCAN Test Register functions are set to their reset values when bit MCAN_CCCR.TEST is cleared.

Loop Back mode and software control of pin CANTX are hardware test modes. Programming of TX ≠ 0 disturbs the message transfer on the CAN bus.

- **LBCK: Loop Back Mode (read/write)**

0 (DISABLED): Reset value. Loop Back mode is disabled.

1 (ENABLED): Loop Back mode is enabled (see Section 49.5.1.9).

- **TX: Control of Transmit Pin (read/write)**

Value	Name	Description
0	RESET	Reset value, CANTX controlled by the CAN Core, updated at the end of the CAN bit time.
1	SAMPLE_POINT_MONITORIN G	Sample Point can be monitored at pin CANTX.
2	DOMINANT	Dominant ('0') level at pin CANTX.
3	RECESSIVE	Recessive ('1') at pin CANTX.

- **RX: Receive Pin (read-only)**

Monitors the actual value of pin CANRX.

0: The CAN bus is dominant (CANRX = '0').

1: The CAN bus is recessive (CANRX = '1').

49.6.6 MCAN RAM Watchdog Register

Name: MCAN_RWD

Address: 0x40030014 (0), 0x40034014 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WDV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDC							

The RAM Watchdog monitors the Message RAM response time. A Message RAM access via the MCAN's Generic Master Interface starts the Message RAM Watchdog Counter with the value configured by MCAN_RWD.WDC. The counter is reloaded with MCAN_RWD.WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR.WDI is set. The RAM Watchdog Counter is clocked by the system bus clock (peripheral clock).

- **WDC: Watchdog Configuration (read/write)**

Start value of the Message RAM Watchdog Counter. The counter is disabled when WDC is cleared.

- **WDV: Watchdog Value (read-only)**

Watchdog Counter Value for the current message located in RAM.

49.6.7 MCAN CC Control Register

Name: MCAN_CCCR

Address: 0x40030018 (0), 0x40034018 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NISO	TXP	EFBI	PXHD	–	–	BRSE	FDOE
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

- **INIT: Initialization (read/write)**

0 (DISABLED): Normal operation.

1 (ENABLED): Initialization is started.

Note: Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to ensure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.

- **CCE: Configuration Change Enable (read/write, write protection)**

0 (PROTECTED): The processor has no write access to the protected configuration registers.

1 (CONFIGURABLE): The processor has write access to the protected configuration registers (while MCAN_CCCR.INIT = ‘1’).

- **ASM: Restricted Operation Mode (read/write, write protection against ‘1’)**

For a description of the Restricted Operation mode see Section 49.5.1.5.

0 (NORMAL): Normal CAN operation.

1 (RESTRICTED): Restricted Operation mode active.

- **CSA: Clock Stop Acknowledge (read-only)**

0: No clock stop acknowledged.

1: MCAN may be set in power down by stopping the peripheral clock and the CAN core clock.

- **CSR: Clock Stop Request (read/write)**

0 (NO_CLOCK_STOP): No clock stop is requested.

1 (CLOCK_STOP): Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

- **MON: Bus Monitoring Mode (read/write, write protection against ‘1’)**

0 (DISABLED): Bus Monitoring mode is disabled.

1 (ENABLED): Bus Monitoring mode is enabled.

- **DAR: Disable Automatic Retransmission (read/write, write protection)**

0 (AUTO_RETX): Automatic retransmission of messages not transmitted successfully enabled.

1 (NO_AUTO_RETX): Automatic retransmission disabled.

- **TEST: Test Mode Enable (read/write, write protection against ‘1’)**

0 (DISABLED): Normal operation, MCAN_TEST register holds reset values.

1 (ENABLED): Test mode, write access to MCAN_TEST register enabled.

- **FDOE: CAN FD Operation Enable (read/write, write protection)**

0 (DISABLED): FD operation disabled.

1 (ENABLED): FD operation enabled.

- **BRSE: Bit Rate Switching Enable (read/write, write protection)**

0 (DISABLED): Bit rate switching for transmissions disabled.

1 (ENABLED): Bit rate switching for transmissions enabled.

- **PXHD: Protocol Exception Event Handling (read/write, write protection)**

0: Protocol exception handling enabled.

1: Protocol exception handling disabled.

- **EFBI: Edge Filtering during Bus Integration (read/write, write protection)**

0: Edge filtering is disabled.

1: Edge filtering is enabled. Two consecutive dominant tq required to detect an edge for hard synchronization.

- **TXP: Transmit Pause (read/write, write protection)**

If this bit is set, the MCAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see Section 49.5.5).

0: Transmit pause disabled.

1: Transmit pause enabled.

- **NISO: Non-ISO Operation**

If this bit is set, the MCAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0.

0: CAN FD frame format according to ISO11898-1 (default).

1: CAN FD frame format according to Bosch CAN FD Specification V1.0.

49.6.8 MCAN Nominal Bit Timing and Prescaler Register

Name: MCAN_NBTP

Address: 0x4003001C (0), 0x4003401C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
NSJW							NBRP
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
NBRP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
NTSEG1							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	NTSEG2						

This register can only be written if the bits CCE and INIT are set in MCAN_CCCR.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 CAN core clock periods. $t_q = t_{core\ clock}$ x (NBRP + 1).

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [NTSEG1 + NTSEG2 + 3] t_q
or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

- **NTSEG2: Nominal Time Segment After Sample Point**

0 to 127: The duration of time segment is t_q x (NTSEG2 + 1).

- **NTSEG1: Nominal Time Segment Before Sample Point**

0: Forbidden.

1 to 255: The duration of time segment is t_q x (NTSEG1 + 1).

- **NBRP: Nominal Bit Rate Prescaler**

0 to 511: The value by which the oscillator frequency is divided for generating the CAN time quanta. The CAN time is built up from a multiple of this quanta. CAN time quantum (tq) = $t_{core\ clock}$ x (NBRP + 1)

Note: With a CAN core clock frequency of 8 MHz, the reset value of 0x06000A03 configures the MCAN for a bit rate of 500 kbit/s.

- **NSJW: Nominal (Re) Synchronization Jump Width**

0 to 127: The duration of a synchronization jump is t_q x (NSJW + 1).

49.6.9 MCAN Timestamp Counter Configuration Register

Name: MCAN_TSCC

Address: 0x40030020 (0), 0x40034020 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	TCP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	–	–	–	–	–	TSS	

For a description of the Timestamp Counter see Section 49.5.2.

- **TSS: Timestamp Select**

Value	Name	Description
0	ALWAYS_0	Timestamp counter value always 0x0000
1	TCP_INC	Timestamp counter value incremented according to TCP
2	EXT_TIMESTAMP	External timestamp counter value used
3	ALWAYS_0	Timestamp counter value always 0x0000

- **TCP: Timestamp Counter Prescaler**

Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1…16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Note: With CAN FD, an external counter is required for timestamp generation (TSS = 2).

49.6.10 MCAN Timestamp Counter Value Register

Name: MCAN_TSCV

Address: 0x40030024 (0), 0x40034024 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TSC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TSC							

- **TSC: Timestamp Counter (cleared on write)**

The internal/external Timestamp Counter value is captured on start of frame (both Receive and Transmit). When MCAN_TSCC.TSS = 1, the Timestamp Counter is incremented in multiples of CAN bit times [1…16] depending on the configuration of MCAN_TSCC.TCP. A wrap around sets interrupt flag MCAN_IR.TSW. Write access resets the counter to zero.

When MCAN_TSCC.TSS = 2, TSC reflects the external Timestamp Counter value. Thus a write access has no impact.

Note: A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by write access to MCAN_TSCV.

49.6.11 MCAN Timeout Counter Configuration Register

Name: MCAN_TOCC

Address: 0x40030028 (0), 0x40034028 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
TOP							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TOP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	–	–	–	–	TOS		ETOC

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

For a description of the Timeout Counter, see Section 49.5.3.

- **ETOC: Enable Timeout Counter**

0 (NO_TIMEOUT): Timeout Counter disabled.

1 (TOS_CONTROLLED): Timeout Counter enabled.

For use of timeout function with CAN FD, see Section 49.5.3.

- **TOS: Timeout Select**

When operating in Continuous mode, a write to MCAN_TOCV presets the counter to the value configured by MCAN_TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by MCAN_TOCC.TOP. Down-counting is started when the first FIFO element is stored.

Value	Name	Description
0	CONTINUOUS	Continuous operation
1	TX_EV_TIMEOUT	Timeout controlled by Tx Event FIFO
2	RX0_EV_TIMEOUT	Timeout controlled by Receive FIFO 0
3	RX1_EV_TIMEOUT	Timeout controlled by Receive FIFO 1

- **TOP: Timeout Period**

Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

49.6.12 MCAN Timeout Counter Value Register

Name: MCAN_TOCV

Address: 0x4003002C (0), 0x4003402C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TOC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOC							

- **TOC: Timeout Counter (cleared on write)**

The Timeout Counter is decremented in multiples of CAN bit times [1…16] depending on the configuration of MCAN_TSCC.TCP. When decremented to zero, interrupt flag MCAN_IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via MCAN_TOCC.TOS.

49.6.13 MCAN Error Counter Register

Name: MCAN_ECR

Address: 0x40030040 (0), 0x40034040 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CEL							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RP	REC						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEC							

- **TEC: Transmit Error Counter**

Actual state of the Transmit Error Counter, values between 0 and 255.

- **REC: Receive Error Counter**

Actual state of the Receive Error Counter, values between 0 and 127.

- **RP: Receive Error Passive**

0: The Receive Error Counter is below the error passive level of 128.

1: The Receive Error Counter has reached the error passive level of 128.

- **CEL: CAN Error Logging (cleared on read)**

The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO.

Note: When MCAN_CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

49.6.14 MCAN Protocol Status Register

Name: MCAN_PSR

Address: 0x40030044 (0), 0x40034044 (1)

Access: Read-only/

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	TDCV						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	PXE	RFDF	RBRS	RESI	DLEC		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BO	EW	EP	ACT		LEC		

- **LEC: Last Error Code (set to 111 on read)**

The LEC indicates the type of the last error to occur on the CAN bus. This field is cleared when a message has been transferred (reception or transmission) without error.

Value	Name	Description
0	NO_ERROR	No error occurred since LEC has been reset by successful reception or transmission.
1	STUFF_ERROR	More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
2	FORM_ERROR	A fixed format part of a received frame has the wrong format.
3	ACK_ERROR	The message transmitted by the MCAN was not acknowledged by another node.
4	BIT1_ERROR	During transmission of a message (with the exception of the arbitration field), the device tried to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant.
5	BIT0_ERROR	During transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device tried to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the processor to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRC_ERROR	The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match the CRC calculated from the received data.
7	NO_CHANGE	Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows value '7', no CAN bus event was detected since the last processor read access to the Protocol Status Register.

- **ACT: Activity**

Monitors the CAN communication state of the CAN module.

Value	Name	Description
0	SYNCHRONIZING	Node is synchronizing on CAN communication
1	IDLE	Node is neither receiver nor transmitter
2	RECEIVER	Node is operating as receiver
3	TRANSMITTER	Node is operating as transmitter

- **EP: Error Passive**

0: The MCAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected.

1: The MCAN is in the Error_Passive state.

- **EW: Warning Status**

0: Both error counters are below the Error_Warning limit of 96.

1: At least one of error counter has reached the Error_Warning limit of 96.

- **BO: Bus_Off Status**

0: The MCAN is not Bus_Off.

1: The MCAN is in Bus_Off state.

- **DLEC: Data Phase Last Error Code (set to 111 on read)**

Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

- **RESI: ESI Flag of Last Received CAN FD Message (cleared on read)**

This bit is set together with RFDF, independently from acceptance filtering.

0: Last received CAN FD message did not have its ESI flag set.

1: Last received CAN FD message had its ESI flag set.

- **RBRS: BRS Flag of Last Received CAN FD Message (cleared on read)**

This bit is set together with RFDF, independently from acceptance filtering.

0: Last received CAN FD message did not have its BRS flag set.

1: Last received CAN FD message had its BRS flag set.

- **RFDF: Received a CAN FD Message (cleared on read)**

This bit is set independently from acceptance filtering.

0: Since this bit was reset by the CPU, no CAN FD message has been received

1: Message in CAN FD format with FDF flag set has been received

- **PXE: Protocol Exception Event (cleared on read)**

0: No protocol exception event occurred since last read access

1: Protocol exception event occurred

- **TDCV: Transmitter Delay Compensation Value**

0 to 127: Position of the secondary sample point, in CAN core clock periods, defined by the sum of the measured delay from CANTX to CANRX and MCAN_TDCR.TDCO.

49.6.15 MCAN Transmitter Delay Compensation Register

Name: MCAN_TDCR

Address: 0x40030048 (0), 0x40034048 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	TDCO						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	TDCF						

- **TDCF: Transmitter Delay Compensation Filter**

0 to 127: defines the minimum value for the SSP position, in CAN core clock periods. Dominant edges on CANRX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO.

- **TDCO: Transmitter Delay Compensation Offset**

0 to 127: Offset value, in CAN core clock periods, defining the distance between the measured delay from CANTX to CANRX and the secondary sample point.

49.6.16 MCAN Interrupt Register

Name: MCAN_IR

Address: 0x40030050 (0), 0x40034050 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ARA	PED	PEA	WDI	BO	EW
23	22	21	20	19	18	17	16
EP	ELO	–	–	DRX	TOO	MRAF	TSW
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the processor clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. A hard reset will clear the register. The configuration of IE controls whether an interrupt is generated. The configuration of ILS controls on which interrupt line an interrupt is signalled.

- **RF0N: Receive FIFO 0 New Message**

0: No new message written to Receive FIFO 0.

1: New message written to Receive FIFO 0.

- **RF0W: Receive FIFO 0 Watermark Reached**

0: Receive FIFO 0 fill level below watermark.

1: Receive FIFO 0 fill level reached watermark.

- **RF0F: Receive FIFO 0 Full**

0: Receive FIFO 0 not full.

1: Receive FIFO 0 full.

- **RF0L: Receive FIFO 0 Message Lost**

0: No Receive FIFO 0 message lost.

1: Receive FIFO 0 message lost, also set after write attempt to Receive FIFO 0 of size zero.

- **RF1N: Receive FIFO 1 New Message**

0: No new message written to Receive FIFO 1.

1: New message written to Receive FIFO 1.

- **RF1W: Receive FIFO 1 Watermark Reached**

0: Receive FIFO 1 fill level below watermark.

1: Receive FIFO 1 fill level reached watermark.

- **RF1F: Receive FIFO 1 Full**

0: Receive FIFO 1 not full.

1: Receive FIFO 1 full.

- **RF1L: Receive FIFO 1 Message Lost**

0: No Receive FIFO 1 message lost.

1: Receive FIFO 1 message lost, also set after write attempt to Receive FIFO 1 of size zero.

- **HPM: High Priority Message**

0: No high priority message received.

1: High priority message received.

- **TC: Transmission Completed**

0: No transmission completed.

1: Transmission completed.

- **TCF: Transmission Cancellation Finished**

0: No transmission cancellation finished.

1: Transmission cancellation finished.

- **TFE: Tx FIFO Empty**

0: Tx FIFO non-empty.

1: Tx FIFO empty.

- **TEFN: Tx Event FIFO New Entry**

0: Tx Event FIFO unchanged.

1: Tx Handler wrote Tx Event FIFO element.

- **TEFW: Tx Event FIFO Watermark Reached**

0: Tx Event FIFO fill level below watermark.

1: Tx Event FIFO fill level reached watermark.

- **TEFF: Tx Event FIFO Full**

0: Tx Event FIFO not full.

1: Tx Event FIFO full.

- **TEFL: Tx Event FIFO Element Lost**

0: No Tx Event FIFO element lost.

1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

- **TSW: Timestamp Wraparound**

0: No timestamp counter wrap-around.

1: Timestamp counter wrapped around.

- **MRAF: Message RAM Access Failure**

The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.
- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Receive Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN is switched into Restricted Operation mode (see Section 49.5.1.5). To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

0: No Message RAM access failure occurred.

1: Message RAM access failure occurred.

- **TOO: Timeout Occurred**

0: No timeout.

1: Timeout reached.

- **DRX: Message stored to Dedicated Receive Buffer**

The flag is set whenever a received message has been stored into a dedicated Receive Buffer.

0: No Receive Buffer updated.

1: At least one received message stored into a Receive Buffer.

- **ELO: Error Logging Overflow**

0: CAN Error Logging Counter did not overflow.

1: Overflow of CAN Error Logging Counter occurred.

- **EP: Error Passive**

0: Error_Passive status unchanged.

1: Error_Passive status changed.

- **EW: Warning Status**

0: Error_Warning status unchanged.

1: Error_Warning status changed.

- **BO: Bus_Off Status**

0: Bus_Off status unchanged.

1: Bus_Off status changed.

- **WDI: Watchdog Interrupt**

0: No Message RAM Watchdog event occurred.

1: Message RAM Watchdog event due to missing READY.

- **PEA: Protocol Error in Arbitration Phase**

0: No protocol error in arbitration phase

1: Protocol error in arbitration phase detected (MCAN_PSR.LEC differs from 0 or 7)

- **PED: Protocol Error in Data Phase**

0: No protocol error in data phase

1: Protocol error in data phase detected (MCAN_PSR.DLEC differs from 0 or 7)

- **ARA: Access to Reserved Address**

0: No access to reserved address occurred

1: Access to reserved address occurred

49.6.17 MCAN Interrupt Enable Register

Name: MCAN_IE

Address: 0x40030054 (0), 0x40034054 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ARAE	PEDE	PEAE	WDIE	BOE	EWE
23	22	21	20	19	18	17	16
EPE	ELOE	–	–	DRXE	TOOE	MRAFE	TSWE
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

The following configuration values are valid for all listed bit names of this register:

0: Disables the corresponding interrupt.

1: Enables the corresponding interrupt.

- **RF0NE: Receive FIFO 0 New Message Interrupt Enable**

- **RF0WE: Receive FIFO 0 Watermark Reached Interrupt Enable**

- **RF0FE: Receive FIFO 0 Full Interrupt Enable**

- **RF0LE: Receive FIFO 0 Message Lost Interrupt Enable**

- **RF1NE: Receive FIFO 1 New Message Interrupt Enable**

- **RF1WE: Receive FIFO 1 Watermark Reached Interrupt Enable**

- **RF1FE: Receive FIFO 1 Full Interrupt Enable**

- **RF1LE: Receive FIFO 1 Message Lost Interrupt Enable**

- **HPME: High Priority Message Interrupt Enable**

- **TCE: Transmission Completed Interrupt Enable**

- **TCFE: Transmission Cancellation Finished Interrupt Enable**

- **TFEE: Tx FIFO Empty Interrupt Enable**

- **TEFNE: Tx Event FIFO New Entry Interrupt Enable**

- **TEFWE: Tx Event FIFO Watermark Reached Interrupt Enable**

- **TEFFE: Tx Event FIFO Full Interrupt Enable**

- **TEFLE: Tx Event FIFO Event Lost Interrupt Enable**

- **TSWE: Timestamp Wraparound Interrupt Enable**

- **MRAFE: Message RAM Access Failure Interrupt Enable**
- **TOOE: Timeout Occurred Interrupt Enable**
- **DRXE: Message stored to Dedicated Receive Buffer Interrupt Enable**
- **ELOE: Error Logging Overflow Interrupt Enable**
- **EPE: Error Passive Interrupt Enable**
- **EWE: Warning Status Interrupt Enable**
- **BOE: Bus_Off Status Interrupt Enable**
- **WDIE: Watchdog Interrupt Enable**
- **PEAE: Protocol Error in Arbitration Phase Enable**
- **PEDE: Protocol Error in Data Phase Enable**
- **ARAE: Access to Reserved Address Enable**

49.6.18 MCAN Interrupt Line Select Register

Name: MCAN_ILS

Address: 0x40030058 (0), 0x40034058 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	ARAL	PEDL	PEAL	WDIL	BOL	EWL
23	22	21	20	19	18	17	16
EPL	ELOL	–	–	DRXL	TOOL	MRAFL	TSWL
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines.

0: Interrupt assigned to interrupt line MCAN_INT0.

1: Interrupt assigned to interrupt line MCAN_INT1.

- **RF0NL: Receive FIFO 0 New Message Interrupt Line**

- **RF0WL: Receive FIFO 0 Watermark Reached Interrupt Line**

- **RF0FL: Receive FIFO 0 Full Interrupt Line**

- **RF0LL: Receive FIFO 0 Message Lost Interrupt Line**

- **RF1NL: Receive FIFO 1 New Message Interrupt Line**

- **RF1WL: Receive FIFO 1 Watermark Reached Interrupt Line**

- **RF1FL: Receive FIFO 1 Full Interrupt Line**

- **RF1LL: Receive FIFO 1 Message Lost Interrupt Line**

- **HPML: High Priority Message Interrupt Line**

- **TCL: Transmission Completed Interrupt Line**

- **TCFL: Transmission Cancellation Finished Interrupt Line**

- **TFEL: Tx FIFO Empty Interrupt Line**

- **TEFNL: Tx Event FIFO New Entry Interrupt Line**

- **TEFWL: Tx Event FIFO Watermark Reached Interrupt Line**

- **TEFFL: Tx Event FIFO Full Interrupt Line**

- **TEFLL: Tx Event FIFO Event Lost Interrupt Line**

- **TSWL: Timestamp Wraparound Interrupt Line**

- **MRAFL: Message RAM Access Failure Interrupt Line**
- **TOOL: Timeout Occurred Interrupt Line**
- **DRXL: Message stored to Dedicated Receive Buffer Interrupt Line**
- **ELOL: Error Logging Overflow Interrupt Line**
- **EPL: Error Passive Interrupt Line**
- **EWL: Warning Status Interrupt Line**
- **BOL: Bus_Off Status Interrupt Line**
- **WDIL: Watchdog Interrupt Line**
- **PEAL: Protocol Error in Arbitration Phase Line**
- **PEDL: Protocol Error in Data Phase Line**
- **ARAL: Access to Reserved Address Line**

49.6.19 MCAN Interrupt Line Enable

Name: MCAN_ILE

Address: 0x4003005C (0), 0x4003405C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	EINT1	EINT0

Each of the two interrupt lines to the processor can be enabled / disabled separately by programming bits EINT0 and EINT1.

- **EINT0: Enable Interrupt Line 0**

0: Interrupt line MCAN_INT0 disabled.

1: Interrupt line MCAN_INT0 enabled.

- **EINT1: Enable Interrupt Line 1**

0: Interrupt line MCAN_INT1 disabled.

1: Interrupt line MCAN_INT1 enabled.

49.6.20 MCAN Global Filter Configuration

Name: MCAN_GFC

Address: 0x40030080 (0), 0x40034080 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	–	ANFS		ANFE		RRFS	RRFE

Global settings for Message ID filtering. The Global Filter Configuration controls the filter path for standard and extended messages as described in Figure 49-5 and Figure 49-6.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **RRFE: Reject Remote Frames Extended**

0 (FILTER): Filter remote frames with 29-bit extended IDs.

1 (REJECT): Reject all remote frames with 29-bit extended IDs.

- **RRFS: Reject Remote Frames Standard**

0 (FILTER): Filter remote frames with 11-bit standard IDs.

1 (REJECT): Reject all remote frames with 11-bit standard IDs.

- **ANFE: Accept Non-matching Frames Extended**

Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RX_FIFO_0	Accept in Rx FIFO 0
1	RX_FIFO_1	Accept in Rx FIFO 1
2-3	REJECTED	Message rejected

- **ANFS: Accept Non-matching Frames Standard**

Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RX_FIFO_0	Accept in Rx FIFO 0
1	RX_FIFO_1	Accept in Rx FIFO 1
2-3	REJECTED	Message rejected

49.6.21 MCAN Standard ID Filter Configuration

Name: MCAN_SIDFC

Address: 0x40030084 (0), 0x40034084 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
LSS							
15	14	13	12	11	10	9	8
FLSSA							
7	6	5	4	3	2	1	0
FLSSA						–	–

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages as described in Figure 49-5.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **FLSSA: Filter List Standard Start Address**

Start address of standard Message ID filter list (32-bit word address, see Figure 49-12).

Write FLSSA with the bits [15:2] of the 32-bit address.

- **LSS: List Size Standard**

0: No standard Message ID filter.

1-128: Number of standard Message ID filter elements.

>128: Values greater than 128 are interpreted as 128.

49.6.22 MCAN Extended ID Filter Configuration

Name: MCAN_XIDFC

Address: 0x40030088 (0), 0x40034088 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	LSE						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FLESA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FLESA						–	–

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for standard messages as described in Figure 49-6.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **FLESA: Filter List Extended Start Address**

Start address of extended Message ID filter list (32-bit word address, see Figure 49-12).

Write FLESA with the bits [15:2] of the 32-bit address.

- **LSE: List Size Extended**

0: No extended Message ID filter.

1-64: Number of extended Message ID filter elements.

>64: Values greater than 64 are interpreted as 64.

49.6.23 MCAN Extended ID AND Mask

Name: MCAN_XIDAM

Address: 0x40030090 (0), 0x40034090 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	EIDM				
23	22	21	20	19	18	17	16
EIDM							
15	14	13	12	11	10	9	8
EIDM							
7	6	5	4	3	2	1	0
EIDM							

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **EIDM: Extended ID Mask**

For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

49.6.24 MCAN High Priority Message Status

Name: MCAN_HPMS

Address: 0x40030094 (0), 0x40034094 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FLST	FIDX						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MSI		BIDX					

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

- **BIDX: Buffer Index**

Index of Receive FIFO element to which the message was stored. Only valid when MSI[1] = ‘1’.

- **MSI: Message Storage Indicator**

Value	Name	Description
0	NO_FIFO_SEL	No FIFO selected.
1	LOST	FIFO message lost.
2	FIFO_0	Message stored in FIFO 0.
3	FIFO_1	Message stored in FIFO 1.

- **FIDX: Filter Index**

Index of matching filter element. Range is 0 to MCAN_SIDFC.LSS - 1 resp. MCAN_XIDFC.LSE - 1.

- **FLST: Filter List**

Indicates the filter list of the matching filter element.

0: Standard filter list

1: Extended filter list

49.6.25 MCAN New Data 1

Name: MCAN_NDAT1

Address: 0x40030098 (0), 0x40034098 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
23	22	21	20	19	18	17	16
ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
15	14	13	12	11	10	9	8
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
7	6	5	4	3	2	1	0
ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0

- **NDx: New Data**

The register holds the New Data flags of Receive Buffers 0 to 31. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

0: Receive Buffer not updated

1: Receive Buffer updated from new message

49.6.26 MCAN New Data 2

Name: MCAN_NDAT2

Address: 0x4003009C (0), 0x4003409C (1)

Access: Read/Write

31	30	29	28	27	26	25	24
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
23	22	21	20	19	18	17	16
ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
15	14	13	12	11	10	9	8
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
7	6	5	4	3	2	1	0
ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32

- **NDx: New Data**

The register holds the New Data flags of Receive Buffers 32 to 63. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

0: Receive Buffer not updated.

1: Receive Buffer updated from new message.

49.6.27 MCAN Receive FIFO 0 Configuration

Name: MCAN_RXF0C

Address: 0x400300A0 (0), 0x400340A0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
F0OM	F0WM						
23	22	21	20	19	18	17	16
–	F0S						
15	14	13	12	11	10	9	8
F0SA							
7	6	5	4	3	2	1	0
F0SA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **F0SA: Receive FIFO 0 Start Address**

Start address of Receive FIFO 0 in Message RAM (32-bit word address, see Figure 49-12).

Write F0SA with the bits [15:2] of the 32-bit address.

- **F0S: Receive FIFO 0 Size**

0: No Receive FIFO 0

1-64: Number of Receive FIFO 0 elements.

>64: Values greater than 64 are interpreted as 64.

The Receive FIFO 0 elements are indexed from 0 to F0S-1.

- **F0WM: Receive FIFO 0 Watermark**

0: Watermark interrupt disabled.

1-64: Level for Receive FIFO 0 watermark interrupt (MCAN_IR.RF0W).

>64: Watermark interrupt disabled.

- **F0OM: FIFO 0 Operation Mode**

FIFO 0 can be operated in Blocking or in Overwrite mode (see Section 49.5.4.2).

0: FIFO 0 Blocking mode.

1: FIFO 0 Overwrite mode.

49.6.28 MCAN Receive FIFO 0 Status

Name: MCAN_RXF0S

Address: 0x400300A4 (0), 0x400340A4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	RF0L	F0F
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	F0PI					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	F0GI					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	F0FL						

- **F0FL: Receive FIFO 0 Fill Level**

Number of elements stored in Receive FIFO 0, range 0 to 64.

- **F0GI: Receive FIFO 0 Get Index**

Receive FIFO 0 read index pointer, range 0 to 63.

- **F0PI: Receive FIFO 0 Put Index**

Receive FIFO 0 write index pointer, range 0 to 63.

- **F0F: Receive FIFO 0 Full**

0: Receive FIFO 0 not full.

1: Receive FIFO 0 full.

- **RF0L: Receive FIFO 0 Message Lost**

This bit is a copy of interrupt flag MCAN_IR.RF0L. When MCAN_IR.RF0L is reset, this bit is also reset.

0: No Receive FIFO 0 message lost

1: Receive FIFO 0 message lost, also set after write attempt to Receive FIFO 0 of size zero

Note: Overwriting the oldest message when MCAN_RXF0C.F0OM = ‘1’ will not set this flag.

49.6.29 MCAN Receive FIFO 0 Acknowledge

Name: MCAN_RXF0A

Address: 0x400300A8 (0), 0x400340A8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	–	F0AI					

- **F0AI: Receive FIFO 0 Acknowledge Index**

After the processor has read a message or a sequence of messages from Receive FIFO 0 it has to write the buffer index of the last element read from Receive FIFO 0 to F0AI. This will set the Receive FIFO 0 Get Index MCAN_RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level MCAN_RXF0S.F0FL.

49.6.30 MCAN Receive Buffer Configuration

Name: MCAN_RXBC

Address: 0x400300AC (0), 0x400340AC (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RBSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBSA						–	–

- **RBSA: Receive Buffer Start Address**

Configures the start address of the Receive Buffers section in the Message RAM (32-bit word address, see Figure 49-12). Also used to reference debug messages A,B,C.

Write RBSA with the bits [15:2] of the 32-bit address.

49.6.31 MCAN Receive FIFO 1 Configuration

Name: MCAN_RXF1C

Address: 0x400300B0 (0), 0x400340B0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
F1OM	F1WM						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	F1S						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
F1SA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F1SA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **F1SA: Receive FIFO 1 Start Address**

Start address of Receive FIFO 1 in Message RAM (32-bit word address, see Figure 49-12).

Write F1SA with the bits [15:2] of the 32-bit address.

- **F1S: Receive FIFO 1 Size**

0: No Receive FIFO 1

1-64: Number of elements in Receive FIFO 1.

>64: Values greater than 64 are interpreted as 64.

The elements in Receive FIFO 1 are indexed from 0 to F1S - 1.

- **F1WM: Receive FIFO 1 Watermark**

0: Watermark interrupt disabled

1-64: Level for Receive FIFO 1 watermark interrupt (MCAN_IR.RF1W).

>64: Watermark interrupt disabled.

- **F1OM: FIFO 1 Operation Mode**

FIFO 1 can be operated in Blocking or in Overwrite mode (see Section 49.5.4.2).

0: FIFO 1 Blocking mode.

1: FIFO 1 Overwrite mode.

49.6.32 MCAN Receive FIFO 1 Status

Name: MCAN_RXF1S

Address: 0x400300B4 (0), 0x400340B4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
DMS		–	–	–	–	RF1L	F1F
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	F1PI					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	F1GI					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	F1FL						

- **F1FL: Receive FIFO 1 Fill Level**

Number of elements stored in Receive FIFO 1, range 0 to 64.

- **F1GI: Receive FIFO 1 Get Index**

Receive FIFO 1 read index pointer, range 0 to 63.

- **F1PI: Receive FIFO 1 Put Index**

Receive FIFO 1 write index pointer, range 0 to 63.

- **F1F: Receive FIFO 1 Full**

0: Receive FIFO 1 not full.

1: Receive FIFO 1 full.

- **RF1L: Receive FIFO 1 Message Lost**

This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset.

0: No Receive FIFO 1 message lost.

1: Receive FIFO 1 message lost, also set after write attempt to Receive FIFO 1 of size zero.

Note: Overwriting the oldest message when MCAN_RXF1C.F1OM = ‘1’ will not set this flag.

- **DMS: Debug Message Status**

Value	Name	Description
0	IDLE	Idle state, wait for reception of debug messages, DMA request is cleared.
1	MSG_A	Debug message A received.
2	MSG_AB	Debug messages A, B received.
3	MSG_ABC	Debug messages A, B, C received, DMA request is set.

49.6.33 MCAN Receive FIFO 1 Acknowledge

Name: MCAN_RXF1A

Address: 0x400300B8 (0), 0x400340B8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
–	–	F1AI					

- **F1AI: Receive FIFO 1 Acknowledge Index**

After the processor has read a message or a sequence of messages from Receive FIFO 1 it has to write the buffer index of the last element read from Receive FIFO 1 to F1AI. This will set the Receive FIFO 1 Get Index MCAN_RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level MCAN_RXF1S.F1FL.

49.6.34 MCAN Receive Buffer / FIFO Element Size Configuration

Name: MCAN_RXESC

Address: 0x400300BC (0), 0x400340BC (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	RBDS		
7	6	5	4	3	2	1	0
–	F1DS			–	F0DS		

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Configures the number of data bytes belonging to a Receive Buffer / Receive FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

- **F0DS: Receive FIFO 0 Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

- **F1DS: Receive FIFO 1 Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

- **RBDS: Receive Buffer Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Receive Buffer or Receive FIFO, only the number of bytes as configured by MCAN_RXESC are stored to the Receive Buffer resp. Receive FIFO element. The rest of the frame's data field is ignored.

49.6.35 MCAN Tx Buffer Configuration

Name: MCAN_TXBC

Address: 0x400300C0 (0), 0x400340C0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	TFQM	TFQS					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	NDTB					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TBSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TBSA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **TBSA: Tx Buffers Start Address**

Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 49-12).

Write TBSA with the bits [15:2] of the 32-bit address.

- **NDTB: Number of Dedicated Transmit Buffers**

0: No dedicated Tx Buffers.

1-32: Number of dedicated Tx Buffers.

>32: Values greater than 32 are interpreted as 32.

- **TFQS: Transmit FIFO/Queue Size**

0: No Tx FIFO/Queue.

1-32: Number of Tx Buffers used for Tx FIFO/Queue.

>32: Values greater than 32 are interpreted as 32.

- **TFQM: Tx FIFO/Queue Mode**

0: Tx FIFO operation.

1: Tx Queue operation.

Note: The sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

49.6.36 MCAN Tx FIFO/Queue Status

Name: MCAN_TXFQS

Address: 0x400300C4 (0), 0x400340C4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	TFQF	TFQPI				
15	14	13	12	11	10	9	8
–	–	–	TFGI				
7	6	5	4	3	2	1	0
–	–	TFFL					

The Tx FIFO/Queue status is related to the pending Tx requests listed in register MCAN_TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (MCAN_TXBRP not yet updated).

- **TFFL: Tx FIFO Free Level**

Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (MCAN_TXBC.TFQM = ‘1’).

- **TFGI: Tx FIFO Get Index**

Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (MCAN_TXBC.TFQM = ‘1’).

- **TFQPI: Tx FIFO/Queue Put Index**

Tx FIFO/Queue write index pointer, range 0 to 31.

- **TFQF: Tx FIFO/Queue Full**

0: Tx FIFO/Queue not full.

1: Tx FIFO/Queue full.

Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers.
Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

49.6.37 MCAN Tx Buffer Element Size Configuration

Name: MCAN_TXESC

Address: 0x400300C8 (0), 0x400340C8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	TBDS		

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

- **TBDS: Tx Buffer Data Field Size**

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48- byte data field
7	64_BYTE	64-byte data field

Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size MCAN_TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

49.6.38 MCAN Transmit Buffer Request Pending

Name: MCAN_TXBRP

Address: 0x400300CC (0), 0x400340CC (1)

Access: Read-only

31	30	29	28	27	26	25	24
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
23	22	21	20	19	18	17	16
TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
15	14	13	12	11	10	9	8
TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
7	6	5	4	3	2	1	0
TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0

- **TRPx: Transmission Request Pending for Buffer x**

Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register MCAN_TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register MCAN_TXBCR.

TXBRP bits are set only for those Tx Buffers configured via MCAN_TXBC. After a MCAN_TXBRP bit has been set, a Tx scan (see Section 49.5.5) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).

A cancellation request resets the corresponding transmission request pending bit of register MCAN_TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.

After a cancellation has been requested, a finished cancellation is signalled via MCAN_TXBCF.

- after successful transmission together with the corresponding MCAN_TXBTO bit.
- when the transmission has not yet been started at the point of cancellation.
- when the transmission has been aborted due to lost arbitration.
- when an error occurred during frame transmission.

In DAR mode, all transmissions are automatically cancelled if they are not successful. The corresponding MCAN_TXBCF bit is set for all unsuccessful transmissions.

0: No transmission request pending

1: Transmission request pending

Note: MCAN_TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding MCAN_TXBRP bit is reset.

49.6.39 MCAN Transmit Buffer Add Request

Name: MCAN_TXBAR

Address: 0x400300D0 (0), 0x400340D0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
23	22	21	20	19	18	17	16
AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
15	14	13	12	11	10	9	8
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
7	6	5	4	3	2	1	0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

- **ARx: Add Request for Transmit Buffer x**

Each Transmit Buffer has its own Add Request bit. Writing a ‘1’ will set the corresponding Add Request bit; writing a ‘0’ has no impact. This enables the processor to set transmission requests for multiple Transmit Buffers with one write to MCAN_TXBAR. MCAN_TXBAR bits are set only for those Transmit Buffers configured via TXBC. When no Transmit scan is running, the bits are reset immediately, else the bits remain set until the Transmit scan process has completed.

0: No transmission request added.

1: Transmission requested added.

Note: If an add request is applied for a Transmit Buffer with pending transmission request (corresponding MCAN_TXBRP bit already set), this Add Request is ignored.

49.6.40 MCAN Transmit Buffer Cancellation Request

Name: MCAN_TXBCR

Address: 0x400300D4 (0), 0x400340D4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
23	22	21	20	19	18	17	16
CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
15	14	13	12	11	10	9	8
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
7	6	5	4	3	2	1	0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

- **CRx: Cancellation Request for Transmit Buffer x**

Each Transmit Buffer has its own Cancellation Request bit. Writing a ‘1’ will set the corresponding Cancellation Request bit; writing a ‘0’ has no impact. This enables the processor to set cancellation requests for multiple Transmit Buffers with one write to MCAN_TXBCR. MCAN_TXBCR bits are set only for those Transmit Buffers configured via TXBC. The bits remain set until the corresponding bit of MCAN_TXBRP is reset.

0: No cancellation pending.

1: Cancellation pending.

49.6.41 MCAN Transmit Buffer Transmission Occurred

Name: MCAN_TXBTO

Address: 0x400300D8 (0), 0x400340D8 (1)

Access: Read-only

31	30	29	28	27	26	25	24
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
23	22	21	20	19	18	17	16
TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
15	14	13	12	11	10	9	8
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
7	6	5	4	3	2	1	0
TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

- **TOx: Transmission Occurred for Buffer x**

Each Transmit Buffer has its own Transmission Occurred bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit of register MCAN_TXBAR.

0: No transmission occurred.

1: Transmission occurred.

49.6.42 MCAN Transmit Buffer Cancellation Finished

Name: MCAN_TXBCF

Address: 0x400300DC (0), 0x400340DC (1)

Access: Read-only

31	30	29	28	27	26	25	24
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
23	22	21	20	19	18	17	16
CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
15	14	13	12	11	10	9	8
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
7	6	5	4	3	2	1	0
CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0

- **CFx: Cancellation Finished for Transmit Buffer x**

Each Transmit Buffer has its own Cancellation Finished bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a cancellation was requested via MCAN_TXBCR. In case the corresponding MCAN_TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register MCAN_TXBAR.

0: No transmit buffer cancellation.

1: Transmit buffer cancellation finished.

49.6.43 MCAN Transmit Buffer Transmission Interrupt Enable

Name: MCAN_TXBTIE

Address: 0x400300E0 (0), 0x400340E0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
23	22	21	20	19	18	17	16
TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
15	14	13	12	11	10	9	8
TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
7	6	5	4	3	2	1	0
TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0

- **TIEx: Transmission Interrupt Enable for Buffer x**

Each Transmit Buffer has its own Transmission Interrupt Enable bit.

0: Transmission interrupt disabled

1: Transmission interrupt enable

49.6.44 MCAN Transmit Buffer Cancellation Finished Interrupt Enable

Name: MCAN_TXBCIE

Address: 0x400300E4 (0), 0x400340E4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
23	22	21	20	19	18	17	16
CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
15	14	13	12	11	10	9	8
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
7	6	5	4	3	2	1	0
CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0

- **CFIEx: Cancellation Finished Interrupt Enable for Transmit Buffer x**

Each Transmit Buffer has its own Cancellation Finished Interrupt Enable bit.

0: Cancellation finished interrupt disabled.

1: Cancellation finished interrupt enabled.

49.6.45 MCAN Transmit Event FIFO Configuration

Name: MCAN_TXEFC

Address: 0x400300F0 (0), 0x400340F0 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	EFWM					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	EFS					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EFSA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EFSA						–	–

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

- **EFSA: Event FIFO Start Address**

Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 49-12).

Write EFSA with the bits [15:2] of the 32-bit address.

- **EFS: Event FIFO Size**

0: Tx Event FIFO disabled.

1-32: Number of Tx Event FIFO elements.

>32: Values greater than 32 are interpreted as 32.

The Tx Event FIFO elements are indexed from 0 to EFS - 1.

- **EFWM: Event FIFO Watermark**

0: Watermark interrupt disabled.

1-32: Level for Tx Event FIFO watermark interrupt (MCAN_IR.TEFW).

>32: Watermark interrupt disabled.

49.6.46 MCAN Tx Event FIFO Status

Name: MCAN_TXEFS

Address: 0x400300F4 (0), 0x400340F4 (1)

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	TEFL	EFF
23	22	21	20	19	18	17	16
–	–	–	EFPI				
15	14	13	12	11	10	9	8
–	–	–	EFGI				
7	6	5	4	3	2	1	0
–	–	EFFL					

- **EFFL: Event FIFO Fill Level**

Number of elements stored in Tx Event FIFO, range 0 to 32.

- **EFGI: Event FIFO Get Index**

Tx Event FIFO read index pointer, range 0 to 31.

- **EFPI: Event FIFO Put Index**

Tx Event FIFO write index pointer, range 0 to 31.

- **EFF: Event FIFO Full**

0: Tx Event FIFO not full

1: Tx Event FIFO full

- **TEFL: Tx Event FIFO Element Lost**

This bit is a copy of interrupt flag MCAN_IR.TEFL. When MCAN_IR.TEFL is reset, this bit is also reset.

0: No Tx Event FIFO element lost

1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

49.6.47 MCAN Tx Event FIFO Acknowledge

Name: MCAN_TXEFA

Address: 0x400300F8 (0), 0x400340F8 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	EFAI				

- **EFAI: Event FIFO Acknowledge Index**

After the processor has read an element or a sequence of elements from the Tx Event FIFO, it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index MCAN_TXEFS.EFGI to EFAI + 1 and update the FIFO 0 Fill Level MCAN_TXEFS.EFFL.

50. Timer Counter (TC)

50.1 Description

A Timer Counter (TC) module includes three identical TC channels. The number of implemented TC modules is device-specific.

Each TC channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs and two multipurpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC embeds a quadrature decoder (QDEC) connected in front of the timers and driven by TIOA0, TIOB0 and TIOB1 inputs. When enabled, the QDEC performs the input lines filtering, decoding of quadrature signals and connects to the timers/counters in order to read the position and speed of the motor through the user interface.

The TC block has two global registers which act upon all TC channels:

- Block Control register (TC_BCR)—allows channels to be started simultaneously with the same instruction
- Block Mode register (TC_BMR)—defines the external clock inputs for each channel, allowing them to be chained

50.2 Embedded Characteristics

- Total of 12 Channels
- 16-bit Channel Size
- Wide Range of Functions Including:
  - Frequency measurement
  - Event counting
  - Interval measurement
  - Pulse generation
  - Delay timing
  - Pulse Width Modulation
  - Up/down capabilities
  - Quadrature decoder
  - 2-bit Gray up/down count for stepper motor
- Each Channel is User-Configurable and Contains:
  - Three external clock inputs
  - Five Internal clock inputs
  - Two multipurpose input/output signals acting as trigger event
  - Trigger/capture events can be directly synchronized by PWM signals
- Internal Interrupt Signal
- Read of the Capture Registers by the DMAC
- Compare Event Fault Generation for PWM
- Register Write Protection