# 32. CAN with Flexible Data-rate (CANFD)

## 32.1 Overview

The CAN with Flexible Data-rate (CANFD) supports the following functions:

- Multiple channel operation
- Gateway function
- CAN with Flexible Data-rate.*1

Note 1. This feature is not available in the classical CAN function.

The CANFD module has a flexible message buffer and FIFO structure that meet the requirements of various applications. It also provides test modes to achieve high testability of the module that can be useful for power-on testing.

This specification describes a 2 channel implementation of the CANFD module.

The CANFD mode is only available in certain products that support it.

### 32.1.1 CAN-FD Module

**Table 32.1 CAN-FD module specifications (1 of 2)**

| Parameter | Specifications |
|---|---|
| Communication | CAN functionality conforms to CAN-FD ISO 11898-1 (2015) |
| Gateway function | CAN 2.0 ↔ CAN 2.0<br>CAN 2.0 ↔ CAN-FD gateway (only 8-byte payload)*1<br>CAN-FD ↔ CAN-FD*1 |
| Data transfer rate | Up to 1 Mbps for arbitration phase and up to 8 Mbps for data phase, individually for each CAN channel |
| Operation frequency<br>Peripheral clock/APB clock | 50 MHz (PCLKB)<br>RAM clock: 100 MHz (PCLKA) |
| Data Link Layer (DLL) clock | Max ≤ 40 MHz |
| Input/Output pins | TX/RX |
| CAN channels | 2 channels |
| Selectable ID type | 11-bit Standard ID |
|  | 11-bit Standard ID + 18-bit Extended ID |
| Selectable frame type | Data frame (RTR = 0) (CAN and CAN-FD frames) |
|  | Remote frame (RTR = 1) (only CAN frames) |
| Variable data byte count for data frames | DLC range: 0 to F |
| Message buffer | Up to 16 × 2 reception message buffers, shared among all the CAN channels |
|  | 16 transmit message buffers per channel |
|  | 4 transmission queues per channel<br>Automatic message transfer into transmission queues supported |
| FIFO number | 8 reception FIFO buffers<br>Up to 3 × 2 FIFOs individually configurable as:<br>● Reception FIFO<br>● Transmission FIFO<br>● CAN-to-CAN Gateway FIFO |
| Automatic delay interval timer for transmission | The delay timer can be applied to:<br>● Transmission FIFO<br>● CAN-to-CAN Gateway FIFO |

**Table 32.1 CAN-FD module specifications (2 of 2)**

| Parameter | Specifications |
|---|---|
| Enhanced reception filtering | Support of 11 bits and 29 bits CAN identifier |
| | Programmable 29 bits CAN identifier acceptance filter mask for each entry |
| | Programmable gateway routing capability for each channel (up to 8 routing destinations) |
| | RTR and IDE masking |
| | Data Length Code (DLC) filter |
| | Message buffer payload overload protection |
| | Payload filter |
| | Updating Acceptance Filter List (AFL) entry during communication |
| General software support | Automatic label information added to receive message (for upper software layer support) |
| Timer | TX and RX Timestamp function |
| Power down function | Module start stop function for each CAN node (Channel and Global Sleep modes) |
| RAM | RAM ECC protected (2 bits error detection, 1-bit error correction) |
| Bus traffic measurement | CAN bus traffic measurement of each channel is possible |
| TrustZone Filter | One security attribution can be set, and the attribution of the two channels are the same |

Note 1. This feature is not available in the classical CAN function.

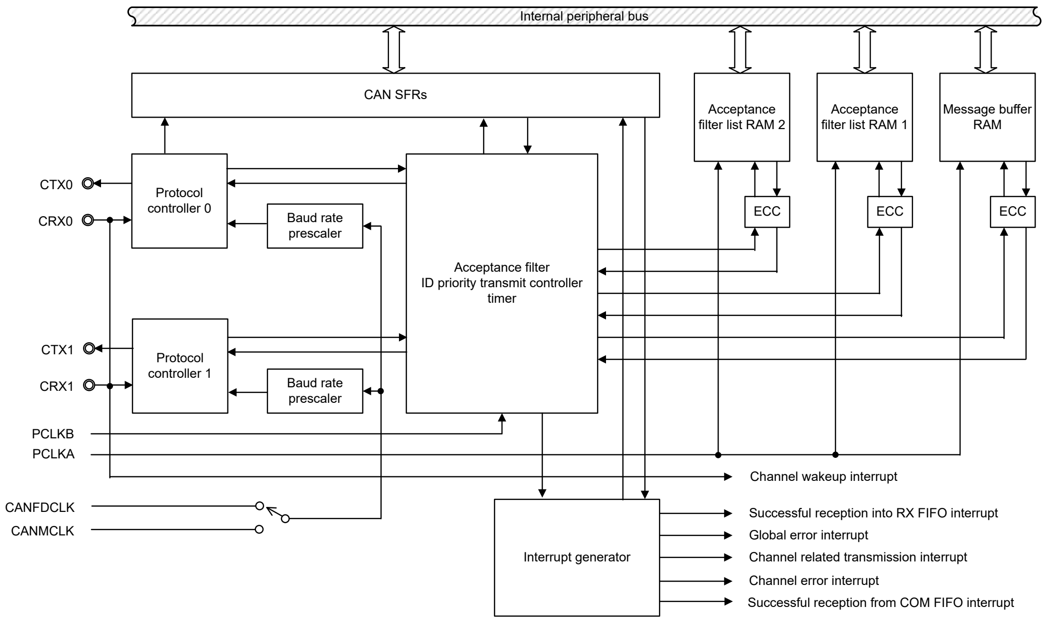


**Figure 32.1 Overview of the CAN-FD module**

- TX/RX:
  Input/Output, CRXn/CTXn, pins of the CAN module
- Protocol controller:
  Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling
- Acceptance filter list RAM:

This RAM is used to store the message acceptance filtering entries for all channels. Each acceptance filter entry has an individual ID, data length code, data field, message pointer for upper layer application usage and message direction pointer. The AFLRAM is divided in two parts to accelerate the Acceptance Filter List (AFL) access process.

- Message buffer RAM:
  This RAM is used to store messages after reception or for transmission using a normal message buffer or a FIFO. Each message entry has an individual ID, data length code, data field, message pointer for upper layer application usage and a time stamp.
- Acceptance filter:
  Performs filtering of received messages. The entries in the Acceptance filter list RAM are used for the filtering process.
- Two timers:
  - Reception Timestamp function
  - Transmission separation time for FIFO buffers
- Interrupt generator:
  Generates several types of global and channel interrupts
- CAN Special Function Registers (SFRs):
  Registers associated with CAN. See section 32.2.87. Message Buffer Component Structure.

## 32.2 Register Descriptions

### 32.2.1 Register Table

The reset value shown for the RAM area, consisting of CFDGAFLIDn, CFDGAFLMn, CFDGAFLP0n, CFDGAFLP1n, CFDRMBCPn, CFDRFMBCPn, CFDCFMBCPn, CFDTMBCPn, CFDTHLACC0n, CFDTHLACC1n and CFDRPGACCn is valid after initialization of a hardware reset. See section 32.4.2. CAN Module Configuration after Hardware Reset for details of the initialization process.

If a write access with a size of 8 or 16 bits is performed for the RAM area, then the CAN-FD module does a read-modify write-access to the RAM location, because the RAM requires a 32-bit access through the ECC module.

For single bit error, the correct data is written back. For multiple bit errors, unknown data is written back.

Do not access the space where the register is not assigned.

The read data from the space where the register is not assigned is unknown.

### 32.2.2 Legend

For all repetitive registers and bits, a lowercase index is used to indicate which slice is being referenced. If an index is being used, it is defined and described in the Register table it is being used in.

There is one global index used across all the registers and bits that need it.

**Table 32.2 CANFD Registers (1 of 5)**

| Register name | Symbol | Reset value | Offset Address | Access size |
|---|---|---|---|---|
| Channel n Nominal Bitrate Configuration Register n | CFDCnNCFG | 0x00000000 | 0x0000 + n × 0x0010 | 8, 16, 32 |
| Channel n Control Registers | CFDCnCTR | 0x00000005 | 0x0004 + n × 0x0010 | 8, 16, 32 |
| Channel n Status Registers | CFDCnSTS | 0x00000005 | 0x0008 + n × 0x0010 | 8, 16, 32 |
| Channel n Error Flag Registers | CFDCnERFL | 0x00000000 | 0x000C + n × 0x0010 | 8, 16, 32 |
| Global Configuration Register | CFDGCFG | 0x00000000 | 0x0084 | 8, 16, 32 |
| Global Control Register | CFDGCTR | 0x00000005 | 0x0088 | 8, 16, 32 |
| Global Status Register | CFDGSTS | 0x0000000D | 0x008C | 8, 16, 32 |
| Global Error Flag Register | CFDGERFL | 0x00000000 | 0x0090 | 8, 16, 32 |
| Global Timestamp Counter Register | CFDGTSC | 0x00000000 | 0x0094 | 16, 32 |

**Table 32.2 CANFD Registers (2 of 5)**

| Register name | Symbol | Reset value | Offset Address | Access size |
|---|---|---|---|---|
| Global Acceptance Filter List Entry Control Register | CFDGAFLECTR | 0x00000000 | 0x0098 | 8, 16, 32 |
| Global Acceptance Filter List Configuration Register n | CFDGAFLCFG0 | 0x00000000 | 0x009C | 8, 16, 32 |
| RX Message Buffer Number Register | CFDRMNB | 0x00000000 | 0x00AC | 8, 16, 32 |
| RX Message Buffer New Data Register n | CFDRMNDn | 0x00000000 | 0x00B0 + n × 0x0004 | 8, 16, 32 |
| RX FIFO Configuration/Control Registers n | CFDRFCCn | 0x00000000 | 0x00C0 + n × 0x0004 | 8, 16, 32 |
| RX FIFO Status Registers n | CFDRFSTSn | 0x00000001 | 0x00E0 + n × 0x0004 | 8, 16, 32 |
| RX FIFO Pointer Control Registers n | CFDRFPCTRn | 0x00000000 | 0x0100 + n × 0x0004 | 8, 16, 32 |
| Common FIFO Configuration/Control Registers n | CFDCFCCn | 0x00000000 | 0x0120 + n × 0x0004 | 8, 16, 32 |
| Common FIFO Configuration/Control Enhancement Registers n | CFDCFCCEn | 0x00000000 | 0x0180 + n × 0x0004 | 8, 16, 32 |
| Common FIFO Status Registers n | CFDCFSTSn | 0x00000001 | 0x01E0 + n × 0x0004 | 8, 16, 32 |
| Common FIFO Pointer Control Registers n | CFDCFPCTRn | 0x00000000 | 0x0240 + n × 0x0004 | 8, 16, 32 |
| FIFO Empty Status Register | CFDFESTS | 0x00003FFF | 0x02A0 | 8, 16, 32 |
| FIFO Full Status Register | CFDFFSTS | 0x00000000 | 0x02A4 | 8, 16, 32 |
| FIFO Message Lost Status Register | CFDFMSTS | 0x00000000 | 0x02A8 | 8, 16, 32 |
| RX FIFO Interrupt Flag Status Register | CFDRFISTS | 0x00000000 | 0x02AC | 8, 16, 32 |
| Common FIFO RX Interrupt Flag Status Register | CFDCFRISTS | 0x00000000 | 0x02B0 | 8, 16, 32 |
| Common FIFO TX Interrupt Flag Status Register | CFDCFTISTS | 0x00000000 | 0x02B4 | 8, 16, 32 |
| Common FIFO One Frame RX Interrupt Flag Status Register | CFDCFOFRISTS | 0x00000000 | 0x02B8 | 8, 16, 32 |
| Common FIFO One Frame TX Interrupt Flag Status Register | CFDCFOFTISTS | 0x00000000 | 0x02BC | 8, 16, 32 |
| Common FIFO Message Over Write Status Register | CFDCFMOWSTS | 0x00000000 | 0x02C0 | 8, 16, 32 |
| FIFO FDC Full Status Register | CFDFFFSTS | 0x00000000 | 0x02C4 | 8, 16, 32 |
| TX Message Buffer Control Registers n | CFDTMCn | 0x00 | 0x02D0 + n × 0x0001 | 8 |
| TX Message Buffer Status Registers n | CFDTMSTSn | 0x00 | 0x07D0 + n × 0x0001 | 8 |
| TX Message Buffer Transmission Request Status Register f | CFDTMTRSTSf | 0x00000000 | 0x0CD0 + f × 0x0004 | 8, 16, 32 |
| TX Message Buffer Transmission Abort Request Status Register f | CFDTMTARSTSf | 0x00000000 | 0x0D70 + f × 0x0004 | 8, 16, 32 |
| TX Message Buffer Transmission Completion Status Register f | CFDTMTCSTSf | 0x00000000 | 0x0E10 + f × 0x0004 | 8, 16, 32 |
| TX Message Buffer Transmission Abort Status Register f | CFDTMTASTSf | 0x00000000 | 0x0EB0 + f × 0x0004 | 8, 16, 32 |
| TX Message Buffer Interrupt Enable Configuration Register f | CFDTMIECf | 0x00000000 | 0x0F50 + f × 0x0004 | 8, 16, 32 |
| TX Queue Configuration/Control Registers 0 [n] | CFDTXQCC0n | 0x00000000 | 0x1000 + n × 0x0004 | 8, 16, 32 |
| TX Queue Status Registers 0 [n] | CFDTXQSTS0n | 0x00000001 | 0x1020 + n × 0x0004 | 8, 16, 32 |
| TX Queue Pointer Control Registers 0 [n] | CFDTXQPCTR0n | 0x00000000 | 0x1040 + n × 0x0004 | 8, 16, 32 |

**Table 32.2 CANFD Registers (3 of 5)**

| Register name | Symbol | Reset value | Offset Address | Access size |
|---|---|---|---|---|
| TX Queue Configuration/Control Registers 1 [n] | CFDTXQCC1n | 0x00000000 | 0x1060 + n × 0x0004 | 8, 16, 32 |
| TX Queue Status Registers 1 [n] | CFDTXQSTS1[n] | 0x00000001 | 0x1080 + n × 0x0004 | 8, 16, 32 |
| TX Queue Pointer Control Registers 1 [n] | CFDTXQPCTR1n | 0x00000000 | 0x10A0 + n × 0x0004 | 8, 16, 32 |
| TX Queue Configuration/Control Registers 2 [n] | CFDTXQCC2n | 0x00000000 | 0x10C0 + n × 0x0004 | 8, 16, 32 |
| TX Queue Status Registers 2 [n] | CFDTXQSTS2n | 0x00000001 | 0x10E0 + n × 0x0004 | 8, 16, 32 |
| TX Queue Pointer Control Registers 2 [n] | CFDTXQPCTR2n | 0x00000000 | 0x1100 + n × 0x0004 | 8, 16, 32 |
| TX Queue Configuration/Control Registers 3 [n] | CFDTXQCC3n | 0x00000000 | 0x1120 + n × 0x0004 | 8, 16, 32 |
| TX Queue Status Registers 3 [n] | CFDTXQSTS3n | 0x00000001 | 0x1140 + n × 0x0004 | 8, 16, 32 |
| TX Queue Pointer Control Registers 3 [n] | CFDTXQPCTR3n | 0x00000000 | 0x1160 + n × 0x0004 | 8, 16, 32 |
| TX Queue Empty Status Register | CFDTXQESTS | 0x000000FF | 0x1180 | 8, 16, 32 |
| TX Queue Full Interrupt Status Register | CFDTXQFISTS | 0x00000000 | 0x1184 | 8, 16, 32 |
| TX Queue Message Lost Status Register | CFDTXQMSTS | 0x00000000 | 0x1188 | 8, 16, 32 |
| TX Queue Interrupt Status Register | CFDTXQISTS | 0x00000000 | 0x1190 | 8, 16, 32 |
| TX Queue One Frame TX Interrupt Status Register | CFDTXQOFTISTS | 0x00000000 | 0x1194 | 8, 16, 32 |
| TX Queue One Frame RX Interrupt Status Register | CFDTXQOFRISTS | 0x00000000 | 0x1198 | 8, 16, 32 |
| TX Queue Full Status Register | CFDTXQFSTS | 0x00000000 | 0x119C | 8, 16, 32 |
| TX History List Configuration/Control Register n | CFDTHLCCn | 0x00000000 | 0x1200 + n × 0x0004 | 8, 16, 32 |
| TX History List Status Register n | CFDTHLSTSn | 0x00000001 | 0x1220 + n × 0x0004 | 8, 16, 32 |
| TX History List Pointer Control Registers n | CFDTHLPCTRn | 0x00000000 | 0x1240 + n × 0x0004 | 8, 16, 32 |
| Global TX Interrupt Status Register 0 | CFDGTINTSTS0 | 0x00000000 | 0x1300 | 8, 16, 32 |
| Global Test Configuration Register | CFDGTSTCFG | 0x00000000 | 0x1308 | 8, 16, 32 |
| Global Test Control Register | CFDGTSTCTR | 0x00000000 | 0x130C | 8, 16, 32 |
| Global FD Configuration Register | CFDGFDCFG | 0x00000000 | 0x1314 | 8, 16, 32 |
| Global FD CRC Configuration Register | CFDGCRCCFG | 0x00000000 | 0x1318 | 8, 16, 32 |
| Global Lock Key Register | CFDGLOCKK | 0x00000000 | 0x131C | 16, 32 |
| Global AFL Ignore Entry Register | CFDGAFLIGNENT | 0x00000000 | 0x1324 | 8, 16, 32 |
| Global AFL Ignore Control Register | CFDGAFLIGNCTR | 0x00000000 | 0x1328 | 16, 32 |
| DMA Transfer Control Register | CFDCDTCT | 0x00000000 | 0x1330 | 8, 16, 32 |
| DMA Transfer Status Register | CFDCDTSTS | 0x00000000 | 0x1334 | 8, 16, 32 |
| DMA TX Transfer Control Register | CFDCDTTCT | 0x00000000 | 0x1340 | 8, 16, 32 |
| DMA TX Transfer Status Register | CFDCDTTSTS | 0x00000000 | 0x1344 | 8, 16, 32 |
| Global RX Interrupt Status Register n | CFDGRINTSTSn | 0x00000000 | 0x1350 + n × 0x0004 | 8, 16, 32 |
| Global SW Reset Register | CFDGRSTC | 0x00000000 | 0x1380 | 16, 32 |
| Channel n Data Bitrate Configuration Register*2 | CFDCnDCFG | 0x00000000 | 0x1400 + n × 0x0020 | 8, 16, 32 |
| Channel n CAN-FD Configuration Register | CFDCnFDCFG | 0x00000000 | 0x1404 + n × 0x0020 | 8, 16, 32 |
| Channel n CAN-FD Control Register | CFDCnFDCTR | 0x00000000 | 0x1408 + n × 0x0020 | 8, 16, 32 |

**Table 32.2 CANFD Registers (4 of 5)**

| Register name | Symbol | Reset value | Offset Address | Access size |
|---|---|---|---|---|
| Channel n CAN-FD Status Register | CFDCnFDSTS | 0x00000000 | 0x140C + n × 0x0020 | 8, 16, 32 |
| Channel n CAN-FD CRC Register*2 | CFDCnFDCRC | 0x00000000 | 0x1410 + n × 0x0020 | 8, 16, 32 |
| Channel n Bus Load Control Register | CFDCnBLCT | 0x00000000 | 0x1418 + n × 0x0020 | 8, 16, 32 |
| Channel n Bus Load Status Register | CFDCnBLSTS | 0x00000000 | 0x141C + n × 0x0020 | 8, 16, 32 |
| Global Acceptance Filter List ID Registers n | CFDGAFLIDn | 0x00000000*1 | 0x1800 + (n - 1) × 0x0010 | 8, 16, 32 |
| Global Acceptance Filter List Mask Registers n | CFDGAFLMn | 0x00000000*1 | 0x1804 + (n - 1) × 0x0010 | 8, 16, 32 |
| Global Acceptance Filter List Pointer 0 Registers n | CFDGAFLP0n | 0x00000000*1 | 0x1808 + (n - 1) × 0x0010 | 8, 16, 32 |
| Global Acceptance Filter List Pointer 1 Registers n | CFDGAFLP1n | 0x00000000*1 | 0x180C + (n - 1) × 0x0010 | 8, 16, 32 |
| Channel n TX History List Access Registers 0 | CFDTHLACC0n | 0x00000000*1 | 0x8000 + n × 0x0008 | 8, 16, 32 |
| Channel n TX History List Access Registers 1 | CFDTHLACC1n | 0x00000000*1 | 0x8004 + n × 0x0008 | 8, 16, 32 |
| RAM Test Page Access Registers n | CFDRPGACCn | 0x00000000*1 | 0x8400 + n × 0x0004 | 8, 16, 32 |
| RX FIFO Access ID Register n | CFDRFIDn | 0x00000000*1 | 0x6000 + n × 0x080 | 8, 16, 32 |
| RX FIFO Access Pointer Register n | CFDRFPTRn | 0x00000000*1 | 0x6004 + n × 0x080 | 8, 16, 32 |
| RX FIFO Access CAN-FD Status Register n | CFDRFFDSTSn | 0x00000000*1 | 0x6008 + n × 0x080 | 8, 16, 32 |
| RX FIFO Access Data Field p Register n | CFDRFDFpn | 0x00000000*1 | 0x600C + p × 0x004 + n × 0x080 | 8, 16, 32 |
| Common FIFO Access ID Register n Channel i | CFDCFIDn_i | 0x00000000*1 | 0x6400 + n × 0x080 + i × 0x180 | 8, 16, 32 |
| Common FIFO Access Pointer Register n Channel i | CFDCFPTRn_i | 0x00000000*1 | 0x6404 + n × 0x080 + i × 0x180 | 8, 16, 32 |
| Common FIFO Access CAN-FD Control/ Status Register n Channel i | CFDCFFDCSTSn_i | 0x00000000*1 | 0x6408 + n × 0x080 + i × 0x180 | 8, 16, 32 |
| Common FIFO Access Data Field p Register n Channel i | CFDCFDFpn | 0x00000000*1 | 0x640C + p × 0x004 + n × 0x080 + i × 0x180 | 8, 16, 32 |
| RX Message Buffer ID Registers | CFDRMID | 0x00000000*1 | See section 32.2.87.2. CFDRMIDn_i : RX Message Buffer ID Register n Channel i (n = 0 to 15, i = 0, 1) | 8, 16, 32 |
| RX Message Buffer Pointer Registers | CFDRMPTR | 0x00000000*1 | See section 32.2.87.3. CFDRMPTRn_i : RX Message Buffer Pointer Register n Channel i (n = 0 to 15, i = 0, 1) | 8, 16, 32 |
| RX Message Buffer CAN-FD Status Register | CFDRMFDSTS | 0x00000000*1 | See section 32.2.87.4. CFDRMFDSTSn_i : RX Message Buffer CAN-FD Status Register n Channel i (n = 0 to 15, i = 0, 1) | 8, 16, 32 |
| RX Message Buffer Data Field p Registers Channel i | CFDRMDFp_n_i | 0x00000000*1 | See section 32.2.87.5. CFDRMDFp_n_i : RX Message Buffer Data Field p Register n Channel i (p = 0 to 15, n = 0 to 15, i = 0, 1) | 8, 16, 32 |
| TX Message Buffer ID Registers | CFDTMID | 0x00000000*1 | See section 32.2.87.14. CFDTMIDn_i : TX Message Buffer ID Register n Channel i (n = 0 to 7, 32 to 39, i = 0, 1) | 8, 16, 32 |

**Table 32.2 CANFD Registers (5 of 5)**

| Register name | Symbol | Reset value | Offset Address | Access size |
|---|---|---|---|---|
| TX Message Buffer Pointer Registers | CFDTMPTR | 0x00000000[*1] | See section 32.2.87.15. CFDTMPTRn_i : TX Message Buffer Pointer Register n Channel i (n = 0 to 7, 32 to 39, i = 0, 1) | 8, 16, 32 |
| TX Message Buffer CAN-FD Control Register | CFDTMFDCTR | 0x00000000[*1] | See section 32.2.87.16. CFDTMFDCTRn_i : TX Message Buffer CANFD Control Register n Channel i (n = 0 to 7, 32 to 39, i = 0, 1) | 8, 16, 32 |
| TX Message Buffer Data Field p Registers n Channel i | CFDTMDFp_n_i | 0x00000000[*1] | See section 32.2.87.17. CFDTMDFp_n_i : TX Message Buffer Data Field p Register n Channel i (p = 0 to 15, n = 0 to 7, 32 to 39, i = 0, 1) | 8, 16, 32 |

Note: For the range of subscripts (n, f, i, p), refer to the explanation of each register.

Note 1. The RAM area is initialized after a hardware reset, see section 32.4.2. CAN Module Configuration after Hardware Reset.

Note 2. These registers are not available in the classical CAN function.

### 32.2.3 CFDCnNCFG : Channel n Nominal Bitrate Configuration Register (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x0000 + 0x10 × n

| Bit position: | 31 25 | 24 17 | 16 10 | 9 0 |
|---|---|---|---|---|
| Bit field: | NTSEG2[6:0] | NTSEG1[7:0] | NSJW[6:0] | NBRP[9:0] |
| Value after reset: | 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 9:0 | NBRP[9:0] | Channel Nominal Baud Rate Prescaler<br>Nominal baud rate prescaler division ratio | R/W |
| 16:10 | NSJW[6:0] | Resynchronization Jump Width<br>0x00: 1 Tq<br>0x01: 2 Tq<br>⋮<br>0x7E: 127 Tq<br>0x7F: 128 Tq | R/W |
| 24:17 | NTSEG1[7:0] | Timing Segment 1<br>0x00: Reserved<br>0x01: 2 Tq<br>0x02: 3 Tq<br>0x03: 4 Tq<br>⋮<br>0xFE: 255 Tq<br>0xFF: 256 Tq | R/W |
| 31:25 | NTSEG2[6:0] | Timing Segment 2<br>0x00: Reserved<br>0x01: 2 Tq<br>⋮<br>0x7E: 127 Tq<br>0x7F: 128 Tq | R/W |

Note: Tq means time quantum.

This register configures the transmission/reception nominal baud rate parameters of the channels.

#### NBRP[9:0] bits (Channel Nominal Baud Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

### NSJW[6:0] bits (Resynchronization Jump Width)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

### NTSEG1[7:0] bits (Timing Segment 1)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 256, inclusive. See section 32.4.1.2. CAN Bit Timing for more details.

### NTSEG2[6:0] bits (Timing Segment 2)

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 128, inclusive.

## 32.2.4 CFDCnCTR : Channel n Control Registers (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x0004 + 0x10 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | ROM | CRCT | — | — | — | CTMS[1:0] | | CTME | ERRD | BOM[1:0] | | — | TDCV FIE | SOCO IE | EOCO IE | TAIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | ALIE | BLIE | OLIE | BORIE | BOEIE | EPIE | EWIE | BEIE | — | — | — | — | RTBO | CSLP R | CHMDC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 1:0 | CHMDC[1:0] | Channel Mode Control<br>0 0: Channel operation mode request<br>0 1: Channel reset request<br>1 0: Channel halt request<br>1 1: Keep current value | R/W |
| 2 | CSLPR | Channel Sleep Request<br>0: Channel sleep request disabled<br>1: Channel sleep request enabled | R/W |
| 3 | RTBO | Return from Bus-Off<br>0: Channel is not forced to return from bus-off<br>1: Channel is forced to return from bus-off | R/W |
| 7:4 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | BEIE | Bus Error Interrupt Enable<br>0: Bus error interrupt disabled<br>1: Bus error interrupt enabled | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 9 | EWIE | Error Warning Interrupt Enable<br>0: Error warning interrupt disabled<br>1: Error warning interrupt enabled | R/W |
| 10 | EPIE | Error Passive Interrupt Enable<br>0: Error passive interrupt disabled<br>1: Error passive interrupt enabled | R/W |
| 11 | BOEIE | Bus-Off Entry Interrupt Enable<br>0: Bus-off entry interrupt disabled<br>1: Bus-off entry interrupt enabled | R/W |
| 12 | BORIE | Bus-Off Recovery Interrupt Enable<br>0: Bus-off recovery interrupt disabled<br>1: Bus-off recovery interrupt enabled | R/W |
| 13 | OLIE | Overload Interrupt Enable<br>0: Overload interrupt disabled<br>1: Overload interrupt enabled | R/W |
| 14 | BLIE | Bus Lock Interrupt Enable<br>0: Bus lock interrupt disabled<br>1: Bus lock interrupt enabled | R/W |
| 15 | ALIE | Arbitration Lost Interrupt Enable<br>0: Arbitration lost interrupt disabled<br>1: Arbitration lost interrupt enabled | R/W |
| 16 | TAIE | Transmission Abort Interrupt Enable<br>0: TX abort interrupt disabled<br>1: TX abort interrupt enabled | R/W |
| 17 | EOCOIE | Error Occurrence Counter Overflow Interrupt Enable<br>0: Error occurrence counter overflow interrupt disabled<br>1: Error occurrence counter overflow interrupt enabled | R/W |
| 18 | SOCOIE | Successful Occurrence Counter Overflow Interrupt Enable<br>0: Successful occurrence counter overflow interrupt disabled<br>1: Successful occurrence counter overflow interrupt enabled | R/W |
| 19 | TDCVFIE*1 | Transceiver Delay Compensation Violation Interrupt Enable<br>0: Transceiver delay compensation violation interrupt disabled<br>1: Transceiver delay compensation violation interrupt enabled | R/W |
| 20 | — | This bit is read as 0. The write value should be 0. | R/W |
| 22:21 | BOM[1:0] | Channel Bus-Off Mode<br>0 0: Normal mode (comply with ISO 11898-1)<br>0 1: Entry to Halt mode automatically at bus-off start<br>1 0: Entry to Halt mode automatically at bus-off end<br>1 1: Entry to Halt mode (during bus-off recovery period) by software | R/W |
| 23 | ERRD | Channel Error Display<br>0: Only the first set of error codes displayed<br>1: Accumulated error codes displayed | R/W |
| 24 | CTME | Channel Test Mode Enable<br>0: Channel test mode disabled<br>1: Channel test mode enabled | R/W |
| 26:25 | CTMS[1:0] | Channel Test Mode Select<br>0 0: Basic test mode<br>0 1: Listen-only mode<br>1 0: Self-test mode 0 (External loopback mode)<br>1 1: Self-test mode 1 (Internal loopback mode) | R/W |
| 29:27 | — | These bits are read as 0. The write value should be 0. | R/W |
| 30 | CRCT | CRC Error Test<br>0: First data bit of reception stream not inverted<br>1: First data bit of reception stream inverted | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 31 | ROM[*1] | Restricted Operation Mode<br>0: Restricted operation mode disabled<br>1: Restricted operation mode enabled | R/W |

Note 1. These bits are not available in the classical CAN function.

Each Channel Control register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

### CHMDC[1:0] bits (Channel Mode Control)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in section 32.3.3. Channel Modes.

Setting CHMDC[1:0] bits to 11b has no effect. When the CANFD module is in GL_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDCnCTR.BOM settings.

If CPU write access to CFDCnCTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM = 01b, or at the end of bus-off when CFDCnCTR.BOM = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDCnCTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDCnCTR.CHMDC value is 00b (Operation mode).

### CSLPR bit (Channel Sleep Request)

When the CSLPR bit is 1, a Sleep mode request is generated for the corresponding CAN channel

When this bit is 0, a request to exit Sleep mode is generated for the related CANFD channel.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

### RTBO bit (Return from Bus-Off)

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDCnSTS.BOSTS) is set to 0.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDCnCTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0.

Return from the Bus-Off command should be used only when CFDCnCTR.BOM is set to 00b.

Only write to this bit when the related CANFD channel is in CH_OPERATION mode. This bit is automatically cleared when set by software.

### BEIE bit (Bus Error Interrupt Enable)

When the BEIE and the CFDCnERFL.BEF bits are both 1, an error interrupt request is generated.

This bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### EWIE bit (Error Warning Interrupt Enable)

When the EWIE and the CFDCnERFL.EWF bits are both 1, an error interrupt request is generated.

The EWIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### EPIE bit (Error Passive Interrupt Enable)

An error interrupt request is generated when the EPIE bit and the CFDCnERFL.EPF are both 1.

The EPIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE and the CFDCnERFL.BOEF bits are both 1, an error interrupt request is generated.

The BOEIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE and the CFDCnERFL.BORF bits are both 1, an error interrupt request is generated.

The BORIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### OLIE bit (Overload Interrupt Enable)

When the OLIE and the CFDCnERFL.OVLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### BLIE bit (Bus Lock Interrupt Enable)

When the BLIE and the CFDCnERFL.BLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### ALIE bit (Arbitration Lost Interrupt Enable)

When the ALIE and the CFDCnERFL.ALF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### TAIE bit (Transmission Abort Interrupt Enable)

When the TAIE bit is 1 and a transmission is successfully aborted from a TX MB belonging to the corresponding CAN channel, an interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)

When the EOCOIE bit is 1 and the CFDCnFDSTS.EOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)

When the SOCOIE bit is 1 and the CFDCnFDSTS.SOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

### TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the TDCVFIE bit is 1 and the CFDCnFDSTS.TDCVF bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

### BOM[1:0] bits (Channel Bus-Off Mode)

The BOM[1:0]bits control the timing of the recovery from Bus-Off mode of the CANFD Channel.

Do not write to these bits in CH_SLEEP mode. Only write to these bits when the related CANFD channel is in CH_RESET mode.

Only write to these bits when the related CANFD channel is in CH_RESET mode.

### ERRD bit (Channel Error Display)

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDCnERFL).

If the ERRD bit is 0 and more than one error occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until CFDCnERFL[14:8] is cleared.

Do not write to the ERRD bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

### CTME bit (Channel Test Mode Enable)

The CTME bit enables the channel test modes.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

### CTMS[1:0] bits (Channel Test Mode Select)

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH_SLEEP or CH_RESET mode. Only write to these bits when the related CANFD channel is in CH_HALT mode.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

### CRCT bit (CRC Error Test)

The CRCT bit checks the internal CRC generator logic of the protocol controller.

The internal generated CRC value is always observed in the following registers:

- CFDCnERFL.CRCREG (Classical CAN frames)
- CFDCnFDCRC.CRCREG (CANFD frames).[*1]

Note 1. This feature is not available in the classical CAN function.

Some restriction exist when using this bit:

- It is not possible to use this feature with CAN nodes connected to the MCU externally, only with nodes connected to the internal CAN bus communication can be used
- One CAN node can send a reference message and the receiver node can invert one bit of the incoming bit stream.

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing.

The CRC Error test mode is enabled if the CRCT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1.

Do not write to the CRCT bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

### ROM bit (Restricted Operation Mode)

When the ROM and CTME bits are both 1, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDCnCTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

### 32.2.5 CFDCnSTS : Channel n Status Registers (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x0008 + 0x10 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | TEC[7:0] | | | | | | | | REC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | ESIF | COMSTS | RECSTS | TRMSTS | BOSTS | EPSTS | CSLPSTS | CHLTSTS | CRSTSTS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | CRSTSTS | Channel Reset Status<br>0: Channel not in Reset mode<br>1: Channel in Reset mode | R |
| 1 | CHLTSTS | Channel Halt Status<br>0: Channel not in Halt mode<br>1: Channel in Halt mode | R |
| 2 | CSLPSTS | Channel Sleep Status<br>0: Channel not in Sleep mode<br>1: Channel in Sleep mode | R |
| 3 | EPSTS | Channel Error Passive Status<br>0: Channel not in error passive state<br>1: Channel in error passive state | R |
| 4 | BOSTS | Channel Bus-Off Status<br>0: Channel not in bus-off state<br>1: Channel in bus-off state | R |
| 5 | TRMSTS | Channel Transmit Status<br>0: Channel is not transmitting<br>1: Channel is transmitting | R |
| 6 | RECSTS | Channel Receive Status<br>0: Channel is not receiving<br>1: Channel is receiving | R |
| 7 | COMSTS | Channel Communication Status<br>0: Channel is not ready for communication<br>1: Channel is ready for communication | R |
| 8 | ESIF[*1] | Error State Indication Flag<br>0: No CANFD message has been received when the ESI flag was set<br>1: At least one CANFD message was received when the ESI flag was set | R/W |
| 15:9 | — | These bits are read as 0. The write value should be 0. | R/W |
| 23:16 | REC[7:0] | Reception Error Count<br>These bits increment or decrement the counter value according to error status of the CAN channel during reception. | R |
| 31:24 | TEC[7:0] | Transmission Error Count<br>These bits increment or decrement the counter value according to error status of the CAN channel during transmission. | R |

Note 1. This bit is not available in the classical CAN function.

Each Channel Status Register shows the mode, error and transmission or reception status of the related channel together with its reception and transmission error count values.

### CRSTSTS bit (Channel Reset Status)

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

### CHLTSTS bit (Channel Halt Status)

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

### CSLPSTS bit (Channel Sleep Status)

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CANFD channel enters Sleep mode, and is cleared automatically when the related CANFD channel exits Sleep mode.

### EPSTS bit (Channel Error Passive Status)

The EPSTS bit indicates whether the related CANFD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 0x7F.

This bit is cleared automatically when the related CANFD channel exits the error passive state or enters Reset mode.

### BOSTS bit (Channel Bus-Off Status)

The BOSTS bit indicates whether the related CANFD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 0xFF and the related CANFD channel is in the bus-off state (CAN Transmission Error Count Register > 0xFF).

This bit is cleared automatically when the related CANFD channel exits bus-off state.

### TRMSTS bit (Channel Transmit Status)

The TRMSTS bit indicates whether the related CANFD channel is transmitting a message.

This bit is set automatically when the related CANFD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a receiver node.

### RECSTS bit (Channel Receive Status)

The RECSTS bit indicates whether the related CANFD channel is receiving a message.

This bit is set automatically when the related CANFD channel is operating as a receiver node.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a transmitter node.

### COMSTS bit (Channel Communication Status)

The COMSTS bit indicates whether the related CANFD channel is ready for communication.

This bit is set automatically when the related CANFD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET or CD_HALT mode.

Note: This bit is 1 during bus-off state.

### ESIF bit (Error State Indication Flag)

The ESIF bit is set when the ESI bit is sampled recessively for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CANFD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0 to it. This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

### REC[7:0] bits (Reception Error Count)

The REC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CANFD module enters GL_RESET or the CANFD channel is in CH_RESET mode.

### TEC[7:0] bits (Transmission Error Count)

The TEC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CANFD channel is in CH_HALT mode.

These bits are cleared automatically when CANFD module is in GL_RESET or CANFD channel is in CH_RESET mode.

## 32.2.6 CFDCnERFL : Channel n Error Flag Registers (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x000C + 0x10 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | CRCREG[14:0] | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | ADERR | B0ERR | B1ERR | CERR | AERR | FERR | SERR | ALF | BLF | OVLF | BORF | BOEF | EPF | EWF | BEF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | BEF | Bus Error Flag<br>0: Channel bus error not detected<br>1: Channel bus error detected | R/W |
| 1 | EWF | Error Warning Flag<br>0: Channel error warning not detected<br>1: Channel error warning detected | R/W |
| 2 | EPF | Error Passive Flag<br>0: Channel error passive not detected<br>1: Channel error passive detected | R/W |
| 3 | BOEF | Bus-Off Entry Flag<br>0: Channel bus-off entry not detected<br>1: Channel bus-off entry detected | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 4 | BORF | Bus-Off Recovery Flag<br>0: Channel bus-off recovery not detected<br>1: Channel bus-off recovery detected | R/W |
| 5 | OVLF | Overload Flag<br>0: Channel overload not detected<br>1: Channel overload detected | R/W |
| 6 | BLF | Bus Lock Flag<br>0: Channel bus lock not detected<br>1: Channel bus lock detected | R/W |
| 7 | ALF | Arbitration Lost Flag<br>0: Channel arbitration lost not detected<br>1: Channel arbitration lost detected | R/W |
| 8 | SERR | Stuff Error<br>0: Channel stuff error not detected<br>1: Channel stuff error detected | R/W |
| 9 | FERR | Form Error<br>0: Channel form error not detected<br>1: Channel form error detected | R/W |
| 10 | AERR | Acknowledge Error<br>0: Channel acknowledge error not detected<br>1: Channel acknowledge error detected | R/W |
| 11 | CERR | CRC Error<br>0: Channel CRC error not detected<br>1: Channel CRC error detected | R/W |
| 12 | B1ERR | Bit 1 Error<br>0: Channel bit 1 error not detected<br>1: Channel bit 1 error detected | R/W |
| 13 | B0ERR | Bit 0 Error<br>0: Channel bit 0 error not detected<br>1: Channel bit 0 error detected | R/W |
| 14 | ADERR | Acknowledge Delimiter Error<br>0: Channel acknowledge delimiter error not detected<br>1: Channel acknowledge delimiter error detected | R/W |
| 15 | — | This bit is read as 0. The write value should be 0. | R/W |
| 30:16 | CRCREG[14:0] | CRC Register value<br>These bits show the CRC value calculated for the CAN2.0 CAN frame. | R |
| 31 | — | This bit is read as 0. The write value should be 0. | R/W |

Each Channel Error Flag register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) to check when each error condition occurs.

For this register, only a single bit can be cleared by software. Do not use the bit clear instruction to clear the bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Example in assembler language to clear the CFDCnERFL.BEF bit:

```
mov.b #0x0FE, CFDCnERFL ;
```

### BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CANFD channel is in CH_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### EWF bit (Error Warning Flag)

The EWF bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when either TEC or REC exceeds 0x5F.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x5F. Therefore, if the TEC or REC remains > 0x5F and the EWF bit is cleared by software, it is not set again until both the TEC and REC go below 0x60 and either TEC or REC crosses over again from a value 0x5F to a value > 0x5F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### EPF bit (Error Passive Flag)

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x7F. Therefore, if the TEC or REC remains > 0x7F and the bit is cleared by software, it is not set again until both the TEC and REC go below 0x80 and either TEC or REC crosses over again from a value ≤ 0x7F to a value > 0x7F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### BOEF bit (Bus-Off Entry Flag)

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### BORF bit (Bus-Off Recovery Flag)

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDCnCTR.BOM is 00b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDCnCTR.BOM is 10b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDCnCTR.BOM is 11b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDCnCTR.RTBO is set to 1 (the CAN channel returns to error active)
- When CFDCnCTR.BOM is 01b
- When CFDCnCTR.BOM is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### OVLF bit (Overload Flag)

The OVLF flag indicates a detection of a CAN channel overload state.

The OVLF bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### BLF bit (Bus Lock Flag)

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

It is cleared automatically when the related CANFD channel is in CH_RESET mode.

### ALF bit (Arbitration Lost Flag)

The ALF bit indicates a detection of a CAN channel bus arbitration lost condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

The bit is set automatically when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### SERR bit (Stuff Error)

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDCnERFL[14:8] is already set. Otherwise, this bit is set if CFDCnERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### FERR bit (Form Error)

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.

3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDCnERFL[14:8] is already set. Otherwise, this bit is set if CFDCnERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### AERR bit (Acknowledge Error)

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDCnERFL[14:8] is already set. Otherwise, this bit is set if CFDCnERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION.

### CERR bit (CRC Error)

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDCnERFL[14:8] is already set. Otherwise, this bit is set if CFDCnERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

### B1ERR bit (Bit 1 Error)

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDCnERFL[14:8] is already set. Otherwise, this bit is set if CFDCnERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

#### B0ERR bit (Bit 0 Error)

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDCnERFL[14:8] is already set. Otherwise, this bit is set if CFDCnERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

#### ADERR bit (Acknowledge Delimiter Error)

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDCnERFL[14:8] is already set. Otherwise, this bit is set if CFDCnERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

#### CRCREG[14:0] bits (CRC Register value)

The CRCREG[14:0] bits read the calculated CRC value when CFDCnCTR.CTME bit is 1 for the channel.

If CFDCnCTR.CTME bit is 0, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the CANFD channel logic when the CTME bit is enabled.

The CFDCnERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

### 32.2.7 CFDCnDCFG : Channel n Data Bitrate Configuration Register (n = 0, 1)

This register is not available in the classical CAN function.

Base address: CANFD = 0x400B_0000

Offset address: 0x1400 + 0x20 × n

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | 27 | | | 24 | | | | | 19 | | | 16 | | | | 12 | | | | 8 | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | DSJW[3:0] | | | | — | — | — | — | DTSEG2[3:0] | | | | — | — | — | DTSEG1[4:0] | | | | | DBRP[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | DBRP[7:0] | Channel Data Baud Rate Prescaler<br>Data Baud Rate Prescaler division ratio | R/W |
| 12:8 | DTSEG1[4:0] | Timing Segment 1<br>0x00: Reserved<br>0x01: 2 Tq<br>0x02: 3 Tq<br>0x03: 4 Tq<br>⋮<br>0x1E: 31 Tq<br>0x1F: 32 Tq | R/W |
| 15:13 | — | These bits are read as 0. The write value should be 0. | R/W |
| 19:16 | DTSEG2[3:0] | Timing Segment 2<br>0x0: Reserved<br>0x1: 2 Tq<br>⋮<br>0xE: 15 Tq<br>0xF: 16 Tq | R/W |
| 23:20 | — | These bits are read as 0. The write value should be 0. | R/W |
| 27:24 | DSJW[3:0] | Resynchronization Jump Width<br>0x0: 1 Tq<br>0x1: 2 Tq<br>⋮<br>0xF: 16 Tq | R/W |
| 31:28 | — | These bits are read as 0. The write value should be 0. | R/W |

Note: Tq means time quantum.

The Channel n Data Bitrate Configuration Register (n = 0, 1) configures the transmission/reception data baud rate parameters of the channels.

The channel of Classical CAN mode does not perform configuration of this register.

### DBRP[7:0] bits (Channel Data Baud Rate Prescaler)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

### DTSEG1[4:0] bits (Timing Segment 1)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits. See section 32.4.1.2. CAN Bit Timing for more details.

### DTSEG2[3:0] bits (Timing Segment 2)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits.

### DSJW[3:0] bits (Resynchronization Jump Width)

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

## 32.2.8 CFDCnFDCFG : Channel n CAN-FD Configuration Register (n = 0, 1)

This register is not available in the classical CAN function.

Base address: CANFD = 0x400B_0000

Offset address: 0x1404 + 0x20 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | CFDTE | CLOE | REFE | FDOE | — | GWBRS | GWFDF | GWEN | TDCO[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | ESIC | TDCE | TDCOC | — | — | — | — | — | EOCCFG[2:0][2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2:0 | EOCCFG[2:0] | Error Occurrence Counter Configuration<br>0 0 0: All transmitter or receiver CAN frames<br>0 0 1: All transmitter CAN frames<br>0 1 0: All receiver CAN frames<br>0 1 1: Reserved<br>1 0 0: Only transmitter or receiver CAN-FD data-phase (fast bits)<br>1 0 1: Only transmitter CAN-FD data-phase (fast bits)<br>1 1 0: Only receiver CAN-FD data-phase (fast bits)<br>1 1 1: Reserved | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | TDCOC[*1] | Transceiver Delay Compensation Offset Configuration<br>0: Measured + offset<br>1: Offset-only | R/W |
| 9 | TDCE[*1] | Transceiver Delay Compensation Enable<br>0: Transceiver delay compensation disabled<br>1: Transceiver delay compensation enabled | R/W |
| 10 | ESIC[*1] | Error State Indication Configuration<br>0: The ESI bit in the frame represents the error state of the node itself<br>1: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself. | R/W |
| 11 | — | This bit is read as 0. The write value should be 0. | R/W |
| 15:12 | — | These bits are read as 0. The write value should be 0. | R/W |
| 23:16 | TDCO[7:0][*1] | Transceiver Delay Compensation Offset | R/W |
| 24 | GWEN[*1] | CAN2.0, CAN-FD Multi-Gateway Enable<br>0: Multi-gateway disabled<br>1: Multi-gateway enable | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 25 | GWFDF[1] | Gateway FDF Configuration Bit<br>0: Gateway frame is transmitted as Classical CAN frame<br>1: Gateway frame is transmitted as CAN-FD frame | R/W |
| 26 | GWBRS[1] | Gateway BRS Configuration Bit<br>0: Gateway frame is transmitted with BRS = 0<br>1: Gateway frame is transmitted with BRS = 1 | R/W |
| 27 | — | This bit is read as 0. The write value should be 0. | R/W |
| 28 | FDOE[1] | FD-Only Enable<br>0: FD-only mode disabled<br>1: FD-only mode enabled | R/W |
| 29 | REFE | RX Edge Filter Enable<br>0: RX edge filter disabled<br>1: RX edge filter enabled | R/W |
| 30 | CLOE[1] | Classical CAN Enable<br>0: Classical CAN mode disabled<br>1: Classical CAN mode enabled | R/W |
| 31 | CFDTE | CAN-FD Frame Distinction Enable<br>0: CAN-FD frame distinction disabled<br>1: CAN-FD frame distinction enabled | R/W |

Note 1. These bits are not available in the classical CAN function.

The Channel n CAN-FD Configuration Register (n = 0, 1) configures which communication direction (transmitter/receiver) errors are counted.

### EOCCFG[2:0] bits (Error Occurrence Counter Configuration)

The EOCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### TDCOC bit (Transceiver Delay Compensation Offset Configuration)[1]

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN-FD channel. If the bit is set to 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

### TDCE bit (Transceiver Delay Compensation Enable)[1]

The TDCE bit enables the transceiver delay compensation for the CAN-FD channel.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

### ESIC bit (Error State Indication Configuration)[1]

Bus controllers that are used as CAN-to-CAN gateway support that in every forwarded CAN-FD message. The ESI flag does not change to reflect the status of the gateway, bridge, or router but instead the flag is sent as it was in the original message.

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTSn.CFESI or CFDTMFDCTRn.TMESI).

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

### TDCO[7:0] bits (Transceiver Delay Compensation Offset)*1

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDCnFDCFG.TDCOC setting.

If CFDCnFDCFG.TDCOC = 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDCnFDCFG.TDCO. See section 32.4.1.5. Transmitter Delay Compensation for details on how CFDCnFDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

### GWEN bit (CAN2.0, CAN-FD Multi-Gateway Enable)*1

When the GWEN bit is enabled, a multi-gateway is enabled. Message received on one node can be routed to another node using the COM FIFO when they are configured as gateway FIFO (CFDCFCCn.CFM = 10b). Furthermore, when TX Queue is set as gateway mode, the message received on one node can be stored in TX Queue and can be sent to another node.

The FDF and BRS bits of the routed message can be changed by the configuration value of the CFDCnFDCFG.GWFDF and CFDCnFDCFG.GWBRS bits. By this, the transmitted value of these bits can be replaced.

Example :

CFDCnFDCFG.GWEN = 1 on channel y

CFDCnFDCFG.GWFDF = 1

If a Classical CAN frame is received on channel x and routed to a gateway FIFO or TX Queue of channel y. Then this CAN frame is sent on channel y as a CAN-FD frame because of the CFDCnFDCFG.GWFDF bit.

Table 32.3 shows how the message information is changed depending on the received and configured data.

**Table 32.3 Modified message information by received and configured data**

| Routed CAN frame | Routed received DLC | CAN BRS bit | Configured CFDCnFDCFG.GWFDF bit | Gateway message DLC | Gateway message BRS bit | Gateway message frame type |
|---|---|---|---|---|---|---|
| CAN2.0 | ≤ 8 | N/A | 1 | ≤ 8 | Based on configuration of CFDCnFDCFG.GWBRS | FD |
| CAN2.0 | > 8 | N/A | 1 | = 8 | Based on configuration of CFDCnFDCFG.GWBRS | FD |
| FD | ≤ 8 | None | 1 | ≤ 8 | Based on configuration of CFDCnFDCFG.GWBRS | FD |
| FD | > 8 | None | 1 | > 8 | Based on configuration of CFDCnFDCFG.GWBRS | FD |
| CAN2.0 | ≤ 8 | N/A | 0 | ≤ 8 | N/A | CAN2.0 |
| CAN2.0 | > 8 | N/A | 0 | > 8 | N/A | CAN2.0 |
| FD | ≤ 8 | None | 0 | ≤ 8 | N/A | CAN2.0 |
| FD | > 8 | None | 0 | = 8 | N/A | CAN2.0 |

Note: This gateway is limited to an 8-byte data payload for different frame type. If routing and target frame type is the same, the data length code (DLC) value remains the same. If the source frame is a CAN-FD with more than 8 data bytes, then on classical destination node, the data payload is reduced to 8 bytes. Only the first 8 bytes of data perform gateway transmission.

Note: Transmission buffers other than the gateway FIFO are not affected by this feature.

Do not route remote frames through the gateway when CFDCnFDCFG.GWEN is set. When a destination node is CFDCnFDCFG.FDOE = 1, set CFDCnFDCFG.GWEN and CFDCnFDCFG.GWFDF to 1. When a destination node is CFDCnFDCFG.CLOE = 1, set CFDCnFDCFG.GWEN=1 and CFDCnFDCFG.GWFDF = 0.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

### GWFDF bit (Gateway FDF Configuration Bit)[*1]

When the GWEN bit is set to 1, the FDF bit of the transmitting gateway frame is replaced by the value of the GWFDF bit.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CAN-FD module is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

### GWBRS bit (Gateway BRS Configuration Bit)[*1]

When the GWEN bit is set to 1, the BRS bit of the transmitting gateway frame is replaced by the value of CFDCnFDCFG.GWBRS.

In classical CAN frames, the GWBRS bit is invalid.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

### FDOE bit (FD-Only Enable)[*1]

The FDOE bit enables the reception and transmission of CAN-FD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don't care (CFDCFFDCSTSn.CFFDF/CFDTMFDCTRn.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with 0xCC.

The FDOE bit cannot be written in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Do not set CFDCnFDCFG.FDOE and CFDCnFDCFG.CLOE simultaneously.

### REFE bit (RX Edge Filter Enable)

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode. Do not set this bit when in Classical CAN mode and when CFDCnFDCFG.CFDTE = 0 (disabled CAN-FD frame distinction).

### CLOE bit (Classical CAN Enable)[*1]

The CLOE bit enables the Classical CAN mode. If this bit is 1, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDCnFDCFG.CLOE and CFDCnFDCFG.FDOE simultaneously.

| CFDCnFDCFG.CLOE | CFDCnFDCFG.FDOE | Channel mode |
|---|---|---|
| 0 | 0 | CAN-FD mode |
| 0 | 1 | FD-only mode |
| 1 | 0 | Classical CAN mode |

| CFDCnFDCFG.CLOE | CFDCnFDCFG.FDOE | Channel mode |
|---|---|---|
| 1 | 1 | Reserved |

The CANFD mode is available only for CANFD supported product.

Do not write to this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the CAN-FD channel is in CH_RESET mode.

### CFDTE bit (CAN-FD Frame Distinction Enable)

The CFDTE bit enables the CAN-FD frame distinction function. The CFDTE bit is required for Classical CAN mode (CFDCnFDCFG.CLOE = 1).

If this bit is 0, the protocol controller can only send Classical frames and response with a Form or CRC error on FD frames.

If this bit is 1, the protocol controller behaves according to the ISO 11898-1 (DIS 2015) specification. If the FDF bit is detected recessive, then the protocol controller enters the protocol exception state, and attempts to integrate back to the CAN communication.

Do not write this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET mode.

Note 1. These bits are not available in the classical CAN function.

## 32.2.9 CFDCnFDCTR : Channel n CANFD Control Register (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1408 + 0x20 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SOCCLR | EOCCLR |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | EOCCLR | Error Occurrence Counter Clear<br>0: No error occurrence counter clear<br>1: Clear error occurrence counter | R/W |
| 1 | SOCCLR | Successful Occurrence Counter Clear<br>0: No successful occurrence counter clear<br>1: Clear successful occurrence counter | R/W |
| 31:2 | — | These bits are read as 0. The write value should be 0. | R/W |

The Channel n CANFD Control Register (n = 0, 1) controls the error and successful occurrence counters.

### EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

### SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

### 32.2.10 CFDCnFDSTS : Channel n CANFD Status Register (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x140C + 0x20 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | SOC[7:0] | | | | | | | | EOC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | TDCVF | — | — | — | — | — | SOCO | EOCO | TDCR[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TDCR[7:0][*1] | Transceiver Delay Compensation Result | R |
| 8 | EOCO | Error Occurrence Counter Overflow<br>0: Error occurrence counter has not overflowed<br>1: Error occurrence counter has overflowed | R/W |
| 9 | SOCO | Successful Occurrence Counter Overflow<br>0: Successful occurrence counter has not overflowed<br>1: Successful occurrence counter has overflowed | R/W |
| 14:10 | — | These bits are read as 0. The write value should be 0. | R/W |
| 15 | TDCVF[*1] | Transceiver Delay Compensation Violation Flag<br>0: Transceiver delay compensation violation has not occurred<br>1: Transceiver delay compensation violation has occurred | R/W |
| 23:16 | EOC[7:0] | Error Occurrence Counter<br>These bits show the error occurrence counter value. | R |
| 31:24 | SOC[7:0] | Successful occurrence counter<br>These bits show the successful occurrence counter value. | R |

Note 1. These bits are not available in the classical CAN function.

The Channel n CANFD Status Register (n = 0, 1) indicates the transceiver compensation delay result and its related FIFO message lost status.

#### TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDCnFDCFG.TDCOC configuration and the offset value in CFDCnFDCFG.TDCO. See section 32.4.1.5. Transmitter Delay Compensation for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between FDF and the RES bit when CFDCnFDCFG.TDCOC = 0 and the transceiver delay compensation is enabled (CFDCnFDCFG.TDCE = 1).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

Note: These bits are not available in the classical CAN function.

#### EOCO bit (Error Occurrence Counter Overflow)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDCnFDSTS.EOC is 0xFF and a CAN bus error is detected based on the configuration defined in CFDCnFDCFG.EOCCFG.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

### SOCO bit (Successful Occurrence Counter Overflow)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDCnFDSTS.SOC is 0xFF and a successful message reception or successful message transmission occurs.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Write to this bit only when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

### TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDCnFDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk_dlc) and the internal bit is overrun.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

### EOC[7:0] bits (Error Occurrence Counter)

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDCnFDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to CFDCnFDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDCnFDCFG.EOCCFG bits. When the counter reaches the value of 0xFF, the update stops.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

### SOC[7:0] bits (Successful occurrence counter)

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CANFD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of 0xFF, the update stops.

Note: In Loopback mode, the counter is incremented twice.

These bits are cleared by writing 1 to CFDCnFDCTR.SOCCLR.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

## 32.2.11 CFDCnFDCRC : Channel n CANFD CRC Register (n = 0, 1)

This register is not available in the classical CAN function.

Base address: CANFD = 0x400B_0000

Offset address: 0x1410 + 0x20 × n

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | 27 | | | 24 | | | | 20 | | | | | | | | | | | | | | | | | | | | 0 |
| Bit field: | — | — | — | — | SCNT[3:0] | | | | — | — | — | CRCREG[20:0] | | | | | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 20:0 | CRCREG[20:0] | CRC Register value<br>These bits show the CRC value calculated for the CANFD frame. | R |
| 23:21 | — | These bits are read as 0. The write value should be 0. | R/W |
| 27:24 | SCNT[3:0] | Stuff bit count<br>These bits shows the stuff bit count (mod 8) for the CANFD frame. | R |
| 31:28 | — | These bits are read as 0. The write value should be 0. | R/W |

The Channel n CANFD CRC Register (n = 0, 1) holds the CRC value calculated for the CANFD frame.

### CRCREG[20:0] bits (CRC Register value)

The CRCREG[20:0] bits contain the CRC value calculated by the CANFD channel logic when the CFDCnCTR.CTME bit is enabled.

The CFDCnFDCRC.CRCREG value is updated in the first bit of the CRC field of the CANFD frame (reception and transmission).

When the CFDCnCTR.CTME bit is 0, the CRCREG[20:0] bits are always read as 0.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

### SCNT[3:0] bits (Stuff bit count)

The SCNT[3:0] bits contain the stuff count value of the CANFD frame. These bits indicate the number of inserted stuff bits (modulo 8, Graycoded) for a CANFD frame when the CFDCnCTR.CTME bit is enabled in CFDCnFDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDCnCTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0.

The SCNT value is updated in the first bit of CRC field of the CANFD frame (reception and transmission).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

## 32.2.12 CFDGCFG : Global Configuration Register

Base address: CANFD = 0x400B_0000

Offset address: 0x0084

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | ITRCP[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | TSBTCS[2:0] | | | TSSS | TSP[3:0] | | | | — | — | CMPOC | DCS | MME | DRE | DCE | TPRI |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TPRI | Transmission Priority<br>0: ID priority<br>1: Message buffer number priority | R/W |
| 1 | DCE | DLC Check Enable<br>0: DLC check disabled<br>1: DLC check enabled | R/W |
| 2 | DRE | DLC Replacement Enable<br>0: DLC replacement disabled<br>1: DLC replacement enabled | R/W |
| 3 | MME | Mirror Mode Enable<br>0: Mirror mode disabled<br>1: Mirror mode enabled | R/W |
| 4 | DCS | Data Link Controller Clock Select<br>0: Internal clean clock<br>1: External clock source connected to CANMCLK pin | R/W |
| 5 | CMPOC[*1] | CAN-FD Message Payload Overflow Configuration<br>0: Message is rejected<br>1: Message payload is cut to fit to configured message size | R/W |
| 7:6 | — | These bits are read as 0. The write value should be 0. | R/W |
| 11:8 | TSP[3:0] | Timestamp Prescaler<br>0x0: Timestamp prescaler = 1<br>0x1: Timestamp prescaler = 2<br>0x2: Timestamp prescaler = 4<br>0x3: Timestamp prescaler = 8<br>⋮<br>0xD: Timestamp prescaler = 8192<br>0xE: Timestamp prescaler = 16384<br>0xF: Timestamp prescaler = 32768 | R/W |
| 12 | TSSS | Timestamp Source Select<br>0: Source clock for timestamp counter is peripheral clock<br>1: Source clock for timestamp counter is bit time clock | R/W |
| 15:13 | TSBTCS[2:0] | Timestamp Bit Time Channel Select<br>0 0 0: Select clock from channel 0<br>0 0 1: Select clock from channel 1<br>Others: Setting prohibited | R/W |
| 31:16 | ITRCP[15:0] | Interval Timer Reference Clock Prescaler<br>FIFO interval timer prescaler value | R/W |

Note 1. This bit are not available in the classical CAN function.

The Global Configuration Register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of all CAN channels. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

### TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

### DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

### DRE bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE is 1, the CAN-FD stores the configured value (CFDGAFLP0n.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

### MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

### DCS bit (Data Link Controller Clock Select)

The DCS bit selects the clock source for CAN communication. Internal clean clock has a smaller clock jitter than the peripheral clock B (PCLKB).

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

### CMPOC bit (CAN-FD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCn.RFPLS, and CFDCFCCn.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

Note: This bit is not available in the classical CAN function.

### TSP[3:0] bits (Timestamp Prescaler)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

### TSSS bit (Timestamp Source Select)

The TSSS bit allows the selection of the clock source for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode. Additionally, do not set this bit to 1 when CAN-FD communication is used.*1

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

### TSBTCS[2:0] bits (Timestamp Bit Time Channel Select)

The TSBTCS[2:0] bits allow the selection of the bit time clock of a particular channel for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

### ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock.

When these bits are 0x0000, the timer is disabled.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

## 32.2.13 CFDGCTR : Global Control Register

Base address: CANFD = 0x400B_0000

Offset address: 0x0088

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSRST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field: | MOWEIE | QMEIE | — | — | CMPOFIE | THLEIE | MEIE | DEIE | — | — | — | — | — | GSLPR | GMDC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 1:0 | GMDC[1:0] | Global Mode Control<br>0 0: Global operation mode request<br>0 1: Global reset mode request<br>1 0: Global halt mode request<br>1 1: Keep current value | R/W |
| 2 | GSLPR | Global Sleep Request<br>0: Global sleep request disabled<br>1: Global sleep request enabled | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | DEIE | DLC Check Interrupt Enable<br>0: DLC check interrupt disabled<br>1: DLC check interrupt enabled | R/W |
| 9 | MEIE | Message Lost Error Interrupt Enable<br>0: Message lost error interrupt disabled<br>1: Message lost error interrupt enabled | R/W |
| 10 | THLEIE | TX History List Entry Lost Interrupt Enable<br>0: TX history list entry lost interrupt disabled<br>1: TX history list entry lost interrupt enabled | R/W |
| 11 | CMPOFIE[*1] | CANFD Message Payload Overflow Flag Interrupt Enable<br>0: CANFD message payload overflow flag interrupt disabled<br>1: CANFD message payload overflow flag interrupt enabled | R/W |
| 13:12 | — | These bits are read as 0. The write value should be 0. | R/W |
| 14 | QMEIE | TXQ Message Lost Error Interrupt Enable<br>0: TXQ message lost error interrupt disabled<br>1: TXQ message lost error interrupt enabled | R/W |
| 15 | MOWEIE | GW FIFO Message Overwrite Error Interrupt Enable<br>0: GW FIFO message overwrite error interrupt disabled<br>1: GW FIFO message overwrite error interrupt enabled | R/W |
| 16 | TSRST | Timestamp Reset<br>0: Timestamp not reset<br>1: Timestamp reset | R/W |
| 31:17 | — | These bits are read as 0. The write value should be 0. | R/W |

Note 1. This bit is not available in the classical CAN function.

The Global Control Register controls the global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

### GMDC bits (Global Mode Control)

The GMDC bits can be used to configure the modes for the CANFD module. Additionally, if CFDGCTR.GSLPR bit is 1 when the CANFD module is in Reset mode, the CANFD module enters Global Sleep mode. Additionally, if CFDGCTR.GSLPR is 1 when the CANFD module is in Reset Mode, then the CANFD module enters Global Sleep Mode.

Setting the GMDC bits to 11b has no effect. Mode transition is described in detail in section 32.3.2. Global Modes.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### GSLPR bit (Global Sleep Request)

The GSLPR bit globally selects the sleep request for CANFD module including all CAN channels. Channel sleep request is set automatically for all channels.

Only write to this bit when the CANFD module is in GL_RESET or GL_SLEEP mode.

### DEIE bit (DLC Check Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### MEIE bit (Message Lost Error Interrupt Enable)

When the MEIE bit is 1, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the THLEIE bit is 1, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### CMPOFIE bit (CANFD Message Payload Overflow Flag Interrupt Enable)

When the CMPOFIE bit is 1, an interrupt is generated when a CANFD message payload overflow condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Note: This bit is not available in the classical CAN function

### QMEIE bit (TXQ Message Lost Error Interrupt Enable)

When the QMEIE bit is 1, an interrupt is generated when a TXQ message lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### MOWEIE bit (GW FIFO Message Overwrite Error Interrupt Enable)

When the MOWEIE bit is 1, an interrupt is generated in GW mode and a GW FIFO message over write condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### TSRST bit (Timestamp Reset)

When the TSRST bit is 1, the Global Timestamp Register is reset to 0x0000.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

Read value is always 0.

This bit is cleared automatically by the CANFD module logic.

## 32.2.14 CFDGSTS : Global Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x008C

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | GRAM INIT | GSLP STS | GHLT STS | GRST STS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | GRSTSTS | Global Reset Status<br>0: Not in Reset mode<br>1: In Reset mode | R |
| 1 | GHLTSTS | Global Halt Status<br>0: Not in Halt mode<br>1: In Halt mode | R |
| 2 | GSLPSTS | Global Sleep Status<br>0: Not in Sleep mode<br>1: In Sleep mode | R |
| 3 | GRAMINIT | Global RAM Initialization<br>0: RAM initialization is complete<br>1: RAM initialization is ongoing | R |
| 31:4 | — | These bits are read as 0. | R |

The Global Status Register indicates the global status of the CANFD module.

### GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CANFD module Reset mode.

This bit is set automatically when the CANFD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CANFD module exits the GL_RESET mode.

### GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CANFD module Halt mode.

This bit is set automatically when the CANFD module enters GL_HALT mode.

This bit is cleared automatically when the CANFD module exits the GL_HALT mode.

### GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CANFD module Sleep mode.

This bit is set automatically when the CANFD module enters GL_SLEEP mode.

This bit is cleared automatically when the CANFD module exits the GL_SLEEP mode.

### GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CANFD module RAM initialization.

This bit is set automatically when the CANFD module enters GL_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CANFD module completed RAM initialization.

This bit is cleared when the test_mode input port is set to 1.

## 32.2.15 CFDGERFL : Global Error Flag Register

Base address: CANFD = 0x400B_0000

Offset address: 0x0090

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | EEF1 | EEF0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | QMES | — | QOWES | CMPOF | THLES | MES | DEF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | DEF | DLC Error Flag<br>0: DLC error not detected<br>1: DLC error detected | R/W |
| 1 | MES | Message Lost Error Status<br>0: Message lost error not detected<br>1: Message lost error detected | R |
| 2 | THLES | TX History List Entry Lost Error Status<br>0: TX history list entry lost error not detected<br>1: TX history list entry lost error detected | R |
| 3 | CMPOF[*1] | CANFD Message Payload Overflow Flag<br>0: CANFD message payload overflow not detected<br>1: CANFD message payload overflow detected | R/W |
| 4 | QOWES | TXQ Message Overwrite Error Status<br>0: TXQ message overwrite error not detected<br>1: TXQ message overwrite error detected | R |
| 5 | — | This bit is read as 0. The write value should be 0. | R |
| 6 | QMES | TXQ Message Lost Error Status<br>0: TXQ message lost error not detected<br>1: TXQ message lost error detected | R |
| 15:7 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | EEF0 | ECC Error Flag for Channel 0<br>0: ECC error not detected during TX-SCAN<br>1: ECC error detected during TX-SCAN | R/W |
| 17 | EEF1 | ECC Error Flag for Channel 1<br>0: ECC error not detected during TX-SCAN<br>1: ECC error detected during TX-SCAN | R/W |
| 31:18 | — | These bits are read as 0. The write value should be 0. | R/W |

Note 1. This bit is not available in the classical CAN function.

The Global Error Flag register indicates the detection of global errors.

### DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when a DLC error is detected in a received frame.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set

The bit is cleared by writing 0 to it.

This bit is cleared automatically in GL_RESET mode.

### MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CANFD module is in GL_RESET mode.

### THLES bit (TX History List Entry Lost Error Status)

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CANFD module is in GL_RESET mode.

### CMPOF bit (CANFD Message Payload Overflow Flag)

The CMPOF bit is set automatically when a CANFD message payload overflow is detected on at least one channel.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

This bit is cleared by writing 0 to it. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared automatically in GL_RESET mode.

Note: This bit is not available in the classical CAN function

### QOWES bit (TXQ Message Overwrite Error Status)

The QOWES bit is set automatically when a TXQ message overwrite error is detected.

This bit is cleared automatically when all TXQ message overwrite flags are cleared.

This bit is cleared automatically in GL_RESET mode.

### QMES bit (TXQ Message Lost Error Status)

The QMES bit is set automatically when a TXQ message lost error is detected.

This bit is cleared automatically when all TXQ message lost flags are cleared.

This bit is cleared automatically in GL_RESET mode.

### EEF0 bit (ECC Error Flag for Channel 0)

The EEF0 bit specifies whether an ECC error has occurred on Channel 0.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL_RESET mode.

### EEF1 bit (ECC Error Flag for Channel 1)

The EEF1 bit specifies whether an ECC error has occurred on Channel 1.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 to this bit has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL_RESET mode.

## 32.2.16 CFDGTINTSTS0 : Global TX Interrupt Status Register 0

Base address: CANFD = 0x400B_0000

Offset address: 0x1300

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | CFOTI<br>F1 | TQOFI<br>F1 | THIF1 | CFTIF<br>1 | TQIF1 | TAIF1 | TSIF1 | — | CFOTI<br>FO | TQOFI<br>FO | THIF0 | CFTIF<br>0 | TQIF0 | TAI0 | TSIF0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TSIF0 | TX Successful Interrupt Flag Channel 0<br>0: Channel n TX Successful Interrupt flag not set<br>1: Channel n TX Successful Interrupt flag set | R |
| 1 | TAI0 | TX Abort Interrupt Flag Channel 0<br>0: Channel n TX Abort Interrupt flag not set<br>1: Channel n TX Abort Interrupt flag set | R |
| 2 | TQIF0 | TX Queue Interrupt Flag Channel 0<br>0: Channel n TX Queue Interrupt flag not set<br>1: Channel n TX Queue Interrupt flag set | R |
| 3 | CFTIF0 | COM FIFO TX/GW Mode Interrupt Flag Channel 0<br>0: Channel n COM FIFO TX/GW Mode Interrupt flag not set<br>1: Channel n COM FIFO TX/GW Mode Interrupt flag set | R |
| 4 | THIF0 | TX History List Interrupt Channel 0<br>0: Channel n TX History List Interrupt flag not set<br>1: Channel n TX History List Interrupt flag set | R |
| 5 | TQOFIFO | TX Queue One Frame Transmission Interrupt Flag Channel 0<br>0: Channel n TX Queue One Frame Transmission Interrupt flag not set<br>1: Channel n TX Queue One Frame Transmission Interrupt flag set | R |
| 6 | CFOTIFO | COM FIFO One Frame Transmission Interrupt Flag Channel 0<br>0: Channel n COM FIFO One Frame Transmission Interrupt flag not set<br>1: Channel n COM FIFO One Frame Transmission Interrupt flag set | R |
| 7 | — | This bit is read as 0. The write value should be 0. | R/W |
| 8 | TSIF1 | TX Successful Interrupt Flag Channel 1<br>0: Channel n TX Successful Interrupt flag not set<br>1: Channel n TX Successful Interrupt flag set | R |
| 9 | TAIF1 | TX Abort Interrupt Flag Channel 1<br>0: Channel n TX Abort Interrupt flag not set<br>1: Channel n TX Abort Interrupt flag set | R |
| 10 | TQIF1 | TX Queue Interrupt Flag Channel 1<br>0: Channel n TX Queue Interrupt flag not set<br>1: Channel n TX Queue Interrupt flag set | R |
| 11 | CFTIF1 | COM FIFO TX/GW Mode Interrupt Flag Channel 1<br>0: Channel n COM FIFO TX/GW Mode Interrupt flag not set<br>1: Channel n COM FIFO TX/GW Mode Interrupt flag set | R |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 12 | THIF1 | TX History List Interrupt Channel 1<br>0: Channel n TX History List Interrupt flag not set<br>1: Channel n TX History List Interrupt flag set | R |
| 13 | TQOFIF1 | TX Queue One Frame Transmission Interrupt Flag Channel 1<br>0: Channel n TX Queue One Frame Transmission Interrupt flag not set<br>1: Channel n TX Queue One Frame Transmission Interrupt flag set | R |
| 14 | CFOTIF1 | COM FIFO One Frame Transmission Interrupt Flag Channel 1<br>0: Channel n COM FIFO One Frame Transmission Interrupt flag not set<br>1: Channel n COM FIFO One Frame Transmission Interrupt flag set | R |
| 31:15 | — | These bits are read as 0. | R |

The Global TX Interrupt Status register indicates the detection of transmit specific interrupts.

### TSIF0 bit (TX Successful Interrupt Flag Channel 0)

The TSIF0 bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### TAI0 bit (TX Abort Interrupt Flag Channel 0)

The TAIF0 bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### TQIF0 bit (TX Queue Interrupt Flag Channel 0)

The TQIF0 bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt is enable disabled)
- When in GL_RESET or CH_RESET mode.

### CFTIF0 bit (COM FIFO TX/GW Mode Interrupt Flag Channel 0)

The CFTIF0 bit is set to 1 when the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### THIF0 bit (TX History List Interrupt Channel 0)

The THIF0 bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### CFDGTINTSTS0.TQOFIFn bit (TX Queue One Frame Transmission Interrupt Flag Channel n (n = 0, 1) )

The CFDGTINTSTS0.TQOFIFn bit is set to 1 when the TX Queue One Frame Transmission Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue One Frame Transmission Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### CFDGTINTSTS0.CFOTIFn bit (COM FIFO One Frame Transmission Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.CFOTIFn bit is set to 1 when the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is set (when the Interrupt is enabled).

This bit is cleared automatically:

- When the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### TSIF1 bit (TX Successful Interrupt Flag Channel 1)

The TSIF1 bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### TAIF1 bit (TX Abort Interrupt Flag Channel 1)

The TAIF1 bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### TQIF1 bit (TX Queue Interrupt Flag Channel 1)

The TQIF1 bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt is enable disabled)
- When in GL_RESET or CH_RESET mode.

### CFTIF1 bit (COM FIFO TX/GW Mode Interrupt Flag Channel 1)

The CFTIF1 bit is set to 1 when the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

### THIF1 bit (TX History List Interrupt Channel 1)

The THIF1 bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

## 32.2.17 CFDGTSC : Global Timestamp Counter Register

Base address: CANFD = 0x400B_0000

Offset address: 0x0094

| Bit position: | 31 | | | | | | | | | | | | | | | | 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 15:0 | TS[15:0] | Timestamp value | R |
| 31:16 | — | These bits are read as 0. | R |

The Global Timestamp Counter register stores the timestamp based on the selected configuration.

### TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS and TSP. The accuracy of the timestamp counter cannot be guaranteed when transitioning to halt state.

The Timestamp value is stored in this register based on the configuration of TSSS, TSBTCS and TSP.

Do not write to bits TS[15:0] when the CANFD module is in GL_RESET or GL_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL_RESET mode.

## 32.2.18 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register

Base address: CANFD = 0x400B_0000

Offset address: 0x0098

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | AFLDAE | — | — | — | — | AFLPN[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 3:0 | AFLPN[3:0] | Acceptance Filter List Page Number<br>Select an Acceptance Filter List page | R/W |
| 7:4 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | AFLDAE | Acceptance Filter List Data Access Enable<br>0: Acceptance Filter List data access disabled<br>1: Acceptance Filter List data access enabled | R/W |
| 31:9 | — | These bits are read as 0. The write value should be 0. | R/W |

The Global Acceptance Filter List Entry Control Register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

### AFLPN[3:0] bits (Acceptance Filter List Page Number)

The AFLPN[3:0] bits select the page number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries.

Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode. Enter only the values between 0x00 and 0x07, inclusive.

### AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

## 32.2.19 CFDGAFLCFG0 : Global Acceptance Filter List Configuration Register 0

Base address: CANFD = 0x400B_0000

Offset address: 0x009C

| Bit position: | 31 | | | | | | | 24 | | | | | | | | 16 | | | | | | | | 8 | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | RNC0[8:0] | | | | | | | | | — | — | — | — | — | — | — | RNC1[8:0] | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 8:0 | RNC1[8:0] | Rule Number for Channel 1<br>Number of rules dedicated to channel 1 | R/W |
| 15:9 | — | These bits are read as 0. The write value should be 0. | R/W |
| 24:16 | RNC0[8:0] | Rule Number for Channel 0<br>Number of rules dedicated to channel 0 | R/W |
| 31:25 | — | These bits are read as 0. The write value should be 0. | R/W |

The Global Acceptance Filter List Configuration Register 0 is used to define the number of rules for entries in the Acceptance Filter List, applicable for channels 0 to 1.

The total number of available entries in the Acceptance Filter List is 64 × (n + 1), 128 for 2 CAN channels. However, the filters can be allocated flexibly to the different channels depending on requirements as long as both of the following conditions are satisfied:

- The maximum number of acceptance filter per channel is 384
- The total number of rules defined for all channels is not exceeding the number of available entries in the Acceptance Filter List.

### RNC0[8:0] bits (Rule Number for Channel n (n = 0, 1))

The RNC0[8:0] bits define the number of rules in the Acceptance Filter List for channel n.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

## 32.2.20 CFDGAFLIDn : Global Acceptance Filter List ID Registers (n = 1 to 16)

Base address: CANFD = 0x400B_0000

Offset address: 0x1800 + 0x10 × (n - 1)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit field: | GAFLIDE | GAFLRTR | GAFLLB | GAFLID[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field: | GAFLID[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 28:0 | GAFLID[28:0] | Global Acceptance Filter List Entry ID Field<br>ID part of the Global Acceptance Filter List entry | R/W |
| 29 | GAFLLB | Global Acceptance Filter List Entry Loopback Configuration<br>0: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX<br>1: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX | R/W |
| 30 | GAFLRTR | Global Acceptance Filter List Entry RTR Field<br>0: Data frame<br>1: Remote frame | R/W |
| 31 | GAFLIDE | Global Acceptance Filter List Entry IDE Field<br>0: Standard identifier of rule entry ID is valid for acceptance filtering<br>1: Extended identifier of rule entry ID is valid for acceptance filtering | R/W |

The Global Acceptance Filter List ID Registers (n = 1 to 16) are used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

### GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See section 32.5.5. Loopback Modes for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### 32.2.21 CFDGAFLMn : Global Acceptance Filter List Mask Registers (n = 1 to 16)

Base address: CANFD = 0x400B_0000

Offset address: 0x1804 + 0x10 × (n - 1)

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | GAFLIDEM | GAFLRTRM | GAFLIFL1 | GAFLIDM[28:16] | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | GAFLIDM[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 28:0 | GAFLIDM[28:0] | Global Acceptance Filter List ID Mask Field<br>Global Acceptance Filter List Mask field bits for ID field | R/W |
| 29 | GAFLIFL1 | Global Acceptance Filter List Information Label 1<br>Global Acceptance Filter List information label bit 1 | R/W |
| 30 | GAFLRTRM | Global Acceptance Filter List Entry RTR Mask<br>0: RTR bit is not used for ID matching<br>1: RTR bit is used for ID matching | R/W |
| 31 | GAFLIDEM | Global Acceptance Filter List IDE Mask<br>0: IDE bit is not used for ID matching<br>1: IDE bit is used for ID matching | R/W |

The Global Acceptance Filter List Mask Registers are used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

#### GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

| | |
|---|---|
| 0 | Corresponding STD-ID/EXT-ID bit is not used for ID matching |
| 1 | Corresponding STD-ID/EXT-ID bit is used for ID matching |

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

#### GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is a MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTSn.RMIFL [1], CFDRFFDSTSn.RFIFL [1], CFDCFFDCSTSn.CFIFL [1]) of the storage location of an incoming message.

Note: This bit is stored in CFDTHLACC1n.TIFL [1] when CFDTHLCCn.THLDGE = 1 is set up using Gateway function.

#### GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0, the STD-ID comparison takes place.

If the received IDE bit is 1, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

## 32.2.22 CFDGAFLP0n : Global Acceptance Filter List Pointer 0 Registers (n = 1 to 16)

Base address: CANFD = 0x400B_0000

Offset address: 0x1808 + 0x10 × (n - 1)

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | GAFLPTR[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | GAFL RMV | — | — | GAFLRMDP[4:0] | | | | | GAFLI FL0 | GAFL SRD2 | GAFL SRD1 | GAFL SRD0 | GAFLDLC[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 3:0 | GAFLDLC[3:0] | Global Acceptance Filter List DLC Field<br>Minimum number of data bytes in a data frame required for acceptance | R/W |
| 4 | GAFLSRD0 | Global Acceptance Filter List Select Routing Destination 0<br>0: Routing target is CFIFO0<br>1: Routing target is TX Queue 0 instead of CFIFO0 | R/W |
| 5 | GAFLSRD1 | Global Acceptance Filter List Select Routing Destination 1<br>0: Routing target is CFIFO1<br>1: Routing target is TX Queue 1 instead of CFIFO1 | R/W |
| 6 | GAFLSRD2 | Global Acceptance Filter List Select Routing Destination 2<br>0: Routing target is CFIFO2<br>1: Routing target is TX Queue 2 instead of CFIFO2 | R/W |
| 7 | GAFLIFL0 | Global Acceptance Filter List Information Label 0 | R/W |
| 12:8 | GAFLRMDP[4:0] | Global Acceptance Filter List RX Message Buffer Direction Pointer<br>RX message buffer number for storage of received messages | R/W |
| 14:13 | — | These bits are read as 0. The write value should be 0. | R/W |
| 15 | GAFLRMV | Global Acceptance Filter List RX Message Buffer Valid<br>0: Single message buffer direction pointer is invalid<br>1: Single message buffer direction pointer is valid | R/W |
| 31:16 | GAFLPTR[15:0] | Global Acceptance Filter List Pointer | R/W |

The Global Acceptance Filter List Pointer 0 Registers (n = 1 to 16) are used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

### GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0.

Table 32.4 shows DLC value that can be configured.

**Table 32.4 Configuration of DLC value**

| Format | DLC[3] | DLC[2] | DLC[1] | DLC[0] | Description |
|---|---|---|---|---|---|
| CAN and CAN-FD | 0 | 0 | 0 | 0 | DLC of received message = 0 or more (DLC filter check is disabled) |
| CAN and CAN-FD | 0 | 0 | 0 | 1 | DLC of received message = 1 or more |
| CAN and CAN-FD | 0 | 0 | 1 | 0 | DLC of received message = 2 or more |
| CAN and CAN-FD | 0 | 0 | 1 | 1 | DLC of received message = 3 or more |
| CAN and CAN-FD | 0 | 1 | 0 | 0 | DLC of received message = 4 or more |
| CAN and CAN-FD | 0 | 1 | 0 | 1 | DLC of received message = 5 or more |
| CAN and CAN-FD | 0 | 1 | 1 | 0 | DLC of received message = 6 or more |
| CAN and CAN-FD | 0 | 1 | 1 | 1 | DLC of received message = 7 or more |
| CAN | 1 | x | x | x | DLC of received message = 8 or more |
| CAN-FD | 1 | 0 | 0 | 0 | DLC of received message = 8 or more*1 |
| CAN-FD | 1 | 0 | 0 | 1 | DLC of received message = 12 or more*1 |
| CAN-FD | 1 | 0 | 1 | 0 | DLC of received message = 16 or more*1 |
| CAN-FD | 1 | 0 | 1 | 1 | DLC of received message = 20 or more*1 |
| CAN-FD | 1 | 1 | 0 | 0 | DLC of received message = 24 or more*1 |
| CAN-FD | 1 | 1 | 0 | 1 | DLC of received message = 32 or more*1 |
| CAN-FD | 1 | 1 | 1 | 0 | DLC of received message = 48 or more*1 |
| CAN-FD | 1 | 1 | 1 | 1 | DLC of received message = 64*1 |

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLSRD0 bit (Global Acceptance Filter List Select Routing Destination 0)

The GAFLSRD0 bit changes a copy destination to CFIFO0 or TXQ0 by routing.

If this bit is set as 1, the preset value of CFDGAFLP1n.GAFLFDP selects TX Queue.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO0.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLSRD1 bit (Global Acceptance Filter List Select Routing Destination 1)

The GAFLSRD1 bit changes a copy destination to CFIFO1 or TXQ1 by routing.

If this bit is set to 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ1.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO1.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLSRD2 bit (Global Acceptance Filter List Select Routing Destination 2)

The GAFLSRD2 bit changes a copy destination to CFIFO2 or TXQ2 by routing.

If this bit is set to 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ2.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO2.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the CAN-FD module is in CH_RESET or CH_HALT mode.

### GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

You cannot write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTSn.RMIFL[0], CFDRFFDSTSn.RFIFL[0],CFDCFFDCSTSn.CFIFL[0]) of the storage location of an incoming message.

This bit is stored in CFDTHLACC1n.TIFL[0] when CFDTHLCCn.THLDGE = 1 is set up using the gateway function.

### GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

CFDRMNB.NRXMB[7:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0n.GAFLRMDP[6:0] bits should only be between 0x00 and CFDMNB.NMB[7:0] to 1 less.

If CFDRMNB.NRXMB[7:0] = 0x00, the GAFLRMV bit should be configured as 0.

### GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

### GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

## 32.2.23 CFDGAFLP1n : Global Acceptance Filter List Pointer 1 Registers (n = 1 to 16)

Base address: CANFD = 0x400B_0000

Offset address: 0x180C + 0x10 × (n - 1)

| Bit position: | 31 | | | | | | | | | | | | | | | | | 13 | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | GAFLFDP[13:0] | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 13:0 | GAFLFDP[13:0] | Global Acceptance Filter List FIFO Direction Pointer<br>FIFO direction pointer bits for received message storage | R/W |
| 31:14 | — | These bits are read as 0. The write value should be 0. | R/W |

The Global Acceptance Filter List Pointer 1 registers are used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

### GAFLFDP[13:0] bits (Global Acceptance Filter List FIFO Direction Pointer)

The GAFLFDP[13:0] bits allow the configuration of FIFO buffers as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. Each bit of the CFDGAFLP1n.GAFLFDP[13:0] is configured as dedicated FIFO.

| Bit | Value (binary) | Function |
|---|---|---|
| 0 | 0 | Disable RX FIFO 0 as target for reception |
| | 1 | Enable RX FIFO 0 as target for reception |
| 1 | 0 | Disable RX FIFO 1 as target for reception |
| | 1 | Enable RX FIFO 1 as target for reception |
| 2 | 0 | Disable RX FIFO 2 as target for reception |
| | 1 | Enable RX FIFO 2 as target for reception |
| 3 | 0 | Disable RX FIFO 3 as target for reception |
| | 1 | Enable RX FIFO 3 as target for reception |
| 4 | 0 | Disable RX FIFO 4 as target for reception |
| | 1 | Enable RX FIFO 4 as target for reception |
| 5 | 0 | Disable RX FIFO 5 as target for reception |
| | 1 | Enable RX FIFO 5 as target for reception |
| 6 | 0 | Disable RX FIFO 6 as target for reception |
| | 1 | Enable RX FIFO 6 as target for reception |
| 7 | 0 | Disable RX FIFO 7 as target for reception |
| | 1 | Enable RX FIFO 7 as target for reception |
| 8 | 0 | Disable Common FIFO 0 and Channel 0 TX Queue 0 as target for reception |
| | 1 | GAFLSRD0 = 0: Enable Common FIFO 0 as target for reception<br>GAFLSRD0 = 1: Enable Channel 0 TX Queue 0 as target for reception |
| 9 | 0 | Disable Common FIFO 1 and Channel 0 TX Queue 1 as target for reception |
| | 1 | GAFLSRD1 = 0: Enable Common FIFO 1 as target for reception<br>GAFLSRD1 = 1: Enable Channel 0 TX Queue 1 as target for reception |
| 10 | 0 | Disable Common FIFO 2 and Channel 0 TX Queue 2 as target for reception |
| | 1 | GAFLSRD2 = 0: Enable Common FIFO 2 as target for reception<br>GAFLSRD2 = 1: Enable Channel 0 TX Queue 2 as target for reception |

| Bit | Value (binary) | Function |
|---|---|---|
| 11 | 0 | Disable Common FIFO 3 and Channel 1 TX Queue 0 as target for reception |
| | 1 | GAFLSRD0 = 0: Enable Common FIFO 3 as target for reception<br>GAFLSRD0 = 1: Enable Channel 1 TX Queue 0 as target for reception |
| 12 | 0 | Disable Common FIFO 4 and Channel 1 TX Queue 1 as target for reception |
| | 1 | GAFLSRD1 = 0: Enable Common FIFO 4 as target for reception<br>GAFLSRD1 = 1: Enable Channel 1 TX Queue 1 as target for reception |
| 13 | 0 | Disable Common FIFO 5 and Channel 1 TX Queue 2 as target for reception |
| | 1 | GAFLSRD2 = 0: Enable Common FIFO 5 as target for reception<br>GAFLSRD2 = 1: Enable Channel 1 TX Queue 2 as target for reception |

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0. Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

For storage in TX queue, TX queue buffers of a target that is in GW mode is possible.

For storage in TX Queue, when these TX Queue buffers of a target are in GW mode, it can do.

Only one of the following configurations is valid:

- Up to 8 destination FIFO buffers
- 7 destination FIFO buffers plus one RX message buffer
- 8 destination TX queue buffers
- 7 destination TX queue buffers plus one RX message buffer
- A maximum of 8 destinations in all at FIFO buffer and TX queue buffer.

## 32.2.24 CFDRMNB : RX Message Buffer Number Register

Base address: CANFD = 0x400B_0000

Offset address: 0x00AC

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | 10 | | 8 | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RMPLS[2:0] | | | NRXMB[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | NRXMB[7:0] | Number of RX Message Buffers | R/W |
| 10:8 | RMPLS[2:0][*1] | Reception Message Buffer Payload Data Size<br>0 0 0: 8 bytes<br>0 0 1: 12 bytes<br>0 1 0: 16 bytes<br>0 1 1: 20 bytes<br>1 0 0: 24 bytes<br>1 0 1: 32 bytes<br>1 1 0: 48 bytes<br>1 1 1: 64 bytes | R/W |
| 31:11 | — | These bits are read as 0. The write value should be 0. | R/W |

Note 1. These bits are not available in the classical CAN function.

The RX Message Buffer Number register is used to configure the total number of RX message buffers allocated to all channels.

### NRXMB[7:0] bits (Number of RX Message Buffers)

The NRXMB[7:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Enter only values between 0 and 32 inclusive, with 0x00 indicating that no RX message buffer is allocated.

#### RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

### 32.2.25 CFDRMNDn : RX Message Buffer New Data Register n (n = 0)

Base address: CANFD = 0x400B_0000

Offset address: 0x00B0

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | RMNS[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 31:0 | RMNS[31:0] | RX Message Buffer New Data Status<br>0: New data not stored in corresponding RX message buffer<br>1: New data stored in corresponding RX message buffer | R/W |

The RX Message Buffer New Data Status Register (n = 0) specifies the new data storage status of the RX message buffers.

#### RMNS[31:0] bits (RX Message Buffer New Data Status)

The RMNSu[31:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

The bit position of CFDRMND corresponds to the buffer number of RXMB.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode. Writing 1 has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.
Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0. These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 PCLKB cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 PCLKB cycles + 1 for each 4 bytes (maximum of 20 PCLKB cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

### 32.2.26 CFDRFCCn : RX FIFO Configuration/Control Registers n (n = 0 to 7)

Base address: CANFD = 0x400B_0000

Offset address: 0x00C0 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RFFIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | RFIGCV[2:0] | | | RFIM | — | RFDC[2:0] | | | — | RFPLS[2:0] | | | — | — | RFIE | RFE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | RFE | RX FIFO Enable<br>0: FIFO disabled<br>1: FIFO enabled | R/W |
| 1 | RFIE | RX FIFO Interrupt Enable<br>0: FIFO interrupt generation disabled<br>1: FIFO interrupt generation enabled | R/W |
| 3:2 | — | These bits are read as 0. The write value should be 0. | R/W |
| 6:4 | RFPLS[2:0][*1] | Rx FIFO Payload Data Size Configuration<br>0 0 0: 8 bytes<br>0 0 1: 12 bytes<br>0 1 0: 16 bytes<br>0 1 1: 20 bytes<br>1 0 0: 24 bytes<br>1 0 1: 32 bytes<br>1 1 0: 48 bytes<br>1 1 1: 64 bytes | R/W |
| 7 | — | This bit is read as 0. The write value should be 0. | R/W |
| 10:8 | RFDC[2:0] | RX FIFO Depth Configuration<br>0 0 0: FIFO Depth = 0 message<br>0 0 1: FIFO Depth = 4 messages<br>0 1 0: FIFO Depth = 8 messages<br>0 1 1: FIFO Depth = 16 messages<br>1 0 0: FIFO Depth = 32 essages<br>1 0 1: FIFO Depth = 48 messages<br>1 1 0: FIFO Depth = 64 messages<br>1 1 1: FIFO Depth = 128 messages | R/W |
| 11 | — | This bit is read as 0. | R |
| 12 | RFIM | RX FIFO Interrupt Mode<br>0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV<br>1: Interrupt generated at the end of every received message storage | R/W |
| 15:13 | RFIGCV[2:0] | RX FIFO Interrupt Generation Counter Value<br>0 0 0: Interrupt generated when FIFO is 1/8th full<br>0 0 1: Interrupt generated when FIFO is 1/4th full<br>0 1 0: Interrupt generated when FIFO is 3/8th full<br>0 1 1: Interrupt generated when FIFO is 1/2 full<br>1 0 0: Interrupt generated when FIFO is 5/8th full<br>1 0 1: Interrupt generated when FIFO is 3/4th full<br>1 1 0: Interrupt generated when FIFO is 7/8th full<br>1 1 1: Interrupt generated when FIFO is full | R/W |
| 16 | RFFIE | RX FIFO Full Interrupt Enable<br>0: FIFO interrupt generation disabled<br>1: FIFO interrupt generation enabled | R/W |
| 31:17 | — | These bits are read as 0. | R |

Note 1. These bits are not available in the classical CAN function.

The RX FIFO Configuration/Control Registers (n = 0 to 7) are used to configure and control the eight RX FIFOs.

### RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the RX FIFO is cleared to empty.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDRFCCn.RFDC > 0x000).

Set the RFE bit with a separate write access to the CFDRFCCn register, after all the other bits in the CFDRFCCn register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

### RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

### RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

### RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

### RFIM bit (RX FIFO Interrupt Mode)

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

### RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

### RFFIE bit (RX FIFO Full Interrupt Enable)

The RFFIE bit enables generation of the RXFIFO full interrupt. Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

- Interruption output in number of arbitrary stages (CFDRFCCn.RFIGCV)
- Interruption output in FIFO full state.

Note: Management of the receiving data of FIFO can be performed by these notices of interruption.

## 32.2.27 CFDRFSTSn : RX FIFO Status Registers n (n = 0 to 7)

Base address: CANFD = 0x400B_0000

Offset address: 0x00E0 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RFFIF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | RFMC[7:0] | | | | | | | | — | — | — | — | RFIF | RFMLT | RFFLL | RFEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | RFEMP | RX FIFO Empty<br>0: FIFO not empty<br>1: FIFO empty | R |
| 1 | RFFLL | RX FIFO Full<br>0: FIFO not full<br>1: FIFO full | R |
| 2 | RFMLT | RX FIFO Message Lost<br>0: No message lost in FIFO<br>1: FIFO message lost | R/W |
| 3 | RFIF | RX FIFO Interrupt Flag<br>0: FIFO interrupt condition not satisfied<br>1: FIFO interrupt condition satisfied | R/W |
| 7:4 | — | These bits are read as 0. The write value should be 0. | R/W |
| 15:8 | RFMC[7:0] | RX FIFO Message Count<br>Number of messages stored in FIFO | R |
| 16 | RFFIF | RX FIFO Full Interrupt Flag<br>0: FIFO full interrupt condition not satisfied<br>1: FIFO full interrupt condition satisfied | R/W |
| 31:17 | — | These bits are read as 0. The write value should be 0. | R/W |

The RX FIFO Status Registers (n = 0 to 7) show the status of messages stored in the corresponding FIFO buffers.

### RFEMP bit (RX FIFO Empty)

The RFEMP bit is set automatically when:

- The RFMC bit is 0
- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to is 0
- The CAN-FD module is in GL_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

### RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth
- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to 0
- The CAN-FD module is in GL_RESET mode.

### RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode.

### RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0 to it. The bit is also cleared when CAN-FD module is in GL_RESET mode.

### RFMC[7:0] bits (RX FIFO Message Count)

The RFMC[7:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled and when the CAN-FD module is in GL_RESET mode.

### RFFIF bit (RX FIFO Full Interrupt Flag)

The RFFIF bit is not cleared automatically when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the FIFO full interrupt condition is satisfied. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared by writing 0 to it.

The bit is also cleared when the CAN-FD module is in GL_RESET mode.

## 32.2.28 CFDRFPCTRn : RX FIFO Pointer Control Registers n (n = 0 to 7)

Base address: CANFD = 0x400B_0000

Offset address: 0x0100 + 0x04 × n

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RFPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RFPC[7:0] | RX FIFO Pointer Control<br>Increments read pointer of the corresponding RX FIFO buffers | W |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The RX FIFO Pointer Control Registers (n = 0 to 7) can be used to increment the read pointer of the corresponding RX FIFO buffers.

### RFPC bits (RX FIFO Pointer Control)

When the value 0xFF is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write 0xFF to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 0x00.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

### 32.2.29 CFDCFCCn : Common FIFO Configuration/Control Registers n (n = 0 to 5)

Base address: CANFD = 0x400B_0000

Offset address: 0x0120 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | CFITT[7:0] | | | | | | | | CFDC[2:0] | | | CFTML[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | CFIGCV[2:0] | | | CFIM | CFITR | CFITSS | CFM[1:0] | | — | CFPLS[2:0] | | | — | CFTXIE | CFRXIE | CFE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | CFE | Common FIFO Enable<br>0: FIFO disabled<br>1: FIFO enabled | R/W |
| 1 | CFRXIE | Common FIFO RX Interrupt Enable<br>0: FIFO interrupt generation disabled for Frame RX<br>1: FIFO interrupt generation enabled for Frame RX | R/W |
| 2 | CFTXIE | Common FIFO TX Interrupt Enable<br>0: FIFO interrupt generation disabled for Frame TX<br>1: FIFO interrupt generation enabled for Frame TX | R/W |
| 3 | — | This bit is read as 0. The write value should be 0. | R/W |
| 6:4 | CFPLS[2:0][*1] | Common FIFO Payload Data Size Configuration<br>0 0 0: 8 bytes<br>0 0 1: 12 bytes<br>0 1 0: 16 bytes<br>0 1 1: 20 bytes<br>1 0 0: 24 bytes<br>1 0 1: 32 bytes<br>1 1 0: 48 bytes<br>1 1 1: 64 bytes | R/W |
| 7 | — | This bit is read as 0. The write value should be 0. | R/W |
| 9:8 | CFM[1:0] | Common FIFO Mode<br>0 0: RX FIFO mode<br>0 1: TX FIFO mode<br>1 0: CAN – CAN GW FIFO mode<br>1 1: Reserved | R/W |
| 10 | CFITSS | Common FIFO Interval Timer Source Select<br>0: Reference clock (× 1 / × 10 period)<br>1: Bit time clock of related channel (FIFO is linked to fixed channel) | R/W |
| 11 | CFITR | Common FIFO Interval Timer Resolution<br>0: Reference clock period × 1<br>1: Reference clock period × 10 | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 12 | CFIM | Common FIFO Interrupt Mode<br>0: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFIGCV value from a lower value<br>TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully<br>GW FIFO mode:<br>For RX interrupt flag:<br>Interrupt generated when FIFO counter increments and reaches the value configured in CFIGCV<br>For TX interrupt flag:<br>Interrupt generated when FIFO transmits the last message successfully<br>1: RX FIFO mode: RX interrupt generated at the end of every received message storage<br>TX FIFO mode: interrupt generated for every successfully transmitted message<br>GW FIFO mode:<br>For RX interrupt flag:<br>Interrupt generated when a message is stored in the FIFO<br>For TX interrupt flag:<br>Interrupt generated when a message is successfully transmitted from the FIFO | R/W |
| 15:13 | CFIGCV[2:0] | Common FIFO Interrupt Generation Counter Value<br>0 0 0: Interrupt generated when FIFO is 1/8th full<br>0 0 1: Interrupt generated when FIFO is 1/4th full<br>0 1 0: Interrupt generated when FIFO is 3/8th full<br>0 1 1: Interrupt generated when FIFO is 1/2 full<br>1 0 0: Interrupt generated when FIFO is 5/8th full<br>1 0 1: Interrupt generated when FIFO is 3/4th full<br>1 1 0: Interrupt generated when FIFO is 7/8th full<br>1 1 1: Interrupt generated when FIFO is full | R/W |
| 20:16 | CFTML[4:0] | Common FIFO TX Message Buffer Link<br>Transmission scan link position of the corresponding channel | R/W |
| 23:21 | CFDC[2:0] | Common FIFO Depth Configuration<br>0 0 0: FIFO Depth = 0 messages<br>0 0 1: FIFO Depth = 4 messages<br>0 1 0: FIFO Depth = 8 messages<br>0 1 1: FIFO Depth = 16 messages<br>1 0 0: FIFO Depth = 32 messages<br>1 0 1: FIFO Depth = 48 messages<br>1 1 0: FIFO Depth = 64 messages<br>1 1 1: FIFO Depth = 128 messages | R/W |
| 31:24 | CFITT[7:0] | Common FIFO Interval Transmission Time<br>Delay the start of transmission from the FIFO if configured in TX or GW mode, delay is a multiple of basic Interval Timer Clock Source unit | R/W |

Note 1. These bits are not available in the classical CAN function.

### CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode or GW mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.

This bit can only be set if the configured FIFO depth is greater than 0 (CFDC bit > 0).

Set the CFE bit with a separate write access to the CFDCFCCn register, after all the other bits in this register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

This bit is also cleared automatically when the related channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

### CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

### CFTXIE bit (Common FIFO TX Interrupt Enable)

The CFTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

### CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see section 32.6. FIFO Buffers and Normal Message Buffer Configuration.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

### CFM[1:0] bits (Common FIFO Mode)

The CFM[1:0] bits select the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode. Do not configure these bits to 11b.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

### CFITSS bit (Common FIFO Interval Timer Source Select)

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1.

Do not write 1 to this bit when CAN-FD communication is used.*1

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

### CFITR bit (Common FIFO Interval Timer Resolution)

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.

### CFIM bit (Common FIFO Interrupt Mode)

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

### CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

### CFTML[4:0] bits (Common FIFO TX Message Buffer Link)

The CFTML[4:0] bits select the normal transmit message buffer position where the TX or GW FIFO is linked to, for transmission scanning.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

### CFDC[2:0] bits (Common FIFO Depth Configuration)

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

### CFITT[7:0] bits (Common FIFO Interval Transmission Time)

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX or GW mode. The delay is a multiple of the basic interval timer clock source period (reference clock × 1, reference clock × 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1.

When CFDGCFG.ITRCP[15:0] = 0x0000, set the CFITT[7:0] bits to 0x0000.

## 32.2.30 CFDCFCCEn : Common FIFO Configuration/Control Enhancement Registers n (n = 0 to 5)

Base address: CANFD = 0x400B_0000

Offset address: 0x0180 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFBME |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | CFMOWM | — | — | — | — | — | CFOFTXIE | CFOFRXIE | CFFIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | CFFIE | Common FIFO Full Interrupt Enable<br>0: FIFO Interrupt generation disabled<br>1: FIFO Interrupt generation enabled | R/W |
| 1 | CFOFRXIE | Common FIFO One Frame Reception Interrupt Enable<br>0: One Frame RX interrupt generation disabled<br>1: One Frame RX interrupt generation enabled | R/W |
| 2 | CFOFTXIE | Common FIFO One Frame Transmission Interrupt Enable<br>0: One Frame TX interrupt generation disabled<br>1: One Frame TX interrupt generation enabled | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | CFMOWM | Common FIFO Message Overwrite Mode<br>0: Message discarded mode<br>1: Message overwrite mode | R/W |
| 15:9 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | CFBME | Common FIFO Buffering Mode Enable<br>0: Transmission from Common FIFO<br>1: Transmission halt from Common FIFO | R/W |
| 31:17 | — | These bits are read as 0. The write value should be 0. | R/W |

The Common FIFO Configuration/Control Enhancement Registers (n = 0 to 5) are used to configure the Common FIFOs.

no_of_channels = 2

no_of_CFIFOs_per_channel = Number of Common FIFOs per channel = 3

Where the total number of CFIFOs = no_of_CFIFOs_per_channel × no_of_channels = 3 × 2 = 6 as shown in Figure 32.28.

n = Common FIFO index = [0 .. no_of_CFIFOs - 1]

### CFFIE bit (Common FIFO Full Interrupt Enable)

The CFFIE bit enables generation of the FIFO full interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

1. Interruption output in number of arbitrary stages (CFDCFCCn.CFIGCV)
2. Interruption output in FIFO full state.

Management of the receiving data of FIFO can be performed by these notices of interruption.

### CFOFRXIE bit (Common FIFO One Frame Reception Interrupt Enable)

The CFOFRXIE bit enables generation of the one frame reception interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### CFOFTXIE bit (Common FIFO One Frame Transmission Interrupt Enable)

The CFOFTXIE bit enables generation of the one frame transmission interrupt when the interrupt flag is set after transmission of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

### CFMOWM bit (Common FIFO Message Overwrite Mode)

When the CFMOWM bit is 0, a receiving message is discarded and FIFO is full. When the CFMOWM bit is 1, a receiving message is overwritten and FIFO is full.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Only write 1 to this bit when the common FIFO is in GW mode.

Do not write 1 to this bit when the CFE bit is 1.

### CFBME bit (Common FIFO Buffering Mode Enable)

When the CFBME bit is 0, messages are transmitted from FIFO. When the CFBME bit is 1, messages are not transmitted from FIFO.

Do not write to this bit when the CANFD module is in GL_SLEEP mode. Additionally, do not write 1 to this bit when the CFE bit is 1.

## 32.2.31 CFDCFSTSn : Common FIFO Status Registers n (n = 0 to 5)

Base address: CANFD = 0x400B_0000

Offset address: 0x01E0 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | CFMOW | — | — | — | — | — | CFOFTXIF | CFOFRXIF | CFFIF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | CFMC[7:0] | | | | | | | | — | — | — | CFTXIF | CFRXIF | CFMLT | CFFLL | CFEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | CFEMP | Common FIFO Empty<br>0: FIFO not empty<br>1: FIFO empty | R |
| 1 | CFFLL | Common FIFO Full<br>0: FIFO not full<br>1: FIFO full | R |
| 2 | CFMLT | Common FIFO Message Lost<br>0: Number of message lost in FIFO<br>1: FIFO message lost | R/W |
| 3 | CFRXIF | Common RX FIFO Interrupt Flag<br>0: FIFO interrupt condition not satisfied after frame reception<br>1: FIFO interrupt condition satisfied after frame reception | R/W |
| 4 | CFTXIF | Common TX FIFO Interrupt Flag<br>0: FIFO interrupt condition not satisfied after frame transmission<br>1: FIFO Interrupt condition satisfied after frame transmission | R/W |
| 7:5 | — | These bits are read as 0. The write value should be 0. | R/W |
| 15:8 | CFMC[7:0] | Common FIFO Message Count<br>Number of messages stored in FIFO | R |
| 16 | CFFIF | Common FIFO Full Interrupt Flag<br>0: Interrupt condition not satisfied for FIFO full interrupt<br>1: Interrupt condition satisfied for FIFO full interrupt | R/W |
| 17 | CFOFRXIF | Common FIFO One Frame Reception Interrupt Flag<br>For each FIFO that receives a frame, a corresponding interrupt is set. | R/W |
| 18 | CFOFTXIF | Common FIFO One Frame Transmission Interrupt Flag<br>For each FIFO that transmits a frame, a corresponding interrupt is set. | R/W |
| 23:19 | — | These bits are read as 0. The write value should be 0. | R/W |
| 24 | CFMOW | Common FIFO Message Overwrite<br>0: No message overwrite occurred in FIFO<br>1: Message overwrite occurred in FIFO | R/W |
| 31:25 | — | These bits are read as 0. The write value should be 0. | R/W |

### CFEMP bit (Common FIFO Empty)

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX or GW mode
- The FIFO is disabled by setting the CFE bit to 0
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET when FIFO configured in TX or GW mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX or GW mode.

### CFFLL bit (Common FIFO Full)

The CFFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0
- The CAN-FD module is in GL_RESET mode

- The related CAN-FD channel is in CH_RESET mode when FIFO buffer is configured in TX or GW mode.

### CFMLT bit (Common FIFO Message Lost)

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX or GW mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMLT bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

### CFRXIF bit (Common RX FIFO Interrupt Flag)

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in GW mode or RX mode.

The CFRXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in GW mode.

### CFTXIF bit (Common TX FIFO Interrupt Flag)

The CFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in GW or TX mode.

The CFTXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

### CFMC[7:0] bits (Common FIFO Message Count)

The CFMC[7:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by RS-CAN-FD to be read by the CPU
- Number of CAN messages stored by the RS-CAN-FD in the GW FIFO pending for transmission.

The CFMC[7:0] bits are cleared automatically when:

- The FIFO is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

### CFFIF bit (Common FIFO Full Interrupt Flag)

The CFFIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the FIFO Full Interrupt condition is satisfied for Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFFIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

### CFOFRXIF bit (Common FIFO One Frame Reception Interrupt Flag)

The CFOFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for the FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the One Frame Reception Interrupt condition is satisfied for the Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFOFRXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

### CFOFTXIF bit (Common FIFO One Frame Transmission Interrupt Flag)

The CFOFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the One Frame Transmission Interrupt condition is satisfied for Common FIFO buffers configured in GW mode or TX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

### CFMOW bit (Common FIFO Message Overwrite)

The CFMOW bit is set automatically whenever a message is an overwrite storage of a new message when CFDCFCCEn.CFMOWM = 1 and FIFO is already full in GW mode.

If the set from the CAN channel occurs simultaneously with the clear by a write access, the bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMOW bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

## 32.2.32 CFDCFPCTRn : Common FIFO Pointer Control Registers n (n = 0 to 5)

Base address: CANFD = 0x400B_0000

Offset address: 0x0240 + 0x04 × n

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | CFPC[7:0] | Common FIFO Pointer Control<br>Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration. | W |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The Common FIFO Pointer Control Registers (n = 0 to 5) can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

### CFPC[7:0] bits (Common FIFO Pointer Control)

When the value 0xFF is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 0x00.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Only write 0xFF to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode
- The Common FIFO buffer is enabled and is not configured in GW mode.

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

## 32.2.33 CFDFESTS : FIFO Empty Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02A0

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | 13 | | | | | 8 | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXEMP[5:0] | | | | | | RFXEMP[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RFXEMP[7:0] | RX FIFO Empty Status<br>0: Corresponding FIFO not empty<br>1: Corresponding FIFO empty | R |
| 13:8 | CFXEMP[5:0] | Common FIFO Empty Status<br>0: Corresponding FIFO not empty<br>1: Corresponding FIFO empty | R |
| 31:14 | — | These bits are read as 0. The write value should be 0. | R |

The FIFO Empty Status register shows status of the empty bits of the FIFO buffers.

### RFXEMP[7:0] bits (RX FIFO Empty Status)

Bit 7 (RFXEMP[7]) is associated with FIFO index 7 and bit 0 (RFXEMP[0]) is associated with FIFO index 0.

The RFXEMP[7:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

### CFXEMP[5:0] bits (Common FIFO Empty Status)

Bit 13 (CFXEMP[5]) is associated with common FIFO index 5 and bit 8 (CFXEMP[0]) is associated with common FIFO index 0.

The CFXEMP[5:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

## 32.2.34 CFDFFSTS : FIFO Full Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02A4

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | 13 | | | | | 8 | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXFLL[5:0] | | | | | | RFXFLL[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RFXFLL[7:0] | RX FIFO Full Status<br>0: Corresponding FIFO not full<br>1: Corresponding FIFO full | R |
| 13:8 | CFXFLL[5:0] | Common FIFO Full Status<br>0: Corresponding FIFO not full<br>1: Corresponding FIFO full | R |
| 31:14 | — | These bits are read as 0. The write value should be 0. | R |

The FIFO Full Status Register shows status of the full bits of the FIFO buffers.

### RFXFLL[7:0] bits (RX FIFO Full Status)

The RFXFLL[7:0] bits are cleared when CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

### CFXFLL[5:0] bits (Common FIFO Full Status)

The CFXFLL[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

## 32.2.35 CFDFMSTS : FIFO Message Lost Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02A8

| Bit position: | 31 | | | | | | | | | | | | | | | | | | 13 8 | 7 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXMLT[5:0] | RFXMLT[7:0] |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RFXMLT[7:0] | RX FIFO Message Lost Status<br>0: Corresponding FIFO Message Lost flag not set<br>1: Corresponding FIFO Message Lost flag set | R |
| 13:8 | CFXMLT[5:0] | Common FIFO Message Lost Status<br>0: Corresponding FIFO Message Lost flag not set<br>1: Corresponding FIFO Message Lost flag set | R |
| 31:14 | — | These bits are read as 0. The write value should be 0. | R |

The FIFO Message Lost Status Register shows status of the Msg Lost bits of the FIFO buffers.

### RFXMLT[7:0] bits (RX FIFO Message Lost Status)

The RFXMLT[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

### CFXMLT[5:0] bits (Common FIFO Message Lost Status)

The CFXMLT[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

## 32.2.36 CFDRFISTS : RX FIFO Interrupt Flag Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02AC

| Bit position: | 31 | | | | | | | | 23 16 | | | | | | | | | 7 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | RFXFFLL[7:0] | — | — | — | — | — | — | — | — | RFXIF[7:0] |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RFXIF[7:0] | RX FIFO[x] Interrupt Flag Status<br>0: Corresponding RX FIFO Interrupt flag not set<br>1: Corresponding RX FIFO Interrupt flag set | R |
| 15:8 | — | These bits are read as 0. The write value should be 0. | R/W |
| 23:16 | RFXFFLL[7:0] | RX FIFO[x] Interrupt Full Flag Status<br>0: Corresponding RX FIFO Interrupt Full flag not set<br>1: Corresponding RX FIFO Interrupt Full flag set | R |
| 31:24 | — | These bits are read as 0. The write value should be 0. | R/W |

The FIFO Interrupt Flag Status Register shows status of the interrupt flag bits of the RX FIFO buffers.

### RFXIF[7:0] bits (RX FIFO[x] Interrupt Flag Status)

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

### RFXFFLL[7:0] bits (RX FIFO[x] Interrupt Full Flag Status)

Each bit is set automatically when the corresponding interrupt full flag bit is set in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt full flag bit is cleared in the RX FIFO Status Registers.

## 32.2.37 CFDCFRISTS : Common FIFO RX Interrupt Flag Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02B0

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | 6 | 5 | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXRXIF[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 5:0 | CFXRXIF[5:0] | Common FIFO RX Interrupt Flag Status<br>0: Corresponding Common FIFO RX Interrupt flag is not set<br>1: Corresponding Common FIFO RX Interrupt flag is set | R |
| 31:6 | — | These bits are read as 0. The write value should be 0. | R/W |

The Common FIFO RX Interrupt Flag Status Register shows status of the interrupt flag bits of the Common FIFO buffers.

### CFXRXIF[5:0] bits (Common FIFO RX Interrupt Flag Status)

Each bit is set automatically when the corresponding RX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXRXIF[5:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding RX interrupt flag bit is cleared in the Common FIFO Status Registers.

## 32.2.38 CFDCFTISTS : Common FIFO TX Interrupt Flag Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02B4

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | 6 | 5 | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXTXIF[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 5:0 | CFXTXIF[5:0] | Common FIFO [x] TX Interrupt Flag Status<br>0: Corresponding Common FIFO TX Interrupt flag is not set<br>1: Corresponding Common FIFO TX Interrupt flag is set | R |
| 31:6 | — | These bits are read as 0. The write value should be 0. | R/W |

The Common FIFO TX Interrupt Flag Status Register shows status of the interrupt flag bits of the Common FIFO buffers.

### CFXTXIF[5:0] bits (Common FIFO [x] TX Interrupt Flag Status)

Each bit is set automatically when the corresponding TX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXTXIF[5:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding TX interrupt flag bit is cleared in the Common FIFO Status Registers.

## 32.2.39 CFDFFFSTS : FIFO FDC Full Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02C4

| Bit position: | 31 | | | | | | | | | | | | | | | | | | 13 | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXFFLL[5:0] | | | | | | RFXFFLL[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RFXFFLL[7:0] | RX FIFO FDC Level Full Status<br>0: Corresponding FIFO full interrupt not set<br>1: Corresponding FIFO full interrupt is set | R |
| 13:8 | CFXFFLL[5:0] | COMMON FIFO FDC Level Full Status<br>0: Corresponding FIFO full interrupt not set<br>1: Corresponding FIFO full interrupt is set | R |
| 31:14 | — | These bits are read as 0. The write value should be 0. | R |

The FIFO FDC Full Status Register shows status of the full interrupt flag bits of the FIFO buffers.

### RFXFFLL[7:0] bits (RX FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CANFD module is in GL_RESET mode.

### CFXFFLL[5:0] bits (COMMON FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

The CFXFFLL[5:0] bits are cleared when the CANFD module is in GL_RESET mode.

## 32.2.40 CFDCFMOWSTS : Common FIFO Message Over Write Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02C0

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | | 5 | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXMOW[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 5:0 | CFXMOW[5:0] | Common FIFO Massage Overwrite Status<br>0: Corresponding FIFO Overwrite flag is not set<br>1: Corresponding FIFO Overwrite flag is set | R |
| 31:6 | — | These bits are read as 0. The write value should be 0. | R |

### CFXMOW[5:0] bits (Common FIFO Massage Overwrite Status)

The CFXMOW[5:0] bits are cleared when the CANFD module is in GL_RESET mode. This register is only valid in GW mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

## 32.2.41 CFDCFOFRISTS : Common FIFO One Frame RX Interrupt Flag Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02B8

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | | 5 | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXOFRXIF[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 5:0 | CFXOFRXIF[5:0] | Common FIFO One Frame RX Interrupt Flag Status<br>0: Corresponding Common FIFO One Frame RX Interrupt flag is not set<br>1: Corresponding Common FIFO One Frame RX Interrupt flag is set | R |
| 31:6 | — | These bits are read as 0. The write value should be 0. | R |

The Common FIFO One Frame RX Interrupt Flag Status register shows status of the interrupt flag bits of the Common FIFO buffers.

### CFXOFRXIF[5:0] bits (Common FIFO One Frame RX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame RX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFRXIF[5:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame RX Interrupt flag bit is cleared in the Common FIFO Status Registers.

## 32.2.42 CFDCFOFTISTS : Common FIFO One Frame TX Interrupt Flag Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x02BC

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | | 5 | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFXOFTXIF[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 5:0 | CFXOFTXIF[5:0] | Common FIFO One Frame TX Interrupt Flag Status<br>0: Corresponding Common FIFO One Frame TX Interrupt flag is not set<br>1: Corresponding Common FIFO One Frame TX Interrupt flag is set | R |
| 31:6 | — | These bits are read as 0. The write value should be 0. | R |

The Common FIFO One Frame TX Interrupt Flag Status Register shows status of the interrupt flag bits of the Common FIFO buffers.

### CFXOFTXIF[5:0] bits (Common FIFO One Frame TX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame TX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFTXIF[5:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame TX Interrupt flag bit is cleared in the Common FIFO Status Registers.

## 32.2.43 CFDCDTCT : DMA Transfer Control Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1330

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field: | — | — | — | — | — | — | CFDM AE1 | CFDM AE0 | RFDM AE7 | RFDM AE6 | RFDM AE5 | RFDM AE4 | RFDM AE3 | RFDM AE2 | RFDM AE1 | RFDM AE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | RFDMAE0 | DMA Transfer Enable for RXFIFO 0<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 1 | RFDMAE1 | DMA Transfer Enable for RXFIFO 1<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 2 | RFDMAE2 | DMA Transfer Enable for RXFIFO 2<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 3 | RFDMAE3 | DMA Transfer Enable for RXFIFO 3<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 4 | RFDMAE4 | DMA Transfer Enable for RXFIFO 4<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 5 | RFDMAE5 | DMA Transfer Enable for RXFIFO 5<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 6 | RFDMAE6 | DMA Transfer Enable for RXFIFO 6<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 7 | RFDMAE7 | DMA Transfer Enable for RXFIFO 7<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 8 | CFDMAE0 | DMA Transfer Enable for Common FIFO 0 of Channel 0<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 9 | CFDMAE1 | DMA Transfer Enable for Common FIFO 0 of Channel 1<br>0: DMA transfer request disabled<br>1: DMA transfer request enabled | R/W |
| 31:10 | — | These bits are read as 0. The write value should be 0. | R/W |

The DMA Transfer Control Register controls the start and stop of DMA transfer operation.

### RFDMAEn (n = 0 to 7) bit (DMA Transfer Enable for RXFIFOn)

The RFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

### CFDMAEn (n = 0, 1) bit (DMA Transfer Enable for Common FIFO 0 of Channel 0, 1)

The CFDMAEn bit enables or disables DMA transfer request for common FIFO 0 of channel 0 or 1. Only Common FIFO 0 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel.

To link Common FIFO 2, see bit CFDCDTTCT.CFDMAEn in section 32.2.45. CFDCDTTCT : DMA TX Transfer Control Register.

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX or GW FIFO.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

## 32.2.44 CFDCDTSTS : DMA Transfer Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1334

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | CFDMASTS1 | CFDMASTS0 | RFDMASTS7 | RFDMASTS6 | RFDMASTS5 | RFDMASTS4 | RFDMASTS3 | RFDMASTS2 | RFDMASTS1 | RFDMASTS0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | RFDMASTS0 | DMA Transfer Status for RX FIFO 0<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 1 | RFDMASTS1 | DMA Transfer Status for RX FIFO 1<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 2 | RFDMASTS2 | DMA Transfer Status for RX FIFO 2<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 3 | RFDMASTS3 | DMA Transfer Status for RX FIFO 3<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 4 | RFDMASTS4 | DMA Transfer Status for RX FIFO 4<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 5 | RFDMASTS5 | DMA Transfer Status for RX FIFO 5<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 6 | RFDMASTS6 | DMA Transfer Status for RX FIFO 6<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 7 | RFDMASTS7 | DMA Transfer Status for RX FIFO 7<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 8 | CFDMASTS0 | DMA Transfer Status only for Common FIFO 0 of Channel 0<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 9 | CFDMASTS1 | DMA Transfer Status only for Common FIFO 0 of Channel 1<br>0: DMA transfer stopped<br>1: DMA transfer on going | R |
| 31:10 | — | These bits are read as 0. | R |

The DMA Transfer Status Register shows the status of the DMA transfer.

### RFDMASTSn (n = 0 to 7) bit (DMA Transfer Status for RX FIFO n)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEn (see CFDCDTCT.RFDMAEn bit in section 32.2.43. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the RFDMASTSn bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

### CFDMASTSn bit (DMA Transfer Status only for Common FIFO 0 of Channel 0, 1)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAEn (see CFDCDTCT.CFDMAEn bit in section 32.2.43. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the CFDMASTSn bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

## 32.2.45 CFDCDTTCT : DMA TX Transfer Control Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1340

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFDM AE1 | CFDM AE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | TQ3D MAE1 | TQ3D MAE0 | — | — | — | — | — | — | TQ0D MAE1 | TQ0D MAE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TQ0DMAE0 | DMA TX Transfer Enable for TXQ 0 of Channel 0<br>0: DMA TX transfer request disabled<br>1: DMA TX transfer request enabled | R/W |
| 1 | TQ0DMAE1 | DMA TX Transfer Enable for TXQ 0 of Channel 1<br>0: DMA TX transfer request disabled<br>1: DMA TX transfer request enabled | R/W |
| 7:2 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | TQ3DMAE0 | DMA TX Transfer Enable for TXQ 3 of Channel 0<br>0: DMA TX transfer request disabled<br>1: DMA TX transfer request enabled | R/W |
| 9 | TQ3DMAE1 | DMA TX Transfer Enable for TXQ 3 of Channel 1<br>0: DMA TX transfer request disabled<br>1: DMA TX transfer request enabled | R/W |
| 15:10 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | CFDMAE0 | DMA TX Transfer Enable for Common FIFO 2 of Channel 0<br>0: DMA TX transfer request disabled<br>1: DMA TX transfer request enabled | R/W |
| 17 | CFDMAE1 | DMA TX Transfer Enable for Common FIFO 2 of Channel 1<br>0: DMA TX transfer request disabled<br>1: DMA TX transfer request enabled | R/W |
| 31:18 | — | These bits are read as 0. The write value should be 0. | R/W |

The DMA TX Transfer Control Register controls the start and stop of DMA transfer operation.

### TQ0DMAEn (n = 0, 1) bit (DMA TX Transfer Enable for TXQ 0 of Channel n)

The TQ0DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CANFD module is in GL_RESET mode.

### TQ3DMAEn (n = 0, 1) bit (DMA TX Transfer Enable for TXQ 3 of Channel n)

The TQ3DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CANFD module is in GL_RESET mode.

### CFDMAEn (n = 0, 1) bit (DMA TX Transfer Enable for Common FIFO 2 of Channel n)

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Only common FIFO 2 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel.

To link Common FIFO 0, see CFDCDTCT.CFDMAEn bit in section 32.2.43. CFDCDTCT : DMA Transfer Control Register.

Do not enable a DMA transfer for a Common FIFO that is configured as RX or GW FIFO.

The CFDMAEn bit is cleared when the CANFD module is in GL_RESET mode.

## 32.2.46 CFDCDTTSTS : DMA TX Transfer Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1344

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFDM<br>ASTS1 | CFDM<br>ASTS0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field: | — | — | — | — | — | — | TQ3D<br>MAST<br>S1 | TQ3D<br>MAST<br>S0 | — | — | — | — | — | — | TQ0D<br>MAST<br>S1 | TQ0D<br>MAST<br>S0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TQ0DMASTS0 | DMA TX Transfer Status for TXQ0 of Channel 0<br>0: DMA transfer stopped<br>1: DMA transfer enabled | R |
| 1 | TQ0DMASTS1 | DMA TX Transfer Status for TXQ0 of Channel 1<br>0: DMA transfer stopped<br>1: DMA transfer enabled | R |
| 7:2 | — | These bits are read as 0. The write value should be 0. | R |
| 8 | TQ3DMASTS0 | DMA TX Transfer Status for TXQ3 of Channel 0<br>0: DMA transfer stopped<br>1: DMA transfer enabled | R |
| 9 | TQ3DMASTS1 | DMA TX Transfer Status for TXQ3 of Channel 1<br>0: DMA transfer stopped<br>1: DMA transfer enabled | R |
| 15:10 | — | These bits are read as 0. The write value should be 0. | R |
| 16 | CFDMASTS0 | DMA TX Transfer Status only for Common FIFO 2 of Channel 0<br>0: DMA transfer stopped<br>1: DMA transfer enabled | R |
| 17 | CFDMASTS1 | DMA TX Transfer Status only for Common FIFO 2 of Channel 1<br>0: DMA transfer stopped<br>1: DMA transfer enabled | R |
| 31:18 | — | These bits are read as 0. The write value should be 0. | R/W |

The DMA TX Transfer Status Register shows the status of the DMA transfer.

### TQ0DMASTSn (n = 0, 1) bit (DMA TX Transfer Status for TXQ 0 of Channel n)

The TQ0DMASTSn bit is set when the CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is set (see section 32.2.45. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CANFD module is in GL_RESET mode.

### TQ3DMASTSn (n = 0, 1) bit (DMA TX Transfer Status for TXQ 3 of Channel n)

The TQ3DMASTSn bit is set when the CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is set (see section 32.2.45. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is cleared

- The CANFD module is in GL_RESET mode.

### CFDMASTSn (n = 0, 1) bit (DMA TX Transfer Status only for Common FIFO 2 of Channel n)

The CFDMASTSn bit is set when the CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is set (see section 32.2.45. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CANFD module is in GL_RESET mode.

## 32.2.47 CFDGRINTSTSn : Global RX Interrupt Status Register n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1350 + 0x04 × n

| Bit position: | 31 | 30 | | 28 | | 26 | | 24 | | | | | | 18 | | 16 | | | | | | 10 | | 8 | | | | | | 2 | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | CFOFRIF[2:0] | | | — | CFRFRIF[2:0] | | | — | — | — | — | — | CFRIF[2:0] | | | — | — | — | — | — | QOFRIF[2:0] | | | — | — | — | — | — | QFIF[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2:0 | QFIF[2:0] | TXQ Full Interrupt Flag Channel n (n = 0, 1)<br>0: Corresponding TXQ Full Interrupt flag is not set<br>1: Corresponding TXQ Full Interrupt flag is set | R |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 10:8 | QOFRIF[2:0] | TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1)<br>0: Corresponding TXQ One Frame RX Interrupt flag is not set<br>1: Corresponding TXQ One Frame RX Interrupt flag is set | R |
| 15:11 | — | These bits are read as 0. The write value should be 0. | R/W |
| 18:16 | CFRIF[2:0] | Common FIFO RX Interrupt Flag Channel n (n = 0, 1)<br>0: Corresponding Common FIFO RX Interrupt flag is not set<br>1: Corresponding Common FIFO RX Interrupt flag is set | R |
| 23:19 | — | These bits are read as 0. The write value should be 0. | R/W |
| 26:24 | CFRFRIF[2:0] | Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0, 1)<br>0: Corresponding Common FIFO Full Interrupt flag is not set<br>1: Corresponding Common FIFO Full Interrupt flag is set | R |
| 27 | — | This bit is read as 0. The write value should be 0. | R/W |
| 30:28 | CFOFRIF[2:0] | Common FIFO One Frame RX Interrupt Flag Channel n (n = 0, 1)<br>0: Corresponding Common FIFO One Frame RX Interrupt flag is not set<br>1: Corresponding Common FIFO One Frame RX Interrupt flag is set | R |
| 31 | — | This bit is read as 0. The write value should be 0. | R/W |

### QFIF[2:0] bits (TXQ Full Interrupt Flag Channel n (n = 0, 1))

The QFIF[2:0] bits are set automatically when the TXQ Full Interrupt flag of the related channel is set when the interrupt is enabled.

The QFIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

### QOFRIF[2:0] bits (TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1))

The QOFRIF[2:0] bits are set automatically when the TXQ One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The QOFRIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled
- The CANFD module is in GL_RESET or CH_RESET mode.

#### CFRIF[2:0] bits (Common FIFO RX Interrupt Flag Channel n (n = 0, 1))

The CFRIF[2:0] bits are set automatically when the Common FIFO RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CANFD module is in GL_RESET or CH_RESET mode.

#### CFRFIF[2:0] bits (Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0, 1))

The CFRFIF[2:0] bits are set automatically when the Common FIFO Full Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRFIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

#### CFOFRIF[2:0] bits (Common FIFO One Frame RX Interrupt Flag Channel n (n = 0, 1))

The CFOFRIF[2:0] bits are set automatically when the Common FIFO One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFOFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

### 32.2.48 CFDTMCn : TX Message Buffer Control Registers n (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

Base address: CANFD = 0x400B_0000

Offset address: 0x02D0 + 0x01 × n

| Bit position: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | TMOM | TMTAR | TMTR |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TMTR | TX Message Buffer Transmission Request<br>0: TX Message buffer transmission not requested<br>1: TX message buffer transmission requested | R/W |
| 1 | TMTAR | TX Message Buffer Transmission Abort Request<br>0: TX message buffer transmission request abort not requested<br>1: TX message buffer transmission request abort requested | R/W |
| 2 | TMOM | TX Message Buffer One-shot Mode<br>0: TX message buffer not configured in one-shot mode<br>1: TX message buffer configured in one-shot mode | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Message Buffer Control Registers configure the TX message buffer functions.

### TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CAN-FD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CAN-FD module is in CH_HALT or CH_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX or GW mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSn.TMTRF) in the CFDTMSTSn register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CAN-FD module logic at the end of a successful transmission
- CAN-FD module logic at the end of a transmission abort, requested by the corresponding CFDTMCn.TMTAR bit
- CAN-FD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCn.TMOM bit is set for the message buffer
- CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

### TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CAN-FD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CAN-FD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CAN-FD module logic at the end of a successful transmission
- The CAN-FD module logic at the end of a transmission abort
- The CAN-FD module logic when there is detection of a CAN bus error or arbitration loss
- The CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel enters CH_RESET mode.

### TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CAN-FD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTSn.TMTRF bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTSn.TMTRF bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

## 32.2.49 CFDTMSTSn : TX Message Buffer Status Registers n (n = 0 to 7, 32 to 39, 64 to 71, 96 to 103)

Base address: CANFD = 0x400B_0000

Offset address: 0x07D0 + 0x01 × n

| Bit position: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | TMTARM | TMTRM | TMTRF[1:0] | | TMTSTS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TMTSTS | TX Message Buffer Transmission Status<br>0: No on-going transmission<br>1: On-going transmission | R |
| 2:1 | TMTRF[1:0] | TX Message Buffer Transmission Result Flag<br>0 0: No result<br>0 1: Transmission aborted from the TX message buffer<br>1 0: Transmission successful from the TX message buffer and transmission abort was not requested<br>1 1: Transmission successful from the TX message buffer and transmission abort was requested | R/W |
| 3 | TMTRM | TX Message Buffer Transmission Request Mirrored<br>0: TX message buffer transmission not requested<br>1: TX message buffer transmission requested | R |
| 4 | TMTARM | TX Message Buffer Transmission Abort Request Mirrored<br>0: TX message buffer transmission request abort not requested<br>1: TX message buffer transmission request abort requested | R |
| 7:5 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Message Buffer Status Registers show status of the transmission and transmission abort for the corresponding message buffers.

### TMTSTS bit (TX Message Buffer Transmission Status)

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

### TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00: Transmission in progress or has not been requested
- 01: Transmission has been aborted from the corresponding TX message buffer
- 10: Transmission was successful from the corresponding TX message buffer and the CFDTMCn.TMTAR bit was not set for this TX message buffer
- 11: Transmission was successful from the corresponding TX message buffer, but the CFDTMCn.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

### TMTRM bit (TX Message Buffer Transmission Request Mirrored)

The TMTRM bit is set when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is cleared.

### TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)

The TMTARM bit is set when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is cleared.

## 32.2.50 CFDTMTRSTSf : TX Message Buffer Transmission Request Status Register f (f = 0 to 3)

Base address: CANFD = 0x400B_0000

Offset address: 0x0CD0 + 0x04 × f

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFDTMTRSTS[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | CFDTMTRSTS[7:0] | TX Message Buffer Transmission Request Status<br>0: Transmission not requested for corresponding TX message buffer<br>1: Transmission requested for corresponding TX message buffer | R |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R |

### CFDTMTRSTS[7:0] bits (TX Message Buffer Transmission Request Status)

The CFDTMTRSTS[7:0] bits show status of the CFDTMCn.TMTR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCn), and only when the message buffer does not belong to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

## 32.2.51 CFDTMTARSTSf : TX Message Buffer Transmission Abort Request Status Register f (f = 0 to 3)

Base address: CANFD = 0x400B_0000

Offset address: 0x0D70 + 0x04 × f

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFDTMTARSTS[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | CFDTMTARSTS[7:0] | TX Message Buffer Transmission Abort Request Status<br>0: Transmission abort not requested for corresponding TX message buffer<br>1: Transmission abort requested for corresponding TX message buffer | R |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R |

### CFDTMTARSTS[7:0] bits (TX Message Buffer Transmission Abort Request Status)

The CFDTMTARSTS[7:0] bits show status of the CFDTMCn.TMTAR bits of the TX Message Buffer Control Registers.

Alignment of the CFDTMTARSTS[7:0] bits is shown in Table 32.5.

**Table 32.5 Alignment of CFDTMTARSTS[7:0] mirror bits**

| Bit position | g = TX message buffer number |
|---|---|
| n*64-fmin*32 | n*64+0 |
| n*64+1-fmin*32 | n*64+1 |
| ⋮ | ⋮ |
| n*64+31-fmin*32 | n*64+31 |
| n*64+32-fmax*32 | n*64+32 |
| n*64+33-fmax*32 | n*64+33 |
| ⋮ | ⋮ |
| n*64+62-fmax*32 | n*64+62 |
| n*64+63-fmax*32 | n*64+63 |

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

## 32.2.52 CFDTMTCSTSf : TX Message Buffer Transmission Completion Status Register f (f = 0 to 3)

Base address: CANFD = 0x400B_0000

Offset address: 0x0E10 + 0x04 × f

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFDTMTCSTS[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | CFDTMTCSTS[7:0] | TX Message Buffer Transmission Completion Status<br>0: Transmission not complete for corresponding TX message buffer<br>1: Transmission completed for corresponding TX message buffer | R |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R |

### CFDTMTCSTS[7:0] bits (TX Message Buffer Transmission Completion Status)

The CFDTMTCSTS[7:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

If a CAN channel enters CH_RESET mode, then the bits related to that channel are cleared.

### 32.2.53 CFDTMTASTSf : TX Message Buffer Transmission Abort Status Register f (f = 0 to 3)

Base address: CANFD = 0x400B_0000

Offset address: 0x0EB0 + 0x04 × f

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CFDTMTASTS[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | CFDTMTASTS[7:0] | TX Message Buffer Transmission Abort Status<br>0: Transmission not aborted for corresponding TX message buffer<br>1: Transmission aborted for corresponding TX message buffer | R |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R |

#### CFDTMTASTS[7:0] bits (TX Message Buffer Transmission Abort Status)

The CFDTMTASTS[7:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Alignment of the CFDTMTASTS[7:0] bits is shown in Table 32.6.

**Table 32.6 Alignment of CFDTMTASTS[7:0] mirror bits**

| Bit position | g = TX message buffer number |
|---|---|
| n*64-fmin*32 | n*64+0 |
| n*64+1-fmin*32 | n*64+1 |
| ⋮ | ⋮ |
| n*64+31-fmin*32 | n*64+31 |
| n*64+32-fmax*32 | n*64+32 |
| n*64+33-fmax*32 | n*64+33 |
| ⋮ | ⋮ |
| n*64+62-fmax*32 | n*64+62 |
| n*64+63-fmax*32 | n*64+63 |

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the CFDTMSTSn.TMTRF bits are set to 01b in the corresponding TX Message Buffer Status Register.

Each bit is cleared automatically when:

- The CFDTMSTSn.TMTRF bits are cleared in the corresponding TX Message Buffer Status Register
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

## 32.2.54 CFDTMIECf : TX Message Buffer Interrupt Enable Configuration Register f (f = 0 to 3)

Base address: CANFD = 0x400B_0000

Offset address: 0x0F50 + 0x04 × f

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TMIE[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TMIE[7:0] | TX Message Buffer Interrupt Enable<br>0: TX message buffer interrupt disabled for corresponding TX message buffer<br>1: TX message buffer interrupt enabled for corresponding TX message buffer | R/W |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R |

### TMIE[7:0] bits (TX Message Buffer Interrupt Enable)

If the TMIE[7:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See section 32.7. Interrupts and DMA for TX Message Buffer Interrupt specification.

Do not write to the TMIE[7:0] bits when:

- The CAN-FD module is in GL_SLEEP mode
- The related CAN-FD channel is in CH_SLEEP mode
- The corresponding TX message buffer is part of a TX Queue
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCCn.CFTML bits.

## 32.2.55 CFDTXQCC0n : TX Queue Configuration/Control Registers 0n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1000 + 0x04 × n

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQOFTXIE | TXQOFRXIE | TXQFIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field: | — | — | — | TXQDC[12:8] | | | | | TXQIM | — | TXQTXIE | — | — | — | TXQGWE | TXQE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQE | TX Queue Enable<br>0: TX Queue disabled<br>1: TX Queue enabled | R/W |
| 1 | TXQGWE | TX Queue Gateway Mode Enable<br>0: TX Queue GW mode disabled<br>1: TX Queue GW mode enabled | R/W |
| 4:2 | — | These bits are read as 0. The write value should be 0. | R/W |
| 5 | TXQTXIE | TX Queue TX Interrupt Enable<br>0: TX Queue TX interrupt disabled<br>1: TX Queue TX interrupt enabled | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 6 | — | This bit is read as 0. The write value should be 0. | R/W |
| 7 | TXQIM | TX Queue Interrupt Mode<br>0: When the last message is successfully transmitted<br>1: At every successful transmission | R/W |
| 12:8 | TXQDC[12:8] | TX Queue Depth Configuration<br>0x00: 0 messages<br>0x01: Reserved<br>0x02: 3 messages<br>0x03: 4 messages<br>⋮<br>0x0F: 16 messages<br>Others: Setting prohibited | R/W |
| 15:13 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | TXQFIE | TXQ Full Interrupt Enable<br>0: TX Queue full interrupt generation disabled<br>1: TX Queue full interrupt generation enabled | R/W |
| 17 | TXQOFRXIE | TXQ One Frame Reception Interrupt Enable<br>0: One Frame RX interrupt generation disabled<br>1: One Frame RX interrupt generation enabled | R/W |
| 18 | TXQOFTXIE | TXQ One Frame Transmission Interrupt Enable<br>0: One Frame TX interrupt generation disabled<br>1: One Frame TX interrupt generation enabled | R/W |
| 31:19 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Configuration/Control Registers 0n (n = 0, 1) are used to configure the TX Queue transmission.

TXQ0 is composed of TXMB0 to TXMB31 (at the maximum) when TXQE is enabled.

### TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC0n.TXQDC == 0x00).

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

The TXQE bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

### TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue GW mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

### TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

### TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

### TXQDC[12:8] bits (TX Queue Depth Configuration)

The TXQDC[12:8] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[7] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 8 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

### TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit in Gateway mode (CFDTXQCC0n.TXQGWE = 1).

### TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit in GW mode (CFDTXQCC0n.TXQGWE = 1).

### TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

## 32.2.56 CFDTXQCC1n : TX Queue Configuration/Control Registers 1n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1060 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQOFTXIE | TXQOFRXIE | TXQFIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | TXQDC[12:8] | | | | | TXQIM | — | TXQTXIE | — | — | — | TXQGWE | TXQE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQE | TX Queue Enable<br>0: TX Queue disabled<br>1: TX Queue enabled | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 1 | TXQGWE | TX Queue Gateway Mode Enable<br>0: TX Queue GW mode disabled<br>1: TX Queue GW mode enabled | R/W |
| 4:2 | — | These bits are read as 0. The write value should be 0. | R/W |
| 5 | TXQTXIE | TX Queue TX Interrupt Enable<br>0: TX Queue TX interrupt disabled<br>1: TX Queue TX interrupt enabled | R/W |
| 6 | — | This bit is read as 0. The write value should be 0. | R/W |
| 7 | TXQIM | TX Queue Interrupt Mode<br>0: When the last message is successfully transmitted<br>1: At every successful transmission | R/W |
| 12:8 | TXQDC[12:8] | TX Queue Depth Configuration<br>0x00: 0 messages<br>0x01: Reserved<br>0x02: 3 messages<br>0x03: 4 messages<br>⋮<br>0x0F: 16 messages<br>Others: Setting prohibited | R/W |
| 15:13 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | TXQFIE | TXQ Full Interrupt Enable<br>0: TX Queue full interrupt generation disabled<br>1: TX Queue full interrupt generation enabled | R/W |
| 17 | TXQOFRXIE | TXQ One Frame Reception Interrupt Enable<br>0: One Frame RX interrupt generation disabled<br>1: One Frame RX interrupt generation enabled | R/W |
| 18 | TXQOFTXIE | TXQ One Frame Transmission Interrupt Enable<br>0: One Frame TX interrupt generation disabled<br>1: One Frame TX interrupt generation enabled | R/W |
| 31:19 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Configuration/Control Registers 1n (n = 0, 1) are used to configure the TX Queue transmission.

TXQ1 is composed of TXMB31 to TXMB0 (at the maximum) when TXQE is enabled.

### TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC1n.TXQDC == 0x00).

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

### TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

### TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

### TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_SLEEP
- CH_OPERATION.

### TXQDC[12:8] bits (TX Queue Depth Configuration)

The TXQDC[12:8] bits select the depth of the transmission queue. The message buffer selection starts from MB[7] down to MB[0] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 8 or less.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

### TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQFIF bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

Only write 1 to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1)..

### TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFRXIF bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

Only write 1 to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1).

### TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFTXIF bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

## 32.2.57 CFDTXQCC2n : TX Queue Configuration/Control Registers 2n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x10C0 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQOFTXIE | TXQOFRXIE | TXQFIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | TXQDC[12:8] | | | | | TXQIM | — | TXQTXIE | — | — | — | TXQGWE | TXQE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQE | TX Queue Enable<br>0: TX Queue disabled<br>1: TX Queue enabled | R/W |
| 1 | TXQGWE | TX Queue Gateway Mode Enable<br>0: TX Queue GW mode disabled<br>1: TX Queue GW mode enabled | R/W |
| 4:2 | — | These bits are read as 0. The write value should be 0. | R/W |
| 5 | TXQTXIE | TX Queue TX Interrupt Enable<br>0: TX Queue TX interrupt disabled<br>1: TX Queue TX interrupt enabled | R/W |
| 6 | — | This bit is read as 0. The write value should be 0. | R/W |
| 7 | TXQIM | TX Queue Interrupt Mode<br>0: When the last message is successfully transmitted<br>1: At every successful transmission | R/W |
| 12:8 | TXQDC[12:8] | TX Queue Depth Configuration<br>0x00: 0 messages<br>0x01: Reserved<br>0x02: 3 messages<br>0x03: 4 messages<br>⋮<br>0x0F: 16 messages<br>Others: Setting prohibited | R/W |
| 15:13 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | TXQFIE | TXQ Full Interrupt Enable<br>0: TX Queue full interrupt generation disabled<br>1: TX Queue full interrupt generation enabled | R/W |
| 17 | TXQOFRXIE | TXQ One Frame Reception Interrupt Enable<br>0: One Frame RX interrupt generation disabled<br>1: One Frame RX interrupt generation enabled | R/W |
| 18 | TXQOFTXIE | TXQ One Frame Transmission Interrupt Enable<br>0: One Frame TX interrupt generation disabled<br>1: One Frame TX interrupt generation enabled | R/W |
| 31:19 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Configuration/Control Registers 2n (n = 0, 1) are used to configure the TX Queue transmission.

TXQ2 is composed of TXMB32 to TXMB63 (at the maximum) when TXQE is enabled.

### TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC2n.TXQDC == 0x00).

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

### TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode. When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

### TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

### TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

### TXQDC[12:8] bits (TX Queue Depth Configuration)

The TXQDC[12:8] bits select the depth of the transmission queue. The message buffer selection starts from MB[32] up to MB[39] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 8 or less.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_HALT
- CH_OPERATION
- CH_SLEEP.

### TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQFIF bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

Only write 1 to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1).

### TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFRXIF bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

Only write 1 to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1).

### TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFTXIF bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

## 32.2.58 CFDTXQCC3n : TX Queue Configuration/Control Registers 3n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1120 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQOFTXIE | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | TXQDC[12:8] | | | | | TXQIM | — | TXQTXIE | — | — | — | — | TXQE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQE | TX Queue Enable<br>0: TX Queue disabled<br>1: TX Queue enabled | R/W |
| 4:1 | — | These bits are read as 0. The write value should be 0. | R/W |
| 5 | TXQTXIE | TX Queue TX Interrupt Enable<br>0: TX Queue TX interrupt disabled<br>1: TX Queue TX interrupt enabled | R/W |
| 6 | — | This bit is read as 0. The write value should be 0. | R/W |
| 7 | TXQIM | TX Queue Interrupt Mode<br>0: When the last message is successfully transmitted<br>1: At every successful transmission | R/W |
| 12:8 | TXQDC[12:8] | TX Queue Depth Configuration<br>0x00: 0 messages<br>0x01: Reserved<br>0x02: 3 messages<br>0x03: 4 messages<br>⋮<br>0x0F: 16 messages<br>Others: Setting prohibited | R/W |
| 17:13 | — | These bits are read as 0. The write value should be 0. | R/W |
| 18 | TXQOFTXIE | TXQ One Frame Transmission Interrupt Enable<br>0: One Frame TX interrupt generation disabled<br>1: One Frame TX interrupt generation enabled | R/W |
| 31:19 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX QueueConfiguration/Control Registers 3n (n = 0, 1) are used to configure the TX Queue transmission.

TXQ3 is composed of TXMB63 to TXMB32 (at the maximum) when TXQE is enabled.

### TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC3n.TXQDC == 0x00).

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

### TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

### TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION mode.

### TXQDC[12:8] bits (TX Queue Depth Configuration)

The TXQDC[12:8] bits select the depth of the transmission queue. The message buffer selection starts from MB[39] down to MB[32] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 8 or less.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION mode.

### TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS3n.TXQOFTXIF bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

## 32.2.59 CFDTXQSTS0n : TX Queue Status Registers 0n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1020 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | TXQMLT | TXQOFTXIF | TXQOFRXIF | TXQFIF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | TXQMC[13:8] | | | | | | — | — | — | — | — | TXQTXIF | TXQFLL | TXQEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQEMP | TX Queue Empty<br>0: TX Queue not empty<br>1: TX Queue empty | R |
| 1 | TXQFLL | TX Queue Full<br>0: TX Queue not full<br>1: TX Queue full | R |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2 | TXQTXIF | TX Queue TX Interrupt Flag<br>0: TX Queue interrupt condition not satisfied after a frame TX<br>1: TX Queue interrupt condition satisfied after a frame TX | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 13:8 | TXQMC[13:8] | TX Queue Message Count<br>Number of messages in the TX Queue. | R |
| 15:14 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | TXQFIF | TXQ Full Interrupt Flag<br>When TXQ is in full status, an interrupt is set. | R/W |
| 17 | TXQOFRXIF | TXQ One Frame Reception Interrupt Flag<br>When TXQ receives one frame, an interrupt is set. | R/W |
| 18 | TXQOFTXIF | TXQ One Frame Transmission Interrupt Flag<br>When one frame transmits from TXQ, an interrupt is set. | R/W |
| 19 | TXQMLT | TXQ Message Lost<br>0: No message lost in TXQ<br>1: TXQ message lost | R/W |
| 31:20 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Status Registers 0n (n = 0, 1) show the status of the TX Queue of corresponding CAN channel.

### TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

### TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

### TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

### TXQMC[13:8] bits (TX Queue Message Count)

The TXQMC[13:8] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

### TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in GW mode (CFDTXQCC0n.TXQGWE = 1) that this bit is set automatically when the TX Queue enters a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

### TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in GW mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

### TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

### TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in GW mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

## 32.2.60 CFDTXQSTS1n : TX Queue Status Registers 1n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1080 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | TXQMLT | TXQOFTXIF | TXQOFRXIF | TXQFIF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | TXQMC[13:8] | | | | | | — | — | — | — | — | TXQTXIF | TXQFLL | TXQEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQEMP | TX Queue Empty<br>0: TX Queue not empty<br>1: TX Queue empty | R |
| 1 | TXQFLL | TX Queue Full<br>0: TX Queue not full<br>1: TX Queue full | R |
| 2 | TXQTXIF | TX Queue TX Interrupt Flag<br>0: TX Queue interrupt condition not satisfied after a frame TX<br>1: TX Queue interrupt condition satisfied after a frame TX | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 13:8 | TXQMC[13:8] | TX Queue Message Count<br>Number of messages in the TX Queue | R |
| 15:14 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | TXQFIF | TXQ Full Interrupt Flag<br>When TXQ is in full status, an interrupt is set. | R/W |
| 17 | TXQOFRXIF | TXQ One Frame Reception Interrupt Flag<br>When TXQ receives one frame, an interrupt is set. | R/W |
| 18 | TXQOFTXIF | TXQ One Frame Transmission Interrupt Flag<br>When one frame transmits from TXQ, an interrupt is set. | R/W |
| 19 | TXQMLT | TXQ Message Lost<br>0: No message lost in TXQ<br>1: TXQ message lost | R/W |
| 31:20 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Status Registers 1n (n = 0, 1) show the status of the TX Queue of corresponding CAN channel.

### TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CANFD channel is in CH_RESET mode.

### TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth

- The related CANFD channel is in CH_RESET mode.

### TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQMC[13:8] bits (TX Queue Message Count)

The TXQMC[13:8] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

### TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Only in Gateway mode (CFDTXQCC1n.TXQGWE = 1) that this bit is set automatically when TX Queue enters a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in GW mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in GW mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

## 32.2.61 CFDTXQSTS2n : TX Queue Status Registers 2n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x10E0 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | TXQMLT | TXQOFTXIF | TXQOFRXIF | TXQFIF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | TXQMC[13:8] | | | | | | — | — | — | — | — | TXQTXIF | TXQFLL | TXQEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQEMP | TX Queue Empty<br>0: TX Queue not empty<br>1: TX Queue empty | R |
| 1 | TXQFLL | TX Queue Full<br>0: TX Queue not full<br>1: TX Queue full | R |
| 2 | TXQTXIF | TX Queue TX Interrupt Flag<br>0: TX Queue interrupt condition not satisfied after a frame TX<br>1: TX Queue interrupt condition satisfied after a frame TX | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 13:8 | TXQMC[13:8] | TX Queue Message Count<br>Number of messages in the TX Queue. | R |
| 15:14 | — | These bits are read as 0. The write value should be 0. | R/W |
| 16 | TXQFIF | TXQ Full Interrupt Flag<br>When TXQ is in full status, an interrupt is set. | R/W |
| 17 | TXQOFRXIF | TXQ One Frame Reception Interrupt Flag<br>When TXQ receives one frame, an interrupt is set. | R/W |
| 18 | TXQOFTXIF | TXQ One Frame Transmission Interrupt Flag<br>When one frame transmits from TXQ, an interrupt is set. | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 19 | TXQMLT | TXQ Message Lost<br>0: No message lost in TXQ<br>1: TXQ message lost | R/W |
| 31:20 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Status Registers 2n (n = 0, 1) show the status of the TX Queue of corresponding CAN Channel.

### TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CANFD channel is in CH_RESET mode.

### TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CANFD channel is in CH_RESET mode.

### TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQMC[13:8] bits (TX Queue Message Count)

The TXQMC[13:8] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

### TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in Gateway mode (CFDTXQCC2n.TXQGWE = 1) that this bit is set automatically when the TX Queue enters a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

## 32.2.62 CFDTXQSTS3n : TX Queue Status Registers 3n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1140 + 0x04 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQO<br>FTXIF | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | TXQMC[13:8] | | | | | | — | — | — | — | — | TXQT<br>XIF | TXQF<br>LL | TXQE<br>MP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TXQEMP | TX Queue Empty<br>0: TX Queue not empty<br>1: TX Queue empty | R |
| 1 | TXQFLL | TX Queue Full<br>0: TX Queue not full<br>1: TX Queue full | R |
| 2 | TXQTXIF | TX Queue TX Interrupt Flag<br>0: TX Queue interrupt condition not satisfied after a frame TX<br>1: TX Queue interrupt condition satisfied after a frame TX | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 13:8 | TXQMC[13:8] | TX Queue Message Count<br>Number of messages in the TX Queue | R |
| 17:14 | — | These bits are read as 0. The write value should be 0. | R/W |
| 18 | TXQOFTXIF | TXQ One Frame Transmission Interrupt Flag<br>When one frame transmits from TXQ, an interrupt is set. | R/W |
| 31:19 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Status Registers 3n (n = 0, 1) show the status of the TX Queue of corresponding CAN Channel.

### TXQEMP bit (TX Queue Empty)

The TXQEMP bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The TX Queue is disabled or no messages are stored in the TX Queue
- The last message is transmitted from the TX Queue
- The related CANFD channel is in CH_RESET mode.

### TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CANFD channel is in CH_RESET mode.

### TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

### TXQMC[13:8] bits (TX Queue Message Count)

The TXQMC[13:8] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

### TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled. When transmission is successful in the TX Queue, this bit is set automatically.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode. When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

## 32.2.63 CFDTXQPCTR0n : TX Queue Pointer Control Registers 0n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1040 + 0x04 × n

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TXQPC[7:0] | TX Queue Pointer Control<br>Increments the write pointer to the TX Queue buffer in the corresponding channel | W |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Pointer Control Registers 0n (n = 0, 1) are used to confirm storage of a full message in the corresponding TX Queue buffers.

### TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

## 32.2.64 CFDTXQPCTR1n : TX Queue Pointer Control Registers 1n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x10A0 + 0x04 × n

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TXQPC[7:0] | TX Queue Pointer Control<br>Increments the write pointer to the TX Queue buffer in the corresponding channel | W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Pointer Control Registers 1n (n = 0, 1) are used to confirm storage of a full message in the corresponding TX Queue buffers.

#### TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

### 32.2.65 CFDTXQPCTR2n : TX Queue Pointer Control Registers 2n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1100 + 0x04 × n

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TXQPC[7:0] | TX Queue Pointer Control<br>Increments the write pointer to the TX Queue buffer in the corresponding channel | W |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Pointer Control Registers 2n (n = 0, 1) are used to confirm storage of a full message in the corresponding TX Queue buffers.

#### TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00.

You cannot write to these bits when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

### 32.2.66 CFDTXQPCTR3n : TX Queue Pointer Control Registers 3n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1160 + 0x04 × n

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQPC[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TXQPC[7:0] | TX Queue Pointer Control<br>Increments the write pointer to the TX Queue buffer in the corresponding channel | W |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Pointer Control Registers 3n (n = 0, 1) are used to confirm storage of a full message in the corresponding TX Queue buffers.

### TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when the corresponding TX Queue is enabled and not full.

## 32.2.67 CFDTXQESTS : TX Queue Empty Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1180

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | 8 | 7 | | | | | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQxEMP[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TXQxEMP[7:0] | TXQ Empty Status<br>0: TXQ not empty<br>1: TXQ empty | R |
| 31:8 | — | These bits are read as 0. | R |

TX Queue Empty Status Register shows the status of the empty bits of the TXQ buffers.

### TXQxEMP (x = (n+1) * TXQ Number = 0 to (((n+1)*4 -1)) bits (TXQ Empty Status)

Each bit is set automatically when the corresponding bit is set in the TX Queue Empty Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Empty Status Register.

This bit is set when the CANFD module is in GL_RESET mode.

| Bit position | Corresponding TX Queue |
|---|---|
| 0 | Channel 0 TX Queue0 |
| 1 | Channel 0 TX Queue1 |
| 2 | Channel 0 TX Queue2 |
| 3 | Channel 0 TX Queue3 |
| 4 | Channel 1 TX Queue0 |
| 5 | Channel 1 TX Queue1 |
| 6 | Channel 1 TX Queue2 |
| 7 | Channel 1 TX Queue3 |

## 32.2.68 CFDTXQFISTS : TX Queue Full Interrupt Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1184

| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 7 | 6 | | 4 | 3 | 2 | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQ1FULL[2:0] | | | — | TXQ0FULL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2:0 | TXQ0FULL[2:0] | TXQ Full Interrupt Status for Channel 0<br>0: TXQ full interrupt is not set<br>1: TXQ full interrupt is set | R |
| 3 | — | This bit is read as 0. The write value should be 0. | R/W |
| 6:4 | TXQ1FULL[2:0] | TXQ Full Interrupt Status for Channel 1<br>0: TXQ full interrupt is not set<br>1: TXQ full interrupt is set | R |
| 31:7 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Full Interrupt Status Register shows the status of the full interrupt bits of the TXQ buffers.

**TXQxFULL (x = 0, 1) bits (TXQ Full Interrupt Status)**

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Interrupt Status Register.

This bit is cleared when the CANFD module is in GL_RESET mode.

| Bit position | Corresponding TX Queue |
|---|---|
| 0 | Channel 0 TX Queue0 |
| 1 | Channel 0 TX Queue1 |
| 2 | Channel 0 TX Queue2 |
| 3 | Reserved |
| 4 | Channel 1 TX Queue0 |
| 5 | Channel 1 TX Queue1 |
| 6 | Channel 1 TX Queue2 |
| 7 | Reserved |

## 32.2.69 CFDTXQMSTS : TX Queue Message Lost Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1188

| Bit position: | 31 | 30 | | 28 | | 26 | | 24 | | 22 | | 20 | | 18 | | 16 | | 14 | | 12 | | 10 | | 8 | | 6 | | 4 | | 2 | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQ1ML[2:0] | | | — | TXQ0ML[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2:0 | TXQ0ML[2:0] | TXQ Message Lost Status for Channel 0<br>0: TXQ message lost flag is not set<br>1: TXQ message lost flag is set | R |
| 3 | — | This bit is read as 0. The write value should be 0. | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 6:4 | TXQ1ML[2:0] | TXQ Message Lost Status for Channel 1<br>0: TXQ message lost flag is not set<br>1: TXQ message lost flag is set | R |
| 31:7 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Message Lost Status Register shows the status of the message lost bits of the TXQ buffers.

### TXQxML (x = 0, 1) bits (TXQ message lost Status)

Each bit is set automatically when the corresponding bit is set in the TX Queue Message Lost Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Message Lost Status Register.

This bit is cleared when the CANFD module is in GL_RESET mode.

| Bit position | Corresponding TX Queue |
|---|---|
| 0 | Channel 0 TX Queue0 |
| 1 | Channel 0 TX Queue1 |
| 2 | Channel 0 TX Queue2 |
| 3 | Reserved |
| 4 | Channel 1 TX Queue0 |
| 5 | Channel 1 TX Queue1 |
| 6 | Channel 1 TX Queue2 |
| 7 | Reserved |

## 32.2.70 CFDTXQISTS : TX Queue Interrupt Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1190

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | 8 | 7 | | | 4 | 3 | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQ1ISF[3:0] | | | | TXQ0ISF[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 3:0 | TXQ0ISF[3:0] | TXQ Interrupt Status Flag for Channel 0<br>0: TXQ Interrupt flag is not set<br>1: TXQ Interrupt flag is set | R |
| 7:4 | TXQ1ISF[3:0] | TXQ Interrupt Status Flag for Channel 1<br>0: TXQ Interrupt flag is not set<br>1: TXQ Interrupt flag is set | R |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue Interrupt Status Register shows the status of the interrupt flag of the TXQ buffers.

### TXQxISF[3:0] (x = 0, 1) bits (TXQ Interrupt Status Flag )

Each bit is set automatically when the corresponding bit is set in the TX Queue Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Interrupt Status Register.

This bit is cleared when the CANFD module is in GL_RESET mode.

| Bit position | Corresponding TX Queue |
|---|---|
| 0 | Channel 0 TX Queue0 |

| Bit position | Corresponding TX Queue |
|---|---|
| 1 | Channel 0 TX Queue1 |
| 2 | Channel 0 TX Queue2 |
| 3 | Channel 0 TX Queue3 |
| 4 | Channel 1 TX Queue0 |
| 5 | Channel 1 TX Queue1 |
| 6 | Channel 1 TX Queue2 |
| 7 | Channel 1 TX Queue3 |

### 32.2.71 CFDTXQOFTISTS : TX Queue One Frame TX Interrupt Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1194

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | 8 | 7 | | | 4 | 3 | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQ1OFTISF[3:0] | | | | TXQ0OFTISF[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 3:0 | TXQ0OFTISF[3:0] | TXQ One Frame TX Interrupt Status Flag for Channel 0<br>0: TXQ One Frame TX Interrupt flag is not set<br>1: TXQ One Frame TX Interrupt flag is set | R |
| 7:4 | TXQ1OFTISF[3:0] | TXQ One Frame TX Interrupt Status Flag for Channel 1<br>0: TXQ One Frame TX Interrupt flag is not set<br>1: TXQ One Frame TX Interrupt flag is set | R |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue One Frame TX Interrupt Status Register shows the status of the One Frame TX Interrupt flag of the TXQ buffers.

**TXQxOFTISF[3:0] (x = 0, 1) bits (TXQ One Frame TX Interrupt Status Flag)**

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame TX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame TX Interrupt Status Register.

This bit is cleared when the CANFD module is in GL_RESET mode.

| Bit position | Corresponding TX Queue |
|---|---|
| 0 | Channel 0 TX Queue0 |
| 1 | Channel 0 TX Queue1 |
| 2 | Channel 0 TX Queue2 |
| 3 | Channel 0 TX Queue3 |
| 4 | Channel 1 TX Queue0 |
| 5 | Channel 1 TX Queue1 |
| 6 | Channel 1 TX Queue2 |
| 7 | Channel 1 TX Queue3 |

## 32.2.72 CFDTXQOFRISTS : TX Queue One Frame RX Interrupt Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1198

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | 6 | | 4 | | 2 | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQ1OFRISF[2:0] | | | — | TXQ0OFRISF[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2:0 | TXQ0OFRISF[2:0] | TXQ One Frame RX Interrupt Status Flag<br>0: TXQ One Frame RX Interrupt flag is not set<br>1: TXQ One Frame RX Interrupt flag is set | R |
| 3 | — | This bit is read as 0. The write value should be 0. | R/W |
| 6:4 | TXQ1OFRISF[2:0] | TXQ One Frame RX Interrupt Status Flag<br>0: TXQ One Frame RX Interrupt flag is not set<br>1: TXQ One Frame RX Interrupt flag is set | R |
| 31:7 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX Queue One Frame RX Interrupt Status Register shows the status of the One Frame RX Interrupt flag of the TXQ buffers.

### TXQxOFRISF[2:0] (x = 0, 1) bits (TXQ One Frame RX Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame RX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame RX Interrupt Status Register.

This bit is cleared when the CANFD module is in GL_RESET mode.

| Bit position | Corresponding TX Queue |
|---|---|
| 0 | Channel 0 TX Queue0 |
| 1 | Channel 0 TX Queue1 |
| 2 | Channel 0 TX Queue2 |
| 3 | Reserved |
| 4 | Channel 1 TX Queue0 |
| 5 | Channel 1 TX Queue1 |
| 6 | Channel 1 TX Queue2 |
| 7 | Reserved |

## 32.2.73 CFDTXQFSTS : TX Queue Full Status Register

Base address: CANFD = 0x400B_0000

Offset address: 0x119C

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | 8 | 7 | | | 4 | 3 | | | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TXQ1FSF[3:0] | | | | TXQ0FSF[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 3:0 | TXQ0FSF[3:0] | TXQ Full Status Flag for Channel 0<br>0: TXQ Full flag is not set<br>1: TXQ Full flag is set | R |
| 7:4 | TXQ1FSF[3:0] | TXQ Full Status Flag for Channel 1<br>0: TXQ Full flag is not set<br>1: TXQ Full flag is set | R |
| 31:8 | — | These bits are read as 0. | R |

The TX Queue Full Status Register shows the status of the Full Status flag bits of the TXQ buffers.

### TXQxFSF[31:0] (x = 0, 1) bits (TXQ Full Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Status Register.

This bit is cleared when the CANFD module is in GL_RESET mode.

| Bit position | Corresponding TX Queue |
|---|---|
| 0 | Channel 0 TX Queue0 |
| 1 | Channel 0 TX Queue1 |
| 2 | Channel 0 TX Queue2 |
| 3 | Channel 0 TX Queue3 |
| 4 | Channel 1 TX Queue0 |
| 5 | Channel 1 TX Queue1 |
| 6 | Channel 1 TX Queue2 |
| 7 | Channel 1 TX Queue3 |

## 32.2.74 CFDTHLCCn : TX History List Configuration/Control Register n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1200 + 0x04 × n (n = 0, 1)

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | THLDGE | THLDTE | THLIM | THLIE | — | — | — | — | — | — | — | THLE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | THLE | TX History List Enable<br>0: TX History List disabled<br>1: TX History List enabled | R/W |
| 7:1 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | THLIE | TX History List Interrupt Enable<br>0: TX History List Interrupt disabled<br>1: TX History List Interrupt enabled | R/W |
| 9 | THLIM | TX History List Interrupt Mode<br>0: Interrupt generated if TX History List level reaches ¾ of the TX History List depth<br>1: Interrupt generated for every successfully stored entry | R/W |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 10 | THLDTE | TX History List Dedicated TX Enable<br>0: TX FIFO + TX Queue<br>1: Flat TX MB + TX FIFO + TX Queue | R/W |
| 11 | THLDGE | TX History List Dedicated Gateway Enable<br>0: Not dedicated Gateway FIFO + Gateway TX Queue<br>1: Dedicated Gateway FIFO + Gateway TX Queue | R/W |
| 31:12 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX History List Configuration/Control Register n (n = 0, 1) configures the TX History List functions.

### THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

### THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

### THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

### THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

### THLDGE bit (TX History List Dedicated Gateway Enable)

The THLDGE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

## 32.2.75 CFDTHLSTSn : TX History List Status Register n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1220 + 0x04 × n (n = 0, 1)

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | THLMC[5:0] | | | | | | — | — | — | — | THLIF | THLELT | THLFLL | THLEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | THLEMP | TX History List Empty<br>0: TX History List not empty<br>1: TX History List empty | R |
| 1 | THLFLL | TX History List Full<br>0: TX History List not full<br>1: TX History List full | R |
| 2 | THLELT | TX History List Entry Lost<br>0: No entry lost in TX History List<br>1: TX History List entry Lost | R/W |
| 3 | THLIF | TX History List Interrupt Flag<br>0: TX History List interrupt condition not satisfied<br>1: TX History List interrupt condition satisfied | R/W |
| 7:4 | — | These bits are read as 0. The write value should be 0. | R/W |
| 13:8 | THLMC[5:0] | TX History List Message Count<br>Number of messages stored in TX History List | R |
| 31:14 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX History List Status register shows the status of data stored in the TX History List buffer.

### THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CAN-FD channel is in CH_RESET mode.

### THLFLL bit (TX History List Full)

The THLFLL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 32 entries (each channel has a dedicated TX History List).

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled
- The related CAN-FD channel is in CH_RESET mode.

### THLELT bit (TX History List Entry Lost)

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

### THLIF bit (TX History List Interrupt Flag)

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

The bit is cleared by writing 0 to it.

This bit is automatically cleared in CH_RESET mode.

### THLMC[5:0] bits (TX History List Message Count)

The THLMC[5:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

## 32.2.76 CFDTHLACC0n : TX History List Access Registers 0 (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x8000 + 0x08 × n (n = 0, 1)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit field: | TMTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field: | TGW | — | — | — | — | — | BN[6:0] | | | | | | | BT[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2:0 | BT[2:0] | Buffer Type<br>0 0 1: Flat TX message buffer<br>0 1 0: TX FIFO message buffer number and gateway FIFO message number<br>1 0 0: TX Queue message buffer number | R |
| 9:3 | BN[6:0] | Buffer Number<br>Number of the message buffer | R |
| 14:10 | — | These bits are read as 0. The write value should be 0. | R |
| 15 | TGW | Transmit Gateway Buffer Indication<br>0: No transmission from gateway<br>1: Transmission from gateway | R |
| 31:16 | TMTS[15:0] | Transmit Timestamp<br>Transmit timestamp value for software drivers | R |

The TX History List Access Registers 0 (n = 0, 1) provide access to the entry in the TX History List based on the read timestamp value.

### BT[2:0] bits (Buffer Type)

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

### BN[6:0] bits (Buffer Number)

The BN[6:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

### TGW bit (Transmit Gateway Buffer Indication)

The TGW bit is automatically set to 1 when transmission is completed in GW mode.

### TMTS[15:0] bits (Transmit Timestamp)

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

## 32.2.77 CFDTHLACC1n : TX History List Access Registers 1 (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x8004 + 0x08 × n (n = 0, 1)

| Bit position: | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  | 17 | 16 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TIFL[1:0] |  | TID[15:0] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 15:0 | TID[15:0] | Transmit ID<br>These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers. | R |
| 17:16 | TIFL[1:0] | Transmit Information Label<br>These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers. | R |
| 31:18 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX History List Access Registers 1 provide access to entry in the TX History List based on the read pointer value.

### TID[15:0] bits (Transmit ID)

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRn.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTSn.CFPTR) is for use by software drivers.

When transmission in GW mode, these bits indicate the AFL pointer field (CFDGAFLP0n.GAFLPTR) instead of the message buffer reference ID (CFDTMFDCTRn.TMPTR).

### TIFL[1:0] bits (Transmit Information Label)

The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRn.TMIFL) or the TX FIFO information label (CFDCFFDCSTSn.CFIFL) is for use by software drivers.

When transmission in GW mode, these bits indicate the AFL pointer field (CFDGAFLMn.GAFLIFL1 and CFDGAFLP0n.GAFLIFL0) instead of the MB information label (CFDTMFDCTRn.TMIFL).

## 32.2.78 CFDTHLPCTRn : TX History List Pointer Control Registers n (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1240 + 0x04 × n (n = 0, 1)

| Bit position: | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |  |  |  |  |  |  | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | THLPC[7:0] |  |  |  |  |  |  |  |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | THLPC[7:0] | TX History List Pointer Control<br>Increments the write pointer to the TX History List in the corresponding channel | W |
| 31:8 | — | These bits are read as 0. The write value should be 0. | R/W |

The TX History List Pointer Control Registers n (n = 0, 1) are used to increment the read pointer of the TX History List.

### THLPC[7:0] bits (TX History List Pointer Control)

When 0xFF is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

The read value from these bits is always 0x00. Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Only write 0xFF to these registers when the corresponding TX History List is enabled and not empty.

## 32.2.79 CFDGRSTC : Global SW reset Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1380

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | KEY[7:0] | | | | | | | | — | — | — | — | — | — | — | SRST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | SRST | SW Reset<br>0: Normal state<br>1: SW reset state | R/W |
| 7:1 | — | These bits are read as 0. The write value should be 0. | R/W |
| 15:8 | KEY[7:0] | Key Code<br>These bits control the validity of rewriting of a SRST bit. | W |
| 31:16 | — | These bits are read as 0. The write value should be 0. | R/W |

### SRST bit (SW Reset)

When the SRST bit is set, the CANFD module is in the same state as hardware reset. When a reset is required, write 1 then write 0 to this bit.

This bit is cleared when the CANFD module is in GL_SLEEP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

### KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[15:8] bits, a write to the SRST bit is valid.

The read value from these bits is always 0x00.

CFDGRSTC.SRST bit and the CFDGRSTC.KEY bit should be written simultaneously.

## 32.2.80 CFDGTSTCFG : Global Test Configuration Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1308

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | 25 | | | | | | | | | 16 |
| Bit field: | — | — | — | — | — | — | RTMPS[9:0] | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | | | | | | | | | | | | | | 1 | 0 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ICBCE[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 1:0 | ICBCE[1:0] | Channel n Internal CAN Bus Communication Test Mode Enable<br>0: Channel n internal CAN bus communication disabled<br>1: Channel n internal CAN bus communication enabled | R/W |
| 15:2 | — | These bits are read as 0. The write value should be 0. | R/W |
| 25:16 | RTMPS[9:0] | RAM Test Mode Page Select<br>Select a RAM test mode page | R/W |
| 31:26 | — | These bits are read as 0. The write value should be 0. | R/W |

The Global Test Configuration Register is used to configure the CAN channels joining the internal CAN bus communication test mode and the RAM test mode page.

### ICBCE[1:0] bits (Channel n Internal CAN Bus Communication Test Mode Enable)

When the ICBCE[1:0] bits are set and CAN-FD module is configured in the internal CAN bus communication test mode, then CAN channel n joins the internal CAN bus communication test mode operation.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

### RTMPS[9:0] bits (RAM Test Mode Page Select)

The RTMPS[9:0] bits select the RAM page mode for CPU read/write access when the CAN-FD module is configured in RAM test mode.

See section 32.9.2.1. RAM Test Mode for the RAM test mode specification.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only enter values from 0 to 7 (0x007) for the AFL RAM and 8 to 39 (0x027) for the message buffer RAM.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the related CAN-FD channel is in GL_RESET mode.

## 32.2.81 CFDGTSTCTR : Global Test Control Register

Base address: CANFD = 0x400B_0000

Offset address: 0x130C

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | RTME | — | ICBCTME |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | ICBCTME | Internal CAN Bus Communication Test Mode Enable<br>0: Internal CAN Bus Communication test mode disabled<br>1: Internal CAN Bus Communication test mode enabled | R/W |
| 1 | — | This bit is read as 0. The write value should be 0. | R/W |
| 2 | RTME | RAM Test Mode Enable<br>0: RAM test mode disabled<br>1: RAM test mode enabled | R/W |
| 31:3 | — | These bits are read as 0. The write value should be 0. | R/W |

The Global Test Control register is used to control the global test modes of the CANFD module.

### ICBCTME bit (Internal CAN Bus Communication Test Mode Enable)

When the ICBCTME bit is set, internal CAN bus communication is enabled for the CAN channels that are configured for internal CAN bus communication participation. See section 32.9.2.2. Internal CAN Bus Communication Test Mode for the specification of internal CAN bus communication test mode.

Only write to this bit when the CANFD module is in GL_HALT mode.

Clear this bit when the CANFD module is in GL_HALT mode.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

### RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CANFD module is configured in RAM test mode. See section 32.9.2.1. RAM Test Mode for RAM test mode specification.

Only write to this bit when the CANFD module is in GL_HALT mode.

Clear this bit when the CANFD module is in GL_HALT mode.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

## 32.2.82 CFDGFDCFG : Global FD Configuration Register

Base address: CANFD = 0x400B_0000

Offset address: 0x1314

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | TSCCFG[1:0] | | — | — | — | — | — | — | — | RPED |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | RPED | RES Bit Protocol Exception Disable<br>0: Protocol exception event detection enabled<br>1: Protocol exception event detection disabled | R/W |
| 7:1 | — | These bits are read as 0. The write value should be 0. | R/W |
| 9:8 | TSCCFG[1:0] | Timestamp Capture Configuration<br>0 0: Timestamp capture at the sample point of SOF (start of frame)<br>0 1: Timestamp capture at frame valid indication<br>1 0: Timestamp capture at the sample point of RES bit<br>1 1: Reserved | R/W |
| 31:10 | — | These bits are read as 0. The write value should be 0. | R/W |

### RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CANFD module is in GL_RESET mode.

### TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When CFDGFDCFG.TSCCFG[1:0] = 10b, the timestamp capture is performed for CANFD frames at RES bit and for Classical frames at the start of frame.

Only write to these bits when the CANFD module is in GL_RESET mode.

## 32.2.83 CFDGLOCKK : Global Lock Key Register

Base address: CANFD = 0x400B_0000

Offset address: 0x131C

| Bit position: | 31 | | | | | | | | | | | | | | | | 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | LOCK[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 15:0 | LOCK[15:0] | Lock Key<br>Key bits for unlocking the protection of test modes | W |
| 31:16 | — | The write value should be 0. | W |

The Global Lock Key register is a write-only register that is used to unlock the protection for special test bits.

See section 32.9.2. Global Test Modes for Lock key specification.

### LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CANFD module in FIFO OTB disable and RAM test modes.

The read value from these bits is always 0x0000.

You cannot write to these bits when the CANFD module is in GL_SLEEP or GL_RESET mode.

Do not write to these bits when the CANFD module is in GL_OPERATION mode.

## 32.2.84 CFDRPGACCn : RAM Test Page Access Registers n (n = 0 to 63)

Base address: CANFD = 0x400B_0000

Offset address: 0x8400 + 0x04 × n (n = 0 to 63)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
| Bit field: | RDTA[31:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 31:0 | RDTA[31:0] | RAM Data Test Access<br>RAM data bytes | R/W |

### RDTA[31:0] bits (RAM Data Test Access)

Data can be read from or written into the RDTA[31:0] bits when the CAN-FD module is configured in RAM test mode.

Only write to this bit when the CAN-FD module is in GL_HALT mode and RAM test mode is enabled.

Software data should be read/written in the RAM Test Page Access registers during RAM test mode.

## 32.2.85 CFDCnBLCT : Channel n Bus Load Control Register (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1418 + 0x20 × n (n = 0, 1)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Bit field: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field: | — | — | — | — | — | — | — | BLCLD | — | — | — | — | — | — | — | BLCE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | BLCE | Bus Load Counter Enable<br>0: Bus load counter disable<br>1: Bus load counter enable | R/W |
| 7:1 | — | These bits are read as 0. The write value should be 0. | R/W |
| 8 | BLCLD | BUS Load Counter Load<br>When CFDCnBLCT.BLCLD is set, it is reset after a Bus load counter value is loaded. | W |
| 31:9 | — | These bits are read as 0. The write value should be 0. | R/W |

### BLCE bit (Bus Load Counter Enable)

The BLCE bit indicates the enabling signal of a bus load counter. Write to this bit after setting up the CFDCnNCFG register.

When this bit is 0, the bus load counter stops but the counter is not clear.

The bit is cleared when the related CANFD channel is in CH_RESET mode.

You cannot write to this bit in CH_SLEEP mode.

### BLCLD bit (BUS Load Counter Load)

When the BLCLD bit is set, the bus load counter value is loaded into CFDCnBLSTS.BLC and the bus load counter is reset.

The read value is always 0.

## 32.2.86 CFDCnBLSTS : Channel n Bus Load Status Register (n = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x141C + 0x20 × n (n = 0, 1)

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | 3 | | | 0 |
| Bit field: | BLC[31:3] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 2:0 | — | These bits are read as 0. The write value should be 0. | R/W |
| 31:3 | BLC[31:3] | Bus Load Counter<br>These bits indicate the bus load counter value. | R |

### BLC[31:3] bits (Bus Load Counter)

When the CFDCnBLCT.BLCLD bit is set, the bus load counter value is loaded to BLC bit and the bus load counter is reset.

Bits [2:0] are fixed to 000b.

The bus load counter increases by CANFDCLK period while the CAN bus is in an idle state.

These bits are set only by the CANFD module. Writing any value has no effect.

## 32.2.87 Message Buffer Component Structure

### 32.2.87.1 Start Addresses

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components and the number of channels.

| | |
|---|---|
| no_of_channels | 2 |
| no_of_RMBCPs_per_channel | 16 |
| no_of_RFMBCPs | 8 |
| no_of_CFMBCPs_per_channel | 3 |
| no_of_TMBCPs_per_channel | 16 |

The start addresses for each register in the Message Buffer component are depicted in Table 32.7.

**Table 32.7 Message Buffer Component Register Start Addresses (1 of 2)**

| b = Message buffer component index | MBCP | Register | p | Regular Start Address<br>n = [0…no_of_channels-1] |
|---|---|---|---|---|
| [0…no_of_RMBCPs_per_channel-1] | RMBCPb[i] | RMID | x | 0x2000 + b*0x0080 + n*0x800 |
| | | RMPTR | x | 0x2004 + b*0x0080 + n*0x800 |
| | | RMFDSTS | x | 0x2008 + b*0x0080 + n*0x800 |
| | | RMDFp | [0…15] | 0x200C + p*0x0004 + b*0x0080 + n*0x800 |

**Table 32.7 Message Buffer Component Register Start Addresses (2 of 2)**

| b = Message buffer component index | MBCP | Register | p | Regular Start Address<br>n = [0…no_of_channels-1] |
|---|---|---|---|---|
| [0…no_of_RFMBCPs-1] | RFMBCPb[i] | RFIDE | x | 0x6000 + b*0x0080 |
| | | RFPTRE | x | 0x6004 + b*0x0080 |
| | | RFFDSTSE | x | 0x6008 + b*0x0080 |
| | | RFDFpE | [0…15] | 0x600C + p*0x0004 + b*0x0080 |
| [0…no_of_CFMBCPs_per_channel-1] | CFMBCPb[i] | CFIDE | x | 0x6400 + b*0x0080 + n*0x180 |
| | | CFPTRE | x | 0x6404 + b*0x0080 + n*0x180 |
| | | CFFDCSTSE | x | 0x6408 + b*0x0080 + n*0x180 |
| | | CFDFpE | [0…15] | 0x640C + p*0x0004 + b*0x0080 + n*0x180 |
| [0…no_of_TMBCPs_per_channel-1] | TMBCPb[i] | TMID | x | 0x10000 + b*0x0080 + n*0x2000 |
| | | TMPTR | x | 0x10004 + b*0x0080 + n*0x2000 |
| | | TMFDCTR | x | 0x10008 + b*0x0080 + n*0x2000 |
| | | TMDFp | [0…15] | 0x1000C + p*0x0004 + b*0x0080 + n*0x2000 |

Note: ‘-‘ means Not Applicable

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[i])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[i])
- Common FIFO Access Message Buffer Component (CFDCFMBCPb[i])
- TX Message Buffer Component (CFDTMBCPb[i]).

Where b = the Message Buffer component index that has a range that varies based on the type of Message Buffer component and i = channel index that has a range from 0 to n.

For a summary of this configuration, see Figure 32.28. For a detailed description of the number of and the different types of message buffers, see section 32.6. FIFO Buffers and Normal Message Buffer Configuration.

As described in section 32.2. Register Descriptions, each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

Rc is the Message Buffer Component register where c = Message Buffer Component register index that has a range that varies based on the type of Message Buffer component.

A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains ‘-‘ means reserved and has the same behavio as reserved bits for registers in section 32.2.87. Message Buffer Component Structure.

## 32.2.87.2 CFDRMIDn_i : RX Message Buffer ID Register n Channel i (n = 0 to 15, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x2000 + 0x080 × n + 0x800 × i

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | 30 | | 28 | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
| Bit field: | RMIDE | RMRTR | — | RMID[28:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 28:0 | RMID[28:0] | RX Message Buffer ID Field<br>STD-ID/EXT-ID fields | R |
| 29 | — | This bit is read as 0. The write value should be 0. | R |
| 30 | RMRTR | RX Message Buffer RTR Bit<br>0: Data frame<br>1: Remote frame | R |
| 31 | RMIDE | RX Message Buffer IDE Bit<br>0: STD-ID is stored<br>1: EXT-ID is stored | R |

The RX Message Buffer ID Registers n (n = 0 to 15) store the ID field, IDE bit, and RTR bit of the received message.

### RMID[28:0] bits (RX Message Buffer ID Field)

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

See section 32.2.87.1. Start Addresses for details on how to interpret the structure of this buffer component.

### RMRTR bit (RX Message Buffer RTR Bit)

The RMRTR bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

### RMIDE bit (RX Message Buffer IDE Bit)

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

## 32.2.87.3 CFDRMPTRn_i : RX Message Buffer Pointer Register n Channel i (n = 0 to 15, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x2004 + 0x080 × n + 0x800 × i

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit position: | 31 | | | 28 | | | | | | | | | | | | | 15 | | | | | | | | | | | | | | | 0 |
| Bit field: | RMDLC[3:0] | | | | — | — | — | — | — | — | — | — | — | — | — | — | RMTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 15:0 | RMTS[15:0] | RX Message Buffer Timestamp Field<br>Timestamp value stored for the message in the RX message buffer | R |
| 27:16 | — | These bits are read as 0. The write value should be 0. | R |
| 31:28 | RMDLC[3:0] | RX Message Buffer DLC Field<br>Number of data bytes received in a CAN frame. | R |

The RX Message Buffer Pointer Registers n (n = 0 to 15) store the DLC and Timestamp fields for the received message.

### RMTS[15:0] bits (RX Message Buffer Timestamp Field)

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDGFDCFG.TSCCFG of the received message.

### RMDLC[3:0] bits (RX Message Buffer DLC Field)

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

## 32.2.87.4 CFDRMFDSTSn_i : RX Message Buffer CAN-FD Status Register n Channel i (n = 0 to 15, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x2008 + 0x080 × n + 0x800 × i

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | RMPTR[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | RMIFL[1:0] | | — | — | — | — | — | RMFDF | RMBRS | RMESI |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | RMESI[*1] | Error State Indicator bit<br>0: CAN-FD frame received from error active node<br>1: CAN-FD frame received from error passive node | R |
| 1 | RMBRS[*1] | Bit Rate Switch bit<br>0: CAN-FD frame received with no bit rate switch<br>1: CAN-FD frame received with bit rate switch | R |
| 2 | RMFDF[*1] | CAN FD Format bit<br>0: Non CAN-FD frame received<br>1: CAN-FD frame received | R |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R |
| 9:8 | RMIFL[1:0] | RX Message Buffer Information Label Field | R |
| 15:10 | — | These bits are read as 0. The write value should be 0. | R |
| 31:16 | RMPTR[15:0] | RX Message Buffer Pointer Field | R |

Note 1. This bit is not available in the classical CAN function.

The RX Message Buffer CAN-FD Status Registers n (n = 0 to 2) show the status of the FDF, BRS and ESI bits, and pointer of the received CAN-FD frame.

### RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

### RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

#### RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

#### RMIFL[1:0] bits (RX Message Buffer Information Label Field)

The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

#### RMPTR[15:0] bits (RX Message Buffer Pointer Field)

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

### 32.2.87.5 CFDRMDFp_n_i : RX Message Buffer Data Field p Register n Channel i (p = 0 to 15, n = 0 to 15, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x200C + 0x004 × p + 0x080 × n + 0x800 × i

| Bit position: | 31 24 | 23 16 | 15 8 | 7 0 |
|---|---|---|---|---|
| Bit field: | RMDB_HH[7:0] | RMDB_HL[7:0] | RMDB_LH[7:0] | RMDB_LL[7:0] |
| Value after reset: | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RMDB_LL[7:0] | RX Message Buffer Data Byte (p * 4) | R |
| 15:8 | RMDB_LH[7:0] | RX Message Buffer Data Byte ((p * 4) + 1) | R |
| 23:16 | RMDB_HL[7:0] | RX Message Buffer Data Byte ((p * 4) + 2) | R |
| 31:24 | RMDB_HH[7:0][*1] | RX Message Buffer Data Byte ((p * 4) + 3) | R |

Note 1. These bits are not available in the classical CAN function.

The RX Message Buffer Data Field p Registers n (p = 0 to 15, n = 0 to 15) store the data bytes (p * 4) to data bytes ((p * 4) + 3) of the received message.

#### RMDB_LL[7:0] bits (RX Message Buffer Data Byte (p * 4))

The RMDB(p * 4) [7:0] bits store data bytes (p*4) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

#### RMDB_LH[7:0] bits (RX Message Buffer Data Byte ((p * 4) + 1))

The RMDB((p * 4) + 1)[7:0] bits store data bytes ((p * 4) + 1) of the message in the RX message buffer.

Unused Data Bytes will be filled with 0x00.

#### RMDB_HL[7:0] bits (RX Message Buffer Data Byte ((p * 4) + 2))

The RMDB((p * q) +2)[7:0] bits store data bytes ((p * 4 + 2) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

#### RMDB_HH[7:0] bits (RX Message Buffer Data Byte ((p * 4) + 3))

The RMDB((p * 4) +3)[7:0] bits store data bytes ((p * 4) + 3) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

## 32.2.87.6 CFDRFIDn : RX FIFO Access ID Register n (n = 0 to 7)

Base address: CANFD = 0x400B_0000

Offset address: 0x6000 + 0x080 × n

Bit position: 31 30 28 0

| Bit field: | RFIDE | RFRTR | — | RFID[28:0] |
|---|---|---|---|---|

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 28:0 | RFID[28:0] | RX FIFO Buffer ID Field<br>STD-ID/EXT-ID fields | R |
| 29 | — | This bit is read as 0. | R |
| 30 | RFRTR | RX FIFO Buffer RTR bit<br>0: Data frame<br>1: Remote frame | R |
| 31 | RFIDE | RX FIFO Buffer IDE bit<br>0: STD-ID has been received<br>1: EXT-ID has been received | R |

The RX FIFO Access ID Registers n (n = 0 to 7) store the ID field, IDE bit and RTR bit of the message.

### RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see Identifier Bits Alignment.

### RFRTR bit (RX FIFO Buffer RTR bit)

The RFRTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

### RFIDE bit (RX FIFO Buffer IDE bit)

The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

## 32.2.87.7 CFDRFPTRn : RX FIFO Access Pointer Register n (n = 0 to 7)

Base address: CANFD = 0x400B_0000

Offset address: 0x6004 + 0x080 × n

Bit position: 31 28 15 0

| Bit field: | RFDLC[3:0] | — | — | — | — | — | — | — | — | — | — | — | — | RFTS[15:0] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 15:0 | RFTS[15:0] | RX FIFO Timestamp Value<br>Timestamp value of the received CAN frame | R |
| 27:16 | — | These bits are read as 0. The write value should be 0. | R |
| 31:28 | RFDLC[3:0] | RX FIFO Buffer DLC Field<br>Number of data bytes received in a CAN frame | R |

The FIFO Access Pointer Registers n (n = 0 to 7) store the DLC and Timestamp fields for the received message.

### RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message.

### RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

## 32.2.87.8 CFDRFFDSTSn : RX FIFO Access CAN-FD Status Register n (n = 0 to 7)

Base address: CANFD = 0x400B_0000

Offset address: 0x6008 + 0x080 × n

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | CFDRFPTR[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | RFIFL[1:0] | | — | — | — | — | — | RFFDF | RFBRS | RFESI |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | RFESI[*1] | Error State Indicator bit<br>0: CAN-FD frame received from error active node<br>1: CAN-FD frame received from error passive node | R |
| 1 | RFBRS[*1] | Bit Rate Switch bit<br>0: CAN-FD frame received with no bit rate switch<br>1: CAN-FD frame received with bit rate switch | R |
| 2 | RFFDF[*1] | CAN FD Format bit<br>0: Non CAN-FD frame received<br>1: CAN-FD frame received | R |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R |
| 9:8 | RFIFL[1:0] | RX FIFO Buffer Information Label Field | R |
| 15:10 | — | These bits are read as 0. The write value should be 0. | R |
| 31:16 | CFDRFPTR[15:0] | RX FIFO Buffer Pointer Field | R |

Note 1. This bit is not available in the classical CAN function.

The RX FIFO Access CAN-FD Status Registers n (n = 0 to 7) show the status of the FDF, BRS, and ESI bits, including the pointer of the received CAN-FD frame.

### RFESI bit (Error State Indicator bit)

The RFESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

### RFBRS bit (Bit Rate Switch bit)

The RFBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

### RFFDF bit (CAN FD Format bit)

The RFFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

### RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)

The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

### CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

## 32.2.87.9 CFDRFDFpn : RX FIFO Access Data Field p Register n (p = 0 to 15, n = 0 to 7)

Base address: CANFD = 0x400B_0000

Offset address: 0x600C + 0x004 × p + 0x080 × n

| | | | | |
|---|---|---|---|---|
| Bit position: | 31 24 | 23 16 | 15 8 | 7 0 |
| Bit field: | RFDB_HH[7:0] | RFDB_HL[7:0] | RFDB_LH[7:0] | RFDB_LL[7:0] |
| Value after reset: | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | RFDB_LL[7:0] | RX FIFO Buffer Data Byte (p * 4) | R |
| 15:8 | RFDB_LH[7:0] | RX FIFO Buffer Data Byte ((p * 4) + 1) | R |
| 23:16 | RFDB_HL[7:0] | RX FIFO Buffer Data Byte ((p * 4) + 2) | R |
| 31:24 | RFDB_HH[7:0] | RX FIFO Buffer Data Byte ((p * 4) + 3) | R |

The RX FIFO Access Data Field p Registers n (p = 0 to 15, n = 0 to 7) store data bytes ((p * 4) to data byte ((p * 4) + 3) of the received message.

### RFDB_LL[7:0] bits (RX FIFO Buffer Data Byte (p * 4))

The RFDB(p * 4)[7:0] bits store data bytes (p * 4) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00 according to the configured data payload size CFDRFCCn.RFPLS.

### RFDB_LH[7:0] bits (RX FIFO Buffer Data Byte ((p * 4) + 1))

The RFDB((p * 4) + 1))[7:0] bits store data bytes ((p * 4) + 1)) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

### RFDB_HL[7:0] bits (RX FIFO Buffer Data Byte ((p * 4) + 2))

The RFDB((p * 4) + 2)[7:0] bits store data bytes ((p * 4) + 2) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

### RFDB_HH[7:0] bits (RX FIFO Buffer Data Byte ((p * 4) + 3))

The RFDB((p * 4) + 3)[7:0] bits store data bytes ((p * 4) + 3) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

## 32.2.87.10 CFDCFIDn_i : Common FIFO Access ID Register n Channel i (n = 0 to 2, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x6400 + 0x080 × n + 0x180 × i

| | | | | |
|---|---|---|---|---|
| Bit position: | 31 | 30 | 29 | 28 0 |
| Bit field: | CFIDE | CFRTR | THLEN | CFID[28:0] |
| Value after reset: | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 28:0 | CFID[28:0] | Common FIFO Buffer ID Field<br>STD-ID/EXT-ID fields | R/W |
| 29 | THLEN | THL Entry Enable<br>TX FIFO mode:<br>0: Entry is not to be stored in THL after successful TX<br>1: Entry is to be stored in THL after successful TX<br>RX FIFO mode:<br>Reserved, this bit is read as 0. | R/W |
| 30 | CFRTR | Common FIFO Buffer RTR bit<br>0: Data frame<br>1: Remote frame | R/W |
| 31 | CFIDE | Common FIFO Buffer IDE bit<br>0: STD-ID is to be transmitted or has been received<br>1: EXT-ID is to be transmitted or has been received | R/W |

The Common FIFO Access ID Registers n (n = 0 to 5) store the ID field, IDE bit and RTR bit of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

### CFID[28:0] bits (Common FIFO Buffer ID Field)

The CFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

In TX mode, you can write and read from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

### THLEN bit (THL Entry Enable)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, you can write and read from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

### CFRTR bit (Common FIFO Buffer RTR bit)

The CFRTR bit selects whether a data frame or a remote frame is to be transmitted from or was received in the FIFO buffer.

Note: There are no remote frames in CAN FD format. When a CANFD frame is received (RX mode), the register reflects the state of the received value (RRS bit in FD frame format). When CANFD transmission (TX or GW mode CFDCFID.CFFDF = 1), the bit is always transmitted dominant (data frame).

In TX mode, you can write and read from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

### CFIDE bit (Common FIFO Buffer IDE bit)

The CFIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from or was received in the FIFO buffer.

In TX mode, you can write and read from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

### 32.2.87.11 CFDCFPTRn_i : Common FIFO Access Pointer Register n Channel i (n = 0 to 2, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x6404 + 0x080 × n + 0x180 × i

| Bit position: | 31 | | | 28 | | | | | | | | | | | | | 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | CFDLC[3:0] | | | | — | — | — | — | — | — | — | — | — | — | — | — | CFTS[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 15:0 | CFTS[15:0] | Common FIFO Timestamp Value<br>Timestamp value of the received CAN frame (FIFO in RX mode). | R/W |
| 27:16 | — | These bits are read as 0. The write value should be 0. | R/W |
| 31:28 | CFDLC[3:0] | Common FIFO Buffer DLC Field<br>Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame. | R/W |

The Common FIFO Access Pointer Registers n (n = 0 to 2) store the DLC and Timestamp fields.

In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

#### CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

#### CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

### 32.2.87.12 CFDCFFDCSTSn_i : Common FIFO Access CAN-FD Control/Status Register n Channel i (n = 0 to 2, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x6408 + 0x080 × n + 0x180 × i

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | CFPTR[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | CFIFL[1:0] | | — | — | — | — | — | CFFDF | CFBRS | CFESI |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | CFESI[1] | Error State Indicator bit<br>0: CAN-FD frame received or to transmit by error active node<br>1: CAN-FD frame received or to transmit by error passive node | R/W |
| 1 | CFBRS[1] | Bit Rate Switch bit<br>0: CAN-FD frame received or to transmit with no bit rate switch<br>1: CAN-FD frame received or to transmit with bit rate switch | R/W |
| 2 | CFFDF[1] | CAN FD Format bit<br>0: Non CAN-FD frame received or to transmit<br>1: CAN-FD frame received or to transmit | R/W |
| 7:3 | — | These bits are read as 0. The write value should be 0. | R/W |
| 9:8 | CFIFL[1:0] | COMMON FIFO Buffer Information Label Field | R/W |
| 15:10 | — | These bits are read as 0. The write value should be 0. | R/W |
| 31:16 | CFPTR[15:0] | Common FIFO Buffer Pointer Field | R/W |

Note 1. This bit is not available in the classical CAN function.

The Common FIFO Access CAN-FD Control/Status Registers n (n = 0 to 2) show the status of the FDF, BRS and ESI bits, including the pointer of the received CAN-FD frame or the CAN-FD frame to transmit.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

### CFESI bit (Error State Indicator bit)

In TX mode, you can read and write from FIFO buffers. In this mode, when the CAN-FD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don’t care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFESI bit is updated with the ESI bit value of the CAN-FD frame when it has been received, indicating the error state of the transmitting node. In RX or GW mode, 0 is stored to this bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

### CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CAN-FD module either transmits a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFBRS bit is updated with the BRS bit value of the CAN-FD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CAN-FD frame.

In RX or GW mode, 0 is stored to the CFBRS bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

### CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CAN-FD module either transmits a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CAN-FD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CAN-FD frame (1).

Note: This bit is not available in the classical CAN function.

### CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTSn.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

### CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTSn.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

## 32.2.87.13 CFDCFDFpn_i : Common FIFO Access Data Field p Register n Channel i (p = 0 to 15, n = 0 to 2, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x640C + 0x004 × p + 0x080 × n + 0x180 × i

| | | | | |
|---|---|---|---|---|
| Bit position: | 31 24 | 23 16 | 15 8 | 7 0 |
| Bit field: | CFDB_HH[7:0] | CFDB_HL[7:0] | CFDB_LH[7:0] | CFDB_LL[7:0] |
| Value after reset: | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | CFDB_LL[7:0] | Common FIFO Buffer Data Bytes ((p * 4) | R/W |
| 15:8 | CFDB_LH[7:0] | Common FIFO Buffer Data Bytes ((p * 4) + 1) | R/W |
| 23:16 | CFDB_HL[7:0] | Common FIFO Buffer Data Bytes ((p * 4) + 2) | R/W |
| 31:24 | CFDB_HH[7:0] | Common FIFO Buffer Data Bytes ((p * 4) + 3) | R/W |

i = Channel number

The FIFO Access Data Field p Registers n (p = 0 to 15, n = 0 to 2) store data bytes (p * 4) to data bytes ((p * q) + 3) of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

### CFDB_LL[7:0] bits (Common FIFO Buffer Data Bytes ((p * 4))

The CFDB((p * 4)[7:0] bits store data bytes ((p * 4) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

### CFDB_LH[7:0] bits (Common FIFO Buffer Data Bytes ((p * 4) + 1))

The CFDB((p * 4) + 1)[7:0] bits store data bytes ((p * 4) + 1) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

### CFDB_HL[7:0] bits (Common FIFO Buffer Data Bytes ((p * 4) + 2))

The CFDB((p * 4) + 2)[7:0] bits store data bytes ((p * 4) + 2) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

### CFDB_HH[7:0] bits (Common FIFO Buffer Data Bytes ((p * 4) + 3))

The CFDB((p * 4) + 3)[7:0] bits store data bytes ((p * 4) + 3) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

Note 1. In RX or GW mode, unused data bytes are filled with 0x00 according to the configured data payload size CFDCFCCn.CFPLS, which is a CAN-FD feature not found in classical CAN.

## 32.2.87.14 CFDTMIDn_i : TX Message Buffer ID Register n Channel i (n = 0 to 7, 32 to 39, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1_0000 + 0x080 × n + 0x2000 × i

| Bit position: | 31 | 30 | 29 | 28 | | 0 |
|---|---|---|---|---|---|---|
| Bit field: | TMIDE | TMRTR | THLEN | TMID[28:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 28:0 | TMID[28:0] | TX Message Buffer ID Field<br>STD-ID/EXT-ID fields | R/W |
| 29 | THLEN | Tx History List Entry<br>0: Entry not stored in THL after successful TX<br>1: Entry stored in THL after successful TX | R/W |
| 30 | TMRTR | TX Message Buffer RTR bit<br>0: Data frame<br>1: Remote frame | R/W |
| 31 | TMIDE | TX Message Buffer IDE bit<br>0: STD-ID is transmitted<br>1: EXT-ID is transmitted | R/W |

Each TX Message Buffer ID Registers n (n = 0 to 7, 32 to 39) are used to store the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

### TMID[28:0] bits (TX Message Buffer ID Field)

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

### THLEN bit (Tx History List Entry)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

### TMRTR bit (TX Message Buffer RTR bit)

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CAN-FD format. For a CAN-FD transmission (CFDTMFDCTRn.CFFDF = 1), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

### TMIDE bit (TX Message Buffer IDE bit)

The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

## 32.2.87.15 CFDTMPTRn_i : TX Message Buffer Pointer Register n Channel i (n = 0 to 7, 32 to 39, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1_0004 + 0x080 × n + 0x2000 × i

| Bit position: | 31 |  |  | 28 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | TMDLC[3:0] |  |  |  | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 27:0 | — | The read values are undefined. The write value should be 0. | R/W |
| 31:28 | TMDLC[3:0] | TX Message Buffer DLC Field<br>Number of data bytes to be transmitted in a CAN frame. | R/W |

Each TX Message Buffer Pointer Register n (n = 0 to 7, 32 to 39) is used to store the DLC fields of the message to transmit from the associated buffer.

### TMDLC[3:0] bits (TX Message Buffer DLC Field)

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

### 32.2.87.16 CFDTMFDCTRn_i : TX Message Buffer CANFD Control Register n Channel i (n = 0 to 7, 32 to 39, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1_0008 + 0x080 × n + 0x2000 × i

| Bit position: | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | TMPTR[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit position: | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit field: | — | — | — | — | — | — | TMIFL[1:0] | | — | — | — | — | — | TMFDF | TMBRS | TMESI |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 0 | TMESI*1 | Error State Indicator bit<br>0: CANFD frame to transmit by error active node<br>1: CANFD frame to transmit by error passive node | R/W |
| 1 | TMBRS*1 | Bit Rate Switch bit<br>0: CANFD frame to transmit with no bit rate switch<br>1: CANFD frame to transmit with bit rate switch | R/W |
| 2 | TMFDF*1 | CAN FD Format bit<br>0: Non CANFD frame to transmit<br>1: CANFD frame to transmit | R/W |
| 7:3 | — | The read values are undefined. The write value should be 0. | R/W |
| 9:8 | TMIFL[1:0] | TX Message Buffer Information Label Field | R/W |
| 15:10 | — | The read values are undefined. The write value should be 0. | R/W |
| 31:16 | TMPTR[15:0] | TX Message Buffer Pointer Field | R/W |

Note 1. This bit is not available in the classical CAN function.

The TX Message Buffer CANFD Control Registers n (n = 0 to 7, 32 to 39) show the status of the FDF, BRS and ESI bits, including the pointer fields of the CANFD frame to be transmitted.

#### TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don’t care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

#### TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

#### TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

#### TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

### TMPTR[15:0] bits (TX Message Buffer Pointer Field)

The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

## 32.2.87.17 CFDTMDFp_n_i : TX Message Buffer Data Field p Register n Channel i (p = 0 to 15, n = 0 to 7, 32 to 39, i = 0, 1)

Base address: CANFD = 0x400B_0000

Offset address: 0x1_000C + 0x004 × p + 0x080 × n + 0x2000 × i

| | | | | |
|---|---|---|---|---|
| Bit position: | 31 24 | 23 16 | 15 8 | 7 0 |
| Bit field: | TMDB_HH[7:0] | TMDB_HL[7:0] | TMDB_LH[7:0] | TMDB_LL[7:0] |
| Value after reset: | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 |

| Bit | Symbol | Function | R/W |
|---|---|---|---|
| 7:0 | TMDB_LL[7:0] | TX Message Buffer Data Byte ((p * 4) | R/W |
| 15:8 | TMDB_LH[7:0] | TX Message Buffer Data Byte ((p * 4) + 1) | R/W |
| 23:16 | TMDB_HL[7:0] | TX Message Buffer Data Byte ((p * 4) + 2) | R/W |
| 31:24 | TMDB_HH[7:0] | TX Message Buffer Data Byte ((p * 4) + 3) | R/W |

i = Channel number

Each TX Message Buffer Data Field p Register n (p = 0 to 15, n = 0 to 7, 32 to 39) is used to store data bytes ((p * q) + (q - 4)) to data bytes ((p * 4) + (q - 1)) of the message to transmit from the associated buffer.

### TMDB_LL[7:0] bits (TX Message Buffer Data Byte ((p * 4))

Data bytes ((p * 4)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

### TMDB_LH[7:0] bits (TX Message Buffer Data Byte ((p * 4) + 1))

Data bytes ((p * 4) + 1)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

### TMDB_HL[7:0] bits (TX Message Buffer Data Byte ((p * 4) + 2))

Data bytes ((p * 4) + 2)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

### TMDB_HH[7:0] bits (TX Message Buffer Data Byte ((p * 4) + 3))

Data bytes ((p * 4) + 3)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD hannel is in CH_SLEEP mode.

## 32.3 Modes of Operation

## 32.3.1 Overview

The modes of the CANFD module can be classified into 2 groups:

- Global modes
- Channel modes

## 32.3.2 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes. The global modes of the CANFD module are:

- Global Sleep
- Global Reset
- Global Halt
- Global Operation.

Figure 32.2 shows the possible transitions between the Global modes.

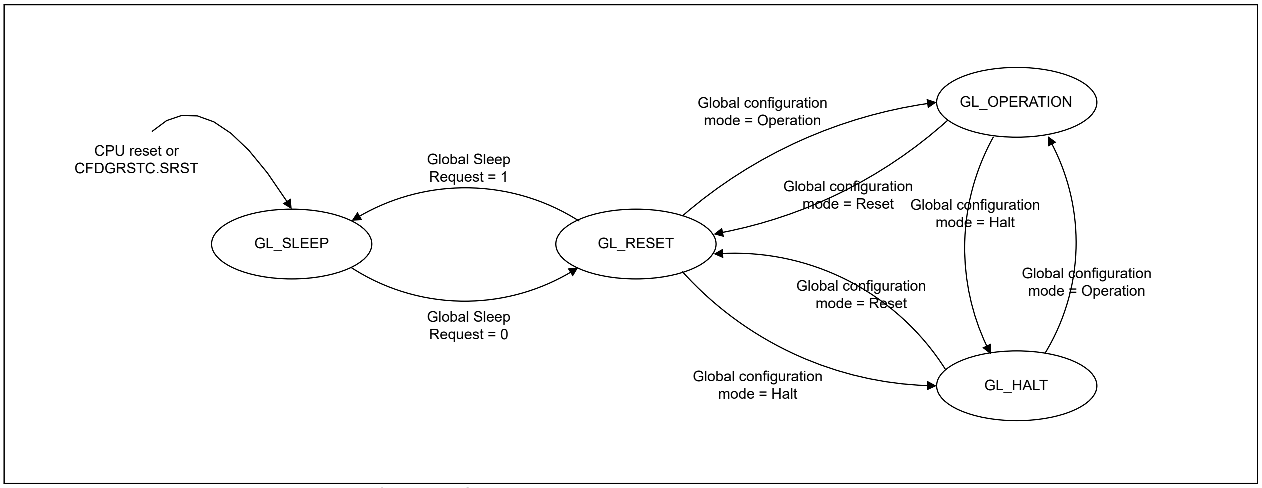


**Figure 32.2 Transition between CANFD Global modes**

Change in the Global mode can affect the Channel mode. Table 32.8 shows the effect of a Global mode transition on a Channel mode.

**Table 32.8 Possible CANFD Channel modes and Global modes**

| Current Global mode | Target Global mode | | | |
|---|---|---|---|---|
| | Sleep | Reset | Halt | Operation |
| Sleep | | Ch-Sleep: Keep<br>Ch-Reset: N/A<br>Ch-Halt: N/A<br>Ch-Oper: N/A | | |
| Reset | Ch-Sleep: Keep<br>Ch-Reset: → Ch-Sleep<br>Ch-Halt: N/A<br>Ch-Oper: N/A | | Ch-Sleep: Keep<br>Ch-Reset: Keep<br>Ch-Halt: N/A<br>Ch-Oper: N/A | Ch-Sleep: Keep<br>Ch-Reset: Keep<br>Ch-Halt: N/A<br>Ch-Oper: N/A |
| Halt | | Ch-Sleep: Keep<br>Ch-Reset: Keep<br>Ch-Halt: → Ch-Reset<br>Ch-Oper: N/A | | Ch-Sleep:Keep<br>Ch-Reset: Keep<br>Ch-Halt: Keep<br>Ch-Oper: N/A |
| Operation | | Ch-Sleep: Keep<br>Ch-Reset: Keep<br>Ch-Halt: → Ch-Reset<br>Ch-Oper: → Ch-Reset | Ch-Sleep: Keep<br>Ch-Reset: Keep<br>Ch-Halt: Keep<br>Ch-Oper: → Ch-Halt | |

### 32.3.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module automatically enters Global Sleep mode.

The CANFD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets all Channel Sleep Request bits and forces all channels into the Channel Sleep mode.

Sleep mode is used for power saving purpose. When CANFD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.

Read access from all registers is still possible and all register values are preserved.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

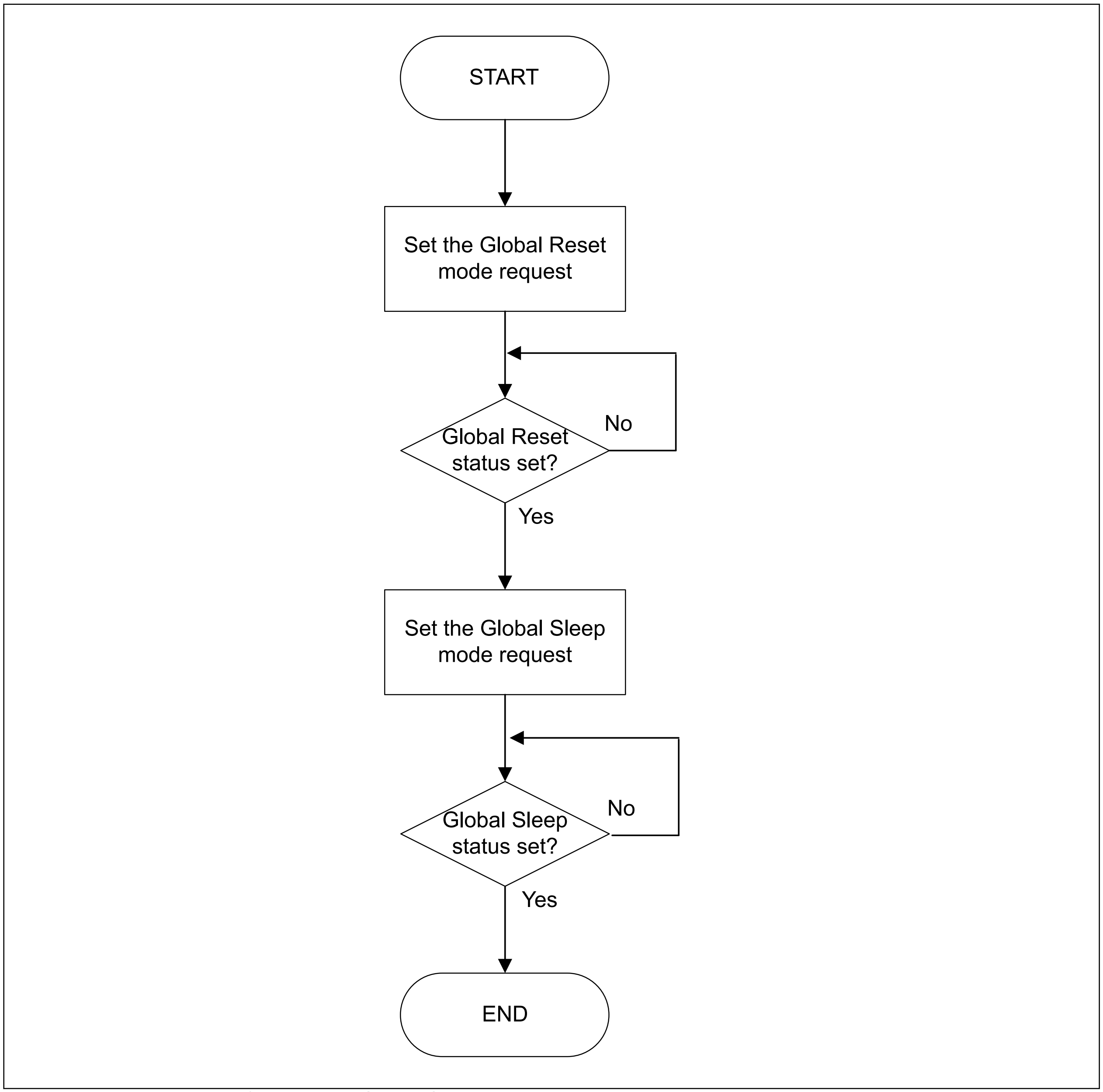

START
Set the Global Reset mode request
Global Reset status set?
No
Yes
Set the Global Sleep mode request
Global Sleep status set?
No
Yes
END

**Figure 32.3 Procedure for entering Global Sleep mode**

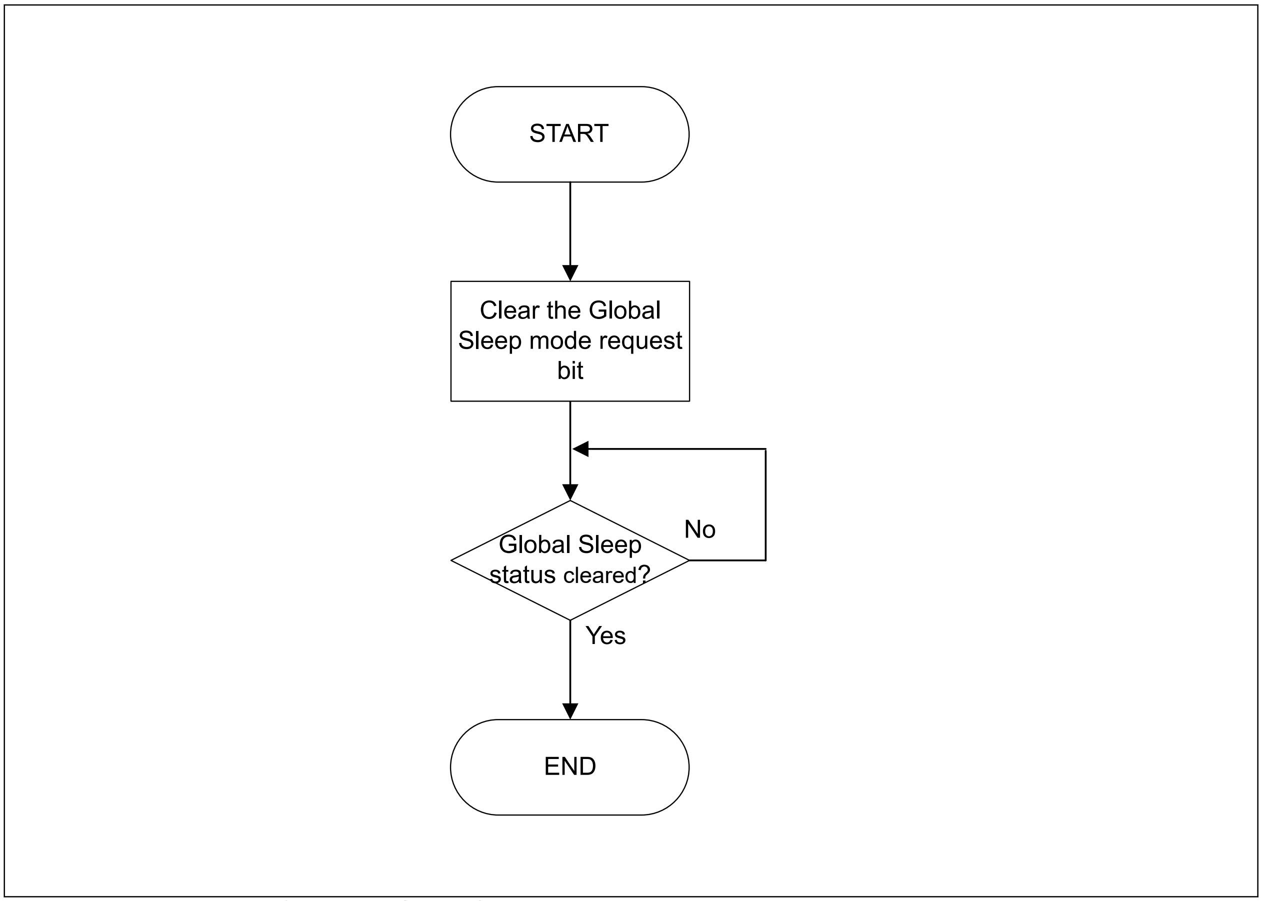

START
Clear the Global Sleep mode request bit
Global Sleep status cleared?
No
Yes
END

**Figure 32.4 Procedure for exiting Global Sleep mode**

### 32.3.2.2 Global Reset Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Reset mode while the CANFD module is in Global Halt or Global Operation mode
- Global Sleep Mode Request bit is cleared while CANFD module is in Global Sleep mode.

In Global Reset mode, all CANFD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and all channel TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their MCU reset values and the CANFD module can be configured.

See section 32.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Reset mode is performed.

Setting the Global mode to Reset by setting the Global Mode Control bits CFDGCTR.GMDC in the Global Control Register to 01b sets all Channel Mode Control bits CFDCnCTR.CHMDC in the Channel Control Registers to 01b and forces all channels into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (CFDCnCTR.CHMDC of related channel already set to 01b).

After setting Global Mode Control bit CFDGCTR.GMDC to Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDGSTS.GRSTSTS in the Global Status Register has been updated, indicating successful transition to Global Reset mode before CFDGCTR.GMDC can be changed again.

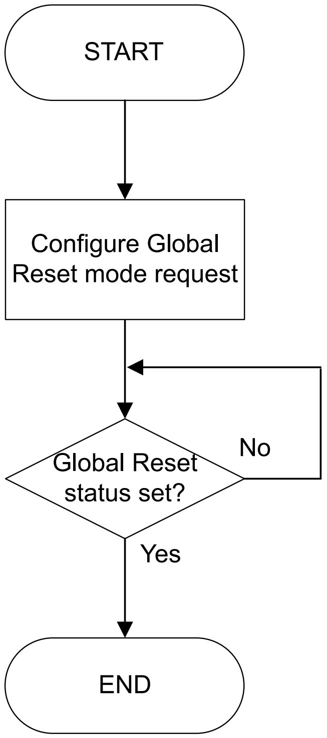


**Figure 32.5 Procedure for entering Global Reset mode**

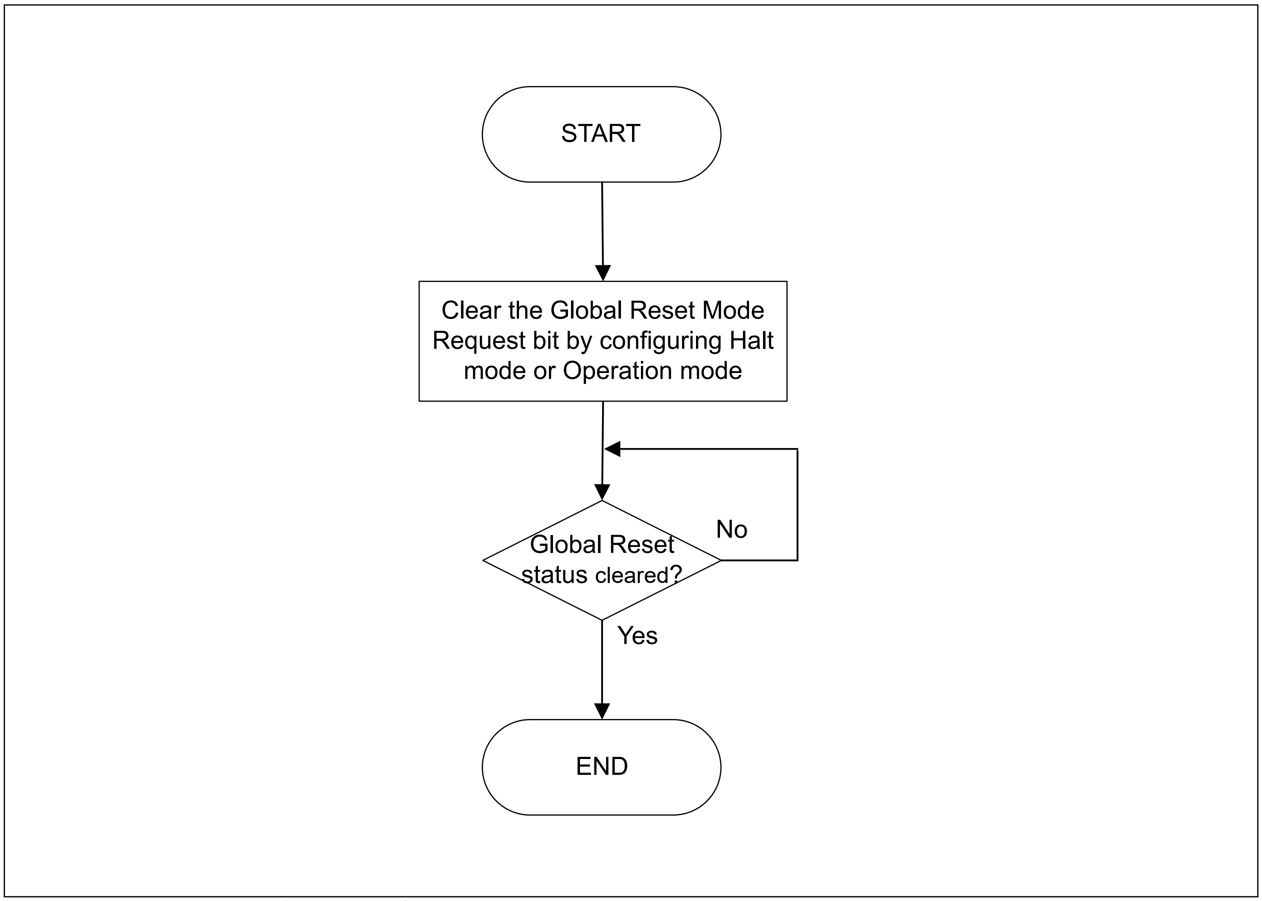

START
Clear the Global Reset Mode Request bit by configuring Halt mode or Operation mode
Global Reset status cleared?
No
Yes
END

**Figure 32.6 Procedure for exiting Global Reset mode**

### 32.3.2.3 Global Halt Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Reset mode:
  - the channels in either Channel Reset or Channel Sleep mode remain in this mode
- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Operation mode:
  - all channels in Channel Reset, Channel Halt, or Channel Sleep mode remain in this mode
  - all channels in Channel Operation mode transition to Channel Halt mode
  - Global Halt Mode Status bit is set when all channels have left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CANFD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See section 32.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt by setting the Global Mode Control bit CFDGCTR.GMDC in the Global Control Register to 10b sets all Channel Mode Control bits CFDCnCTR.CHMDC in the Channel Control Registers to 10b for the channels that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For channels that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CANFD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bit CFDGCTR.GMDC to Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming CFDGSTS.GHLTSTS is set.

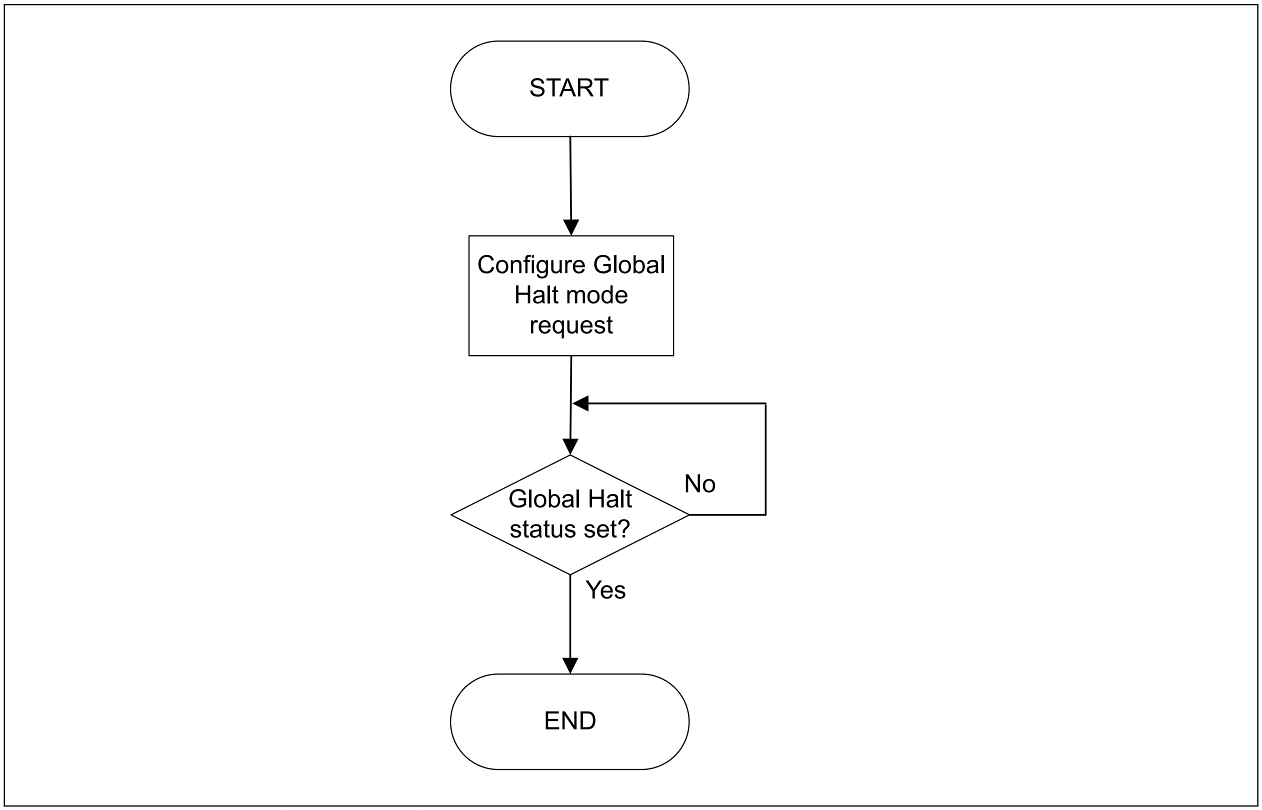


**Figure 32.7 Procedure for entering Global Halt mode**

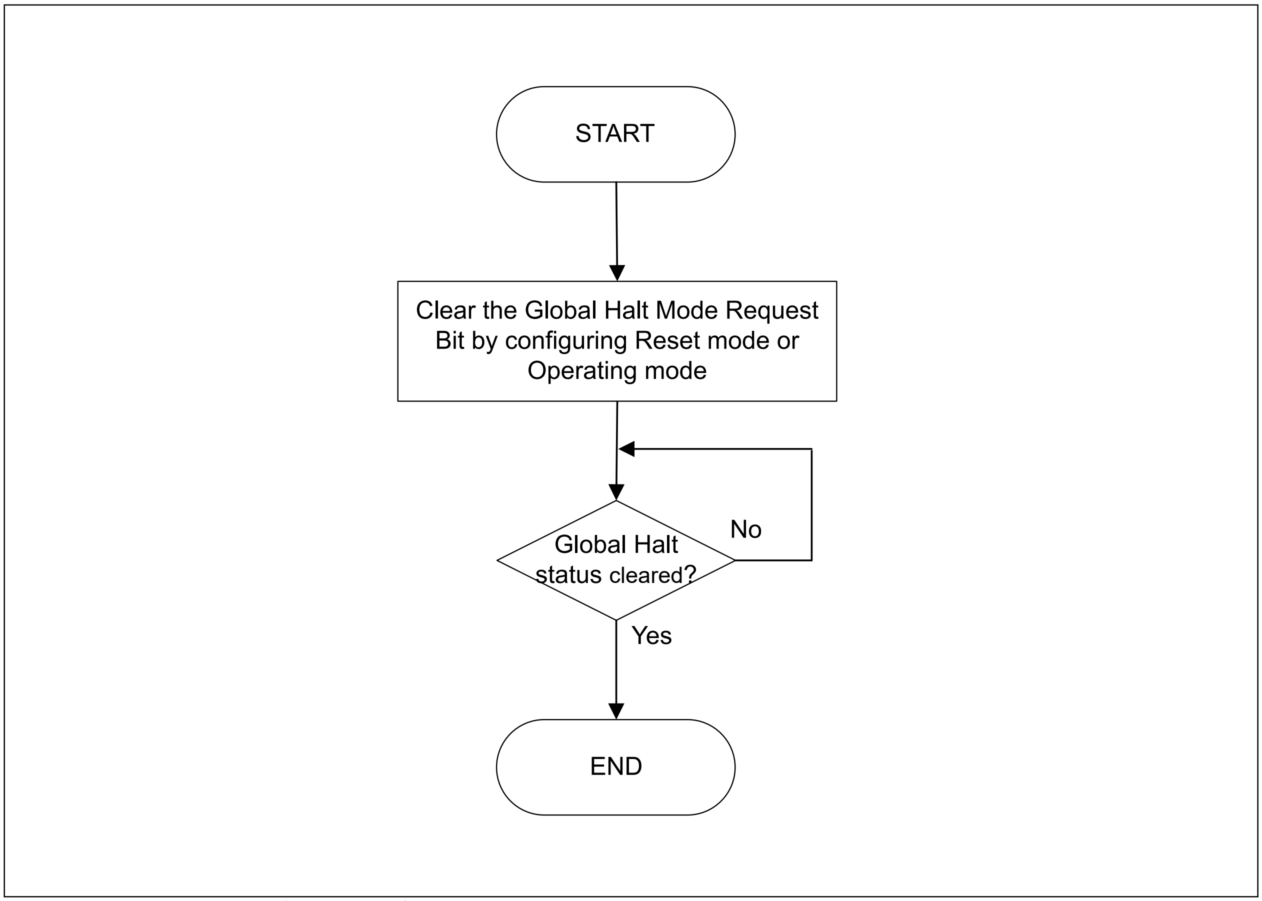

START
Clear the Global Halt Mode Request Bit by configuring Reset mode or Operating mode
Global Halt status cleared?
No
Yes
END

**Figure 32.8 Procedure for exiting Global Halt mode**

### 32.3.2.4 Global Operation Mode

The CANFD module enters this mode when the Global Mode Configuration bits are set to Global Operation mode.

The CANFD channels can only be set to Channel Operation mode and start CAN communication when CANFD is in Global Operation mode.

After setting the Global Mode Control bit CFDGCTR.GMDC to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit CFDGSTS.GRSTSTS and the Global Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before CFDGCTR.GMDC can be modified again.

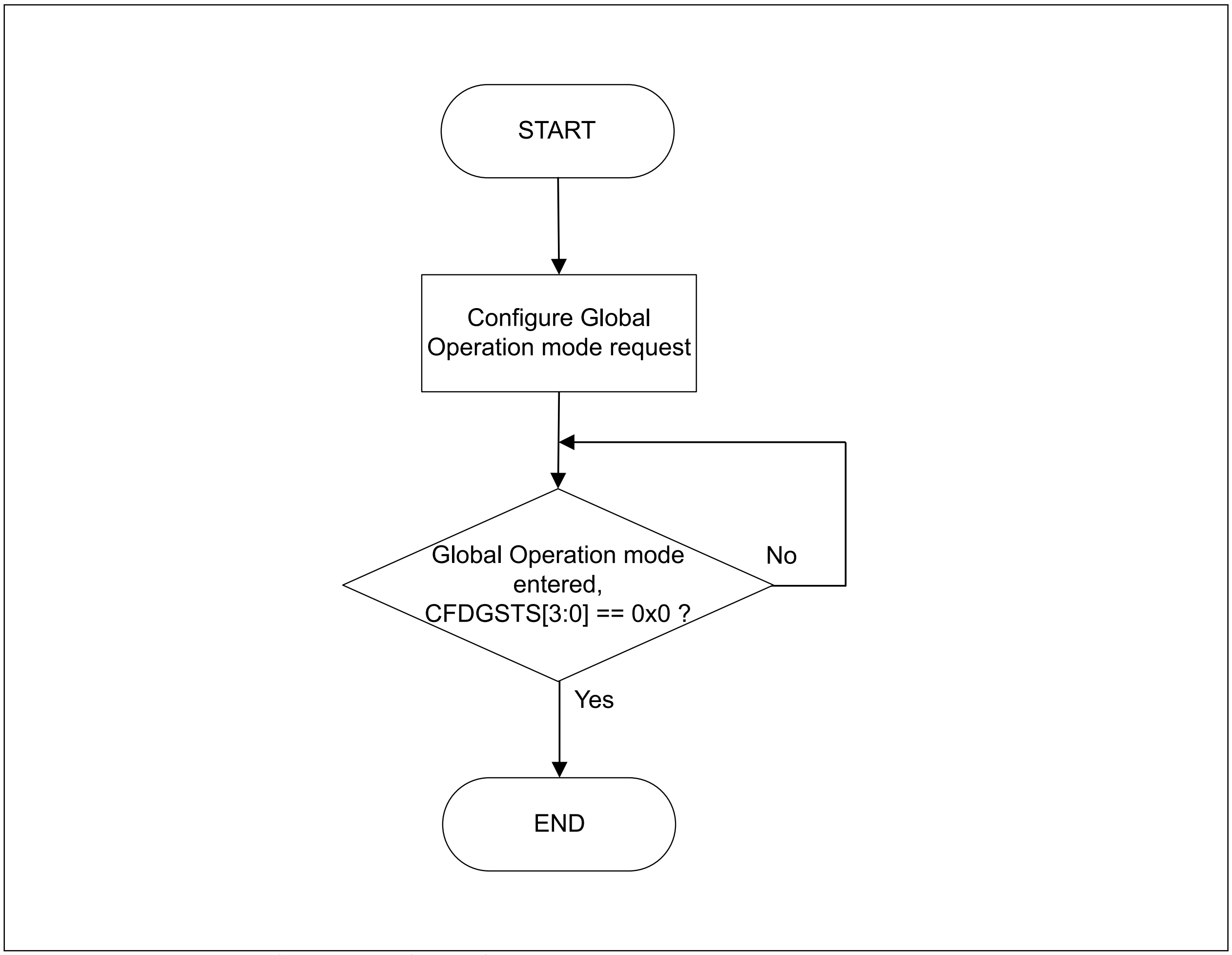

START
Configure Global Operation mode request
Global Operation mode entered, CFDGSTS[3:0] == 0x0 ?
No
Yes
END

**Figure 32.9 Procedure for entering Global Operation mode**

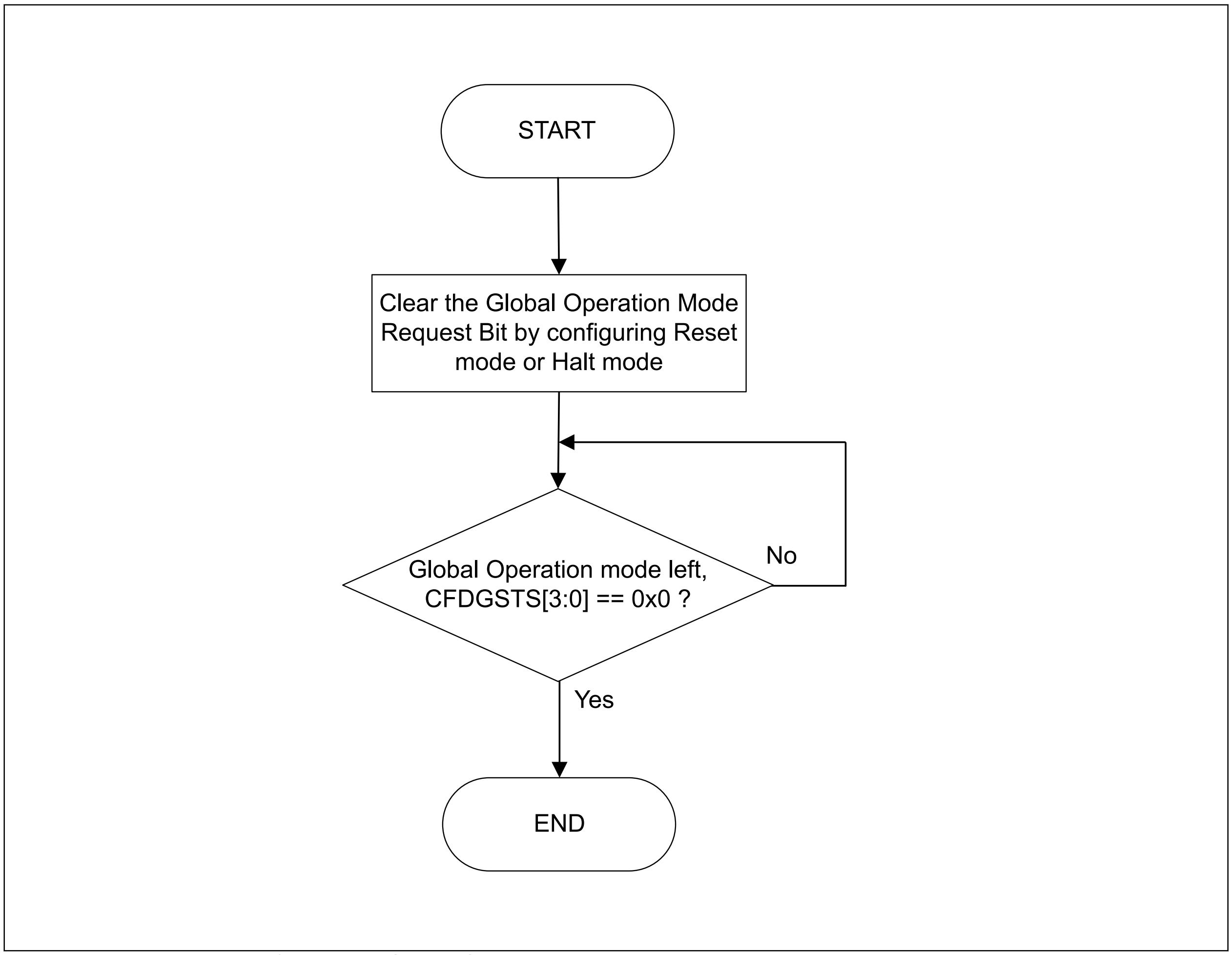

START
Clear the Global Operation Mode Request Bit by configuring Reset mode or Halt mode
Global Operation mode left, CFDGSTS[3:0] == 0x0 ?
No
Yes
END

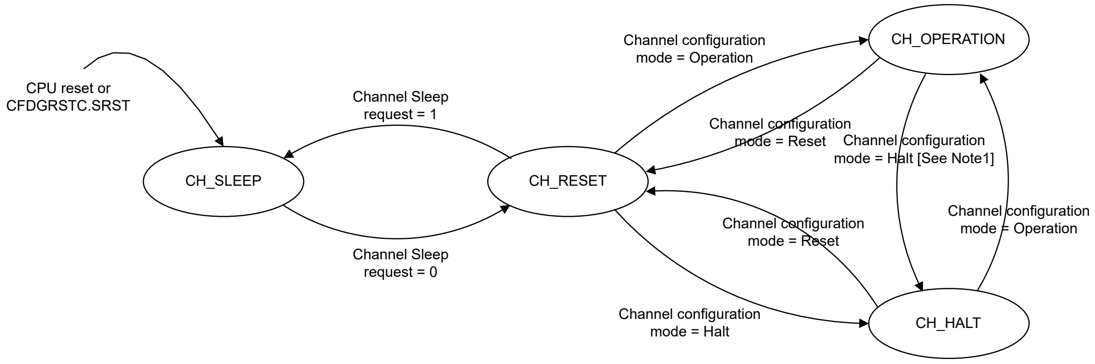
**Figure 32.10 Procedure for exiting Global Operation mode**

### 32.3.3 Channel Modes

Each CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation
- Sleep.

Figure 32.11 shows the possible transitions between the channel modes.



Note 1. The transition timing depends on BOM setting in the Channel Configuration Register:

- when BOM = 01b, the state transition timing is immediate after entering bus-off state
- when BOM = 10b, the state transition timing is at the end of bus-off state
- when BOM = 11b, the state transition timing is matching with the setting of CHMDC to Halt mode.

**Figure 32.11 Transition between CAN channel modes**

### 32.3.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, each CAN channel of the CANFD module automatically enters Channel Sleep mode.

Each CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

### 32.3.3.2 CAN Channel Reset Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDCnCTR.CHMDC in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally all channel related transmission control bits are cleared and the channel related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See section 32.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bit CFDCnCTR.CHMDC to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDCnSTS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDCnCTR.CHMDC bit can be modified again.

See Table 32.9 for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

### 32.3.3.3 CAN Channel Halt Mode

An CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDCnCTR.CHMDC in the Channel Control Registers is configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for this channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure channel test modes.

See section 32.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bit CFDCnCTR.CHMDC to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDCnSTS.CHLTSTS in the related Channel Status Register has been updated to indicate a successful transition to Channel Halt mode before the related CFDCnCTR.CHMDC can be modified again.

See Table 32.9 for the transition behavior to Channel Halt mode while CAN communication is ongoing.

**Table 32.9 Transition behavior in CAN Reset mode and Halt mode**

| | State | | |
|---|---|---|---|
| **Mode** | **Receiver** | **Transmitter** | **Bus-Off** |
| **CAN Channel Reset mode (CFDCnCTR.CHMDC = 01b)** | The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing reception.[*1] | The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing transmission.[*1] | The CAN channel enters Channel Reset mode without waiting for the completion of the bus-off recovery. |
| **CAN Channel Halt mode (CFDCnCTR.CHMDC = 10b)** | CAN channel enters Channel Halt mode at the end of the ongoing reception or error.[*2] | CAN channel enters Channel Halt mode after completion of the ongoing transmission. | When CFDCnCTR.BOM is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence.<br>When CFDCnCTR.BOM is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery.<br>When CFDCnCTR.BOM is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery.<br>When CFDCnCTR.BOM is set to 11b, the CAN channel enters Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery). |

Note 1. If the entry to Channel Reset mode is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to Channel Reset mode.

### 32.3.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDCnCTR.CHMDC bits to 00b. If 11 consecutive recessive bits are detected after entering the CAN Operation mode, the CFDCnSTS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see Figure 32.12):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

  Note: The channel may receive its own message simultaneously when Self-test mode is enabled.
- Channel is in bus-off state: The CAN channel is cut-off from CAN bus communication.

After setting the Channel Mode Control bit CFDCnCTR.CHMDC to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDCnSTS.CRSTSTS and the Channel Halt Mode Status bit CFDCnSTS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDCnCTR.CHMDC bit can be changed again.

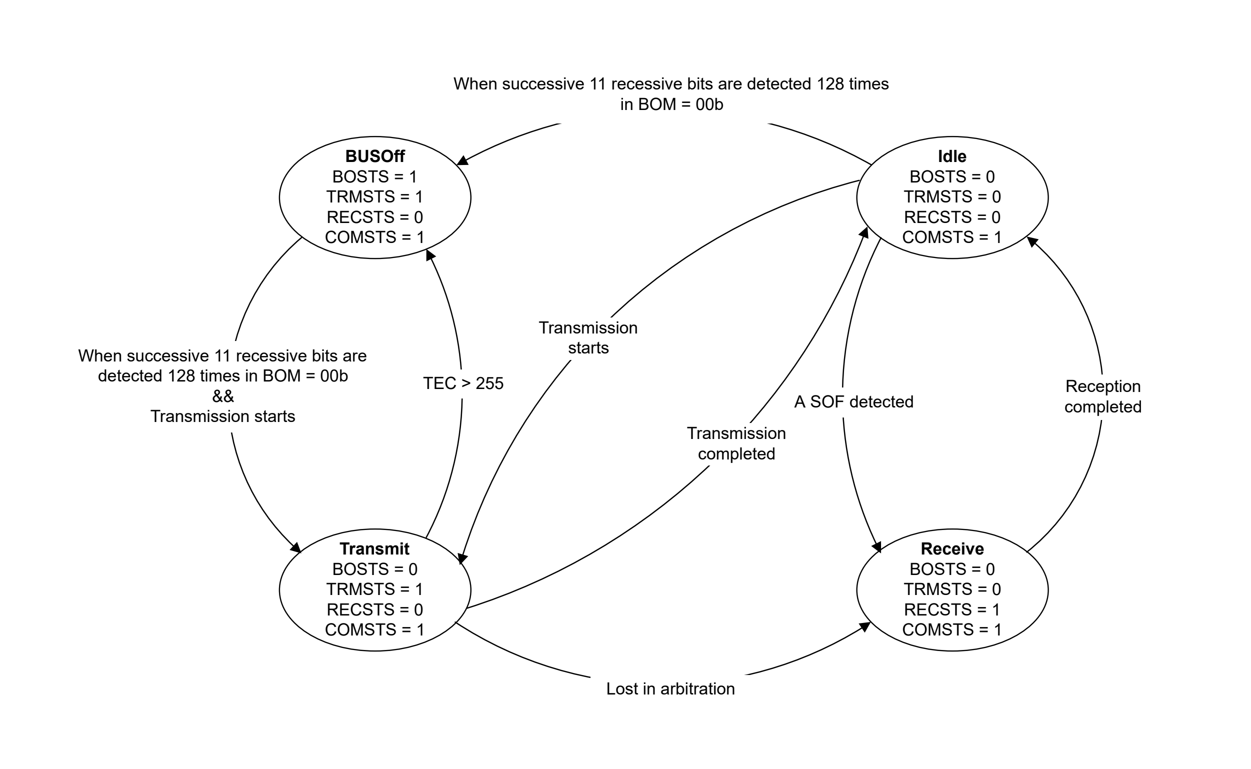


**Figure 32.12 Sub-modes of CAN Channel Operation mode (only when BOM = 00b)**

### 32.3.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDCnCTR.BOM = 00b:
  Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0. The Bus-Off Recovery Flag CFDCnERFL.BORF is set in this case.
- CFDCnCTR.BOM = 01b:

The CAN channel changes the value of the CFDCnCTR.CHMDC bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set in this case.

- CFDCnCTR.BOM = 10b:
  The CAN channel changes the value of the CFDCnCTR.CHMDC bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is set in this case.
- CFDCnCTR.BOM = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.
  TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set.
  Without setting CFDCnCTR.CHMDC [1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CFDCnCTR.BOM = 00b.

Note: If the recovery from bus-off occurs normally in this mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDCnERFL.BORF is set.

When software writes to the CFDCnCTR.CHMDC bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM = 01b, or at the end of bus-off when CFDCnCTR.BOM = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDCnCTR.CHMDC bit to Channel Halt mode request is performed when the CFDCnCTR.CHMDC bit value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDCnCTR.RTBO to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0.

Before setting CFDCnCTR.RTBO to 1, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX or GW mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSn.TMTRF). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTSn.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTSn.CFEMP).

The CFDCnCTR.RTBO bit should be used for bus-off recovery only when CFDCnCTR.BOM is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 32.10 shows the settings for the Bus-Off Entry flag CFDCnERFL.BOEF and the Bus-Off Recovery flag CFDCnERFL.BORF for the different configurations of CFDCnCTR.BOM.

**Table 32.10 Behavior of Bus-off Entry and Recovery flags**

| BOM | BOEF bit set | BORF bit set |
|---|---|---|
| 00b | Always (on entry to bus-off) | Always (on exit from bus-off) |
| 00b<br>CFDCnCTR.RTBO set to 1 | Always (on entry to bus-off) | Only if normal bus-off recovery occurs before software sets CFDCnCTR.RTBO to 1' |
| 01b | Always (on entry to bus-off) | Never |
| 10b | Always (on entry to bus-off) | Always (on exit from bus-off) |
| 11b | Always (on entry to bus-off) | Only if normal bus-off recovery occurs before software issues a Halt request |

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 32.13.

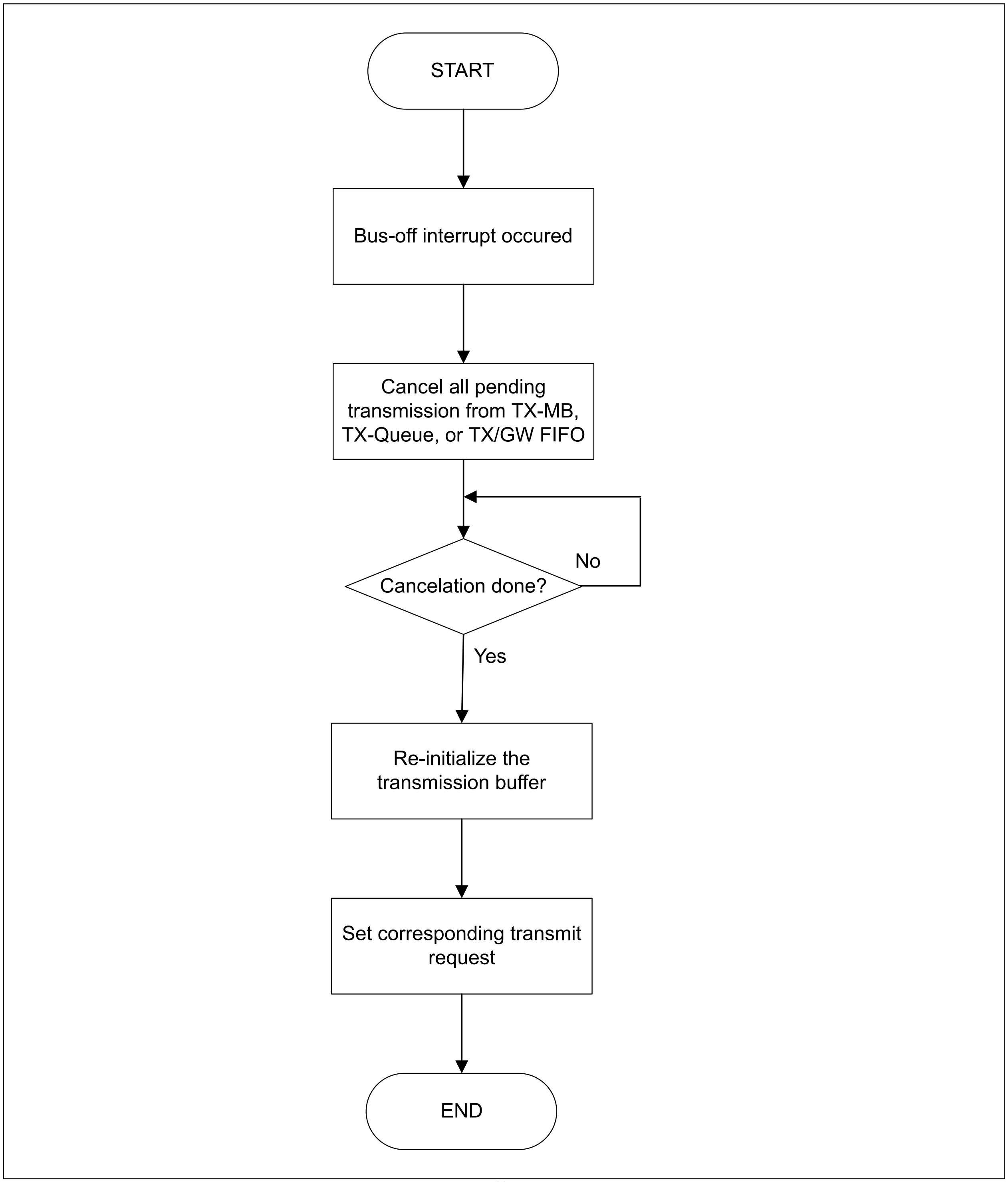

START
Bus-off interrupt occured
Cancel all pending transmission from TX-MB, TX-Queue, or TX/GW FIFO
Cancelation done?
No
Yes
Re-initialize the transmission buffer
Set corresponding transmit request
END

**Figure 32.13 Transmission re-initialization during bus-off**

### 32.3.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the Channel Mode Control bit CFDCnCTR.CHMDC in the Channel Control Registers does not affect the Global Mode Control bit CFDGCTR.GMDC.
- Changing the Global Mode Control bit CFDGCTR.GMDC affects the channel mode control as described in Table 32.11.

**Table 32.11 Interaction between Global and Channel mode transition**

| Global mode change | Channel mode | Channel mode transition action |
|---|---|---|
| Sleep → Reset | Sleep | Channel remains in Sleep mode |
| Sleep → Halt | — (Global mode change not possible) | |
| Sleep → Operation | — (Global mode change not possible) | |
| Reset → Sleep | Sleep | Channels remains in Sleep mode |
| | Reset | Channel Sleep request bit is set automatically, channel enters Sleep Mode |
| Reset → Halt | Sleep | Channel remains in Sleep mode |
| | Reset | Channel remains in Reset mode |
| Reset → Operation | Sleep | Channel remains in Sleep mode |
| | Reset | Channel remains in Reset mode |
| Halt → Sleep | — (Global mode change not possible) | |
| Halt → Reset | Sleep | Channel remains in Sleep mode |
| | Reset | Channel remains in Reset mode |
| | Halt | Channel mode control is set to Reset mode, channel enters Reset mode |
| Halt → Operation | Sleep | Channel remains in Sleep mode |
| | Reset | Channel remains in Reset mode |
| | Halt | Channel remains in Halt mode |
| Operation → Sleep | — (Global mode change not possible) | |
| Operation → Reset | Sleep | Channel remains in Sleep mode |
| | Reset | Channel remains in Reset mode |
| | Halt | Channel mode control is set to Reset mode, channel enters Reset mode |
| | Operation | Channel mode control is set to Reset mode, channel enters Reset mode |
| Operation → Halt | Sleep | Channel remains in Sleep mode |
| | Reset | Channel remains in Reset mode |
| | Halt | Channel remains in Halt mode |
| | Operation | Channel mode control is set to Halt mode, channel enters Halt mode after communication finished |

### 32.3.4.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

**Table 32.12 Maximum transition time for the global mode (1 of 2)**

| From | To | Maximum transition time |
|---|---|---|
| GL_SLEEP | GL_RESET | 3 peripheral clock cycles*2 |
| GL_RESET | GL_SLEEP | 3 peripheral clock cycles |
| GL_RESET | GL_HALT | 10 peripheral clock cycles |
| GL_RESET | GL_OPERATION | 10 peripheral clock cycles |

**Table 32.12 Maximum transition time for the global mode (2 of 2)**

| From | To | Maximum transition time |
|---|---|---|
| GL_HALT | GL_RESET | 2 CAN bit times |
| GL_HALT | GL_OPERATION | 3 peripheral clock cycles |
| GL_OPERATION | GL_RESET | 2 CAN bit times |
| GL_OPERATION | GL_HALT | 3 CAN frames[*1] [*3] |

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 2. Exit GL_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.

Note 3. TQ, CAN frame and CAN bits are related to the individual channels. For the maximum transition time, the channel with the lowest baud rate must be used.

#### 32.3.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes are shown in the following table.

**Table 32.13 Maximum transition time for the channel mode**

| From | To | max. transition time |
|---|---|---|
| CH_SLEEP | CH_RESET | 3 peripheral clock cycles |
| CH_RESET | CH_SLEEP | 3 peripheral clock cycles |
| CH_RESET | CH_HALT | 3 CAN bit times |
| CH_RESET | CH_OPERATION | 4 CAN bit times |
| CH_HALT | CH_RESET | 2 CAN bit times |
| CH_HALT | CH_OPERATION | 4 CAN bit times[*3] |
| CH_OPERATION | CH_RESET | 2 CAN bit times |
| CH_OPERATION | CH_HALT | 2 CAN frames[*1] [*2] |

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDCnCTR.BOM[1:0] bits.

Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 3. In general, if the baudrate prescaler value CFDCnNCFG.NBRP is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

## 32.4 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud Rate setting (nominal and data rate)
- CANFD setting
- Acceptance Filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

### 32.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate

#### 32.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the restriction that apply to the segment setting.

1. Each segment setting
   SS = Fixed to 1 TQ

TSEG1 = See to (CFDCnNCFG) and (CFDCnDCFG)[*1]
TSEG2 = See to (CFDCnNCFG) and (CFDCnDCFG)[*1]
SJW = See to (CFDCnNCFG) and (CFDCnDCFG)[*1]
SS + TSEG1 + TSEG2 = 5 to 49 TQs for Data Bit Rate and 8 to 385 for Nominal Bit Rate

2. Restriction on TSEG1, TSEG2 and SJW
TSEG1(N) > TSEG2(N) ≥ SJW(N)
TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)[*1]
When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDCnDCFG to valid values.

Note 1. This feature is not available in the classical CAN function.

Table 32.14 shows an example of how to set the bit timing to achieve the required Sample Point settings.

**Table 32.14 Bit timing examples**

| | Set value (TQ) | | | | |
|---|---|---|---|---|---|
| 1 bit | SS | TSEG1 | TSEG2 | SJW | Sample point[*1](%) |
| 5TQ | 1 | 2 | 2 | 1 | 60.00 |
| 8TQ | 1 | 4 | 3 | 1 | 62.50 |
| | 1 | 5 | 2 | 1 | 75.00 |
| 10TQ | 1 | 6 | 3 | 1 | 70.00 |
| | 1 | 7 | 2 | 1 | 80.00 |
| 12TQ | 1 | 8 | 3 | 1 | 75.00 |
| | 1 | 9 | 2 | 1 | 83.33 |
| 15TQ | 1 | 10 | 4 | 1 | 73.33 |
| | 1 | 11 | 3 | 1 | 80.00 |
| 16TQ | 1 | 10 | 5 | 1 | 68.75 |
| | 1 | 11 | 4 | 1 | 75.00 |
| 20TQ | 1 | 12 | 7 | 1 | 65.00 |
| | 1 | 13 | 6 | 1 | 70.00 |
| 24TQ | 1 | 15 | 8 | 1 | 66.66 |
| | 1 | 16 | 7 | 1 | 70.83 |
| 50TQ | 1 | 39 | 10 | 4 | 80.00 |

Note 1. Sample point (in case of 75%)

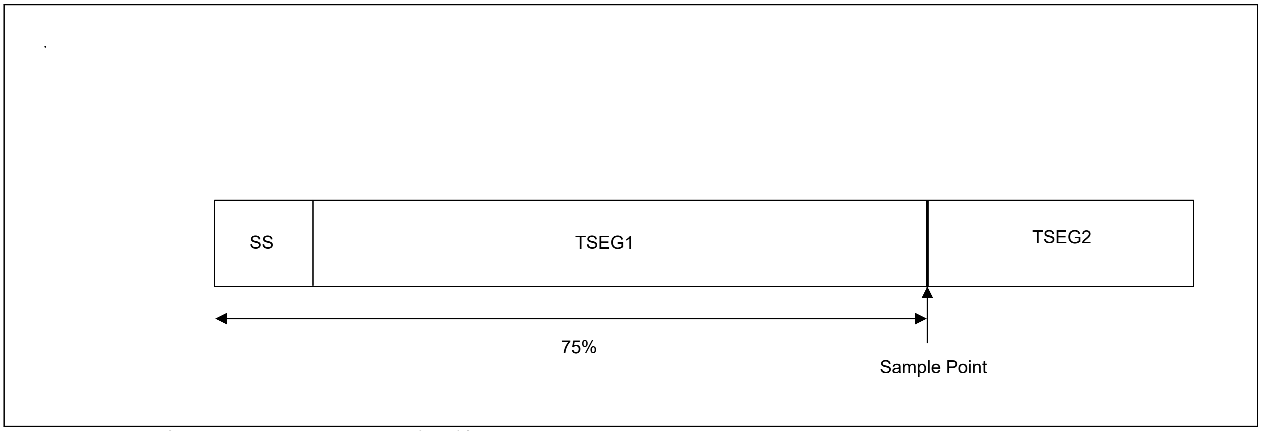


**Figure 32.14 Sample point (in case of 75%)**

### 32.4.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for each channel using the related CFDCnNCFG and CFDCnDCFG[*1] registers.

Note 1. This register is not available in the classical CAN function.

Figure 32.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).

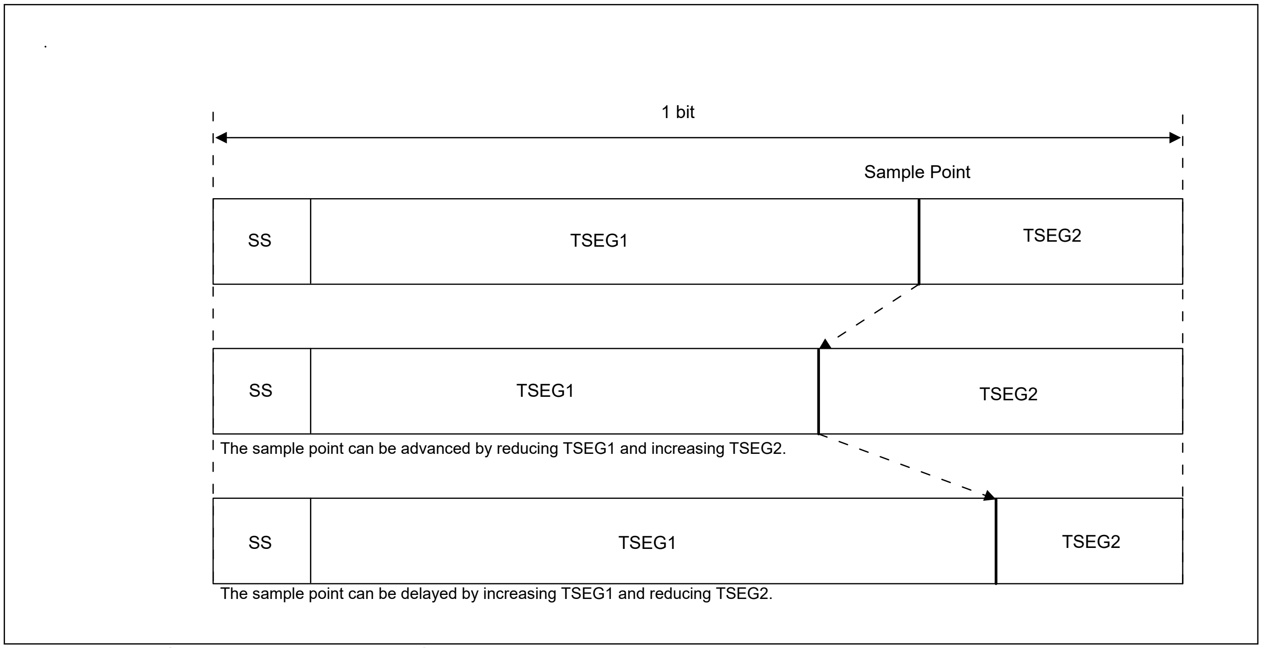

1 bit
Sample Point
SS
TSEG1
TSEG2
SS
TSEG1
TSEG2
The sample point can be advanced by reducing TSEG1 and increasing TSEG2.
SS
TSEG1
TSEG2
The sample point can be delayed by increasing TSEG1 and reducing TSEG2.

**Figure 32.15 Segment composition of a bit and the sample point**

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 32.15 shows only one symbolic sample point.

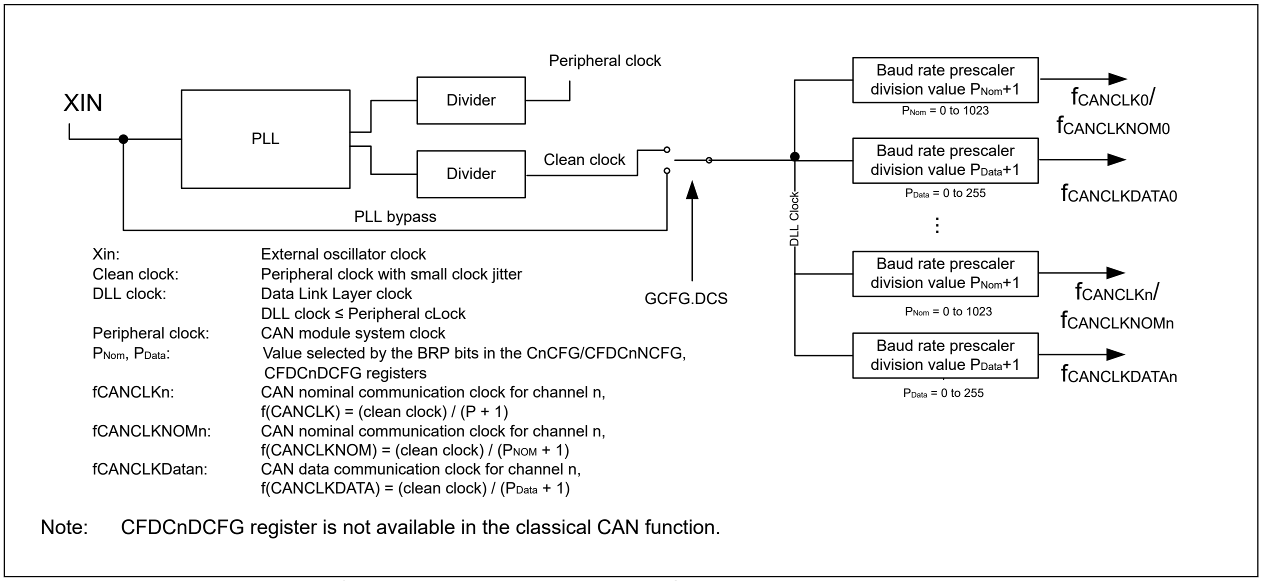
### 32.4.1.3 Baud Rate

Either the CAN channel system clock (clean clock) or the external oscillator clock can be selected globally for all CAN channels as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baudrate} = \frac{\text{DLL\_Clock}}{(\text{number\_of\_time\_quanta\_per\_bit}) \times (\text{BRP} + 1)}$$

Figure 32.16 shows a block diagram of the circuit that generates the CAN channel system clock and Table 32.15 shows a baud rate examples.



Xin: External oscillator clock
Clean clock: Peripheral clock with small clock jitter
DLL clock: Data Link Layer clock
DLL clock ≤ Peripheral cLock
Peripheral clock: CAN module system clock
$P_{Nom}$, $P_{Data}$: Value selected by the BRP bits in the CnCFG/CFDCnNCFG, CFDCnDCFG registers
fCANCLKn: CAN nominal communication clock for channel n, f(CANCLK) = (clean clock) / (P + 1)
fCANCLKNOMn: CAN nominal communication clock for channel n, f(CANCLKNOM) = (clean clock) / ($P_{NOM}$ + 1)
fCANCLKDatan: CAN data communication clock for channel n, f(CANCLKDATA) = (clean clock) / ($P_{Data}$ + 1)

Note: CFDCnDCFG register is not available in the classical CAN function.

**Figure 32.16 Block diagram of the circuit that generates the CAN channel communication clock**

**Table 32.15 Nominal baud rate calculation formula and example CAN communication configurations**

| Baud rate calculation formula | (DLL clock) (baud rate prescaler divide-by-N value[*1]) × (number of TQs in one bit) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 80 MHz | 40 MHz | 32 MHz | 30 MHz | 24 MHz | 20 MHz | 16 MHz | 10 MHz | 8 MHz[*2] |
| 1 Mbps | 8TQ (10)<br>20TQ (4) | 8TQ (5)<br>20TQ (2) | 8TQ (4)<br>16TQ (2) | 10TQ (3)<br>15TQ (2) | 8TQ (3)<br>12TQ (2)<br>24TQ (1) | 10TQ (2)<br>20TQ (1) | 8TQ (2)<br>16TQ (1) | 10TQ (1) | 8TQ (1) |
| 500 Kbps | 8TQ (20)<br>20TQ (8) | 8TQ (10)<br>20TQ (4) | 8TQ (8)<br>16TQ (4) | 10TQ (6)<br>15TQ (4)<br>20TQ (3) | 8TQ (6)<br>12TQ (4)<br>24TQ (2) | 10TQ (4)<br>20TQ (2) | 8TQ (4)<br>16TQ (2) | 10TQ (2)<br>20TQ (1) | 8TQ (2)<br>16TQ (1) |
| 250 Kbps | 8TQ (40)<br>20TQ (16) | 8TQ (20)<br>20TQ (8) | 8TQ (16)<br>16TQ (8) | 10TQ (12)<br>15TQ (8)<br>20TQ (6) | 8TQ (12)<br>12TQ (8)<br>24TQ (4) | 10TQ (8)<br>20TQ (4) | 8TQ (8)<br>16TQ (4) | 10TQ (4)<br>20TQ (2) | 8TQ (4)<br>16TQ (2) |
| 125 Kbps | 8TQ (80)<br>20TQ (32) | 8TQ (40)<br>20TQ (16) | 8TQ (32)<br>16TQ (16) | 10TQ (24)<br>15TQ (16)<br>20TQ (12) | 8TQ (24)<br>12TQ (16)<br>24TQ (8) | 10TQ (16)<br>20TQ (8) | 8TQ (16)<br>16TQ (8) | 10TQ (8)<br>20TQ (4) | 8TQ (8)<br>16TQ (4) |
| 83.3 Kbps | 8TQ (120)<br>12TQ (80)<br>16TQ (60)<br>24TQ (40) | 8TQ (60)<br>12TQ (40)<br>16TQ (30)<br>24TQ (20) | 8TQ (48)<br>12TQ (32)<br>16TQ (24)<br>24TQ (16) | 8TQ (45)<br>10TQ (36)<br>12TQ (30)<br>15TQ (24)<br>20TQ (18)<br>24TQ (15) | 8TQ (36)<br>12TQ (24)<br>16TQ (18)<br>24TQ (12) | 8TQ (30)<br>10TQ (24)<br>12TQ (20)<br>15TQ (16)<br>16TQ (15)<br>20TQ (12)<br>24TQ (10) | 8TQ (24)<br>12TQ (16)<br>16TQ (12)<br>24TQ (8) | 8TQ (15)<br>10TQ (12)<br>12TQ (10)<br>15TQ (8)<br>20TQ (6)<br>24TQ (5) | 8TQ (12) |
| 33.3 Kbps | 8TQ (300)<br>12TQ (200)<br>16TQ (150)<br>20TQ (120)<br>24TQ (100) | 8TQ (150)<br>12TQ (100)<br>16TQ (75)<br>20TQ (60)<br>24TQ (50) | 8TQ (120)<br>10TQ (96)<br>12TQ (80)<br>15TQ (64)<br>16TQ (60)<br>20TQ (48)<br>24TQ (40) | 10TQ (90)<br>12TQ (75)<br>15TQ (60)<br>20TQ (45) | 8TQ (90)<br>10TQ (72)<br>12TQ (60)<br>15TQ (48)<br>16TQ (45)<br>20TQ (36)<br>24TQ (30) | 8TQ (75)<br>10TQ (60)<br>12TQ (50)<br>15TQ (40)<br>20TQ (30)<br>24TQ (25) | 8TQ (60)<br>10TQ (48)<br>12TQ (40)<br>15TQ (32)<br>16TQ (30)<br>20TQ (24)<br>24TQ (20) | 10TQ (30)<br>12TQ (25)<br>15TQ (20)<br>20TQ (15) | 8TQ (30) |

Note: Shown in ( ) are the baud rate prescaler divide-by-N value.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.
Note 2. Minimum frequency to achieve maximum nominal baud rate of 1 Mbps.

Table 32.16 Baud rate calculation example for nominal and data bit rate CAN communication configurations

| | (DLL clock)<br>(baud rate prescaler divide-by-N value[*1]) × (number of TQs in one bit) | | |
|---|---|---|---|
| **Baud rate calculation formula** | **80 MHz** | **40 MHz** | **20 MHz** |
| Nominal 1 Mbps<br>Data 8 Mbps | 80TQ (1) | 40TQ (1) | 20TQ (1) |
| | 10TQ (1) | 5TQ (1) | Not possible |
| Nominal 1 Mbps<br>Data 5 Mbps | 80TQ (1) | 40TQ (1) | 20TQ (1) |
| | 16TQ (1) | 8TQ (1) | Not possible |
| Nominal 500 Kbps<br>Data 2 Mbps | 160TQ (1) | 80TQ (1) | 40TQ (1) |
| | 40TQ (1) | 20TQ (1) | 10TQ (1) |

Note: Shown in ( ) are the baud rate prescaler divide-by-N values and this table is not available in the classical CAN function.

Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means CFDCnNCFG.NBRP = CFDCnDCFG.DBRP.

Additionally, if transceiver delay compensation is used, do not program the CFDCnDCFG.DBRP bit to be greater than 1, as 1 means divide by 2.

### 32.4.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 32.17 shows the procedure for setting the CAN clock and the baud rate for each channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.

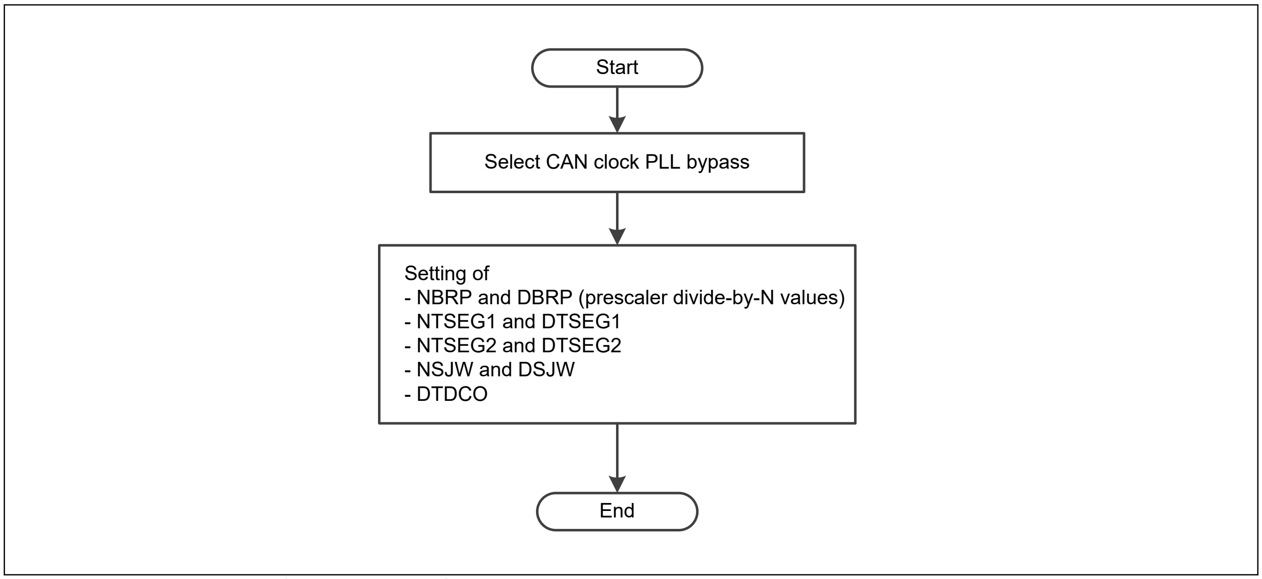


Figure 32.17 Procedure for setting the CAN bit timing and baud rate

### 32.4.1.5 Transmitter Delay Compensation

This chapter is not valid for classical CAN.

When a high baud rate is used such as 5 to 8 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CANFD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CANFD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDCnFDSTS.TDCR) as shown in Figure 32.18.

The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

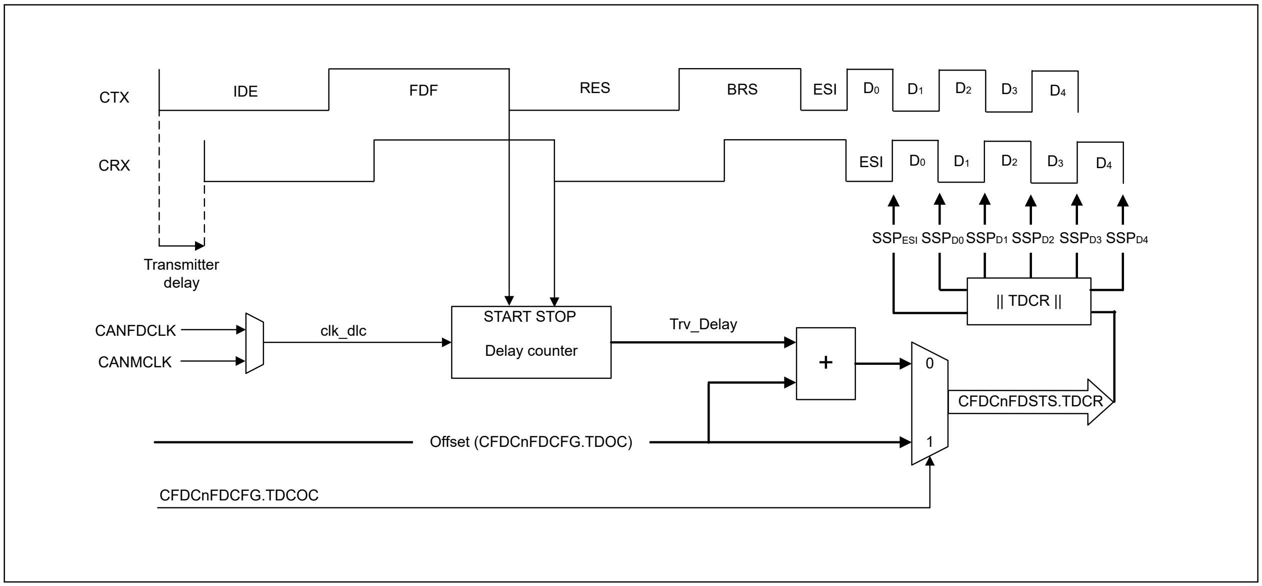


**Figure 32.18 Transmitter delay compensation**

The measured Trv_Delay is based on the number of clk_dlc clock cycles.The delay is counted up by one for each started clock until the dominant value is seen on CAN_RX. Figure 32.19 shows the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

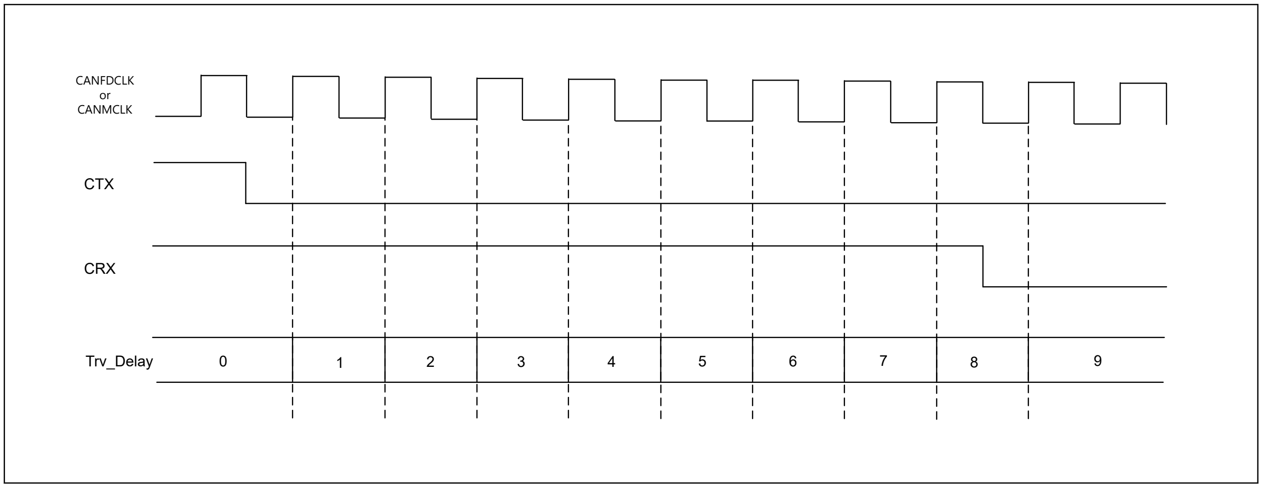


**Figure 32.19 Trv_Delay measurement example**

The SSP is calculated by taking the result from CFDCnFDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 32.20 shows the positioning of the secondary sample point. When CFDCnFDCFG.TDCOC is equal to 0, the SSP is equal to the Trv_Delay (measured delay) + CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (SyncSegmentdata + TSEG1data) to position the SSP to a theoretical location of the sample point.

If the CFDCnFDCFG.TDCOC is equal to 1, the SSP is defined by CFDCnFDCFG.TDCO. If CFDCnDCFG.DBRP is greater than 0, the value is also rounded down to the nearest integer number of time quanta.

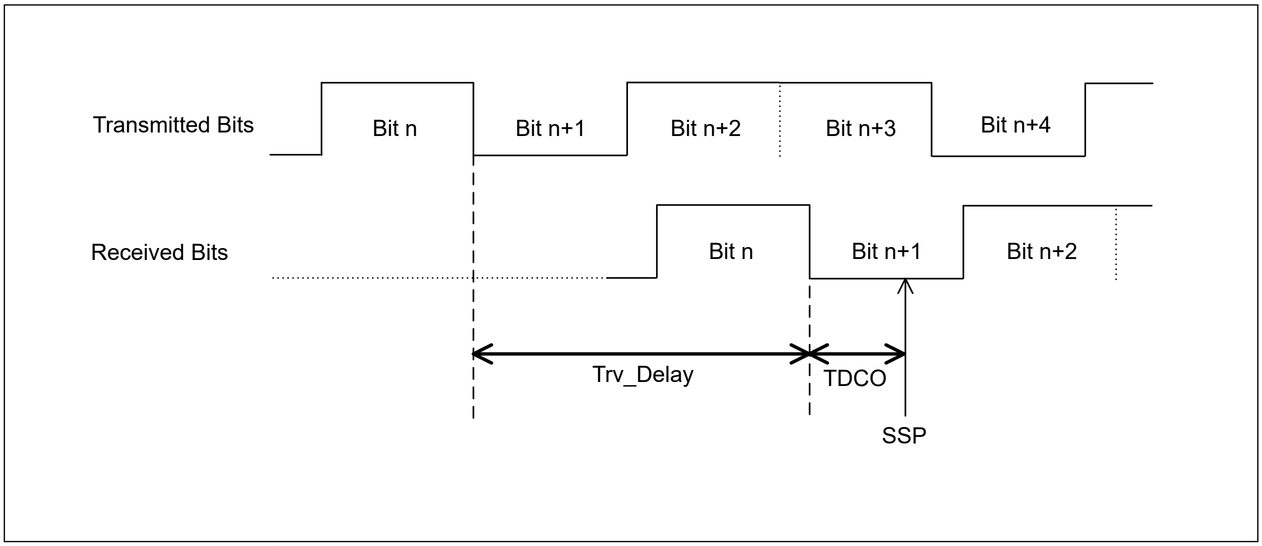


**Figure 32.20 Position of the secondary sample point**

The maximum delay (Trv_Delay + TDCO) which can be compensated by the CANFD module is (6 data bits - 2clk_dlc).

The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.

If different values are used for CFDCnNCFG.NBRP and CFDCnDCFG.DBRP, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 32.21.

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means CFDCnNCFG.NBRP = CFDCnDCFG.DBRP.

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.

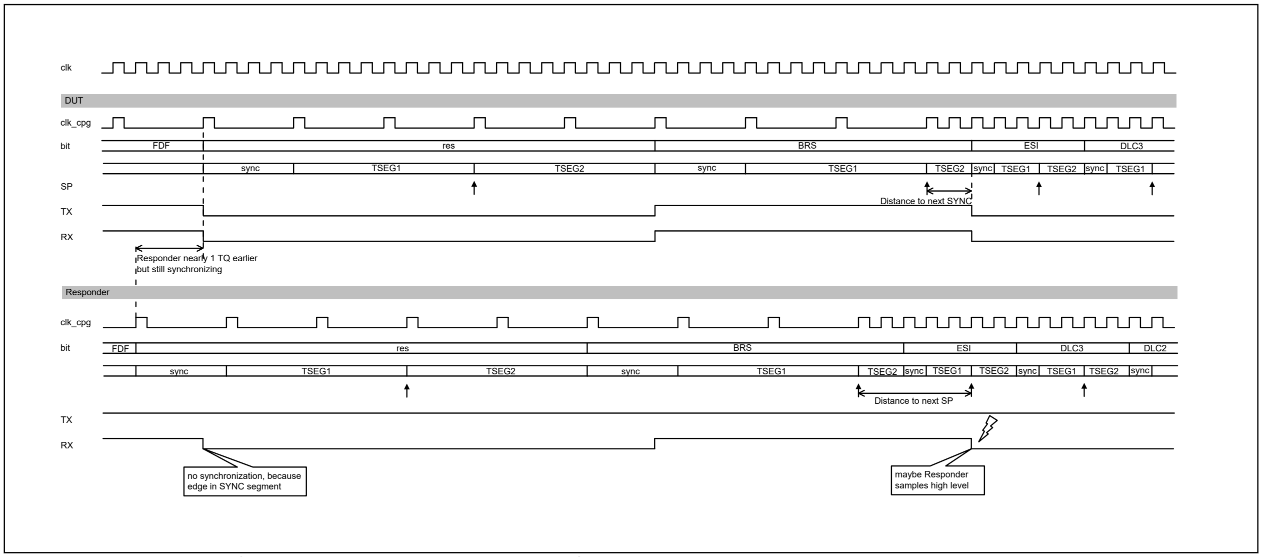


**Figure 32.21 Loss of synchronization between two CAN nodes**

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly (CFDCnFDCFG.TDCE = 1, CFDCnFDCFG.TDCOC = 0).

Figure 32.22 shows the read flow to get the measured transmitter delay compensation result.

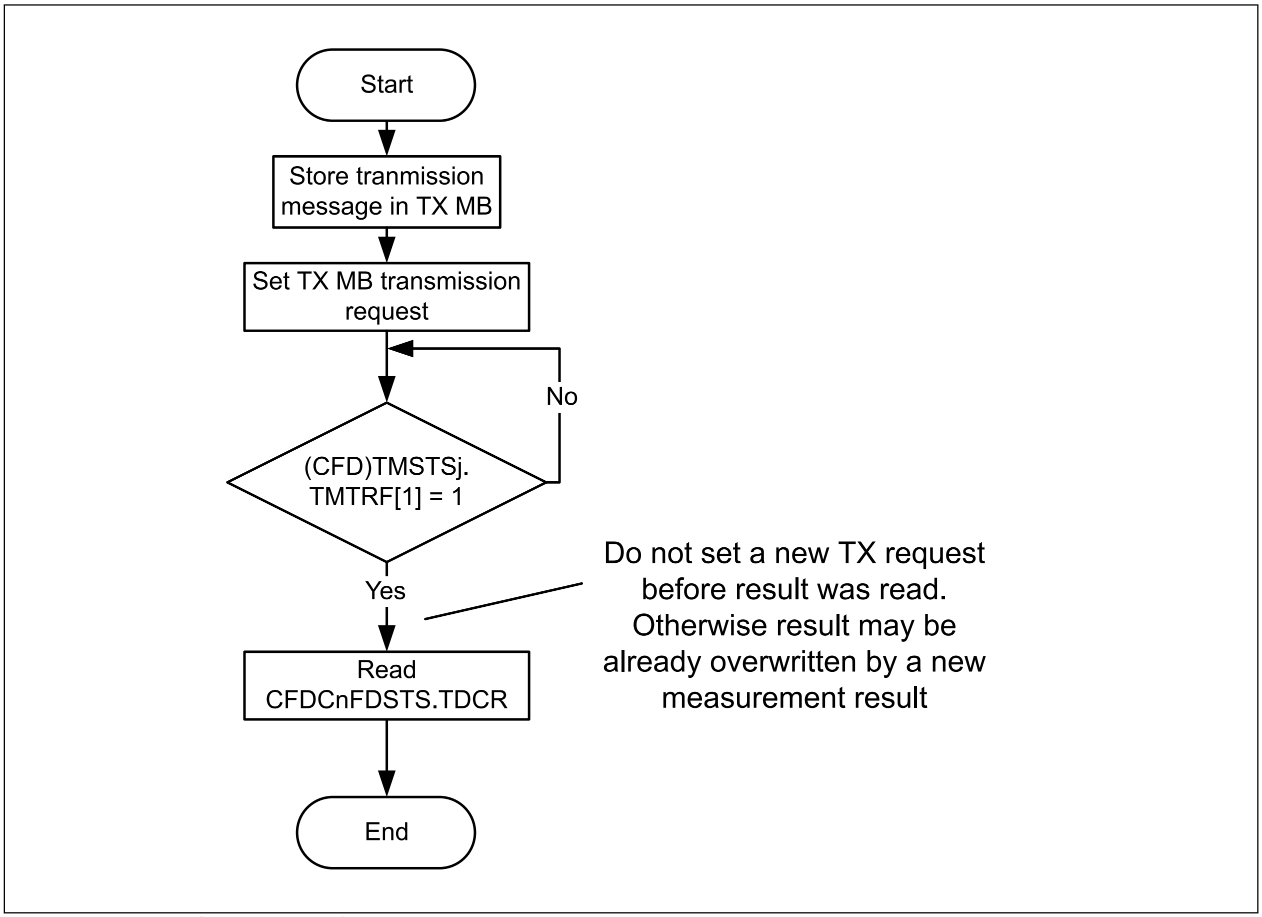

Start
Store tranmission message in TX MB
Set TX MB transmission request
No
(CFD)TMSTSj. TMTRF[1] = 1
Yes
Read CFDCnFDSTS.TDCR
End
Do not set a new TX request before result was read. Otherwise result may be already overwritten by a new measurement result

**Figure 32.22 TDC result read flow**

## 32.4.2 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module enters Global Sleep mode automatically.

To enable configuration of the CANFD module, you must exit Sleep mode by clearing the Global Sleep Request bit CFDGCTR.GSLPR to 0.

After a hardware reset, the module starts RAM initialization, the CFDGSTS.GRAMINIT bit in the Global Status Register is set automatically to indicate that the CANFD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

RAM initialization is necessary to avoid setting of false ECC error flag after HW reset the random data presented in the RAM.

Do not access registers of CANFD in either read or write until RAM initialization is complete and the CFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, each required CAN channel must be configured such as CAN bit timing. For this configuration, all required CAN channels must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 32.23 shows the configuration procedure. For details about each step, seesection 32.5. Acceptance Filtering Function using Global Acceptance Filter List (AFL), section 32.6. FIFO Buffers and Normal Message Buffer Configuration, section 32.7. Interrupts and DMA and section 32.4.1.3. Baud Rate.

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting CFDGRSTC.SRST.

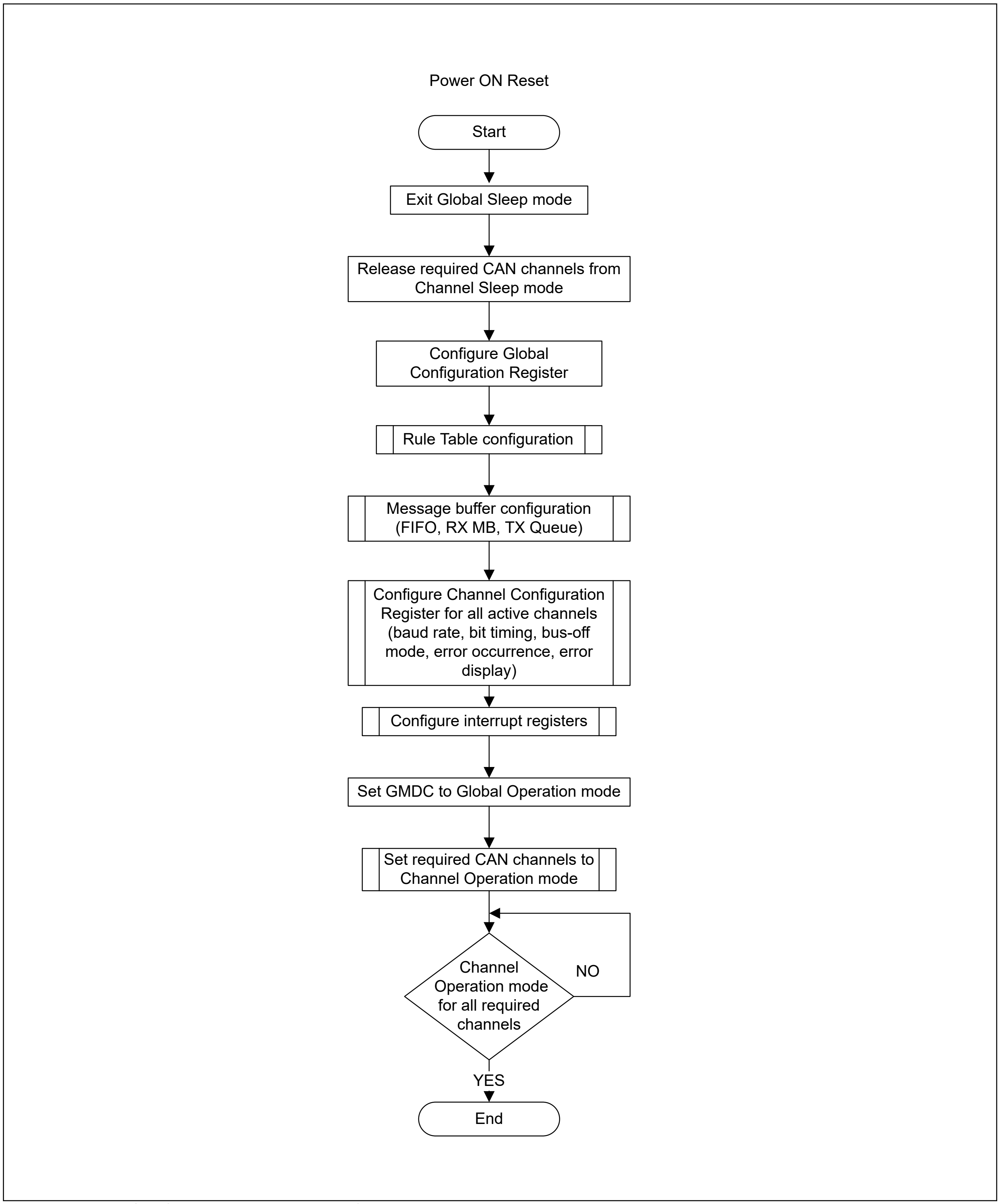

Power ON Reset
Start
Exit Global Sleep mode
Release required CAN channels from Channel Sleep mode
Configure Global Configuration Register
Rule Table configuration
Message buffer configuration (FIFO, RX MB, TX Queue)
Configure Channel Configuration Register for all active channels (baud rate, bit timing, bus-off mode, error occurrence, error display)
Configure interrupt registers
Set GMDC to Global Operation mode
Set required CAN channels to Channel Operation mode
Channel Operation mode for all required channels
NO
YES
End

**Figure 32.23 Configuration procedure after a hardware reset**

## 32.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

### 32.5.1 Overview

The CANFD module can handle message acceptance filtering for all channels with a global Acceptance Filter List (called AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN Identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the CFDGCFG.CMPOC bit[*1]
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

Note 1. This feature is not available in the classical CAN function.

The 2-channel CANFD module allows a maximum of 128 AFL entries across all channels with a maximum of 64 AFL entries per channel.

During acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (CFDGCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0x0 and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN Bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 0x00 in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0x0, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the CFDGCFG.DRE bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception or gateway function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 8 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than eight target destinations is not allowed. If more destinations are programmed, then the internal timing might lead to a race condition that prevents the storage of received messages in the message RAM.. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload Bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

If CFDGCFG.CMPOC = 0, the message is completely rejected and is stored in the target destination. When CFDGCFG.CMPOC = 0 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCn.RFPLS or CFDCFCCn.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is not set to 1, respectively.

When CFDGCFG.CMPOC = 1, the received data bytes greater than CFDRMNB.RMPLS is rejected. When CFDGCFG.CMPOC = 1 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCn.RFPLS or CFDCFCCn.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is set to 1, respectively.

Depending on the CFDGCFG.DRE bit, the original received DLC or the DLC value configured at the AFL entry is stored.

Regardless of the CFDGCFG.CMPOC bit setting, CFDGERFL.CMPOF is set to 1 if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with CFDGERFL.DEF or CFDGERFL.CMPOF*1.

Note 1. This bit is not available in the classical CAN function.

### 32.5.2 Allocation of AFL Entries to each CAN Channel

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see Figure 32.24).

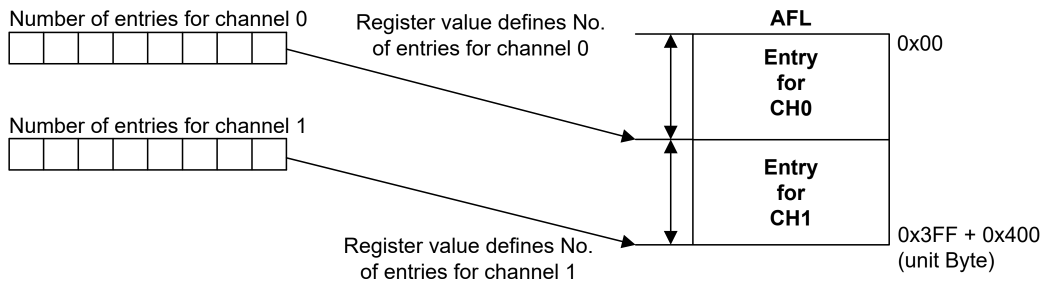

Number of entries for channel 0
Register value defines No. of entries for channel 0
AFL
0x00
Entry for CH0
Number of entries for channel 1
Entry for CH1
Register value defines No. of entries for channel 1
0x3FF + 0x400 (unit Byte)

**Figure 32.24 Configuration of AFL for each channel**

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries for one channel is 64. The total number of entries for all channels should not exceed the maximum limit of (n + 1) × 64.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CANFD module does not flag errors related to the configuration of the AFL.

### 32.5.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).

- IDE bit:
  Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).

- RTR bit:

Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).

- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.

- Mask for Identifier bits (29 bits):
  Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see Figure 32.25.

- Mask for IDE bit:
  If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.

- Mask for RTR bit:
  If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.

- Pointer information (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.

- Information label (2 bits):
  This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.

- DLC value for automatic DLC filtering:
  If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed.

If the DLC value in this AFL entry is configured to 0, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

Note: A message received on channel A can be routed to Common FIFO buffer of another channel. If this Common FIFO buffer is configured in Gateway mode, then the message stored in this Common FIFO Buffer is transmitted on that channel because Common FIFO buffer is associated with channel.

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.

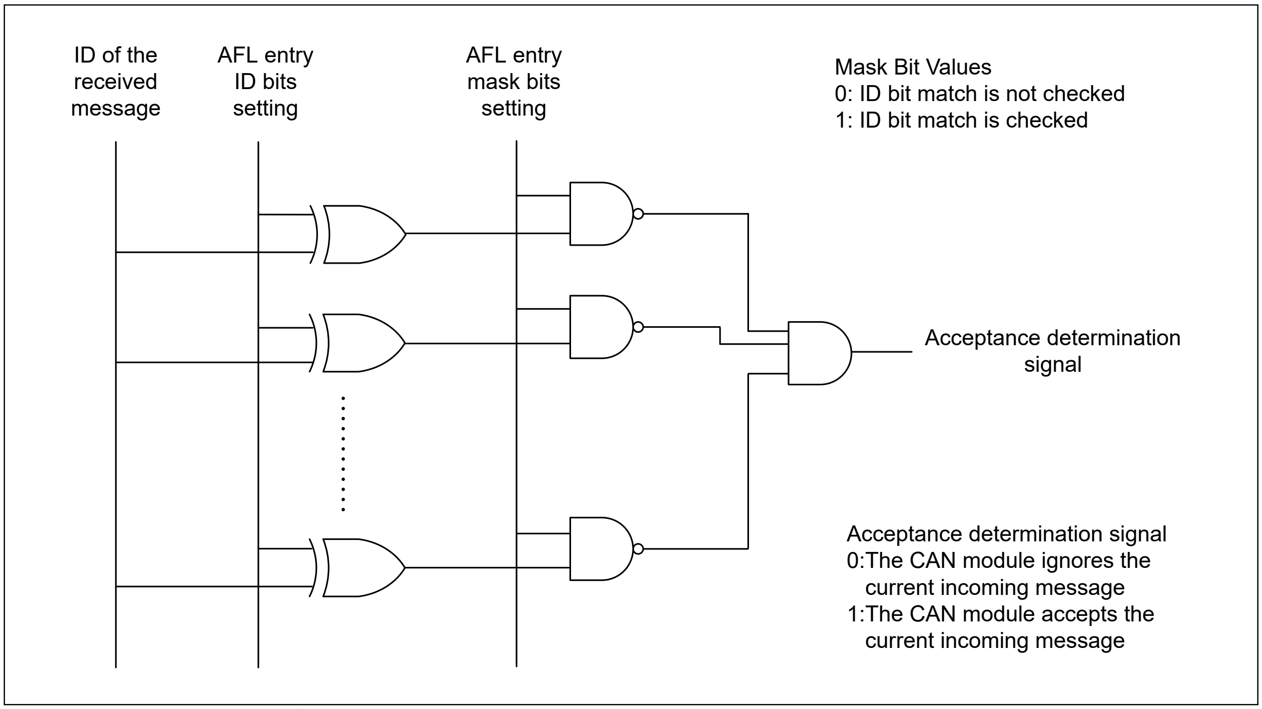

ID of the received message
AFL entry ID bits setting
AFL entry mask bits setting
Mask Bit Values
0: ID bit match is not checked
1: ID bit match is checked
Acceptance determination signal
Acceptance determination signal
0:The CAN module ignores the current incoming message
1:The CAN module accepts the current incoming message

**Figure 32.25 Acceptance function**

### 32.5.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry.

16 sets of these registers form a group of AFL entries. Each group can be accessed through a page mechanism. For the CANFD module, 2-channel version, 8 of these pages exist to allow access to the whole AFL range. The AFL should only be configured in CH_RESET or CH_HALT mode. Pages are linked to the AFL entries in the following way:

| | |
|---|---|
| Page 0 | Entry 0 — 15 |
| Page 1 | Entry 16 — 31 |
| Page 2 | Entry 32 — 47 |
| Page 3 | Entry 48 — 63 |
| Page 4 | Entry 64 — 79 |
| Page 5 | Entry 80 — 95 |
| Page 6 | Entry 96 — 111 |
| Page 7 | Entry 112 — 127 |

The selection of the AFL access page is done using the Global Acceptance Filter List Entry Control Register (CFDGAFLECTR) (Figure 32.26). This register has the following fields:

- 7 bits to select the AFL page number
- 1 bit to enable or disable the AFL data access to prevent unwanted write access to the AFL.

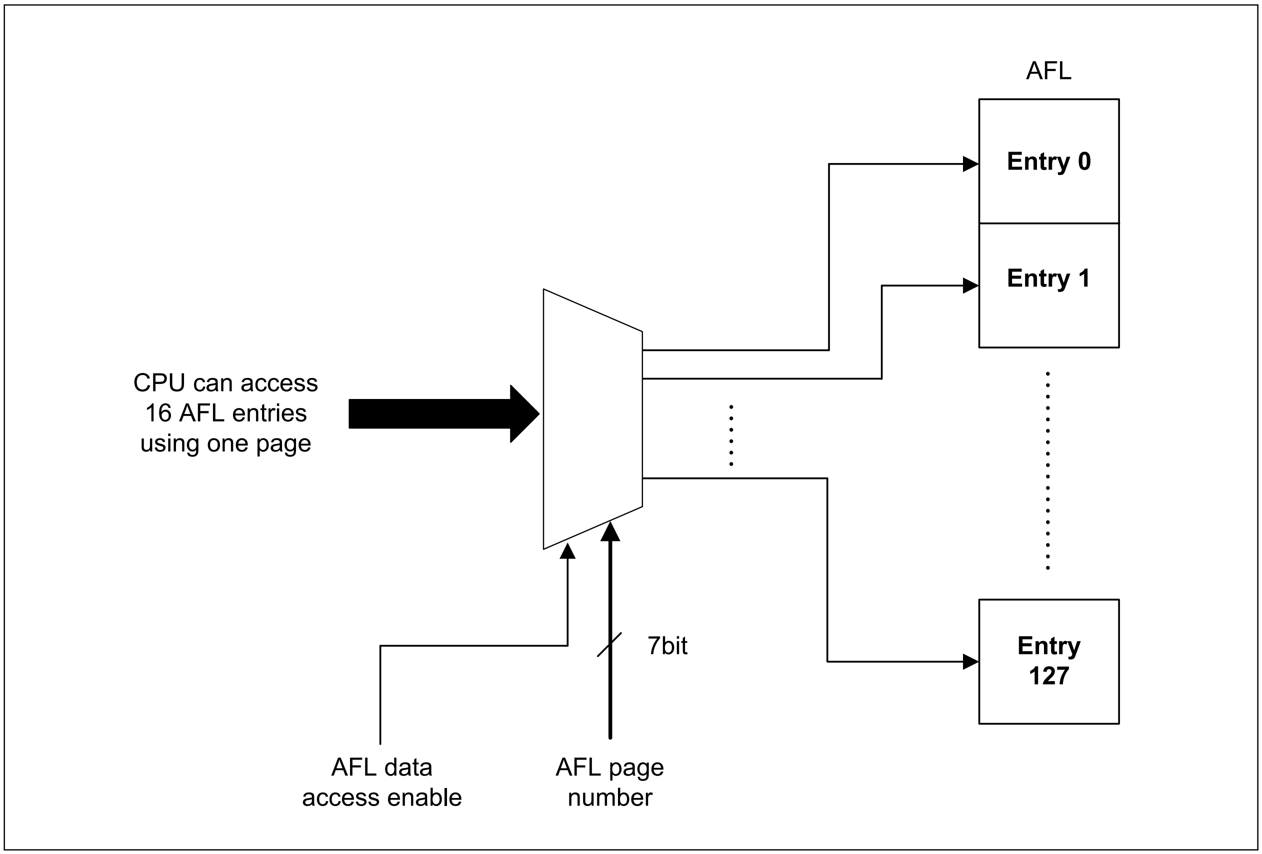


**Figure 32.26 AFL page access**

Application software should not write numbers higher than 0x5F for the AFL page number.

Follow the configuration shown in Figure 32.27 to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

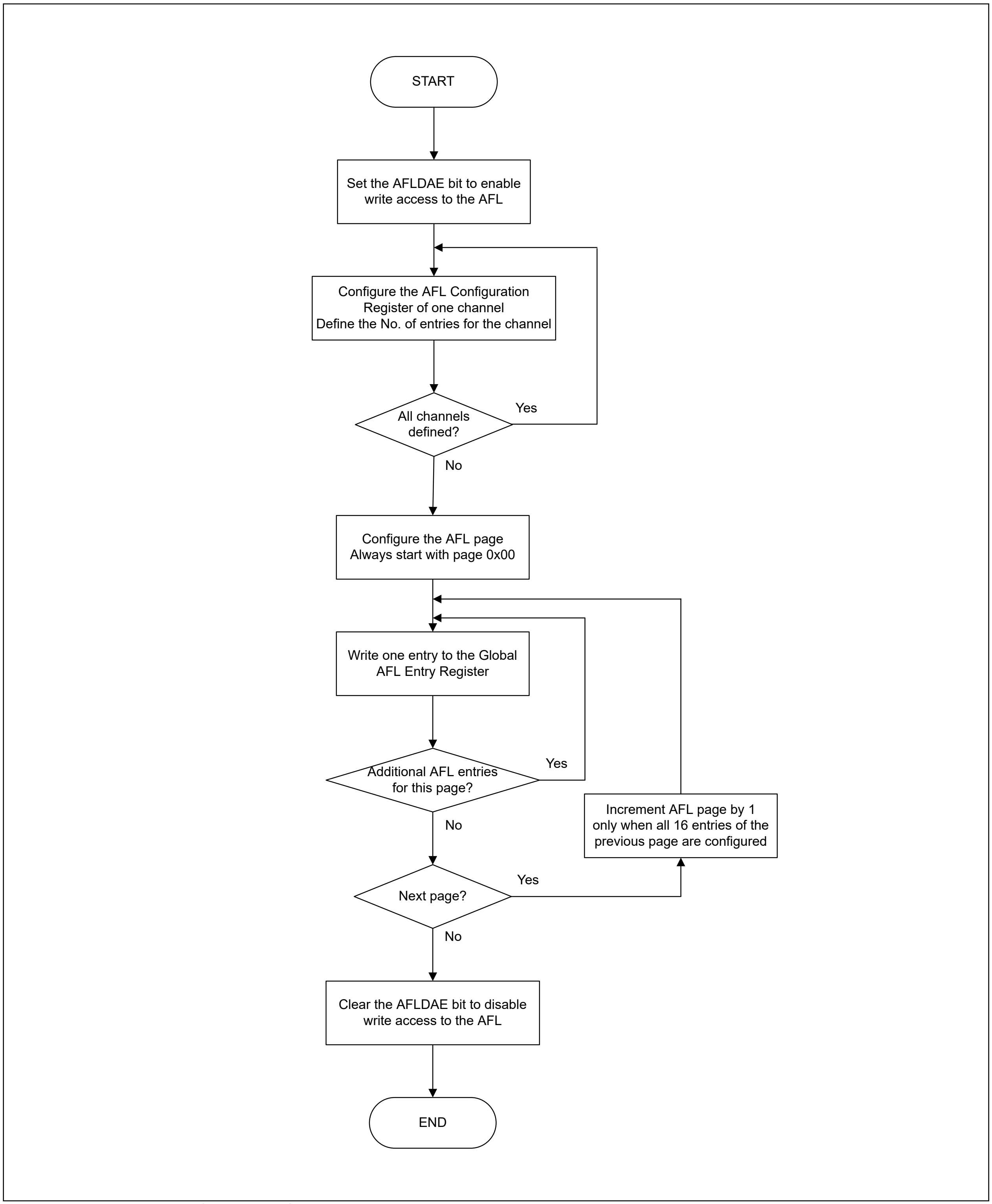

START
Set the AFLDAE bit to enable write access to the AFL
Configure the AFL Configuration Register of one channel
Define the No. of entries for the channel
All channels defined?
Yes
No
Configure the AFL page
Always start with page 0x00
Write one entry to the Global AFL Entry Register
Additional AFL entries for this page?
Yes
No
Increment AFL page by 1 only when all 16 entries of the previous page are configured
Next page?
Yes
No
Clear the AFLDAE bit to disable write access to the AFL
END

**Figure 32.27 AFL configuration flow**

### 32.5.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in Loopback test mode (Self-test mode 0 or Self-test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in Loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If Mirror mode and Loopback test mode are configured at the same time, the Loopback test mode behavior applies.

Table 32.17 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

**Table 32.17 Behavior of acceptance filter based on the loopback configuration setting in AFL entry**

| Mirror Mode Enable (MME Configuration bit) | Loopback in test mode (Self-test mode 0 or Self-test mode 1) | Channel mode | Loopback Configuration bit in AFL entry | AFL entry |
|---|---|---|---|---|
| 0 | 0 | Receiver | 0 | Valid |
| | | | 1 | Invalid |
| | | Transmitter | 0 | Invalid |
| | | | 1 | Invalid |
| | 1 | Receiver | 0 | Valid |
| | | | 1 | Invalid |
| | | Transmitter | 0 | Valid |
| | | | 1 | Valid |
| 1 | 0 | Receiver | 0 | Valid |
| | | | 1 | Invalid |
| | | Transmitter | 0 | Invalid |
| | | | 1 | Valid |
| | 1 | Receiver | 0 | Valid |
| | | | 1 | Invalid |
| | | Transmitter | 0 | Valid |
| | | | 1 | Valid |

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

### 32.5.6 IDE Masking

When the GAFLIDEM bit is 0 in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
  - CFDGAFLID [x] = 0xC0553A20 → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 0x220 / ID[28:0] = 0x00553A20
  - CFDGAFLMn = 0x0000FFFF → IDEM = 0. RTRM = 0, IDM[10:0] = 0x7FF / IDM[28:0] = 0x0000FFFF
- The comparison result for the four different received IDs with AFL entry x is described as follows:
  - If a frame with IDE = 0 and ID = 0x220 is received, this is considered as a match
  - If a frame with IDE = 0 and ID = 0x320 is received, this is not a match

- If a frame with IDE = 1 and ID = 0x1FFF3A20 is received, this is considered as a match
- If a frame with IDE = 1 and ID = 0x08803220 is received, this is not a match.

## 32.6 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CANFD module. The message buffers are mapped as shown in Figure 32.28.

The RX message buffers can be accessed with the RX Message Buffer Registers.

The RX FIFO buffers and the common FIFO buffers configured in RX mode, TX mode, or GW mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in GW mode or RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.

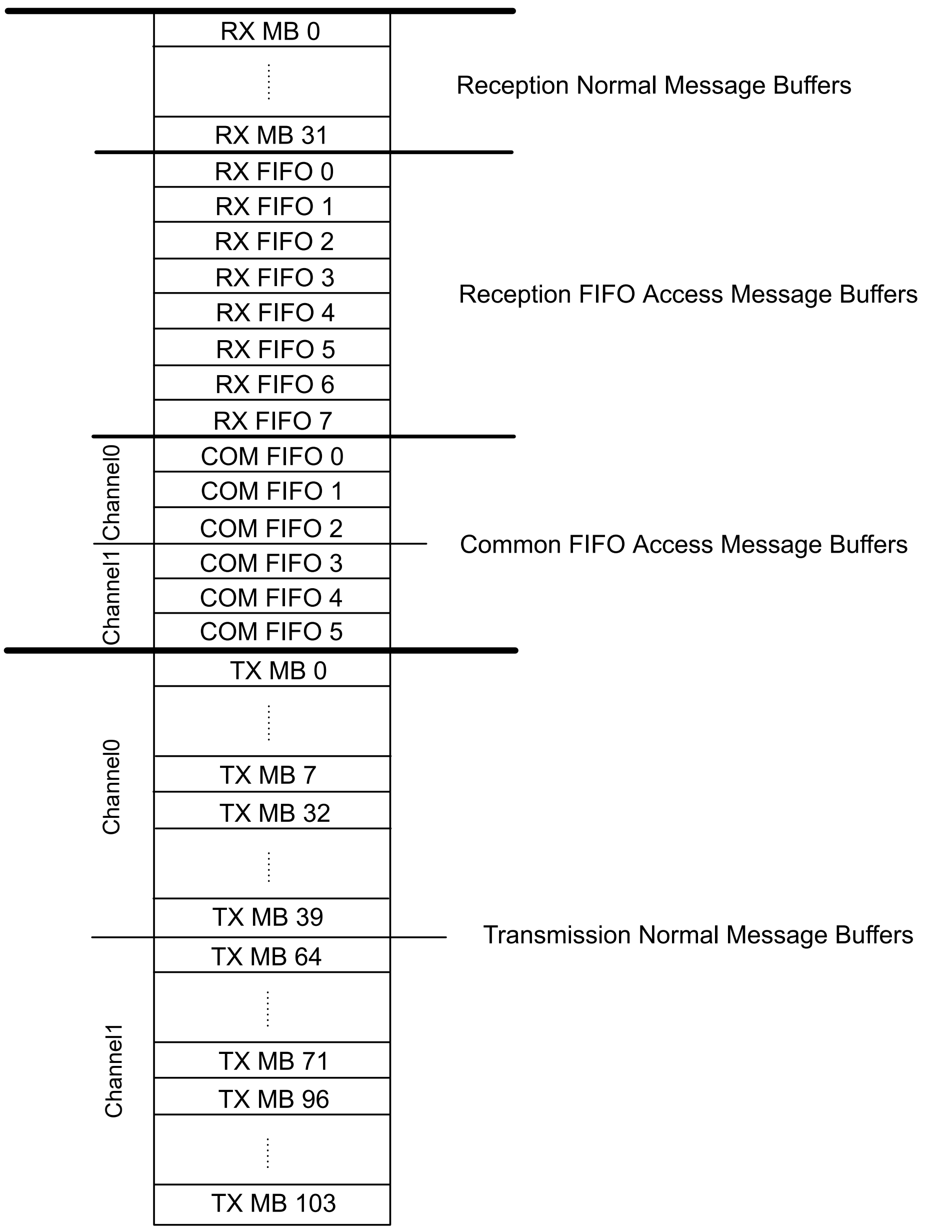


**Figure 32.28 Message buffer configuration**

## 32.6.1 Normal RX Message Buffers

In CANFD module, the frames received by various channels can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

### 32.6.1.1 Normal RX Message Buffer Configuration

In CANFD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 0x00 (no normal RX MB)
- Maximum value = (16 × No. of CAN channels) = 16 × 2 = 32
  = 0x20 (32 flat RX MBs for 2 channels)

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

Note: There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS bit. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

Note: RMPLS and CMPOC bit is not available in the classical CAN function, so, these feature is not valid for classical CAN.

### 32.6.2 FIFO Buffers

The CANFD module provides a fixed number of FIFO buffers to support storage of frames for reception, transmission and gateway functions for various CAN channels.

The number of reception-only FIFO buffers is fixed to 8. However, 3 common FIFO buffers per channel can be configured to store messages for transmission, reception, or gateway function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO or GW FIFO.

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the CFDGCFG.CMPOC bit (message rejecting or data payload cut).

#### 32.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 8 RX FIFO buffers + 6 common FIFO buffers = 14 FIFO buffers for 2 channels and message overwrite mechanism.

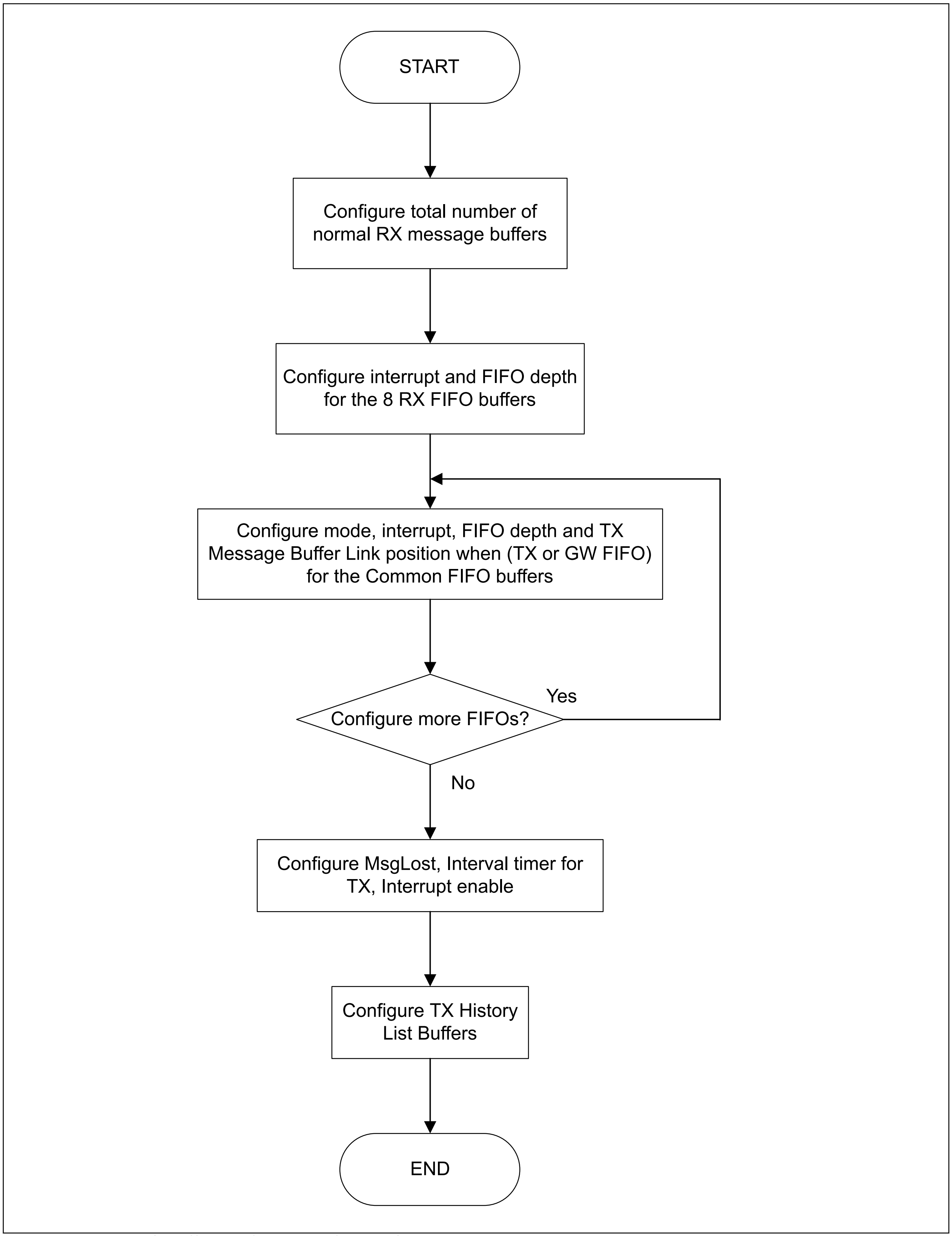

START
Configure total number of normal RX message buffers
Configure interrupt and FIFO depth for the 8 RX FIFO buffers
Configure mode, interrupt, FIFO depth and TX Message Buffer Link position when (TX or GW FIFO) for the Common FIFO buffers
Configure more FIFOs?
Yes
No
Configure MsgLost, Interval timer for TX, Interrupt enable
Configure TX History List Buffers
END

**Figure 32.29 FIFO buffer configuration flow in CANFD module**

As shown in Figure 32.29, the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 8 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size.

For the common FIFO buffers, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position.

### (1) FIFO mode configuration of Common FIFO buffers

The mode of the common FIFO buffers can be configured by writing to the CFDCFCCn.CFM[1:0] bits in the Common FIFO Configuration/Control Registers. The possible modes of configuration for Common FIFO buffers are:

- 0b RX mode (default mode after hardware reset)
- 1b TX mode
- 10b GW mode
- 11b Reserved (Do not write this value to the register bits)

Messages can only be read from the RX FIFO buffers and the Common FIFO buffers configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffers configured in TX mode. These messages are transmitted on the appropriate CAN channel.

Messages can only be read from the Common FIFO buffers configured in GW mode. However, the CPU read access has no impact on the read or write pointers. The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CANFD module.

After a hardware reset, all the Common FIFO buffers are configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffers in the required modes.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting the CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded, and CFDCFSTSn.CFMLT bit is set to 1.

- When CFDCFCCEn.CFMOWM = 1:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.
  The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.
  The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message has been overwritten with the received message.
  In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive FIFO buffer full, the transmitting message is lost and retransmision for the message is not performed. The read point is then moves to the next message automatically.
  Do not write to this bit when the CFDCFCCn.CFE bit is 1.

### (2) FIFO TX message buffer link configuration

When the common FIFO is configured as TX or GW FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan.

The link to a normal TX message buffer must be unique, for example the same TX message buffer cannot be shared between 2 or more common FIFO buffers.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCCn.CFTML[4:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 0x00: TX Message Buffer 32
- 0x01: TX Message Buffer 33
- ⋮
- 0x07: TX Message Buffer 39

### (3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCn.RFDC[2:0] bits and CFDCFCCn.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/ Control Registers. The 8 available options for depth configuration are:

- 0x000: 0 Message (FIFO buffer cannot be enabled)
- 0x001: 4 Messages
- 0x010: 8 Messages
- 0x011: 16 Messages
- 0x100: 32 Messages
- 0x101: 48 Messages
- 0x110: 64 Messages
- 0x111: 128 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to (n + 1) × 256 messages. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCCn.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.

If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

### (4) FIFO payload size configuration

The data size of each FIFO buffer can be configured by writing to the CFDRFCCn.RFPLS[2:0] bits and CFDCFCCn.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/ Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

The RAM allocation for RX message buffers along with FIFO buffers is limited to (n + 1) × 256 messages with 64 data bytes. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: This feature is not available in the classical CAN function.

### (5) FIFO interrupt configuration

The Interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCn.RFIM and CFDCFCCn.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0:
  - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCn.RFIGCV/ CFDCFCCn.CFIGCV value
  - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
  - GW FIFO mode
  - Frame RX: Interrupt generated when message counter increments and reaches the interrupt threshold value
  - Frame TX: Interrupt generated when the last message is transmitted successfully from FIFO.
- 1:
  - RX FIFO mode: Interrupt generated at the end of storage of every received message
  - TX FIFO mode: Interrupt generated for every successfully transmitted message
  - GW FIFO mode
  - Frame RX: Interrupt generated when message is stored in the FIFO
  - Frame TX: Interrupt generated when message is successfully transmitted from the FIFO.

If the Interrupt Mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCn.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCCn.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full
- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCn.RFIGCV[2:0] and CFDCFCCn.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see Table 32.18.

**Table 32.18 FIFO interrupt generation counter and FIFO depth configuration (1 of 2)**

| RFDC[2:0] (CFDC[2:0]) | RFIGCV[2:0] (CFIGCV[2:0]) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 111b | 110b | 101b | 100b | 011b | 010b | 001b | 000b |
| 000b | Don't care (FIFO cannot be enabled) | | | | | | | |
| 001b | Allowed | Not allowed | Allowed | Not allowed | Allowed | Not allowed | Allowed | Not allowed |
| 010b | Allowed | | | | | | | |
| 011b | Allowed | | | | | | | |
| 100b | Allowed | | | | | | | |

Table 32.18 FIFO interrupt generation counter and FIFO depth configuration (2 of 2)

| RFDC[2:0] (CFDC[2:0]) | RFIGCV[2:0] (CFIGCV[2:0]) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 111b | 110b | 101b | 100b | 011b | 010b | 001b | 000b |
| 101b | Allowed | | | | | | | |
| 110b | Allowed | | | | | | | |
| 111b | Allowed | | | | | | | |

Common FIFO buffer can set an interrupt output at the completion time of transmitting one frame, or the completion of reception. In addition, Common FIFO and RX FIFO can set an interrupt output, when stored to the setup number (CFDC/RFDC) of FIFO stages.

#### 32.6.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCn.RFIE
- CFDRFCCn.RFFIE.

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Registers:

- CFDCFCCn.CFRXIE
- CFDCFCCn.CFTXIE
- CFDCFCCEn.CFFIE
- CFDCFCCEn.CFOFRXIE
- CFDCFCCEn.CFOFTXIE.

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCn.RFE and CFDCFCCn.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers to allow transmission and reception of messages.

When CFDCFCCEn.CFBME = 1, it becomes FIFO buffering mode, send data is stored in the Common FIFO, and transmission is stopped. Transmission starts when CFDCFCCEn.CFBME = 0.

Do not write 1 from 0 for this bit when the CFDCFCCn.CFE bit is 1.

## 32.7 Interrupts and DMA

### 32.7.1 Interrupts

The CAN-FD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent from this enable bit.

The channel transmission interrupt has an additional status flag register. The status bits are set when the corresponding interrupt enables are set.

The status flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CAN-FD module can be classified into two groups, global interrupts and channel interrupts:

- Global interrupts:
  The CAN-FD module can generate 2 global interrupts:
  - Global interrupt for successful reception into the 8 RX FIFO buffers
  - Global error interrupt.
- Channel interrupts:
  Each channel of the CAN-FD module can generate 3 channel interrupts:

1. Channel transmission
   - Transmission completion from channel
   - Transmission abort from channel
   - Transmission from TX Queue for a channel
   - Channel THL interrupt
   - Successful transmission from a Common FIFO in TX or GW mode for a channel.
2. Channel error interrupt
3. Successful reception in a Common FIFO in RX or GW mode for a channel or successful routing in a TXQ.

The interrupts are cleared when the corresponding flag bits are cleared or the Interrupt enable bits are cleared.

Table 32.19 gives an overview of interrupt sources for the different interrupt outputs. The interrupt outputs are active-high.

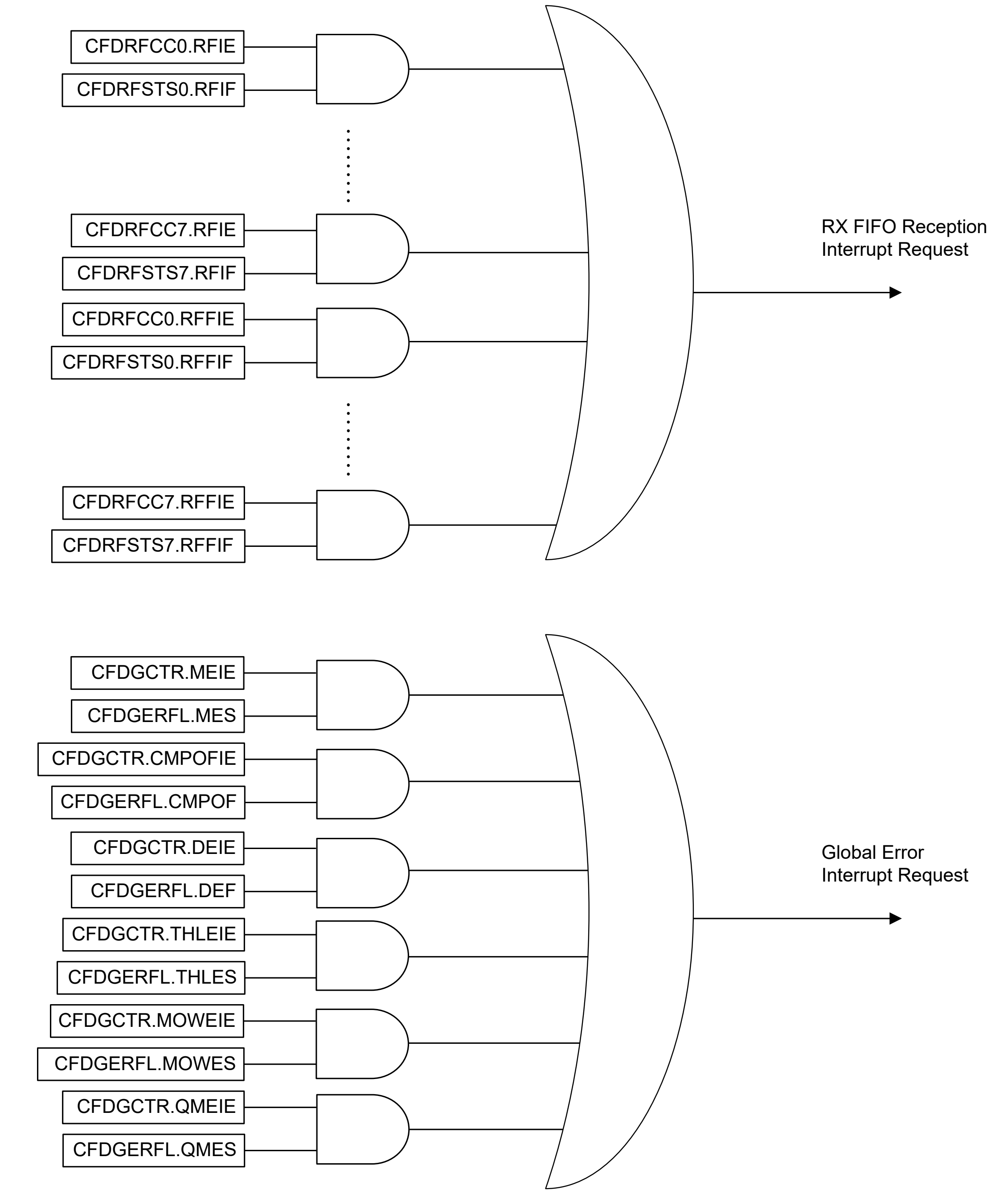
**Table 32.19 Interrupt source overview (1 of 2)**

| Parameter | Interrupt | Interrupt source | Interrupt clearing |
|---|---|---|---|
| Global interrupts | Successful reception into at least one RX FIFO | Interrupt flag of corresponding RX FIFO for which interrupt is enabled | Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled |
| | FIFO full into at least one RX FIFO | FIFO Full Interrupt flag of corresponding RX FIFO for which interrupt is enabled | Clear the FIFO Full Interrupt flags of corresponding RX FIFO buffer for which interrupt is enabled |
| | Global error | Interrupt source is any of the following:<br>● DLC Error flag<br>● Message Lost Status bit<br>● Message Overwrite Status bit<br>● TXQ Message Lost Status bit<br>● TXQ Message Overwrite Status bit<br>● TX History Entry Lost Status bit<br>● CAN-FD Message Payload Overflow flag*1 | Clear all of the following:<br>● DLC Error flag<br>● Message Lost flags in all of the FIFO Status Registers<br>● Message Overwrite flags in all of the Common FIFO Status Registers<br>● Message Lost flags in all of the TXQ Status Registers<br>● Message Overwrite flags in all of the TXQ Status Registers<br>● TX History List Entry Lost flag<br>● CAN-FD Message Payload Overflow flag.*1 |

Table 32.19 Interrupt source overview (2 of 2)

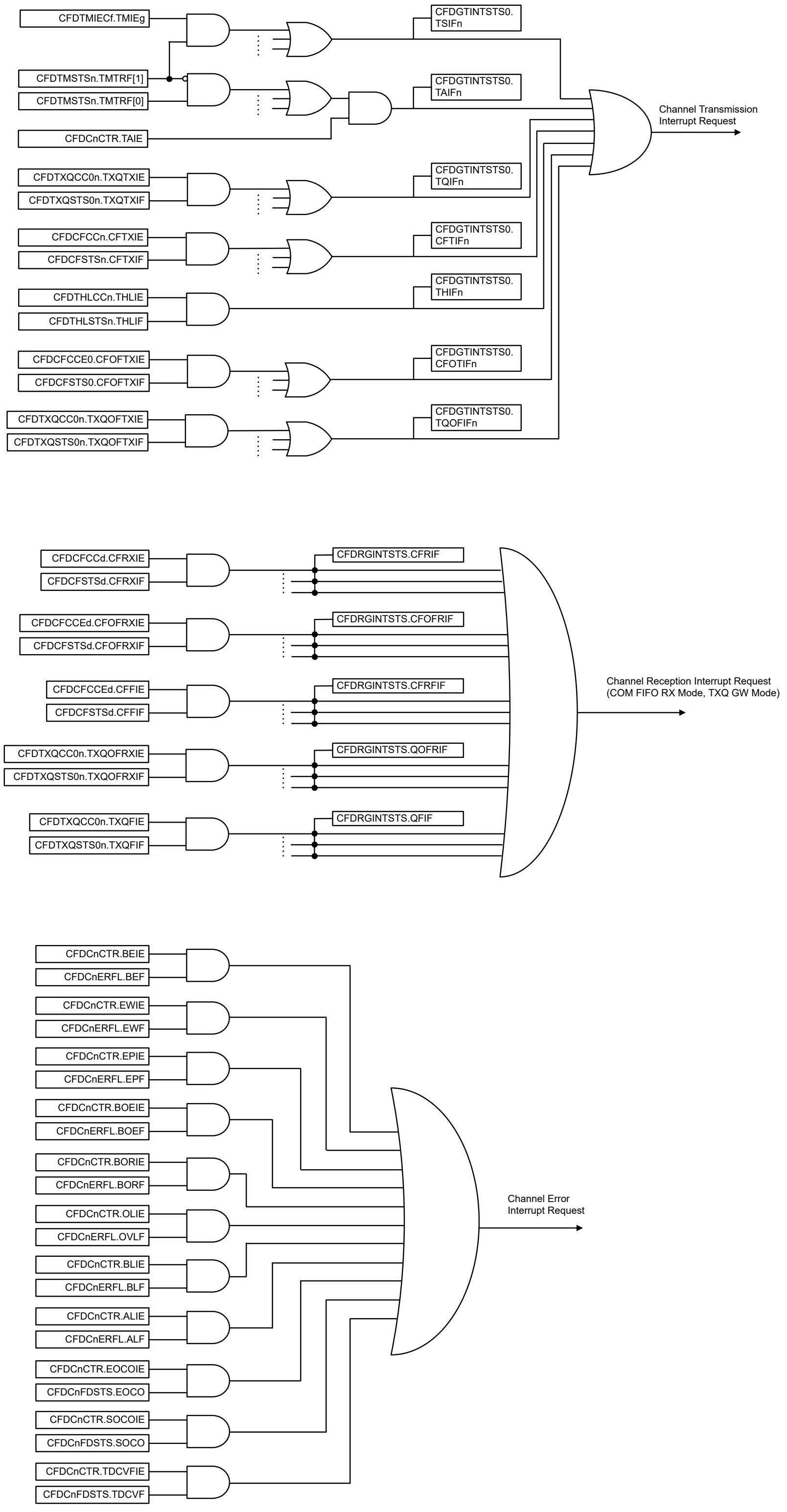
| Parameter | Interrupt | Interrupt source | Interrupt clearing |
|---|---|---|---|
| Channel transmission interrupts | Channel n successful transmission | Any channel related TX MB Successful flag when interrupt is enabled.<br>Separate interrupts are provided for Common FIFO buffers and TX Queue.<br>Note: These interrupts are only set for TX message buffers that do not belong to an enabled TX Queue and are not pointing to a Common FIFO. | Clear all channel related TX MB Result Status bits for which the interrupt is enabled |
| | Channel n abort | Any channel related TX MB Abort flag when interrupt is enabled<br>Separate interrupts are provided for Common FIFO buffers and TX Queue<br>Note: These interrupts are only set for TX message buffers that do not belong to an enabled TX Queue and are not pointing to a Common FIFO. | Clear all channel related TX MB Result Status bits for which the interrupt is enabled globally |
| | Channel n transmission from TX Queue | Related Channel TX Queue Interrupt flag | Clear related Channel TX Queue Interrupt flag |
| | Channel n THL Interrupt | Channel n THL Interrupt Status flag | Clear the relevant THL Interrupt Status flag |
| | Channel n COM FIFO TX Interrupt | Interrupt flag for Common FIFOs in TX or GW mode belonging to the related channel | Clear the interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel |
| | Channel n COM FIFO One Frame TX Interrupt | One Frame Transmission Interrupt flag for Common FIFOs belonging to the related channel | Clear the One Frame Transmission Interrupt flags of Common FIFOs belonging to the related channel |
| | Channel n TXQ One Frame TX Interrupt | One Frame Transmission Interrupt flag for TXQs belonging to the related channel | Clear the One Frame Transmission Interrupt flags of TXQs belonging to the related channel |
| Channel COM RX FIFO Interrupt | Channel n COM FIFO RX Interrupt | Interrupt Flag for Common FIFOs in RX or GW mode belonging to the related channel | Clear the interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel |
| | Channel n COM FIFO One Frame RX Interrupt | One Frame Reception Interrupt flag for Common FIFOs belonging to the related channel | Clear the One Frame Reception Interrupt flags of Common FIFOs belonging to the related channel |
| | Channel n COM FIFO Full Interrupt | FIFO Full Interrupt flag for Common FIFOs in RX or GW mode belonging to the related channel | Clear the FIFO Full Interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel |
| | Channel n TXQ One Frame Routing Interrupt | One Frame Routing Interrupt flag for TXQs in GW mode belonging to the related channel | Clear the One Frame Routing Interrupt flags of TXQs in GW mode belonging to the related channel |
| | Channel n TXQ Full Interrupt | TXQ Full Interrupt flag for TXQs in GW mode belonging to the related channel | Clear the FIFO Full Interrupt flags of TXQs in GW mode belonging to the related channel |
| Channel Error Interrupt | Channel n Error | Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register | Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register |

Note 1. This feature is not available in the classical CAN function.



Note: CFDGCTR.CMPOFIE and CFDGERFL.CMPOF bits are not available in the classical CAN function. For this reason, these signals and the AMD circuit to which these signals are connected are not in the classical CAN function.

**Figure 32.30 Global interrupt block diagram**



Note: CFDCnCTR.TDCVFIE and CFDCnFDSTS.TDCVF bits are not available in the classical CAN function. For this reason, these signals and the AMD circuit to which these signals are connected are not in the classical CAN function.

**Figure 32.31 Channel interrupt block diagram**

### 32.7.2 DMA Transfer

The CAN-FD module has message buffers that can be associated with a DMA channel:

- Reception DMA
  - 8 RX FIFO message buffers
  - 8 Common FIFO message buffers
- Transmission DMA
  - 16 TXQ message buffers (TXQ0, TXQ3)
  - 8 Common FIFO message buffers.

Figure 32.32 shows the potential DMA channels.

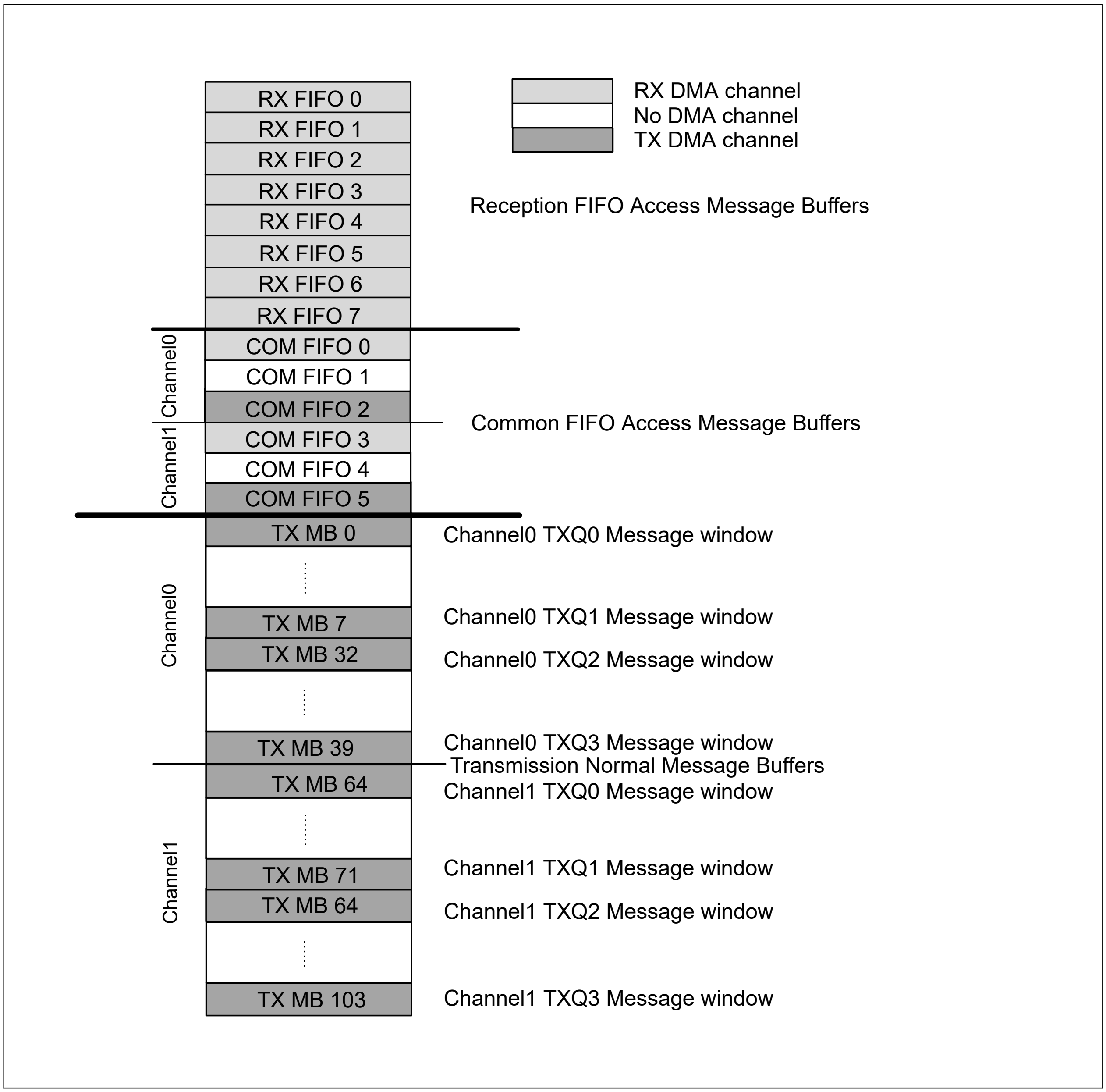

RX DMA channel
No DMA channel
TX DMA channel
RX FIFO 0
RX FIFO 1
RX FIFO 2
RX FIFO 3
RX FIFO 4
RX FIFO 5
RX FIFO 6
RX FIFO 7
Reception FIFO Access Message Buffers
Channel0
COM FIFO 0
COM FIFO 1
COM FIFO 2
Channel1
COM FIFO 3
COM FIFO 4
COM FIFO 5
Common FIFO Access Message Buffers
Channel0
TX MB 0
Channel0 TXQ0 Message window
TX MB 7
Channel0 TXQ1 Message window
TX MB 32
Channel0 TXQ2 Message window
TX MB 39
Channel0 TXQ3 Message window
Transmission Normal Message Buffers
Channel1
TX MB 64
Channel1 TXQ0 Message window
TX MB 71
Channel1 TXQ1 Message window
TX MB 64
Channel1 TXQ2 Message window
TX MB 103
Channel1 TXQ3 Message window

**Figure 32.32 Message buffer connectable to a DMA channel**

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1 and the belonging FIFO is not empty.

Reception FIFO Interrupt should be disabled for this particular FIFO (CFDRFCCn.RFIE or CFDCFCCn.CFRXIE)

Use the regular start address for the DMA access window address and add 0x8000 to the regular start address for the debugger access window. See Figure 32.33.

**Table 32.20 DMA channel access window address**

| b = Message buffer component index | MBCP | Register | P | Regular start address n = [0, 1] |
|---|---|---|---|---|
| [0…7] | FRMBCPb[i] | RFIDE | x | 0x6000 + b*0x0080 |
| | | RFPTRE | x | 0x6004 + b*0x0080 |
| | | RFFDSTSE | x | 0x6008 + b*0x0080 |
| | | RFDFpE | [0…15] | 0x600C + b*0x0004 + b*0x0080 |
| [0…2] | CFMBCPb[i] | CFIDE | x | 0x6400 + b*0x0080 + n*0x180 |
| | | CFPTRE | x | 0x6404 + b*0x0080 + n*0x180 |
| | | CFFDCSTSE | x | 0x6408 + b*0x0080 + n*0x180 |
| | | CFDFpE | [0…15] | 0x640C + b*0x0004 + b*0x0080 + n*0x180 |

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

Note: This feature is not available for classical CAN function because CFDRFCCn.RFPLS and CFDCFCCn.CFPLS are not in classical CAN.

Do not write to the FIFO and TXQ control registers when DMA is enabled. The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time. Figure 32.33 shows a configuration flow for an initial setup.

When CFDCDTTCT.TQ0DMAE or CFDCDTTCT.TQ3DMAE or CFDCDTTCT.CFDMAE is set, the messages of the corresponding TXQ or Common FIFO can be handled by a DMA controller.

The following procedure shows when the TXQ or the Common FIFO can be handled by a DMA controller.

1. CPU checks the TXQ or the Common FIFO is not full.
2. When transmit data can be used, the CPU sets this data to Common FIFO or TXQ.
   When using Common FIFO, transmit data is write in CFDCFID, CFDCFPTR, CFDCFFDCSTS and CFDTMBCPb[i] register.
   When using TXQ, transmit data is write in CFDTMID, CFDTMPTR, CFDTMFDCTR and CFDTMDFp register.
3. For Common FIFO, the common FIFO pointer is incremented automatically when DMA controller writes the last data payload byte configured by CFDCFCCn.CFPLS.
   For TXQ, if the data of 64 data payload is written, a TXQ pointer increases automatically. When payload data is less than 64 bytes, dummy data must be written in and 64 data payload size must be done.

   Note: Only 32-bit write-access can be possible on the DMA message handling.

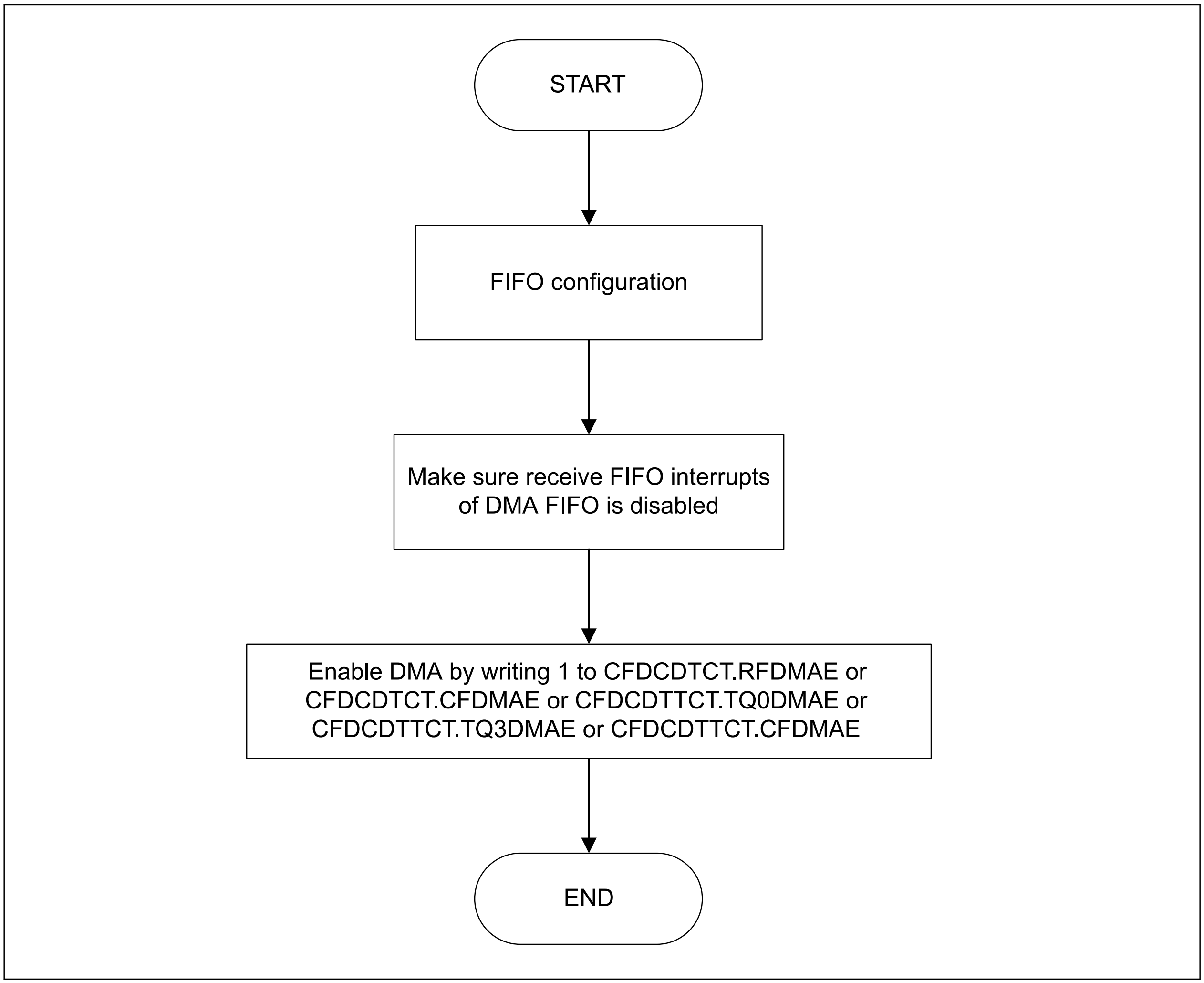

START
FIFO configuration
Make sure receive FIFO interrupts of DMA FIFO is disabled
Enable DMA by writing 1 to CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE or CFDCDTTCT.TQ0DMAE or CFDCDTTCT.TQ3DMAE or CFDCDTTCT.CFDMAE
END

**Figure 32.33 DMA enable flow**

To disable a DMA transfer request, you must disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then the transfer must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS bit. See Figure 32.34 for the DMA disable flow. When the DMA is disabled, consideration should be made for the remaining or new incoming messages to this particular reception FIFO.

When the FIFO is not disabled, reception to the FIFO continues.

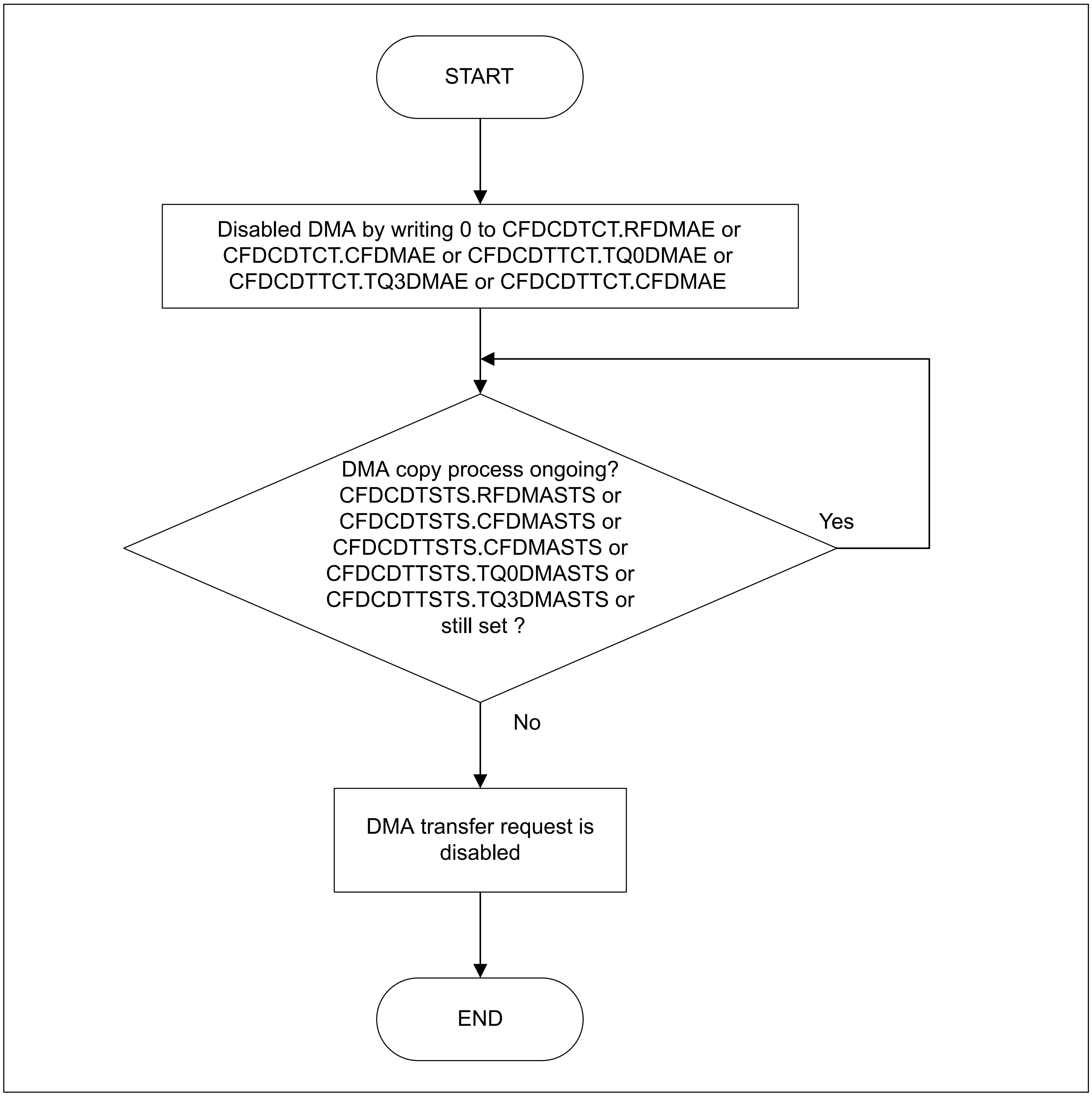

START
Disabled DMA by writing 0 to CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE or CFDCDTTCT.TQ0DMAE or CFDCDTTCT.TQ3DMAE or CFDCDTTCT.CFDMAE
DMA copy process ongoing? CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS or CFDCDTTSTS.CFDMASTS or CFDCDTTSTS.TQ0DMASTS or CFDCDTTSTS.TQ3DMASTS or still set ?
Yes
No
DMA transfer request is disabled
END

**Figure 32.34 DMA disable flow**

## 32.8 Reception and Transmission

### 32.8.1 Reception

In the CANFD module, CAN messages received on any of the channels are stored in RX message buffers, RX FIFO buffers, or Common FIFO buffers configured in RX mode or GW mode depending on the Acceptance Filter List entries.

- Up to 32 RX message buffers can be configured
- 8 RX FIFO buffers available
- Up to 6 Common FIFO buffers can be configured in RX mode or GW mode
- Up to 4 TX Queue can be configured in GW mode.

#### 32.8.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX message buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX message buffer.

If a new message is stored into a RX message buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX message buffer. If such a loss of messages is not acceptable, then RX FIFO should be used for storing related messages.

Note: Interrupts are not provided for the RX message buffers in the CANFD module and therefore, the RX Message Buffer New Data Registers should be accessed periodically to check if a new message has been stored in the RX message buffers.

Note: Unused data bytes are filled with 0x00 depending on the DLC value.

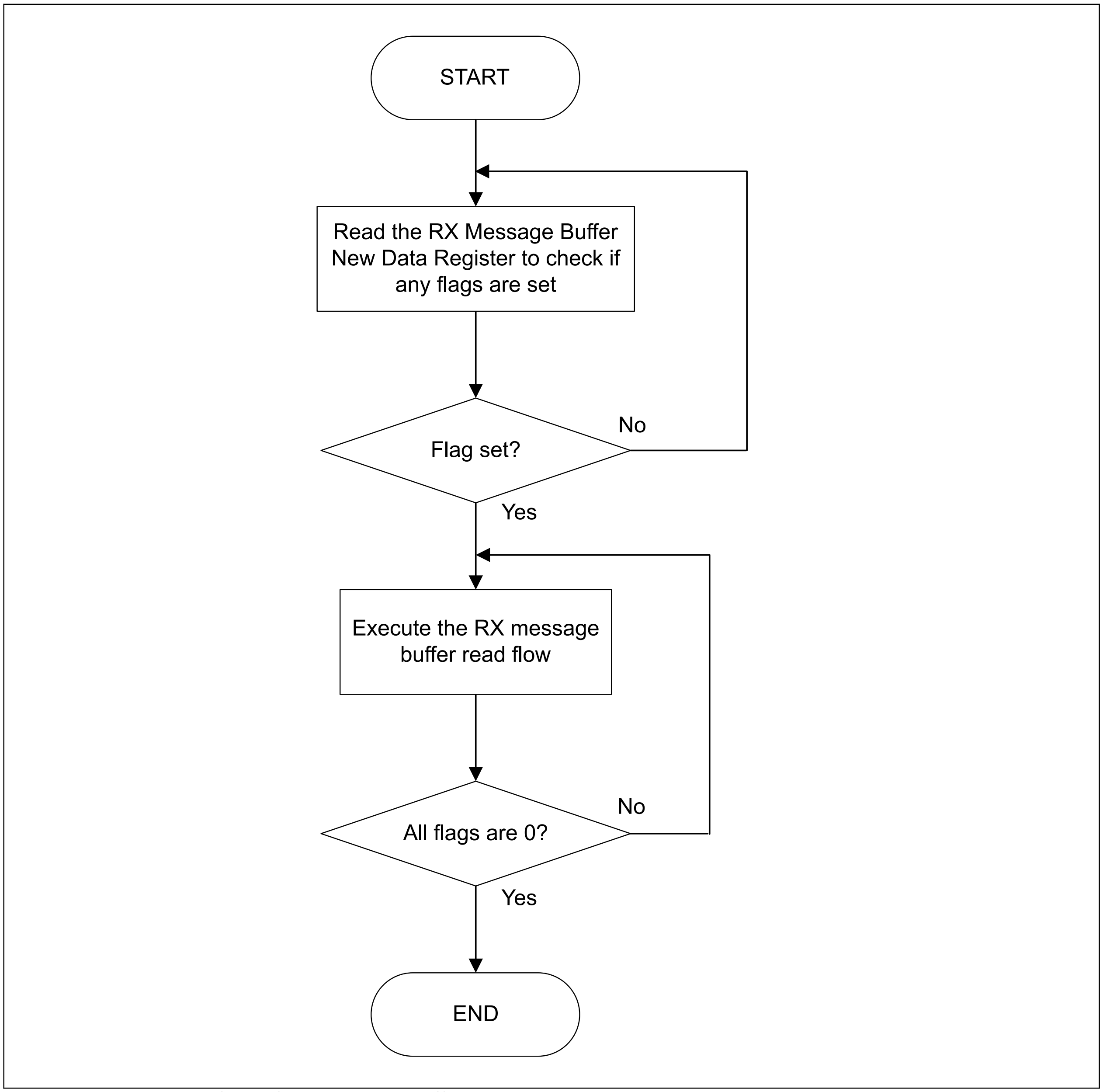

START
Read the RX Message Buffer New Data Register to check if any flags are set
Flag set?
No
Yes
Execute the RX message buffer read flow
All flags are 0?
No
Yes
END

**Figure 32.35 Access flow of RX message buffer**

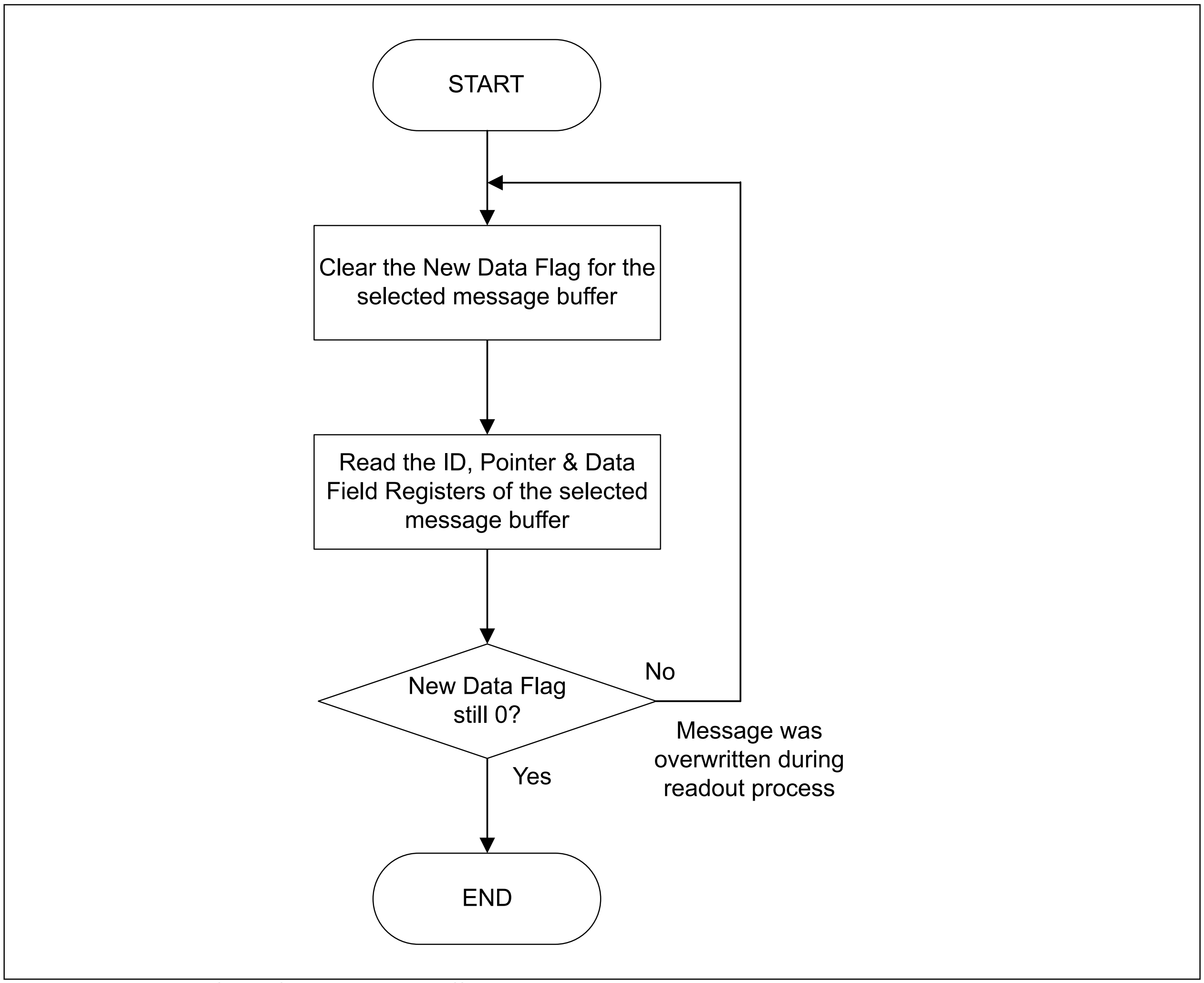

START
Clear the New Data Flag for the selected message buffer
Read the ID, Pointer & Data Field Registers of the selected message buffer
New Data Flag still 0?
No
Message was overwritten during readout process
Yes
END

**Figure 32.36 Read flow of RX message buffer**

### 32.8.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffers configured in RX or GW mode should be configured based on system requirements.

The CFDGAFLP1n.GAFLFDP[31:0] field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffers configured in RX mode or GW mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Registers.

Depending on the configuration of the FIFO buffers, an interrupt might also be generated.

The message can be read from the corresponding FIFO Access registers.

Note: Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value 0xFF is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write 0xFF to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to an overrun condition.

In GW mode, when a transmit/receive FIFO buffer is receiving a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and the CFDCFSTSn.CFMLT bit is set to 1.

- When CFDCFCCEn.CFMOWM = 1:
  When writing data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.
  The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.
  The CFDCFSTSn.CFMOW bit is set to 1, which notifies that the oldest message has been overwritten with the received message.
  In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive FIFO buffer full, the transmitting message is lost and retransmission for the message is not performed. The read point moves to the next message automatically.

Do not modify this bit when the CFDCFCCn.CFE bit is 1.

Common FIFO can set the interrupt when CAN frame reception is completed.

Common FIFO can set the interrupt when FIFO is in full status in RX mode or GW mode.

Note: The message lost can be set only in RX or GW mode by CAN, and the flag is not set when the CPU is overloading the FIFO buffers.

Note: When CFDGAFLP0n.GAFLSRD i (i = 0 to 2) is set and the CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0 to 7) is also set, a receiving frame is stored in the target TXQ as send data by routing.

The RX FIFO buffers and the Common FIFO buffers configured in RX or GW mode can be disabled at any time by clearing the CFDRFCCn.RFE or CFDCFCCn.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

When the CFDRFCCn.RFE or CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO buffers or Common FIFO buffers configured in RX mode is assigned as a DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write 0xFF to the FIFO Pointer Control Register (CFDCFPCTRn.CFPC or CFDRFPCTRn.RFPC). This can lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, the interrupt flag is not cleared automatically. The interrupt flag should be cleared before disabling the FIFO.

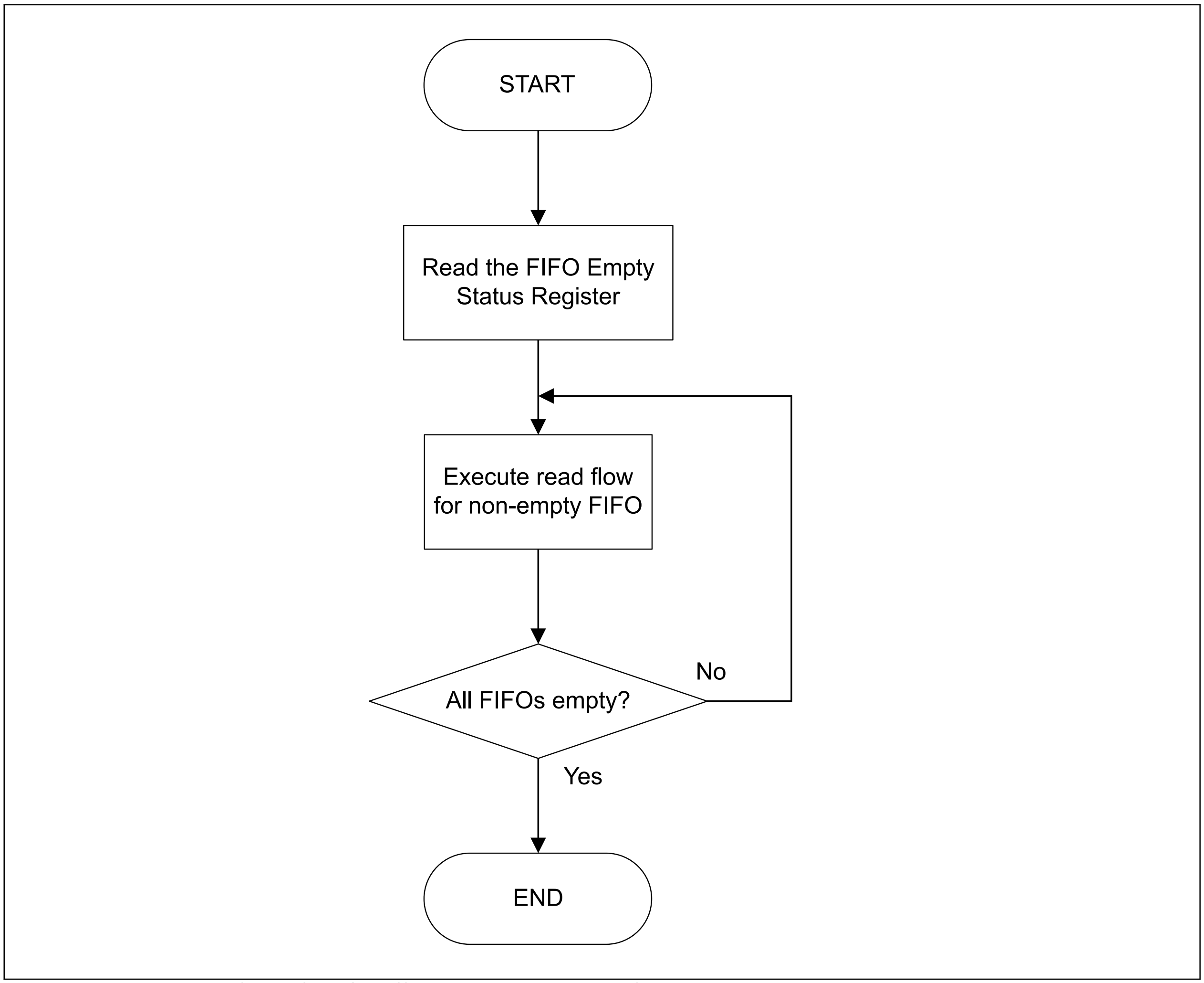

START
Read the FIFO Empty Status Register
Execute read flow for non-empty FIFO
All FIFOs empty?
No
Yes
END

**Figure 32.37 Access flow of FIFO buffer message (example for polling case)**

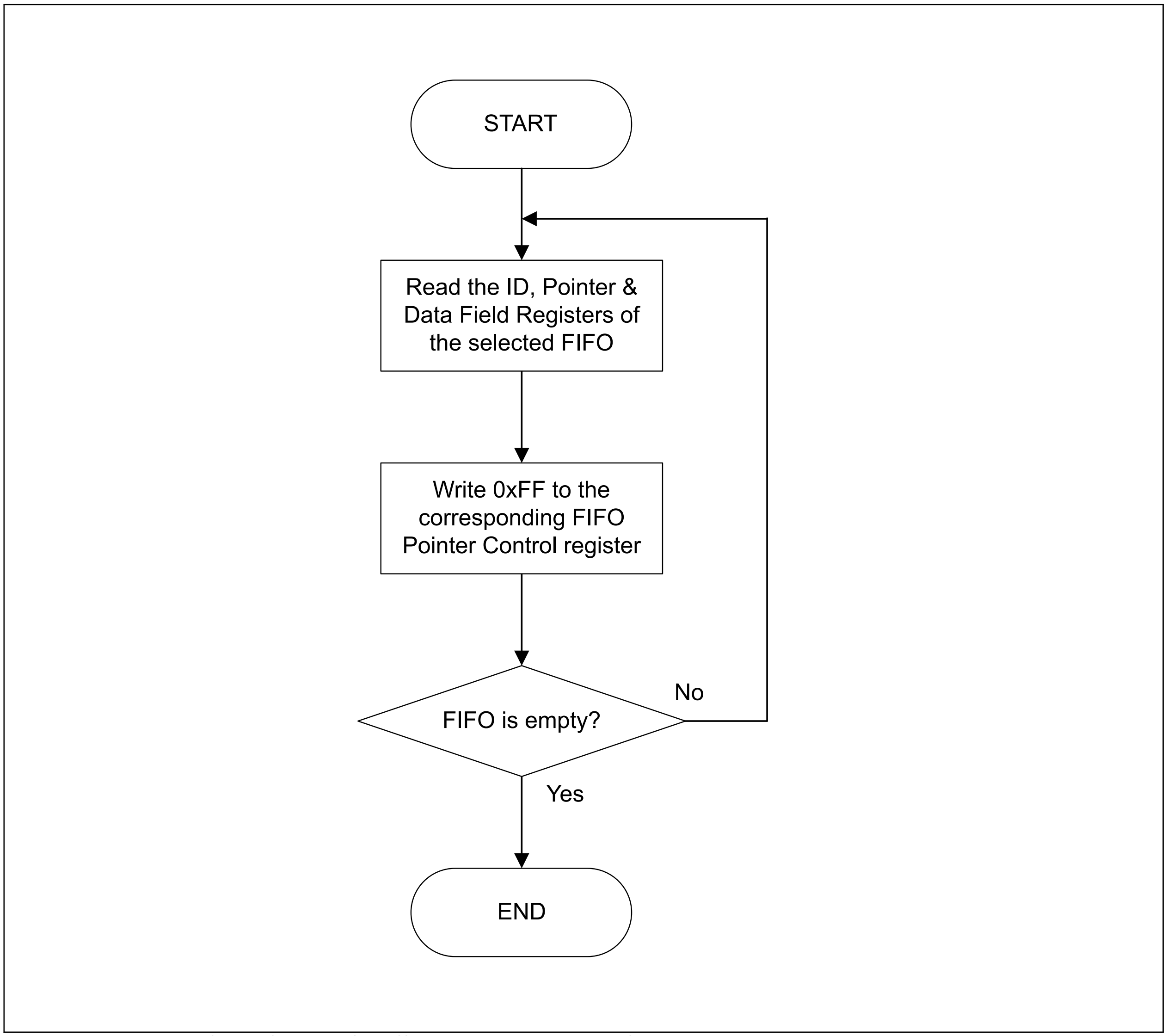

START
Read the ID, Pointer & Data Field Registers of the selected FIFO
Write 0xFF to the corresponding FIFO Pointer Control register
FIFO is empty?
No
Yes
END

**Figure 32.38 Read flow of RX FIFO buffer message (example for polling case)**

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even when an interruption flag is cleared after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the condition, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

### 32.8.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the CFDGFDCFG.TSCCFG[1:0] configuration (at the sample point of start of frame, point in time when the frame is valid, or for CANFD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and data into the target RX message buffer or RX/GW FIFO.

For transmit message, the timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the CFDGCFG.TSSS bit of the Global Configuration Register. If this bit is 0, the peripheral clock is used. If the bit is 1, the selected CAN channel bit time clock is used.

The channel selection is performed with the CFDGCFG.TSBTCS bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset node, for this channel, the timestamp counter is stopped. For other CAN channels, the timestamp counter value is not updated.

If peripheral clock is selected as the timestamp counter clock source, Channel modes do not affect the timestamp counter function.

The source clock for the timestamp counter can be divided by a factor defined by the CFDGCFG.TSP bits (timestamp prescaler) in the Global Configuration Register.

The timestamp counter can be reset to 0x0000 with the CFDGCTR.TSRST bit (timestamp reset).

### 32.8.2 Transmission

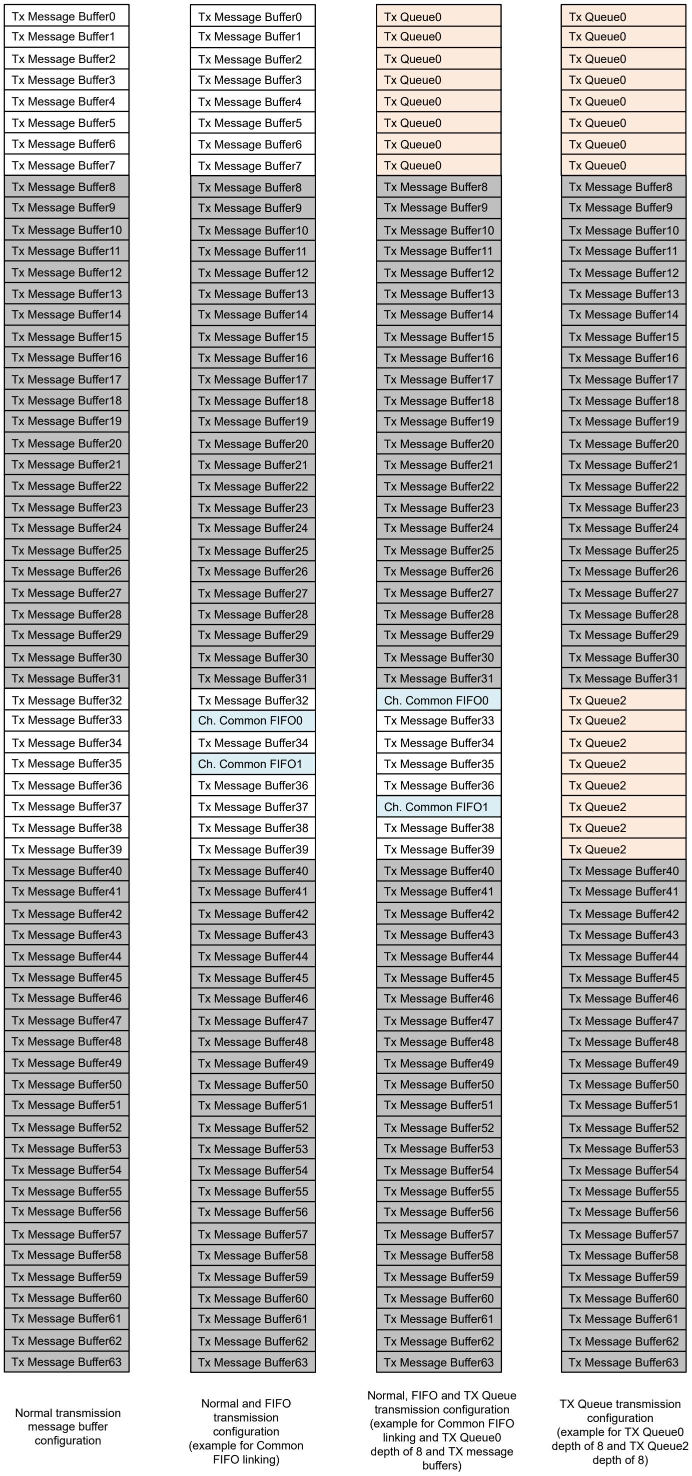
There are several possible transmission configurations for each channel:

- Normal transmission
- FIFO transmission
- Gateway transmission
- TX Queue transmission
  A fixed number of transmission message buffers (16 TX message buffers) are dedicated for each channel. These message buffers are only used for transmission and cannot be configured for reception.
  Additionally transmission from TX Queue or Common FIFO in TX or GW mode can be configured in the following way (see Figure 32.39):
  - TX Queue: Up to eight transmission message buffers for one channel can be grouped to form a TX Queue with a common access window.
    Upper transmission message buffers are used to form the TXQ1 or TXQ3.
    Lower transmission message buffers are used to form the TXQ0 or TXQ2.
    Transmission Control and Status registers of these transmission message buffers should not be used.
    One channel has four TX Queues.
    Each TXQ has an access window.
    - TXQ0 is transmission Message Buffer 0 of each channel.
    - TXQ1 is transmission Message Buffer 7 of each channel.
    - TXQ2 is transmission Message Buffer 32 of each channel.
    - TXQ3 is transmission Message Buffer 39 of each channel.

    When using TXQ1 and TXQ0 simultaneously, the sum of the depths of TXQ1 and TXQ0 should not exceed 8.
    When using TXQ3 and TXQ2 simultaneously, the sum of the depths of TXQ3 and TXQ2 should not exceed 8.
- Common FIFO (TX/GW mode): each Common FIFO in TX or GW mode is linked to a dedicated channel.
  Each channel has a fixed number of three Common FIFO assigned to it. Within the channel, a Common FIFO configured in TX or GW mode, can be freely linked (assigned) between 32 and 39 transmission message buffers (only one FIFO to one transmission message buffer).
  The Common FIFO buffer then replaces the transmission message buffer linked to it.
  Transmission Control and Status registers of these transmission message buffers should not be used.

See Figure 32.28 for information about Common FIFO buffer assignment to related channels.

Note: Common FIFO buffers should not be linked to TX message buffers that are already part of a TX Queue.



| Normal transmission message buffer configuration | Normal and FIFO transmission configuration (example for Common FIFO linking) | Normal, FIFO and TX Queue transmission configuration (example for Common FIFO linking and TX Queue0 depth of 8 and TX message buffers) | TX Queue transmission configuration (example for TX Queue0 depth of 8 and TX Queue2 depth of 8) |
|---|---|---|---|
| Tx Message Buffer0 | Tx Message Buffer0 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer1 | Tx Message Buffer1 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer2 | Tx Message Buffer2 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer3 | Tx Message Buffer3 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer4 | Tx Message Buffer4 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer5 | Tx Message Buffer5 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer6 | Tx Message Buffer6 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer7 | Tx Message Buffer7 | Tx Queue0 | Tx Queue0 |
| Tx Message Buffer8 | Tx Message Buffer8 | Tx Message Buffer8 | Tx Message Buffer8 |
| Tx Message Buffer9 | Tx Message Buffer9 | Tx Message Buffer9 | Tx Message Buffer9 |
| Tx Message Buffer10 | Tx Message Buffer10 | Tx Message Buffer10 | Tx Message Buffer10 |
| Tx Message Buffer11 | Tx Message Buffer11 | Tx Message Buffer11 | Tx Message Buffer11 |
| Tx Message Buffer12 | Tx Message Buffer12 | Tx Message Buffer12 | Tx Message Buffer12 |
| Tx Message Buffer13 | Tx Message Buffer13 | Tx Message Buffer13 | Tx Message Buffer13 |
| Tx Message Buffer14 | Tx Message Buffer14 | Tx Message Buffer14 | Tx Message Buffer14 |
| Tx Message Buffer15 | Tx Message Buffer15 | Tx Message Buffer15 | Tx Message Buffer15 |
| Tx Message Buffer16 | Tx Message Buffer16 | Tx Message Buffer16 | Tx Message Buffer16 |
| Tx Message Buffer17 | Tx Message Buffer17 | Tx Message Buffer17 | Tx Message Buffer17 |
| Tx Message Buffer18 | Tx Message Buffer18 | Tx Message Buffer18 | Tx Message Buffer18 |
| Tx Message Buffer19 | Tx Message Buffer19 | Tx Message Buffer19 | Tx Message Buffer19 |
| Tx Message Buffer20 | Tx Message Buffer20 | Tx Message Buffer20 | Tx Message Buffer20 |
| Tx Message Buffer21 | Tx Message Buffer21 | Tx Message Buffer21 | Tx Message Buffer21 |
| Tx Message Buffer22 | Tx Message Buffer22 | Tx Message Buffer22 | Tx Message Buffer22 |
| Tx Message Buffer23 | Tx Message Buffer23 | Tx Message Buffer23 | Tx Message Buffer23 |
| Tx Message Buffer24 | Tx Message Buffer24 | Tx Message Buffer24 | Tx Message Buffer24 |
| Tx Message Buffer25 | Tx Message Buffer25 | Tx Message Buffer25 | Tx Message Buffer25 |
| Tx Message Buffer26 | Tx Message Buffer26 | Tx Message Buffer26 | Tx Message Buffer26 |
| Tx Message Buffer27 | Tx Message Buffer27 | Tx Message Buffer27 | Tx Message Buffer27 |
| Tx Message Buffer28 | Tx Message Buffer28 | Tx Message Buffer28 | Tx Message Buffer28 |
| Tx Message Buffer29 | Tx Message Buffer29 | Tx Message Buffer29 | Tx Message Buffer29 |
| Tx Message Buffer30 | Tx Message Buffer30 | Tx Message Buffer30 | Tx Message Buffer30 |
| Tx Message Buffer31 | Tx Message Buffer31 | Tx Message Buffer31 | Tx Message Buffer31 |
| Tx Message Buffer32 | Tx Message Buffer32 | Ch. Common FIFO0 | Tx Queue2 |
| Tx Message Buffer33 | Ch. Common FIFO0 | Tx Message Buffer33 | Tx Queue2 |
| Tx Message Buffer34 | Tx Message Buffer34 | Tx Message Buffer34 | Tx Queue2 |
| Tx Message Buffer35 | Ch. Common FIFO1 | Tx Message Buffer35 | Tx Queue2 |
| Tx Message Buffer36 | Tx Message Buffer36 | Tx Message Buffer36 | Tx Queue2 |
| Tx Message Buffer37 | Tx Message Buffer37 | Ch. Common FIFO1 | Tx Queue2 |
| Tx Message Buffer38 | Tx Message Buffer38 | Tx Message Buffer38 | Tx Queue2 |
| Tx Message Buffer39 | Tx Message Buffer39 | Tx Message Buffer39 | Tx Queue2 |
| Tx Message Buffer40 | Tx Message Buffer40 | Tx Message Buffer40 | Tx Message Buffer40 |
| Tx Message Buffer41 | Tx Message Buffer41 | Tx Message Buffer41 | Tx Message Buffer41 |
| Tx Message Buffer42 | Tx Message Buffer42 | Tx Message Buffer42 | Tx Message Buffer42 |
| Tx Message Buffer43 | Tx Message Buffer43 | Tx Message Buffer43 | Tx Message Buffer43 |
| Tx Message Buffer44 | Tx Message Buffer44 | Tx Message Buffer44 | Tx Message Buffer44 |
| Tx Message Buffer45 | Tx Message Buffer45 | Tx Message Buffer45 | Tx Message Buffer45 |
| Tx Message Buffer46 | Tx Message Buffer46 | Tx Message Buffer46 | Tx Message Buffer46 |
| Tx Message Buffer47 | Tx Message Buffer47 | Tx Message Buffer47 | Tx Message Buffer47 |
| Tx Message Buffer48 | Tx Message Buffer48 | Tx Message Buffer48 | Tx Message Buffer48 |
| Tx Message Buffer49 | Tx Message Buffer49 | Tx Message Buffer49 | Tx Message Buffer49 |
| Tx Message Buffer50 | Tx Message Buffer50 | Tx Message Buffer50 | Tx Message Buffer50 |
| Tx Message Buffer51 | Tx Message Buffer51 | Tx Message Buffer51 | Tx Message Buffer51 |
| Tx Message Buffer52 | Tx Message Buffer52 | Tx Message Buffer52 | Tx Message Buffer52 |
| Tx Message Buffer53 | Tx Message Buffer53 | Tx Message Buffer53 | Tx Message Buffer53 |
| Tx Message Buffer54 | Tx Message Buffer54 | Tx Message Buffer54 | Tx Message Buffer54 |
| Tx Message Buffer55 | Tx Message Buffer55 | Tx Message Buffer55 | Tx Message Buffer55 |
| Tx Message Buffer56 | Tx Message Buffer56 | Tx Message Buffer56 | Tx Message Buffer56 |
| Tx Message Buffer57 | Tx Message Buffer57 | Tx Message Buffer57 | Tx Message Buffer57 |
| Tx Message Buffer58 | Tx Message Buffer58 | Tx Message Buffer58 | Tx Message Buffer58 |
| Tx Message Buffer59 | Tx Message Buffer59 | Tx Message Buffer59 | Tx Message Buffer59 |
| Tx Message Buffer60 | Tx Message Buffer60 | Tx Message Buffer60 | Tx Message Buffer60 |
| Tx Message Buffer61 | Tx Message Buffer61 | Tx Message Buffer61 | Tx Message Buffer61 |
| Tx Message Buffer62 | Tx Message Buffer62 | Tx Message Buffer62 | Tx Message Buffer62 |
| Tx Message Buffer63 | Tx Message Buffer63 | Tx Message Buffer63 | Tx Message Buffer63 |

**Figure 32.39 Configuration of channel transmission message buffer**

### 32.8.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, then the transmission priority in the CANFD module can be selected from the following two modes:

- CAN ID priority
- Message buffer number priority.

The transmission priority mode is common for all message buffers and all CAN channels. It can be configured with the CFDGCFG.TPRI bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode and includes the TX Queue message buffers.

If the ID of two or more message buffers is the same, then the smaller message buffer number has higher priority for transmission.

Note: For Common FIFO buffers configured in TX mode or GW mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, then the next pending message within the same FIFO is considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

Figure 32.40 shows the transmission configuration flow.

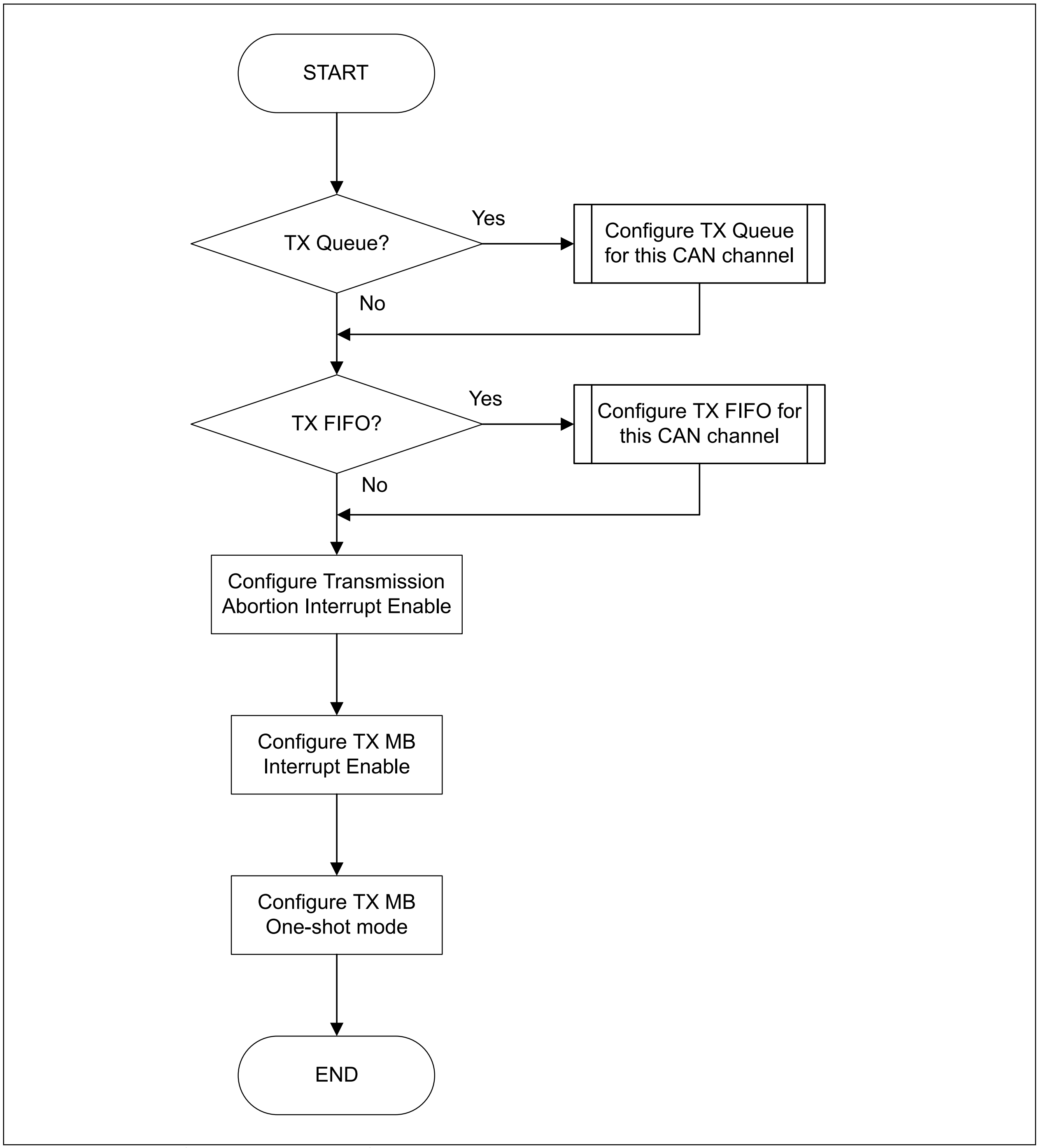

START
TX Queue?
Yes
Configure TX Queue for this CAN channel
No
TX FIFO?
Yes
Configure TX FIFO for this CAN channel
No
Configure Transmission Abortion Interrupt Enable
Configure TX MB Interrupt Enable
Configure TX MB One-shot mode
END

**Figure 32.40 Flow for transmission configuration**

### 32.8.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

1. Regular transmission mode
   If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.
   Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSn.TMTRF) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.

When arbitration is lost or an error occurs, message transmission is further attempted if no transmission abort request is set for this transmission message buffer.
New internal transmission arbitration for this channel is performed for all message buffers with transmission request.

2. One-shot transmission mode

When the CFDTMCn.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, the message buffer is placed in One-shot transmission mode and attempts to transmit a message only once.

Completion of One shot transmission can be checked through the related TX Message Buffer Transmission Result Flag bits (CFDTMSTSn.TMTRF) in the TX Message Buffer Status Registers. The CFDTMSTSn.TMTRF bits are set to 10b or 11b when One-shot transmission is successful.

The CFDTMSTSn.TMTRF bits are set to 01b when arbitration is lost or an error occurs during transmission of the related message buffer.

Additional message transmission is not attempted in this case.

The regular transmission request procedure after a configuration is shown in Figure 32.41.

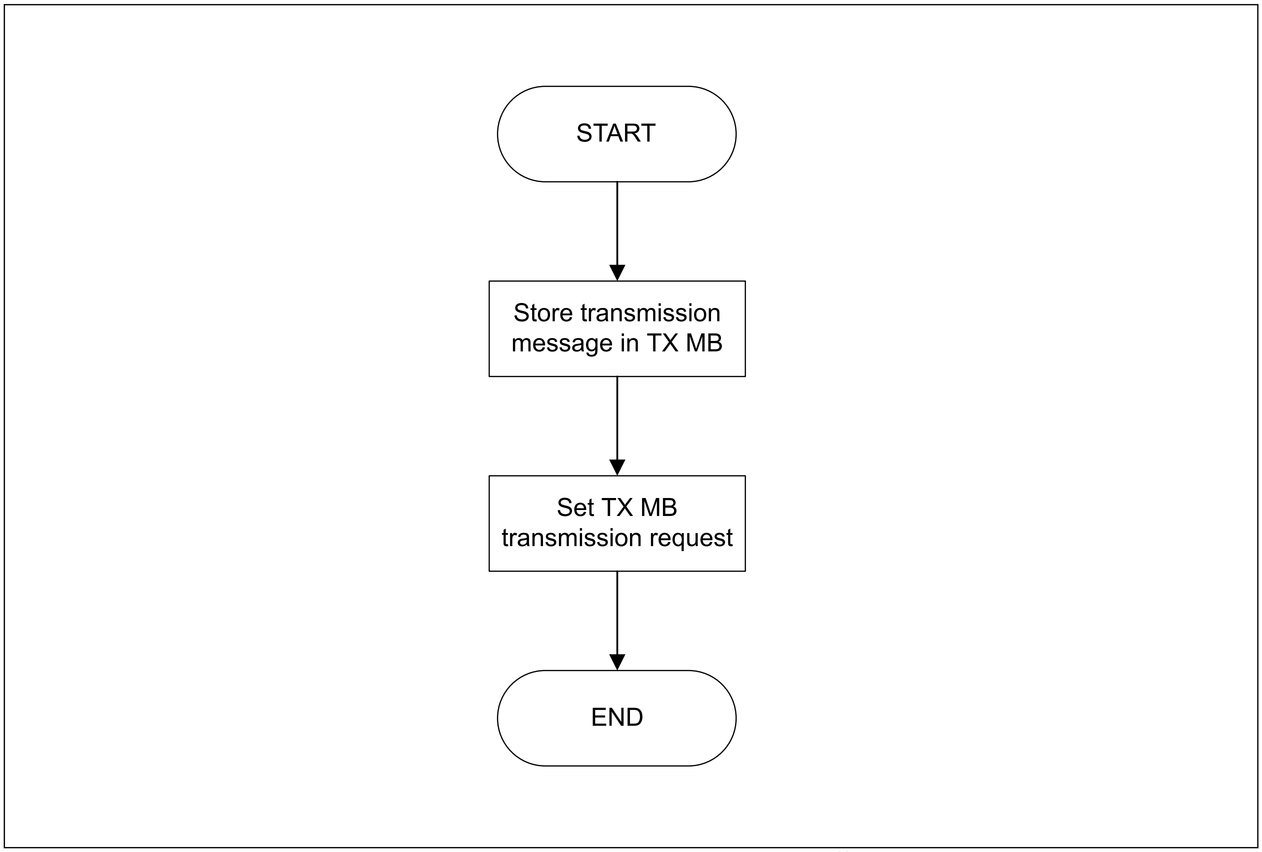

START
Store transmission message in TX MB
Set TX MB transmission request
END

**Figure 32.41 Transmission request procedure using normal TX Message Buffer mode**

### (1) Setting for TX Message Buffer Control Register

Table 32.21 shows configuration of a normal CAN transmission mode.

**Table 32.21 Configuration of CAN transmission mode (1 of 2)**

| Transmission request CFDTMCn.TMTR | Transmission abort request CFDTMCn.TMTAR | One-shot enable CFDTMCn.TMOM | Communication activity |
|---|---|---|---|
| 0 | 0 | 0 | Message buffer disabled |
| 0 | 0 | 1 | Message buffer disabled |

**Table 32.21 Configuration of CAN transmission mode (2 of 2)**

| Transmission request CFDTMCn.TMTR | Transmission abort request CFDTMCn.TMTAR | One-shot enable CFDTMCn.TMOM | Communication activity |
|---|---|---|---|
| 1 | 0 | 0 | Configured as a transmission message buffer for a data frame or a remote frame |
| 1 | 0 | 1 | Configured as a one-shot transmission message buffer for a data frame or a remote frame |
| 1 | 1 | 0 | Transmission abort requested |
| 1 | 1 | 1 | One-shot transmission abort requested |

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 32.42 shows timings for successful transmission for two message buffers of one channel.

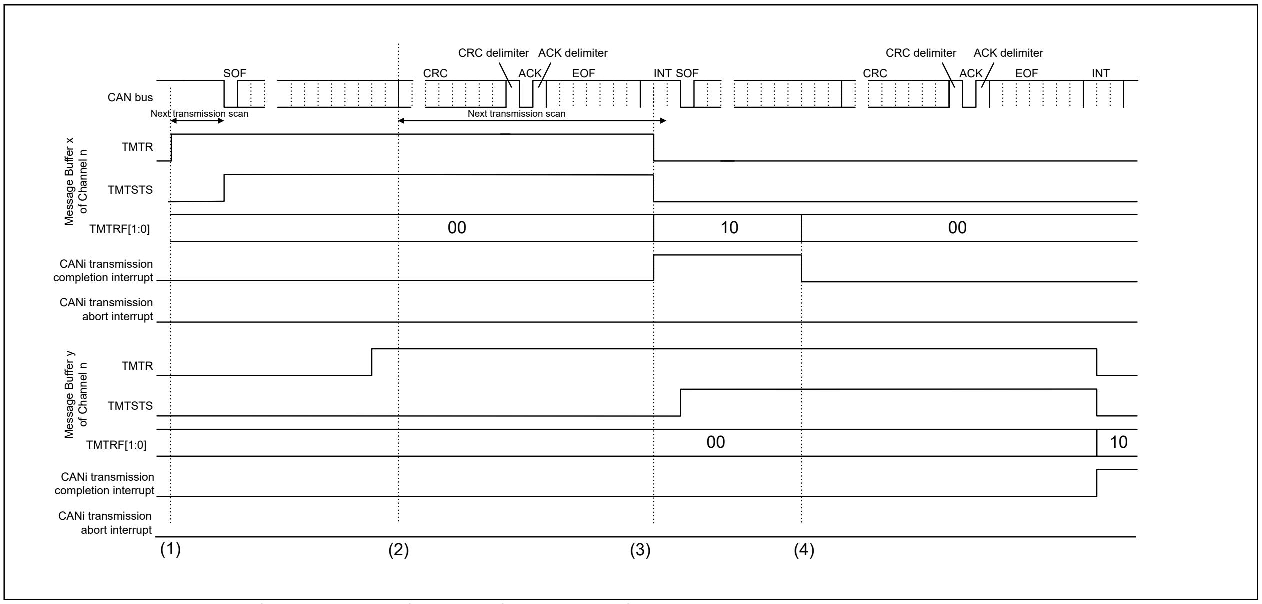

CRC delimiter
ACK delimiter
SOF
CRC
ACK
EOF
INT SOF
CRC delimiter
ACK delimiter
CRC
ACK
EOF
INT
CAN bus
Next transmission scan
Next transmission scan
Message Buffer x of Channel n
TMTR
TMTSTS
TMTRF[1:0]
00
10
00
CANi transmission completion interrupt
CANi transmission abort interrupt
Message Buffer y of Channel n
TMTR
TMTSTS
TMTRF[1:0]
00
10
CANi transmission completion interrupt
CANi transmission abort interrupt
(1)
(2)
(3)
(4)

**Figure 32.42 Timing of request and flag bits for successful transmission**

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the related TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission *1.
2. At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
   The scan time can be delayed due to other transmission scan on other channels, but it finishes before intermission 3 to be able to continue transmission without any gaps.
3. If the message has been successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSn.TMTSTS and the CFDTMCn.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSn.TMTRF flag bits.
4. Before starting the next transmission, clear the CFDTMSTSn.TMTRF bits. Load the next message in the transmission message buffer and set the CFDTMCn.TMTR bit again.
   CFDTMCn.TMTR bit cannot be set again before CFDTMSTSn.TMTRF[1:0] bits are cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs either during transmission or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSn.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 32.43 shows timings for transmission abort for two message buffers of one channel.

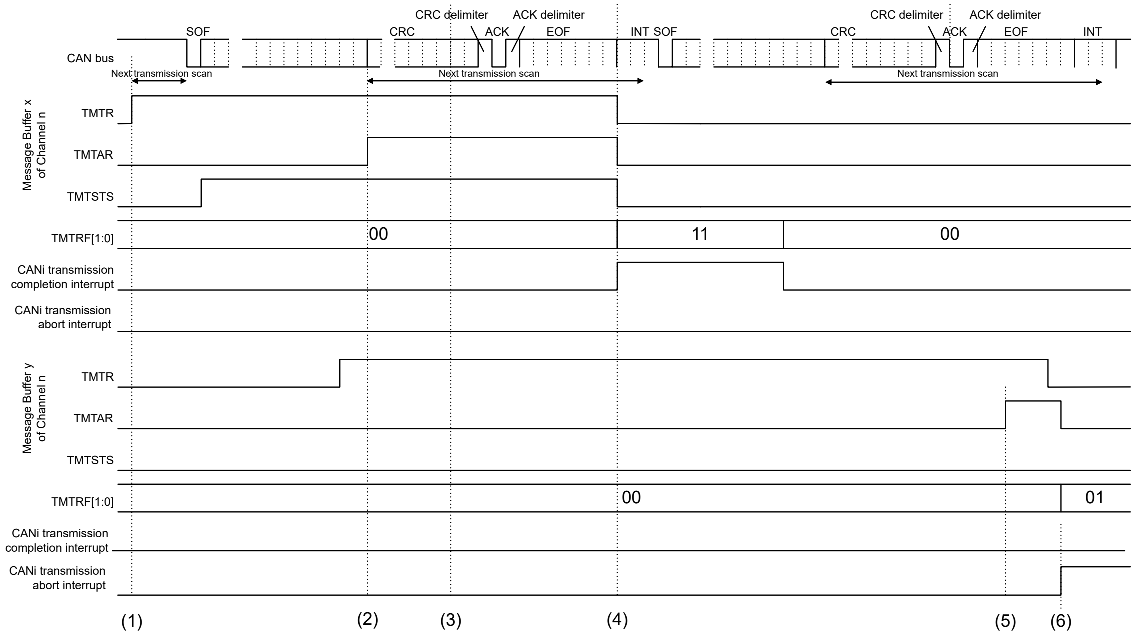


**Figure 32.43 Timing of request and flag bits for transmission abort**

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission*1.
2. If the CFDTMCn.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. At the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example, timing chart message buffer y is not selected as the next transmission message buffer.
   The scan time can be delayed due to other transmission scan on other channels, but it finishes before intermission 3 to be able to continue transmission without any gaps.
4. If the message has been successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and the CFDTMSTSn.TMTSTS and CFDTMCn.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSn.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTSn.TMTSTS is not set). If the CFDTMCn.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTSn.TMTRF[1:0] bits are set to 01b.
   If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b.
   In addition, CFDTMCn.TMTR, and CFDTMCn.TMTAR bits are cleared automatically.
   When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set then an interrupt is generated for successful transmission abort.

To clear the related interrupt line the CFDTMSTSn.TMTRF[1:0] bits have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs, either during transmission, or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

#### 32.8.2.3 TX FIFO or GW FIFO Transmission

Three common FIFO buffers are assigned to each Chanel. The three FIFO buffers can be linked to any normal TX message buffer position for this channel with the CFDCFCCn.CFTML bits in the Common FIFO Configuration/Control Register if configured in TX or GW mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX message buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX message buffer linked to a FIFO buffer configured in TX or GW mode should not be done.

##### (1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value 0xFF is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers. If the message count matches the FIFO depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CANFD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffers can be configured by configuring the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0, then interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCCn.CFIM bit is 1, then interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

The Common FIFO can set interrupt when CAN frame transmission is complete.

The Common FIFO buffer configured in TX Mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Registers. If this bit is cleared to 0, the FIFO Empty flag is set as follows:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

Note: The Common FIFO buffer is considered as disabled after clearing the CFDCFCCn.CFE bit only when the Empty flag is set for the corresponding Common FIFO buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCCn.CFE is set again, ensure that CFDCFSTSn.CFEMP bit is set and that there are no pending abort from the TX FIFO.

When the CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in Figure 32.44.

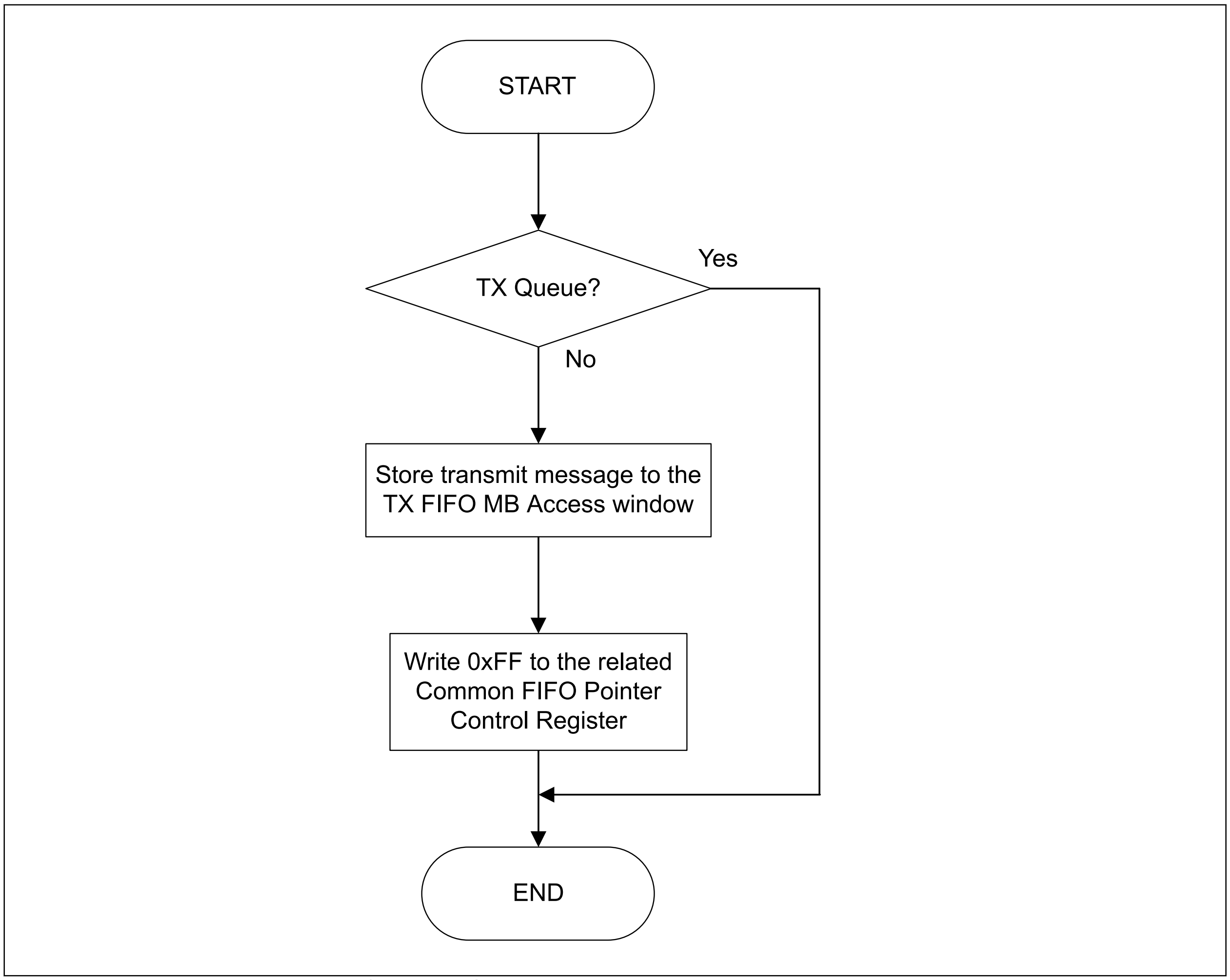

START
TX Queue?
Yes
No
Store transmit message to the TX FIFO MB Access window
Write 0xFF to the related Common FIFO Pointer Control Register
END

**Figure 32.44 Request procedure for TX FIFO transmission**

### (2) GW FIFO Operation

The AFL entries for routing the received messages to GW FIFO buffers should be configured based on the system requirements. The matching AFL entry selects the GW FIFO buffer for storage of a received message on any of the CAN channels.

When a message is successfully received and stored in a GW FIFO buffer, the FIFO message count in the corresponding FIFO Status Register is incremented by 1. If the message count matches the FIFO depth, the FIFO Full flag is set.

The oldest message in the GW FIFO is included in the scan for transmission by the corresponding CANFD module channel logic.

When a message is successfully transmitted from the GW FIFO, the message count value is decremented by 1. When all messages from the GW FIFO are transmitted, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO Full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

In GW mode, when a transmit/receive FIFO buffer is receiving a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting the CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and the CFDCFSTSn.CFMLT bit is set to 1.

- When CFDCFCCEn.CFMOWM = 1:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.
The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message. The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message has been overwritten with the received message.
In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in transmit/receive FIFO buffer full, the transmitting message is lost and retransmission of the message is not performed. The read pointer moves to the next message automatically.

The interrupt generation conditions for the GW FIFO buffers can be configured by setting the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0, the RX Interrupt flag is set when the FIFO counter increments and reaches value configured by CFDCFCCn.CFIGCV and the TX Interrupt flag is set when FIFO transmits the last message successfully.

If CFDCFCCn.CFIM bit is 1, the RX Interrupt flag is set at the end of storage of every received message and the TX Interrupt flag is set if a message is successfully transmitted from the FIFO.

The Common FIFO can set interrupt when:

- CAN frame transmitted is complete
- CAN frame reception is complete
- FIFO is in full status in RX mode or GW mode.

When CFDCFCCEn.CFBME = 1, the mode is FIFO buffering, send data is stored in Common FIFO, and transmission is stopped. Transmission is started if it is set as CFDCFCCEn.CFBME = 0.

The Common FIFO buffers configured in GW mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared, the GW FIFO becomes empty as follows:

- Immediately if the message from the GW FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the GW FIFO is already scheduled for transmission or already in transmission.

Other possible messages pending from the GW FIFO are lost.

Before CFDCFCCn.CFE is set again, ensure that the CFDCFSTSn.CFEMP bit is set and that there are no pending abort from the GW FIFO.

When the CFDCFCCn.CFE bit is cleared and the CFDCFSTSn.CFEMP bit is set, the read and write pointers of the message in GW FIFO are cleared and are no longer active. Therefore, all messages in the GW FIFO buffers are lost and no further message can be stored into the GW FIFO.

In applications intended to be used as CAN-to-CAN gateways, it is recommended that if the Error State Indication (ESI) information of the routing messages is not replaced by the sending node error state indication. For this, each channel has the Control Function register CFDCnFDCFG.ESIC to replace their own ESI information by the routing ESI information.

Note: If the sending node is error-passive, the ESI bit is sent anyway as error-passive (ESI = 1).

Note: This feature is not available in classical CAN function because CFDCnFDCFG is not in classical CAN.

### (3) Interval Timer for FIFO Transmission

For each Common FIFO in TX or GW mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCCn.CFE bit is set.

When the Common FIFO in TX or GW mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCCn.CFE bit.
- CAN channel is in CH_RESET mode.

The interval time is specified by the CFDCFCCn.CFITT value from 0 to 255 timer units in the Common FIFO Configuration/Control Register.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, select a value of 0.

The timer source can be selected with the configuration bit CFITSS in the Common FIFO Configuration/Control Register.

If CAN channel bit time clock is configured as the clock source, and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as the interval timer clock source, the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value CFDGCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See Table 32.22 for CFDGCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency and period.

**Table 32.22 Configuration example for the reference clock of the FIFO interval timer**

| Reference clock/Peripheral clock | 1 µs | 100 µs | 500 µs |
|---|---|---|---|
| 16 MHz/62.5 ns | 16 | 1600 | 8000 |
| 20 MHz/50 ns | 20 | 2000 | 10000 |
| 32 MHz/31.25 ns | 32 | 3200 | 16000 |
| 50 MHz/20 ns | 50 | 5000 | 25000 |

The reference clock resolution can be specified by the interval timer reference clock resolution value CFDCFCCn.CFITR in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value (x1 or x10). The reference clock based interval timer can be used to satisfy the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100 µs to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX/GW FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time. Figure 32.45 shows an example timing of the internal processing.

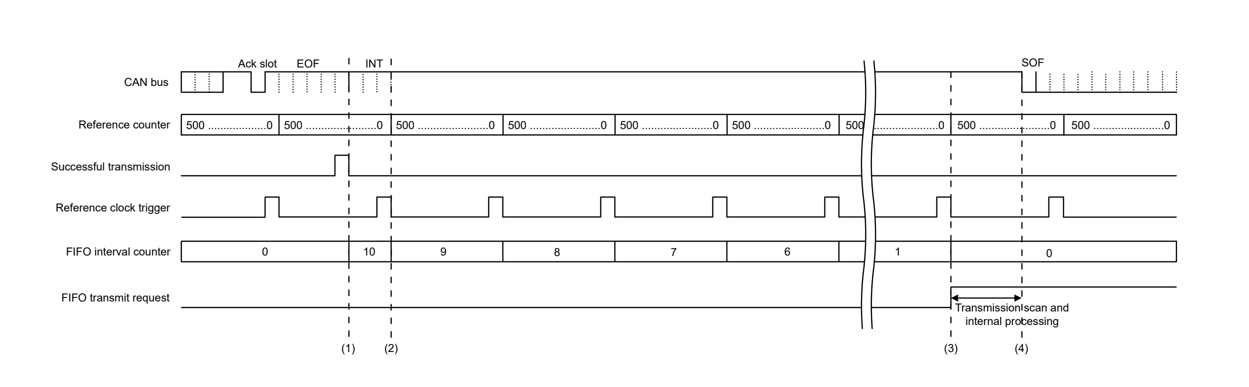


**Figure 32.45 Example for interval processing time**

The configuration for the timing in Figure 32.45 is as follows:

- Peripheral clock frequency = 50 MHz
- Interval timer reference clock (CFDGCFG.ITRCP) = 500 times
- Reference clock from the settings in Figure 32.45 = 10 µs

- Common FIFO interval timer source selection (CFDCFCCn.CFITSS) = 0
- Common FIFO interval timer resolution (CFDCFCCn.CFITR) = 0
- Common FIFO interval transmission time (CFDCFCCn.CFITT) = 10 times
- Theoretical message separation interval = 100 µs

1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to 1 reference clock interval.
2. With the next reference clock trigger the FIFO interval timer is decremented.
3. When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing, this usually takes less than 3 CAN bit time, between the internal FIFO transmit request set in step 3. and the actual transmission.

In the worst case when multiple events such as a reception scan, an internal message routing, a transmit scan on all channels occur, it can take up to 1152 peripheral clock cycles.

As shown in Figure 32.45, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure CFDCFCCn.CFITT to the required minimum value plus 1.

If additional TX message buffers or TX/GW FIFO are configured for transmission of the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time. This is due to higher priority message transmission from these TX message buffers or TX/GW FIFO.

Figure 32.46 shows a block diagram of the FIFO interval time generation circuit.

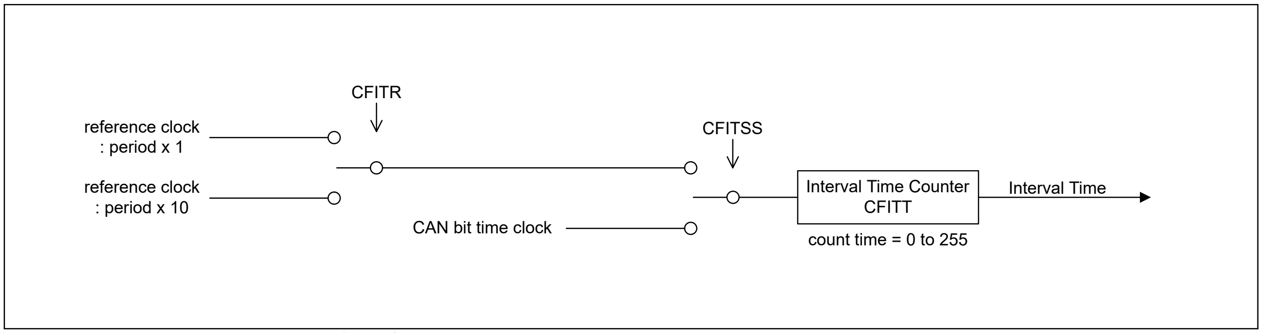


**Figure 32.46 Block diagram of FIFO interval timer**

### 32.8.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of three to 16 TX message buffers, which are accessed through one access window. One channel has four TX Queues:

- The first TX Queue can be configured with a depth of three up to eight buffers and uses TX Message Buffer No. 0 as access window (referred to as TXQ0)
- The second TX Queue can be configured with a depth of three up to eight buffers and uses TX Message Buffer No. 7 as access window (referred to as TXQ1)
- The third TX Queue can be configured with a depth of three up to eight buffers and uses TX Message Buffer No. 32 as access window (referred to as TXQ2)
- The fourth TX Queue can be configured with a depth of three up to eight buffers and uses TX Message Buffer No. 39 as access window (referred to as TXQ3).

All the TXQ0, TXQ1, TXQ2 and TXQ3 messages enter the priority comparison for transmission, which should be only ID Priority (CFDGCFG.TPRI = 0).

The registers for TXQ0 are:

- CFDTXQCC0[n]
- CFDTXQSTS0[n]

- CFDTXQPCTR0[n].

The registers for TXQ1 are:

- CFDTXQCC1[n]
- CFDTXQSTS1[n]
- CFDTXQPCTR1[n].

The registers for TXQ2 are:

- CFDTXQCC2 [n]
- CFDTXQSTS2[n]
- CFDTXQPCTR2[n].

The registers for TXQ3 are:

- CFDTXQCC3[n]
- CFDTXQSTS3[n]
- CFDTXQPCTR3[n].

See related access registers TX Message Buffer ID Registers (TMID[m]), TX Message Buffer Pointer Registers (TMPTR[m]), TX Message Buffer Data Field 0 Registers, and TX Message Buffer Data Field 1 Registers (TMDF[0:1][m]) when access window TXQ0, TXQ1, TXQ2, or TXQ3 is used.

The depth of each TXQ0 buffer can be configured by writing to the CFDTXQCC0n.TXQDC[12:8] bits of the TX Queue Configuration/Control Register. TXQ0 can be set from TXMB0 to TXMB7 as a queue buffer at the maximum.

The 15 available options for the depth configuration of TXQ0 buffer are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x0D: 14 messages
- 0x0E: 15 messages
- 0x0F: 16 messages

The depth of each TXQ1 buffer can be configured by writing to the CFDTXQCC1n.TXQDC[12:8] bits of the TX Queue Configuration/Control Register. TXQ1 can be set from TXMB7 to TXMB0 as a queue buffer at the maximum.

The 15 available options for the depth configuration of TXQ1 buffer are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x0D: 14 messages
- 0x0E: 15 messages
- 0x0F: 16 messages

The depth of each TXQ2 buffer can be configured by writing to the CFDTXQCC2n.TXQDC[12:8] bits of the TX Queue Configuration/Control Register. TXQ2 can be set from TXMB32 to TXMB39 as a queue buffer at the maximum.

The 15 available options for the depth configuration of TXQ2 buffer are:

- 0x00: TX Queue disabled
- 0x01: Reserved

- 0x02: 3 messages
- ⋮
- 0x0D: 14 messages
- 0x0E: 15 messages
- 0x0F: 16 messages

The depth of each TXQ3 buffer can be configured by writing to the CFDTXQCC3n.TXQDC[12:8] bits of the TX Queue Configuration/Control Register. TXQ3 can be set from TXMB39 to TXMB32 as a queue buffer at the maximum.

The 15 available options for the depth configuration of TXQ3 buffer are:

- 00000b: TX Queue disabled
- 00001b: Reserved
- 00010b: 3 messages
- ⋮
- 0x0D: 14 messages
- 0x0E: 15 messages
- 0x0F: 16 messages

When using TXQ1 and TXQ0 simultaneously, the depth of TXQ is 8 or less in total.

When using TXQ3 and TXQ2 simultaneously, the depth of TXQ is 8 or less in total.

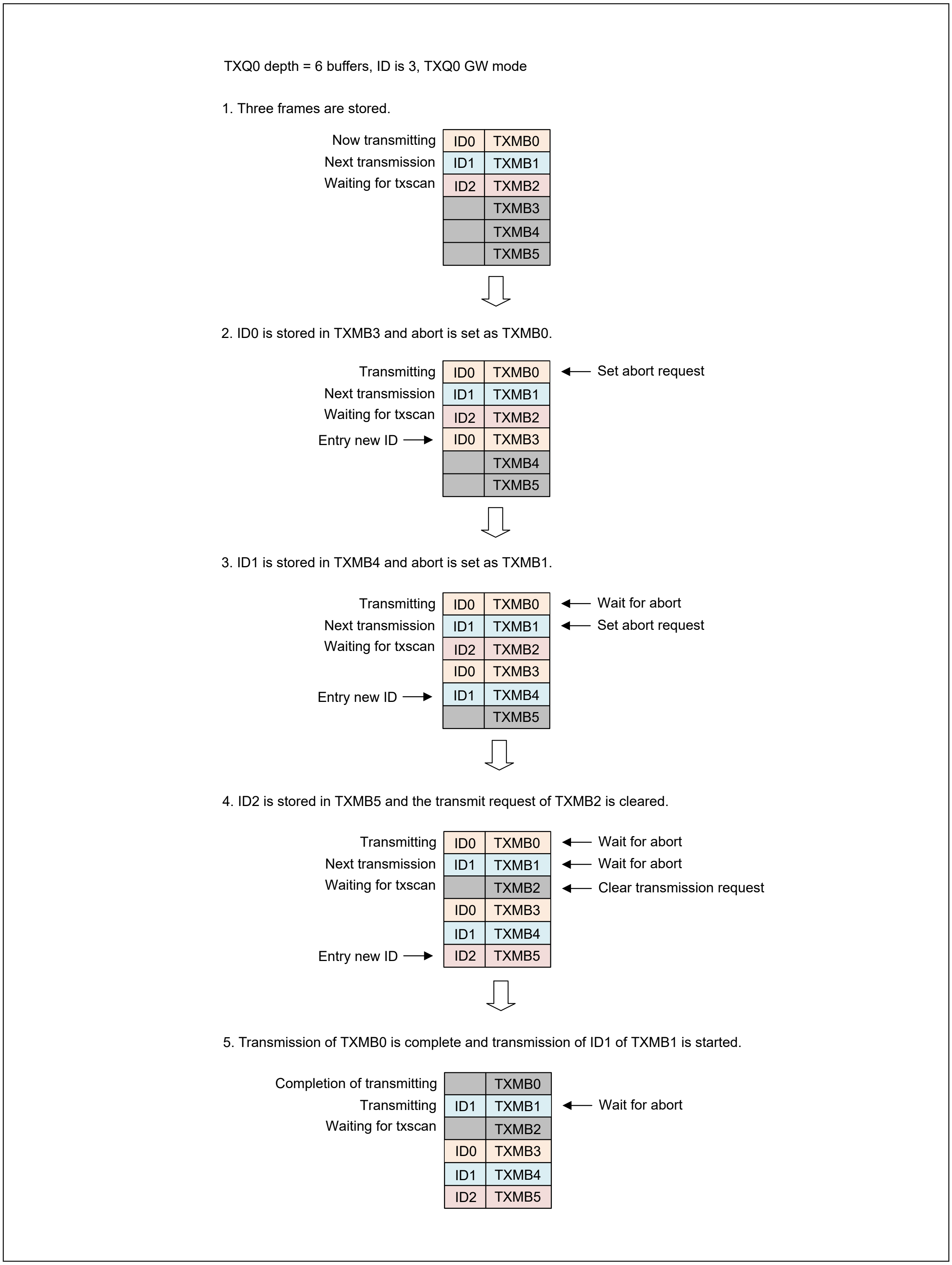
Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 39, TX Message Buffer No. 32, TX Message Buffer No. 7 and TX Message Buffer No. 0, which act as TX Queue access window).

When CFDGAFLP0n.GAFLSRD i (i = 0 to 2) is set and the CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0 to 7) is also set, a receiving frame is stored in the target TXQ as send data by routing.

When CFDTXQCCn.TXQOWE bit is 1, the TX Queue is in TX Queue Overwrite mode. If the message of the same ID is stored in TX Queue when a frame is received and it is stored in TX Queue, an old message is overwritten by a new message. Therefore, an old message is not transmitted. When the old message of the same ID is transmitting and a CAN bus error and an arbitration-lost occur, the message of old ID is not resent.

When using the function in GW mode and TX Queue Overwrite mode, the depth of TXQ (CFDTXQCC0n.TXQDC) should be configured to a value that is the various number of ID used in the TX Queue plus 3. If it accesses by routing in GW mode when a TXQ buffer is full, CFDTXQSTS.TXQMLT is set and send data is thrown away. The function is valid for the standard ID frame and is invalid for the extended ID frame.

Operation of the TXQ same ID over-writing function in GW mode is shown in the following figure.



TXQ0 depth = 6 buffers, ID is 3, TXQ0 GW mode

1. Three frames are stored.

| | | | |
|---|---|---|---|
| Now transmitting | ID0 | TXMB0 | |
| Next transmission | ID1 | TXMB1 | |
| Waiting for txscan | ID2 | TXMB2 | |
| | | TXMB3 | |
| | | TXMB4 | |
| | | TXMB5 | |

2. ID0 is stored in TXMB3 and abort is set as TXMB0.

| | | | |
|---|---|---|---|
| Transmitting | ID0 | TXMB0 | ← Set abort request |
| Next transmission | ID1 | TXMB1 | |
| Waiting for txscan | ID2 | TXMB2 | |
| Entry new ID → | ID0 | TXMB3 | |
| | | TXMB4 | |
| | | TXMB5 | |

3. ID1 is stored in TXMB4 and abort is set as TXMB1.

| | | | |
|---|---|---|---|
| Transmitting | ID0 | TXMB0 | ← Wait for abort |
| Next transmission | ID1 | TXMB1 | ← Set abort request |
| Waiting for txscan | ID2 | TXMB2 | |
| | ID0 | TXMB3 | |
| Entry new ID → | ID1 | TXMB4 | |
| | | TXMB5 | |

4. ID2 is stored in TXMB5 and the transmit request of TXMB2 is cleared.

| | | | |
|---|---|---|---|
| Transmitting | ID0 | TXMB0 | ← Wait for abort |
| Next transmission | ID1 | TXMB1 | ← Wait for abort |
| Waiting for txscan | | TXMB2 | ← Clear transmission request |
| | ID0 | TXMB3 | |
| | ID1 | TXMB4 | |
| Entry new ID → | ID2 | TXMB5 | |

5. Transmission of TXMB0 is complete and transmission of ID1 of TXMB1 is started.

| | | | |
|---|---|---|---|
| Completion of transmitting | | TXMB0 | |
| Transmitting | ID1 | TXMB1 | ← Wait for abort |
| Waiting for txscan | | TXMB2 | |
| | ID0 | TXMB3 | |
| | ID1 | TXMB4 | |
| | ID2 | TXMB5 | |

When a system writes in TXQ, it writes in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full, no further access should be done to the queue, until it is no longer full. If access is a software write when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTSn.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored in the TX Queue.

When a message has been stored to the TX Queue, write 0xFF in the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

Note: If two messages with the same ID are stored in the TX Queue, the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the TX Queue.

If TX Queue Overwrite mode is used, the frame of the same ID is rewritten on a new frame.

For the TX Queue, a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCCn.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in Figure 32.48.

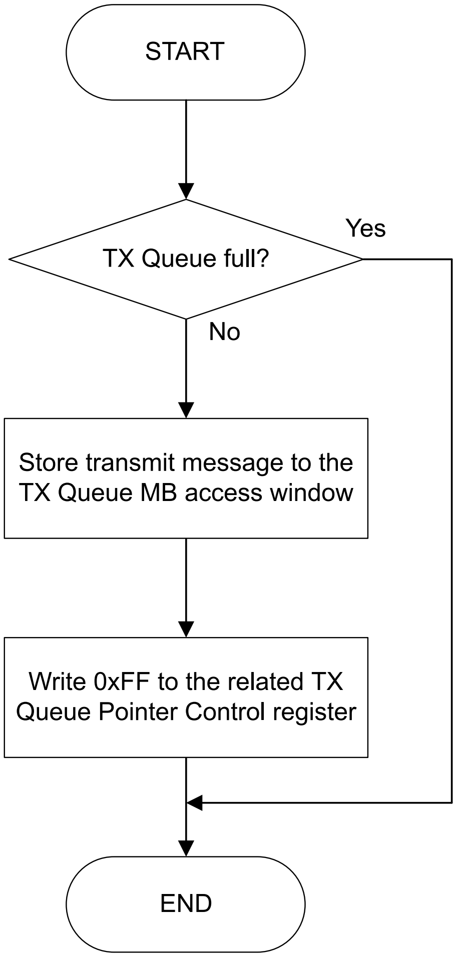

START
TX Queue full?
Yes
No
Store transmit message to the TX Queue MB access window
Write 0xFF to the related TX Queue Pointer Control register
END

**Figure 32.48 TX Queue transmission request**

| TXQ name | Access window | Range width | Direction | Hardware routing access point | CPU access point | DMA access point | Note |
|---|---|---|---|---|---|---|---|
| TXQ0 | TXMB0 | 0, 3-16 | TXMB0 → TXMB7 | Yes | Yes | Yes | When using both TXQ0 and TXQ1, the total number of stages is 8 or less |
| TXQ1 | TXMB7 | 0, 3-16 | TXMB7 → TXMB0 | Yes | No | No | |
| TXQ2 | TXMB32 | 0, 3-16 | TXMB32 → TXMB39 | Yes | No | No | When using both TXQ2 and TXQ3, the total number of stages is 8 or less |
| TXQ3 | TXMB39 | 0, 3-16 | TXMB39 → TXMB32 | No | Yes | Yes | |

TXQ0 can use hardware routing access, CPU access, and DMA access. However, these access methods should not be used simultaneously. Only choose one access method.

### 32.8.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List Buffers for each CAN channel. Two TX History List buffers are provided for each CAN channel and each THL buffer can store up to 16 TX History List entries for a CAN channel.

The CFDTHLCCn.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFO or TX Queue is stored, or if all transmit message information from TX Queue, TX FIFO, or normal TX message buffers is stored in the TX History List for a CAN channel.

When a CFDTHLCCn.THLDGE bit is set, the information on all the frames transmitted in Gateway mode is stored in TX History List. Each transmit message can be individually configured for acceptance to the TX History List with the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted on that CAN channel.

Storing to the list is not synchronized with the status of CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the list can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF is set to 1 when the THLIE bit is configured to 1 or when the TX History List counter CFDTHLSTSn.THLMC[5:0] is increased.

The delay time is dependent on the number of channels due to internal processing.

- Maximum delay time from setting the CFDTMSTSn.TMTRF to store the TX History List data is 224 peripheral bus clock cycles.

The History list records the following information of a transmitted message:

- Buffer type:
  - 001: TX Message Buffer
  - 010: TX FIFO
  - 100: TX Queue
- Buffer number:
  TX message buffer, TX Queue message buffer or TX message buffer link for the Common FIFO buffer from which transmission occurred. The number depends on the buffer type. See Table 32.23.
- Transmission ID:
  Transmission pointer stored in the transmission message
- Transmit timestamp:
  Message timestamp captured at capture point as configured by CFDGFDCFG.TSCCFG.
- Transmission information label:
  Transmission information label stored in the transmission message.
- Transmit gateway buffer indication:
  When data is transmitted from gateway, CFDTHLACC0n.TGW bit is set to 1.

**Table 32.23 TX History List Buffer number entry**

| BT[2:0] Buffer Type | | |
|---|---|---|
| **001b** | **101b** | **100b** |
| **TX Message Buffer** | **TX FIFO** | **TX Queue** |
| TXMB0 | Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO configuration | Number shown corresponds to the message buffer belonging to the TX Queue for which the frame was transmitted |
| TXMB1 | | |
| TXMB2 | | |
| TXMB3 | | |
| TXMB4 | | |
| TXMB5 | | |
| TXMB6 | | |
| TXMB7 | | |
| TXMB32 | | |
| TXMB33 | | |
| TXMB34 | | |
| TXMB35 | | |
| TXMB36 | | |
| TXMB37 | | |
| TXMB38 | | |
| TXMB39 | | |

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDCFFDCSTSn.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, this identification number is stored together with the other message related information to the TX History List and can be read using the Transmission ID (TID) of the TX History List Access Register.

Also, for normal TX message buffers, the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List and the information label is the same.

Figure 32.48 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, 0xFF must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 32.49 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCCn.THLIM bit of the corresponding TX History List Configuration/Control Registers and enabled with the CFDTHLCCn.THLIE bit of the same registers, either to generate an interrupt when the History List reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTSn.THLELT bit in the TX History List Status Register. The status of this bit is also shown by the THLES bit in the Global Error Flag Register.

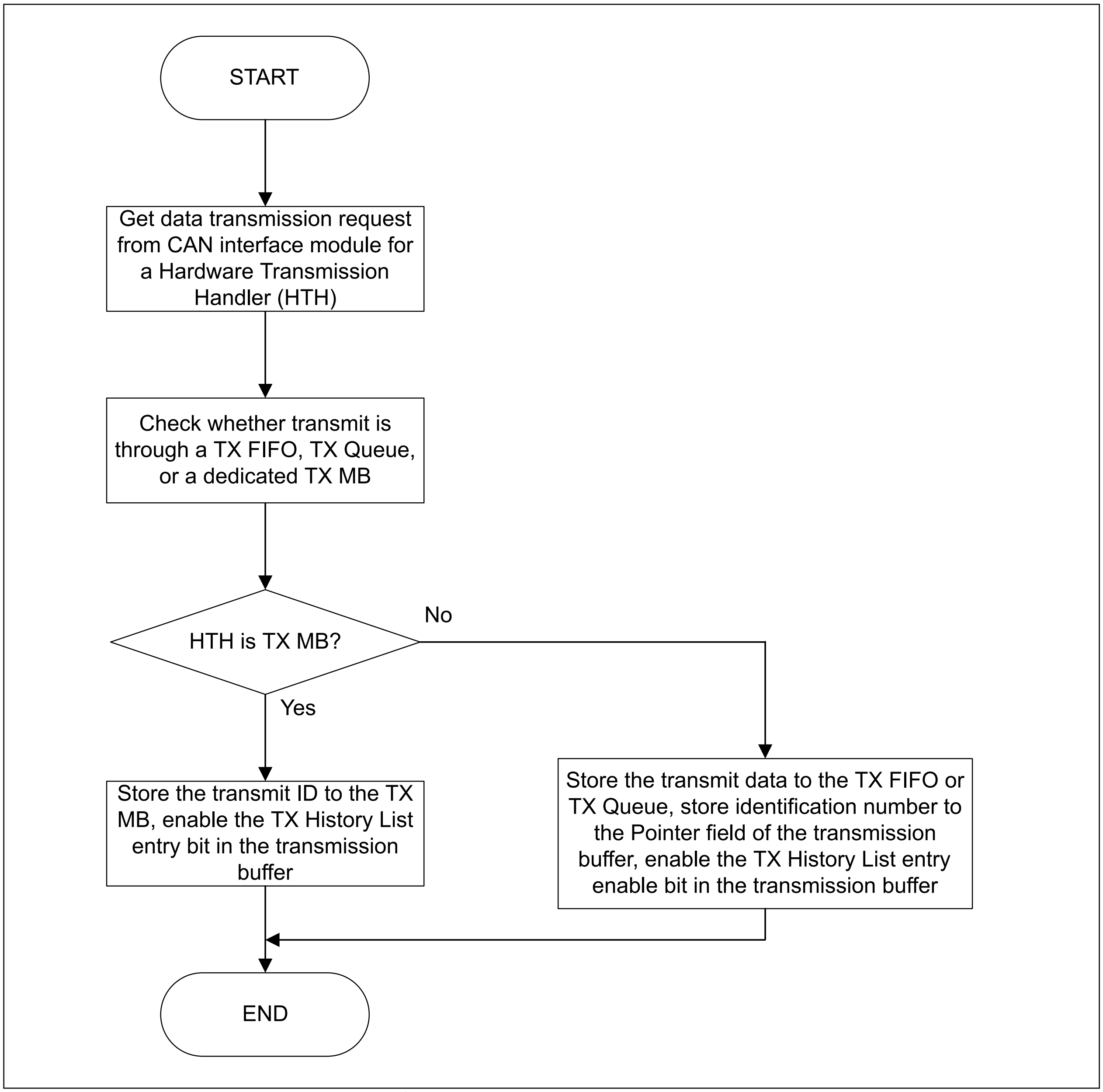

START
Get data transmission request from CAN interface module for a Hardware Transmission Handler (HTH)
Check whether transmit is through a TX FIFO, TX Queue, or a dedicated TX MB
HTH is TX MB?
No
Yes
Store the transmit ID to the TX MB, enable the TX History List entry bit in the transmission buffer
Store the transmit data to the TX FIFO or TX Queue, store identification number to the Pointer field of the transmission buffer, enable the TX History List entry enable bit in the transmission buffer
END

**Figure 32.48 TX History List preparation flow**

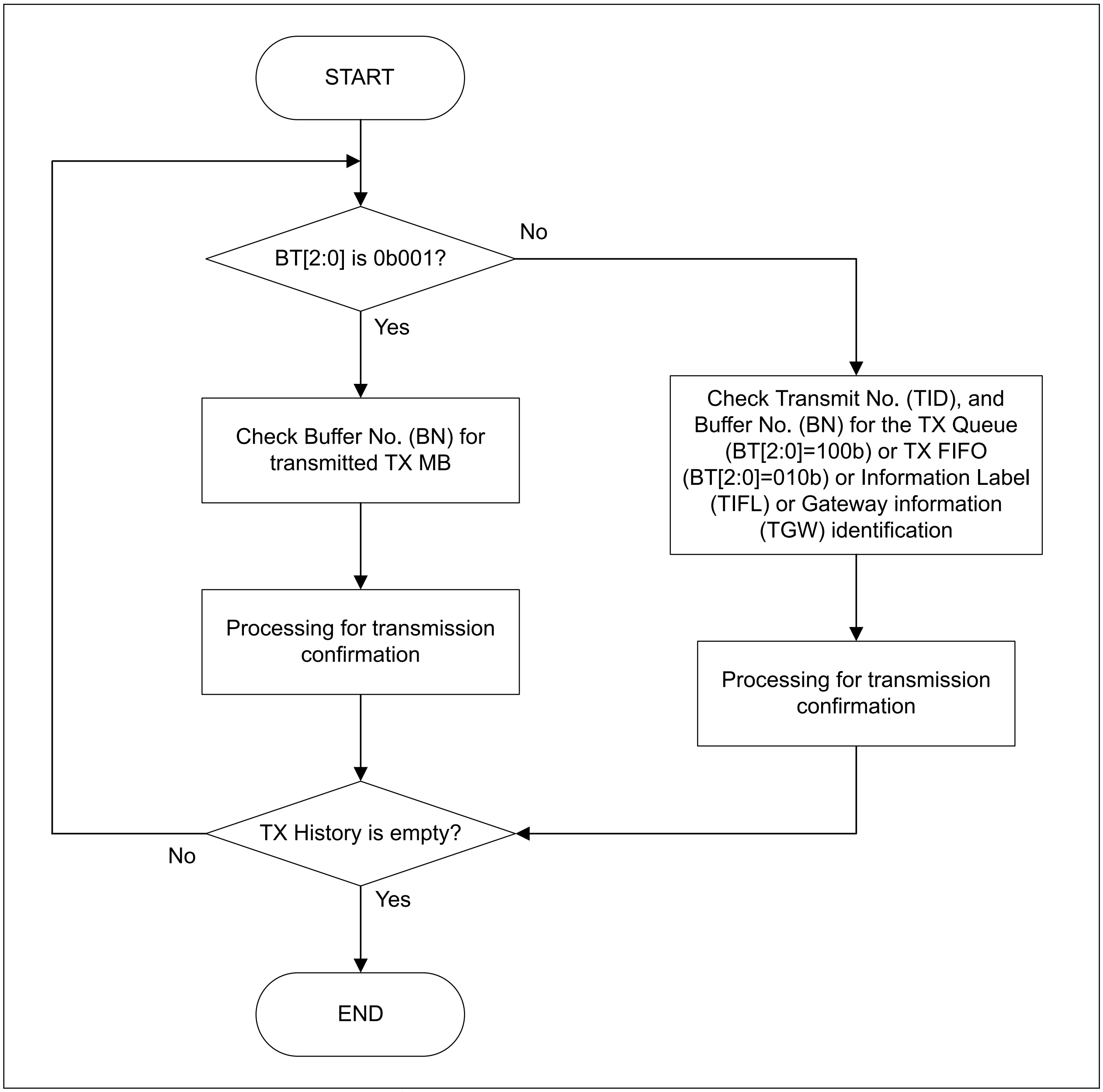

START
BT[2:0] is 0b001?
No
Yes
Check Buffer No. (BN) for transmitted TX MB
Check Transmit No. (TID), and Buffer No. (BN) for the TX Queue (BT[2:0]=100b) or TX FIFO (BT[2:0]=010b) or Information Label (TIFL) or Gateway information (TGW) identification
Processing for transmission confirmation
Processing for transmission confirmation
TX History is empty?
No
Yes
END

**Figure 32.49 TX History List processing flow**

### 32.8.2.6 TX Data Padding

This chapter is not valid for classical CAN.

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, the data bytes beyond the restricted range are replaced by bytes with the value of 0xCC.

This can happen for Common FIFOs configured as (TX or GW mode) when the transmit message DLC is higher than the CFDCFCCn.CFPLS.

This can also happen in FD only mode, if a Classical frame is configured with a DLC bigger than 8.

## 32.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combination of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes.

### 32.9.1 Channel Specific Test Modes

Each CAN channel can be configured into the following test modes:

- Basic test mode
- Listen-only mode
- Self-test mode 0 (External loop back mode)
- Self-test mode 1 (Internal loop back mode)
- Restricted operation mode.

#### 32.9.1.1 Basic Test Mode

The basic test mode should be used when there is requirement for a particular test setting to be enabled other than when in Listen-only and Self-test modes.

#### 32.9.1.2 Listen-only Mode

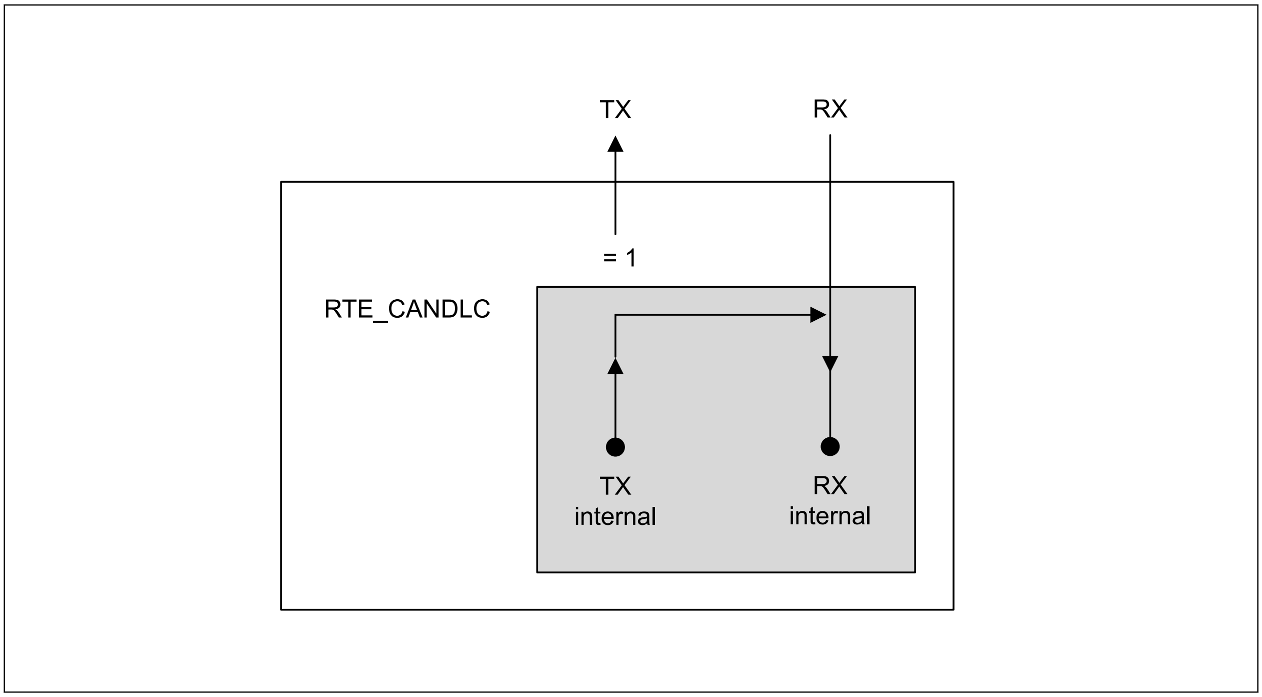
The ISO 11898-1 recommends an optional bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX message buffer or TX/GW FIFO of this channel.

Note: If a message is stored in GW FIFO or routing TXQ, ensure that the transmitting channel is not in Listen-only mode so that transmission is not requested for this channel from the GW FIFO or routing TXQ.

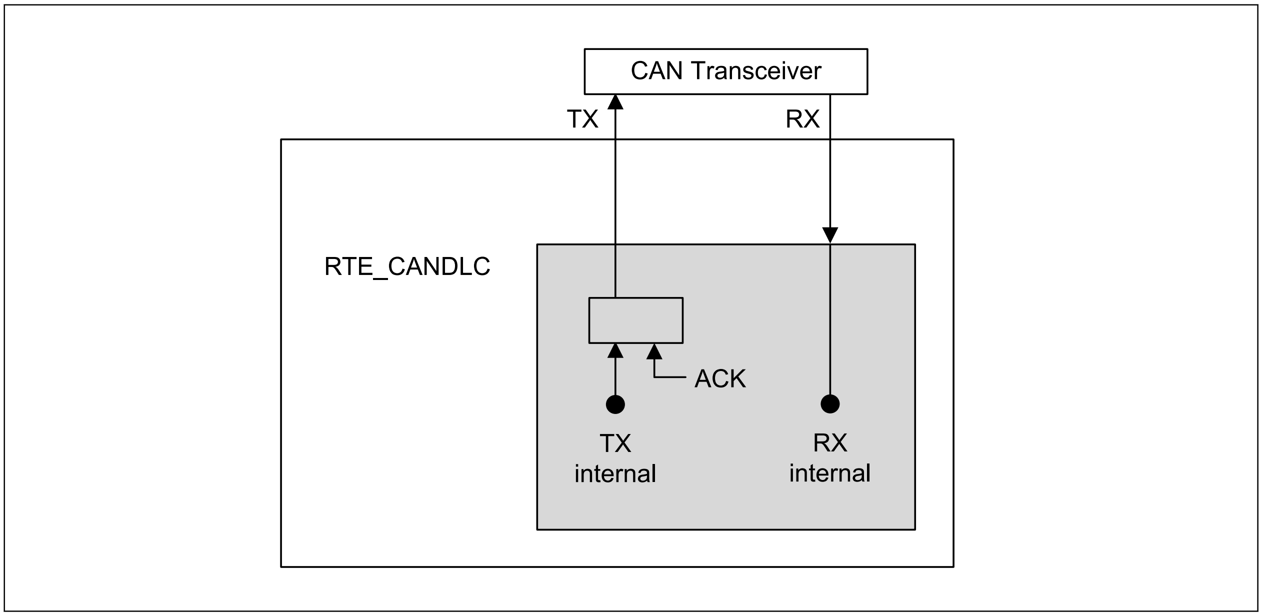


### 32.9.1.3 Self-test Mode 0 (External loopback mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and stores them into its receive message buffers.

To be independent from external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests and the RX/TX pins should be connected to the transceiver.

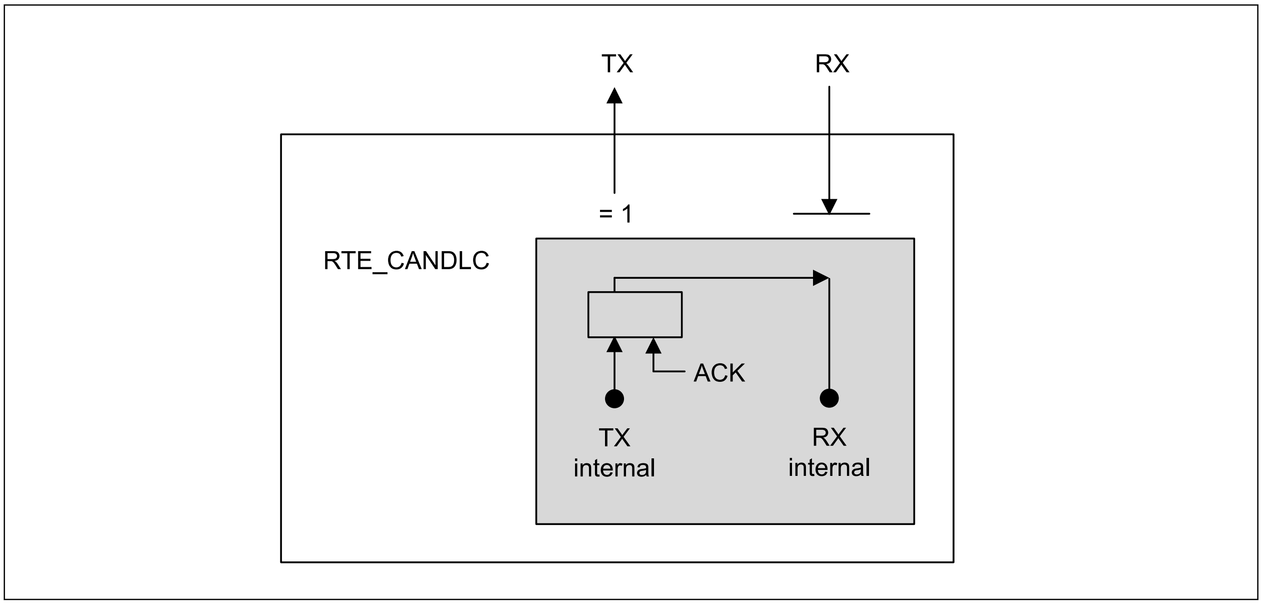


### 32.9.1.4 Self-test Mode 1 (Internal loopback mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits. The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.



### 32.9.1.5 Restricted Operation Mode

This chapter is not valid for classical CAN.

In Restricted operation mode, the CAN node is able to receive valid data and remote frames generating the Acknowledge bit.

Active error or overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors. The mode is specified in ISO 11898-1 and the setting of transmit request is permitted.

### 32.9.2 Global Test Modes

The CANFD module can be configured into the following test modes:

- RAM test mode
- Internal CAN bus communication mode
- CRC error test

The test modes in the following table are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key as shown in the table.

| Test mode | Unlock key 1 | Unlock key 2 |
|---|---|---|
| RAM test mode | 0x7575 | 0x8A8A |

If the software sequence of the two consecutive unlock key write accesses (half-word or word access) is interrupted by any other write access to the register or if incorrect data is written to the Global Unlock Key Register, the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism reset and the test mode enable bit cannot be set and the unlock sequence must be restarted.

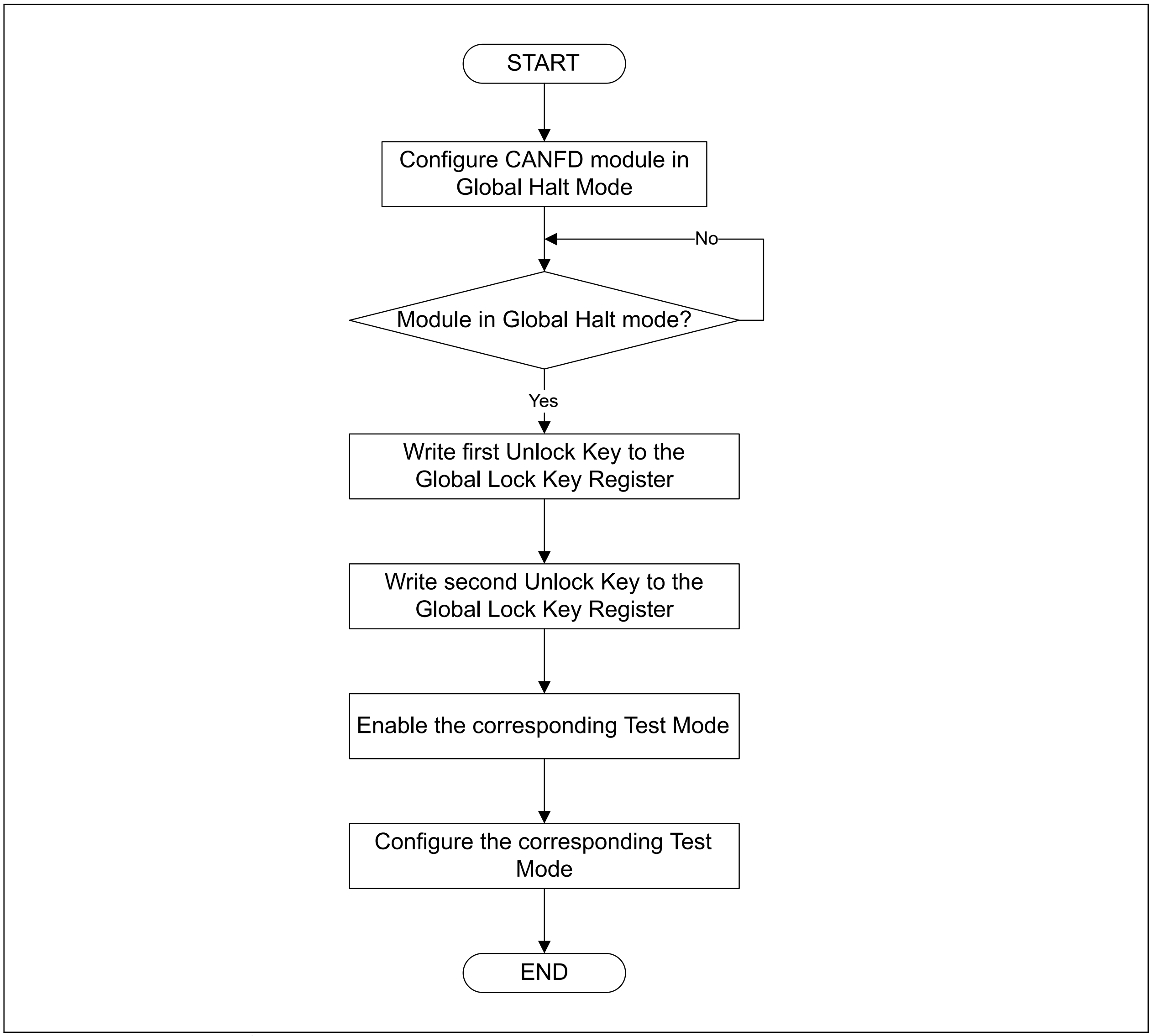

START
Configure CANFD module in Global Halt Mode
No
Module in Global Halt mode?
Yes
Write first Unlock Key to the Global Lock Key Register
Write second Unlock Key to the Global Lock Key Register
Enable the corresponding Test Mode
Configure the corresponding Test Mode
END

**Figure 32.50 Unlock software protection routine**

### 32.9.2.1 RAM Test Mode

The CANFD module can be configured in RAM test mode by setting the CFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after a hardware reset. Therefore, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes, each which can be accessed with the CFDRPGACCn register.

The page should be selected for read/write access by writing to the CFDGTSTCFG.RTMPS[9:0] bits in the Global Test Control Register. Data can then be read from or written in to the RAM Test Page Access Registers.

Figure 32.51 shows the structure of the pages in the RAM when performing a RAM test mode.

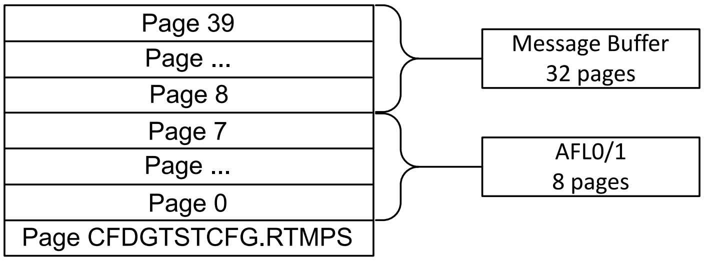


**Figure 32.51 RAM page structure**

The total available RAM size for a 2-CAN channel version is 2048 bytes for the AFL RAM and 8192 bytes for the Message Buffer RAM.

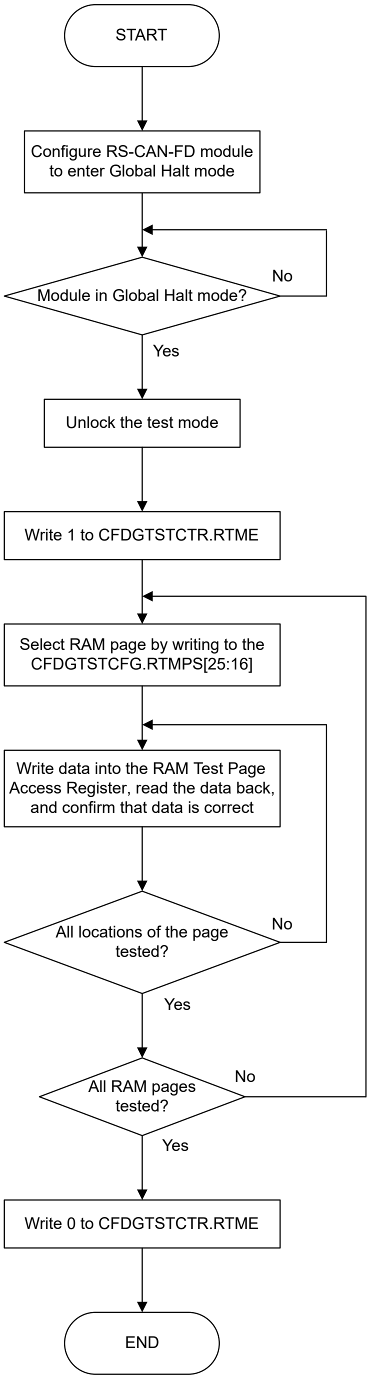
AFL RAM0/1 can treat RAM test mode as one RAM.

The pn and CFDGTSTCFG.RTMPS[9:0] values for the AFL and MB RAMs are calculated in the following way:

pn = ceil (total RAM size in bytes / number of bytes per page)

- AFL RAM:
  pn = ceil (2048 / 256) = 8 pages
  CFDGTSTCFG.RTMPS[9:0] = 0 to 7 (0x00F) inclusive
- MB RAM:
  pn = ceil (8192 / 256) = 32 pages
  CFDGTSTCFG.RTMPS[9:0] = 8 to 39 (0x27) inclusive

Figure 32.52 shows the software flow for RAM test mode.


START
Configure RS-CAN-FD module to enter Global Halt mode
Module in Global Halt mode?
No
Yes
Unlock the test mode
Write 1 to CFDGTSTCTR.RTME
Select RAM page by writing to the CFDGTSTCFG.RTMPS[25:16]
Write data into the RAM Test Page Access Register, read the data back, and confirm that data is correct
All locations of the page tested?
No
Yes
All RAM pages tested?
No
Yes
Write 0 to CFDGTSTCTR.RTME
END

Note 1. Change into the following status before changing to RAMTEST.

- Cancel of a request for transmission.
- Disable of all the FIFO and TXQ.
- Clear of the receiving flag for a receiving buffer.

**Figure 32.52 Software flow for RAM test mode**

To exit this test mode, the CFDGTSTCTR.RTME bit must be cleared. The CFDGTSTCTR.RTME bit is cleared by writing 0 to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CANFD module enters Global Reset mode from the test mode.

### 32.9.2.2 Internal CAN Bus Communication Test Mode

The CANFD module can be configured in internal CAN bus communication test mode by setting the CFDGTSTCTR.ICBCTME bit in the Global Test Control Register. This is a special test mode, in which the CAN channels can be connected together internally to generate a CAN cluster within the CANFD module.

Only use the following sequence to enter internal CAN bus communication test mode:

1. Configure all channels in Halt mode and check that all channels have entered Halt mode (Global Halt mode).
2. Write data into the Global Test Configuration Register to select the channels participating in the internal CAN bus communication test.
3. Set the CFDGTSTCTR.ICBCTME bit of the Global Test Control Register.
4. Check that CFDGTSTCTR.ICBCTME bit is set in the Global Test Control Register.

In this mode, the TXD outputs of the channels participating (configured) in internal CAN bus communication mode are connected together using AND gate. The output of the AND gate is connected to the RXD inputs of all participating channels to create a CAN cluster within the CANFD module. The channels are isolated from the external CAN bus while the CANFD module is in this test mode.

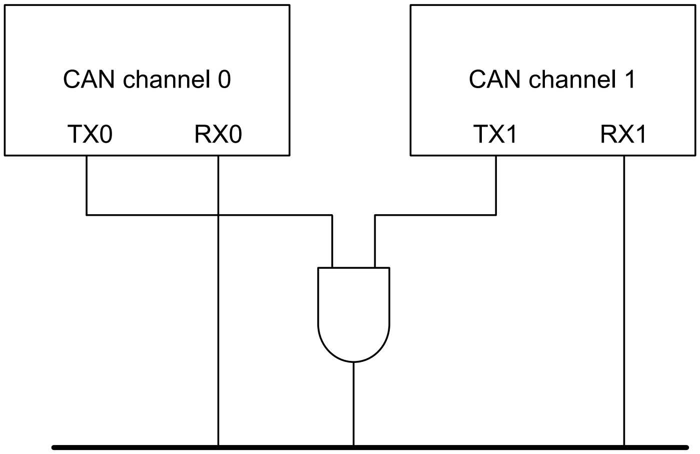


**Figure 32.53 Internal CAN bus connections**

The AFL, Flat RX message buffers, FIFO buffers, Flat TX message buffers and various registers can now be configured as normal to start communication between channels.

The channels not participating in internal CAN bus should only be configured in Halt mode.

#### (1) CRC Error Test

After the CANFD module has been configured in internal CAN bus communication test mode, use the following sequence to perform CRC Error testing. In this sequence, channel x is the reference transmitter CANFD module and channel y is the receiver CANFD module where (x,y = [0…n] and x ≠ y):

1. Configure channel x node to transmit 1 reference message.
2. Set the CFDCyCTR.CRCT bit to 1 to invert the first bit of the incoming bit stream from channel x.
3. Set the CFDTMCx.TMTR.
4. Read either the CFDCyERFL.CRCREG or the CFDCyCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from channel x.

5. Check that CFDCyERFL.CERR is 1

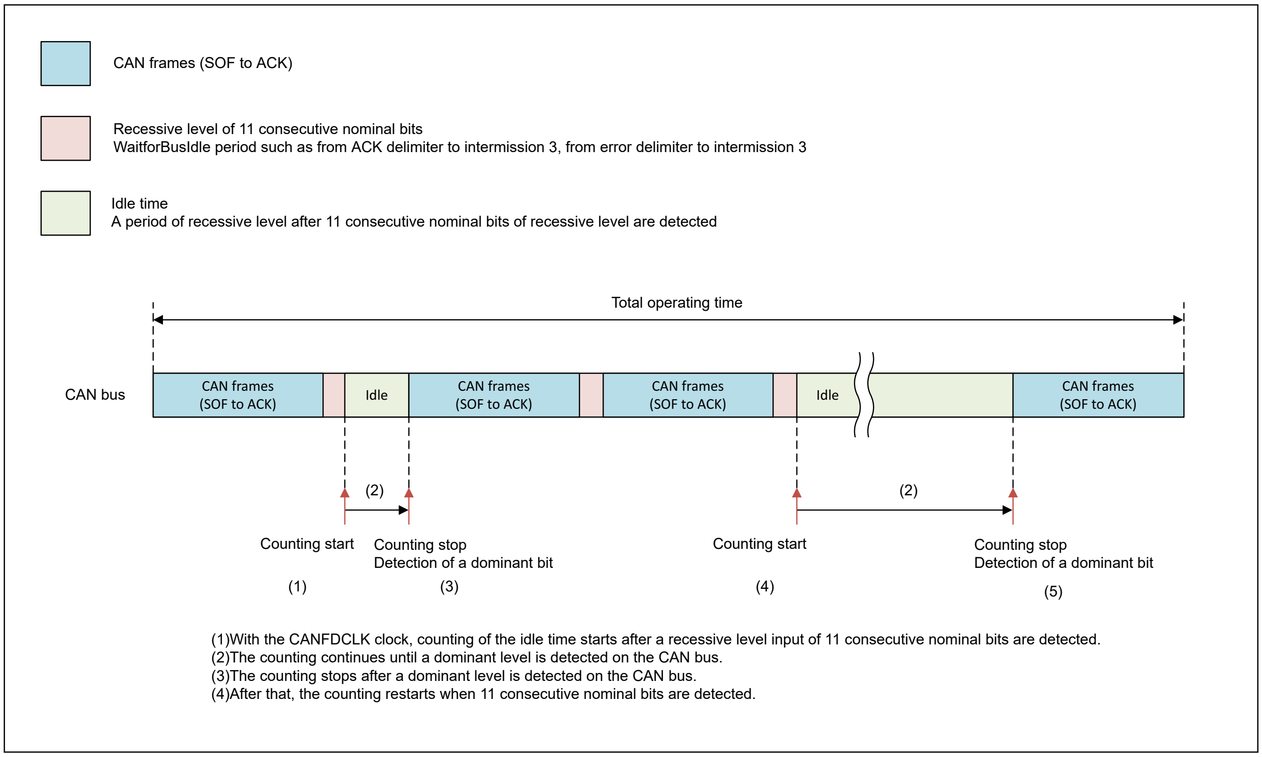
The CRC generator logic is shared between RX and TX so there is no need to create a separate TX CRC error test.

## 32.10 Bus Traffic Measurement

The idle time of the CAN bus can be measured using the CANFDCLK or CANMCLK. Bus traffic can be calculated based on the measurement results.

### 32.10.1 How to Count the CAN Bus Idle Time

The following figure shows the concept for measuring idle time of the CAN bus.


CAN frames (SOF to ACK)
Recessive level of 11 consecutive nominal bits
WaitforBusIdle period such as from ACK delimiter to intermission 3, from error delimiter to intermission 3
Idle time
A period of recessive level after 11 consecutive nominal bits of recessive level are detected
Total operating time
CAN bus
CAN frames (SOF to ACK)
Idle
CAN frames (SOF to ACK)
CAN frames (SOF to ACK)
Idle
CAN frames (SOF to ACK)
(2)
(2)
Counting start
Counting stop
Detection of a dominant bit
Counting start
Counting stop
Detection of a dominant bit
(1)
(3)
(4)
(5)

(1)With the CANFDCLK clock, counting of the idle time starts after a recessive level input of 11 consecutive nominal bits are detected.
(2)The counting continues until a dominant level is detected on the CAN bus.
(3)The counting stops after a dominant level is detected on the CAN bus.
(4)After that, the counting restarts when 11 consecutive nominal bits are detected.

### 32.10.2 Operations and Measurement Procedure

The following procedure shows the steps for measuring idle time of the CAN bus.

1. The channel to be measured exters operation mode.
2. Write 1 to CFDCnBLCT.BLCE bit to set the measuring counter to operating mode.
3. Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register.
4. Detect a recessive level input of 11 consecutive nominal bits.
5. Start counting the bus idle time.
6. Detect a dominant level.
7. The counting stops.
8. Detect a recessive level input of 11 consecutive nominal bits.
9. The counting starts.
10. Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the counter value to CFDCnBLSTS.
11. Read the value of CFDCnBLSTS.

To stop the measurement counter, write 0 to CFDCnBLCT.BLCE bit.

To initialize the counter, write 1 to CFDCnBLCT.BLCLD bit.

This measurement is enabled when the channel to be measured is in operation mode.

When the relevant channels are in reset mode, the counter does not operate.

Also, accurate measurements are not available in test mode.

Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the value of the counter to CFDCnBLSTS.

The lower three bits of CFDCnBLSTS are fixed to 0.

Based on the values of the counter, software can calculate the CAN bus traffic according to the following formulas.

$$\frac{(\text{total operating time} - \text{total idle time})}{\text{Total operating time}} = \frac{\text{bus operating time}}{\text{total operating time}} = \text{Bus usage ratio}$$

- Total idle time: a value read from CFDCnBLSTS × a clock cycle of CANFDCLK
- Total operating time: a setting interval of CFDCnBLCT.BLCLD bit

Example: Below is a calculation example under the following conditions.

- Conditions: nominal bit rate = 1 Mbps
- CANFDCLK clock = 40 MHz (= 25 ns)
- A setting interval of CFDCnBLCT.BLCLD bit = cycle of 1 ms
- A read value of CFDCnBLSTS register = 0x4E20

$$\frac{(\text{total operating time} - \text{total idle time})}{\text{Total operating time}} = \frac{(1000000\text{ns} - 20000 * 25\text{ns})}{1000000\text{ns}} = 50\,\%$$

## 32.11 Usage Notes

### 32.11.1 Settings for the Module-Stop State

The MSTPCRC can enable or disable CANFD operation. The CANFD module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.