

- If required, disable frame self reception by setting the SRX\_DIS bit
- Enable the FIFO by setting the RFEN bit
- Enable the abort mechanism by setting the AEN bit
- Enable the local priority feature by setting the LPRIO\_EN bit
- Initialize the Control Register
  - Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, RJW
  - Determine the bit rate by programming the PRESDIV field
  - Determine the internal arbitration mode (LBUF bit)
- Initialize the Message Buffers
  - The Control and Status word of all Message Buffers must be initialized
  - If FIFO was enabled, the 8-entry ID table must be initialized
  - Other entries in each Message Buffer should be initialized as required
- Initialize the Rx Individual Mask Registers
- Set required interrupt mask bits in the IMASK Registers (for all MB interrupts), in CTRL Register (for Bus Off and Error interrupts) and in MCR Register for Wake-Up interrupt
- Negate the HALT bit in MCR

Starting with the last event, FLEXCAN attempts to synchronize to the CAN bus.

## 26.8 FLEXCAN Memory Map/Register Definition

The complete memory map for a FLEXCAN module with 64 MBs capability is shown in the following table. Each individual register is identified by its complete name and the corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming the SUPV bit in the MCR Register. The MCR register allows only Supervisor access regardless the SUPV bit state.

The FLEXCAN module stores CAN messages for transmission and reception using a Mailboxes and Rx FIFO structure.

**FLEXCAN memory map**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_0000	Module Configuration Register (FLEXCAN1_MCR)	32	R/W	5980_000Fh	<a href="#">26.8.1/1326</a>
209_0004	Control 1 Register (FLEXCAN1_CTRL1)	32	R/W	0000_0000h	<a href="#">26.8.2/1331</a>
209_0008	Free Running Timer Register (FLEXCAN1_TIMER)	32	R/W	0000_0000h	<a href="#">26.8.3/1334</a>

*Table continues on the next page...*

**FLEXCAN memory map (continued)**

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
209_0010	Rx Mailboxes Global Mask Register (FLEXCAN1_RXMGMASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.4/1334</a>
209_0014	Rx Buffer 14 Mask Register (FLEXCAN1_RX14MASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.5/1335</a>
209_0018	Rx Buffer 15 Mask Register (FLEXCAN1_RX15MASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.6/1336</a>
209_001C	Error Counter Register (FLEXCAN1_ECR)	32	R/W	0000_0000h	<a href="#">26.8.7/1337</a>
209_0020	Error and Status 1 Register (FLEXCAN1_ESR1)	32	R/W	0000_0000h	<a href="#">26.8.8/1338</a>
209_0024	Interrupt Masks 2 Register (FLEXCAN1_IMASK2)	32	R/W	0000_0000h	<a href="#">26.8.9/1342</a>
209_0028	Interrupt Masks 1 Register (FLEXCAN1_IMASK1)	32	R/W	0000_0000h	<a href="#">26.8.10/1342</a>
209_002C	Interrupt Flags 2 Register (FLEXCAN1_IFLAG2)	32	R/W	0000_0000h	<a href="#">26.8.11/1343</a>
209_0030	Interrupt Flags 1 Register (FLEXCAN1_IFLAG1)	32	R/W	0000_0000h	<a href="#">26.8.12/1343</a>
209_0034	Control 2 Register (FLEXCAN1_CTRL2)	32	R/W	0000_0000h	<a href="#">26.8.13/1345</a>
209_0038	Error and Status 2 Register (FLEXCAN1_ESR2)	32	R	0000_0000h	<a href="#">26.8.14/1351</a>
209_0044	CRC Register (FLEXCAN1_CRCCR)	32	R	0000_0000h	<a href="#">26.8.15/1353</a>
209_0048	Rx FIFO Global Mask Register (FLEXCAN1_RXFGMASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.16/1354</a>
209_004C	Rx FIFO Information Register (FLEXCAN1_RXFIR)	32	R	0000_0000h	<a href="#">26.8.17/1355</a>
209_0880	Rx Individual Mask Registers (FLEXCAN1_RXIMR0_RXIMR63)	32	R/W	0000_0000h	<a href="#">26.8.18/1356</a>
209_09E0	Glitch Filter Width Registers (FLEXCAN1_GFWR)	32	R/W	0000_007Fh	<a href="#">26.8.19/1356</a>
209_4000	Module Configuration Register (FLEXCAN2_MCR)	32	R/W	5980_000Fh	<a href="#">26.8.1/1326</a>
209_4004	Control 1 Register (FLEXCAN2_CTRL1)	32	R/W	0000_0000h	<a href="#">26.8.2/1331</a>
209_4008	Free Running Timer Register (FLEXCAN2_TIMER)	32	R/W	0000_0000h	<a href="#">26.8.3/1334</a>
209_4010	Rx Mailboxes Global Mask Register (FLEXCAN2_RXMGMASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.4/1334</a>
209_4014	Rx Buffer 14 Mask Register (FLEXCAN2_RX14MASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.5/1335</a>
209_4018	Rx Buffer 15 Mask Register (FLEXCAN2_RX15MASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.6/1336</a>
209_401C	Error Counter Register (FLEXCAN2_ECR)	32	R/W	0000_0000h	<a href="#">26.8.7/1337</a>
209_4020	Error and Status 1 Register (FLEXCAN2_ESR1)	32	R/W	0000_0000h	<a href="#">26.8.8/1338</a>
209_4024	Interrupt Masks 2 Register (FLEXCAN2_IMASK2)	32	R/W	0000_0000h	<a href="#">26.8.9/1342</a>
209_4028	Interrupt Masks 1 Register (FLEXCAN2_IMASK1)	32	R/W	0000_0000h	<a href="#">26.8.10/1342</a>
209_402C	Interrupt Flags 2 Register (FLEXCAN2_IFLAG2)	32	R/W	0000_0000h	<a href="#">26.8.11/1343</a>

Table continues on the next page...

**FLEXCAN memory map (continued)**

<b>Absolute address (hex)</b>	<b>Register name</b>	<b>Width (in bits)</b>	<b>Access</b>	<b>Reset value</b>	<b>Section/ page</b>
209_4030	Interrupt Flags 1 Register (FLEXCAN2_IFLAG1)	32	R/W	0000_0000h	<a href="#">26.8.12/1343</a>
209_4034	Control 2 Register (FLEXCAN2_CTRL2)	32	R/W	0000_0000h	<a href="#">26.8.13/1345</a>
209_4038	Error and Status 2 Register (FLEXCAN2_ESR2)	32	R	0000_0000h	<a href="#">26.8.14/1351</a>
209_4044	CRC Register (FLEXCAN2_CRCR)	32	R	0000_0000h	<a href="#">26.8.15/1353</a>
209_4048	Rx FIFO Global Mask Register (FLEXCAN2_RXFGMASK)	32	R/W	FFFF_FFFFh	<a href="#">26.8.16/1354</a>
209_404C	Rx FIFO Information Register (FLEXCAN2_RXFIR)	32	R	0000_0000h	<a href="#">26.8.17/1355</a>
209_4880	Rx Individual Mask Registers (FLEXCAN2_RXIMR0_RXIMR63)	32	R/W	0000_0000h	<a href="#">26.8.18/1356</a>
209_49E0	Glitch Filter Width Registers (FLEXCAN2_GFWR)	32	R/W	0000_007Fh	<a href="#">26.8.19/1356</a>

**26.8.1 Module Configuration Register (FLEXCANx\_MCR)**

This register defines global system configurations, such as the module operation mode (e.g., low power) and maximum message buffer configuration.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0
	MDIS	FRZ	RFEN	HALT	NOT_RDY	WAK_MSK	SOFT_RST	FRZ_ACK	SUPV	SLF_WAK	WRN_EN	LPM_ACK	WAK_SRC	Reserved	SRX_DIS	IRMQ

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	Reserved	Reserved	LPRIO_EN	AEN	Reserved	IDAM	Reserved	Reserved	Reserved	MAXMB	MAXMB	MAXMB	MAXMB	MAXMB	MAXMB	MAXMB

**FLEXCANx\_MCR field descriptions**

Field	Description
31 MDIS	<p>This bit controls whether FLEXCAN is enabled or not. When disabled, FLEXCAN shuts down the clocks to the CAN Protocol Interface and Message Buffer Management sub-modules. This is the only bit in MCR not affected by soft reset. See <a href="#">Module Disable Mode</a> for more information.</p> <p>1 Disable the FLEXCAN module 0 Enable the FLEXCAN module</p>
30 FRZ	<p>The FRZ bit specifies the FLEXCAN behavior when the HALT bit in the MCR Register is set or when Debug Mode is requested at ARM level. When FRZ is asserted, FLEXCAN is enabled to enter Freeze Mode. Negation of this bit field causes FLEXCAN to exit from Freeze Mode.</p> <p>1 Enabled to enter Freeze Mode 0 Not enabled to enter Freeze Mode</p>
29 RFEN	<p>This bit controls whether the Rx FIFO feature is enabled or not. When RFEN is set, MBs 0 to 5 cannot be used for normal reception and transmission because the corresponding memory region (0x80-0xDC) is used by the FIFO engine as well as additional MBs (up to 32, depending on CTRL2[RFFN] setting) which are used as Rx FIFO ID Filter Table elements. RFEN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in <a href="#">Table 26-18</a> (see <a href="#">Arbitration and Matching Timing</a>). This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 FIFO enabled 0 FIFO not enabled</p>
28 HALT	<p>Assertion of this bit puts the FLEXCAN module into Freeze Mode. The ARM should clear it after initializing the Message Buffers and Control Register. No reception or transmission is performed by FLEXCAN before this bit is cleared. Freeze Mode can not be entered while FLEXCAN is in any of the low power modes. See <a href="#">Freeze Mode</a> for more information.</p> <p>1 Enters Freeze Mode if the FRZ bit is asserted. 0 No Freeze Mode request.</p>
27 NOT_RDY	<p>This read-only bit indicates that FLEXCAN is either in Disable Mode, Stop Mode or Freeze Mode. It is negated once FLEXCAN has exited these modes.</p> <p>1 FLEXCAN module is either in Disable Mode, Stop Mode or Freeze Mode 0 FLEXCAN module is either in Normal Mode, Listen-Only Mode or Loop-Back Mode</p>
26 WAK_MSK	<p>This bit enables the Wake Up Interrupt generation.</p> <p>1 Wake Up Interrupt is enabled 0 Wake Up Interrupt is disabled</p>
25 SOFT_RST	<p>When this bit is asserted, FlexCAN resets its internal state machines and some of the memory mapped registers. The following registers are reset: MCR (except the MDIS bit), TIMER, ECR, ESR1, ESR2, IMASK1, IMASK2, IFLAG1, IFLAG2 and CRCR. Configuration registers that control the interface to the CAN bus are not affected by soft reset. The following registers are unaffected: CTRL1, CTRL2, RXIMR0_RXIMR63, RXGMASK, RX14MASK, RX15MASK, RXFGMASK, RXFIR and all Message Buffers.</p> <p>The SOFT_RST bit can be asserted directly by the ARM when it writes to the MCR Register. It may take some time to fully propagate its effect. The SOFT_RST bit remains asserted while reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed.</p> <p>Soft reset cannot be applied while clocks are shut down in any of the low power modes. The module should be first removed from low power mode, and then soft reset can be applied.</p>

*Table continues on the next page...*

**FLEXCANx\_MCR field descriptions (continued)**

Field	Description
	1 Reset the registers 0 No reset request
24 FRZ_ACK	<p>This read-only bit indicates that FLEXCAN is in Freeze Mode and its prescaler is stopped. The Freeze Mode request cannot be granted until current transmission or reception processes have finished. Therefore the software can poll the FRZ_ACK bit to know when FLEXCAN has actually entered Freeze Mode. If Freeze Mode request is negated, then this bit is negated once the FLEXCAN prescaler is running again. If Freeze Mode is requested while FLEXCAN is in any of the low power modes, then the FRZ_ACK bit will only be set when the low power mode is exited. See <a href="#">Freeze Mode</a> for more information</p> 1 FLEXCAN in Freeze Mode, prescaler stopped 0 FLEXCAN not in Freeze Mode, prescaler running
23 SUPV	<p>This bit configures some of the FLEXCAN registers to be either in Supervisor or User Mode. Reset value of this bit is '1', so the affected registers start with Supervisor access allowance only. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> 1 FlexCAN is in Supervisor Mode. Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location 0 FlexCAN is in User Mode. Affected registers allow both Supervisor and Unrestricted accesses
22 SLF_WAK	<p>This bit enables the Self Wake Up feature when FLEXCAN is in Stop Mode. If this bit had been asserted by the time FLEXCAN entered Stop Mode, then FLEXCAN will look for a recessive to dominant transition on the bus during these modes. If a transition from recessive to dominant is detected during Stop Mode, then FLEXCAN generates, if enabled to do so, a Wake Up interrupt to the ARM so that it can resume the clocks globally and FlexCAN can request to resume the clocks. This bit can not be written while the module is in Stop Mode.</p> 1 FLEXCAN Self Wake Up feature is enabled 0 FLEXCAN Self Wake Up feature is disabled
21 WRN_EN	<p>When asserted, this bit enables the generation of the TWRN_INT and RWRN_INT flags in the Error and Status Register. If WRN_EN is negated, the TWRN_INT and RWRN_INT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> 1 TWRN_INT and RWRN_INT bits are set when the respective error counter transition from <96 to ≥ 96. 0 TWRN_INT and RWRN_INT bits are zero, independent of the values in the error counters.
20 LPM_ACK	<p>This read-only bit indicates that FLEXCAN is either in Disable Mode or Stop Mode. Either of these low power modes can not be entered until all current transmission or reception processes have finished, so the ARM can poll the LPM_ACK bit to know when FLEXCAN has actually entered low power mode. See <a href="#">Module Disable Mode</a>, and <a href="#">Stop Mode</a> for more information</p> 1 FLEXCAN is either in Disable Mode, or Stop mode 0 FLEXCAN not in any of the low power modes
19 WAK_SRC	<p>This bit defines whether the integrated low-pass filter is applied to protect the FLEXCAN_RX input from spurious wake up. See <a href="#">Stop Mode</a> for more information. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> 1 FLEXCAN uses the filtered FLEXCAN_RX input to detect recessive to dominant edges on the CAN bus 0 FLEXCAN uses the unfiltered FLEXCAN_RX input to detect recessive to dominant edges on the CAN bus.
18 -	This field is reserved. Reserved

*Table continues on the next page...*

**FLEXCANx\_MCR field descriptions (continued)**

Field	Description
17 SRX_DIS	<p>This bit defines whether FlexCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the module will not be stored in any MB, regardless if the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Self reception disabled 0 Self reception enabled</p>
16 IRMQ	<p>This bit indicates whether Rx matching process will be based either on individual masking and queue or on masking scheme with RXMGMASK, RX14MASK and RX15MASK, RXFGMASK. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Individual Rx masking and queue feature are enabled. 0 Individual Rx masking and queue feature are disabled. For backward compatibility, the reading of C/S word locks the MB even if it is EMPTY.</p>
15–14 -	This field is reserved. Reserved
13 LPRIQ_EN	<p>This bit is provided for backwards compatibility reasons. It controls whether the local priority feature is enabled or not. It is used to extend the ID used during the arbitration process. With this extended ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Local Priority enabled 0 Local Priority disabled</p>
12 AEN	<p>This bit is supplied for backwards compatibility reasons. When asserted, it enables the Tx abort feature. This feature guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This bit can only be written in Freeze mode as it is blocked by hardware in other modes. Write Abort code into Rx Mailboxes can cause unpredictable results when the MCR[AEN] is asserted.</p> <p>1 Abort enabled 0 Abort disabled</p>
11–10 -	This field is reserved. Reserved
9–8 IDAM	<p>This 2-bit field identifies the format of the elements of the Rx FIFO filter table, as shown below. Note that all elements of the table are configured at the same time by this field (they are all the same format). See <a href="#">Rx FIFO Structure</a>. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>00 Format A One full ID (standard or extended) per ID filter Table element. 01 Format B Two full standard IDs or two partial 14-bit extended IDs per ID filter Table element. 10 Format C Four partial 8-bit IDs (standard or extended) per ID filter Table element. 11 Format D All frames rejected.</p>
7 -	This field is reserved. Reserved
MAXMB	<p>This 7-bit field defines the number of the last Message Buffers that will take part in the matching and arbitration processes. The reset value (0x0F) is equivalent to 16 MB configuration. This field can only be written in Freeze Mode as it is blocked by hardware in other modes</p> <p>Number of the last MB = MAXMB.</p>

*Table continues on the next page...*

**FLEXCANx\_MCR field descriptions (continued)**

Field	Description
	<b>NOTE:</b> Additionally, the value of MAXMB must encompass the FIFO size defined by CTRL2[RFFN] MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in <a href="#">Table 26-18</a> (see <a href="#">Arbitration and Matching Timing</a> ).

**26.8.2 Control 1 Register (FLEXCANx\_CTRL1)**

This register is defined for specific FLEXCAN control features related to the CAN bus, such as bit-rate, programmable sampling point within an Rx bit, Loop Back Mode, Listen Only Mode, Bus Off recovery behavior and interrupt enabling (Bus-Off, Error, Warning). It also determines the Division Factor for the clock prescaler.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PRESDIV								RJW		PSEG1			PSEG2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOFF_MSK	ERR_MSK	Reserved	LPB	TWRN_MSK	RWRN_MSK	Reserved		SMP	BOFF_REC	TSYN	LBUF	LOM	PROP_SEG		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**FLEXCANx\_CTRL1 field descriptions**

Field	Description
31–24 PRESDIV	This 8-bit field defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The Maximum value of this register is 0xFF, that gives a minimum Sclock frequency equal to the PE clock frequency divided by 256. For more information refer to <a href="#">Protocol Timing</a> . This field can only be written in Freeze mode as it is blocked by hardware in other modes.

Table continues on the next page...



**FLEXCANx\_CTRL1 field descriptions (continued)**

Field	Description
	Sclock frequency = CPI clock frequency / (PRESDIV+1)
23–22 RJW	This 2-bit field defines the maximum number of time quanta <sup>1</sup> that a bit time can be changed by one re-synchronization. The valid programmable values are 0-3. This field can only be written in Freeze mode as it is blocked by hardware in other modes  Resync Jump Width = RJW + 1.
21–19 PSEG1	This 3-bit field defines the length of Phase Buffer Segment 1 in the bit time. The valid programmable values are 0-7. This field can only be written in Freeze mode as it is blocked by hardware in other modes  Phase Buffer Segment 1 = (PSEG1 + 1) x Time-Quanta.
18–16 PSEG2	This 3-bit field defines the length of Phase Buffer Segment 2 in the bit time. The valid programmable values are 1-7. This field can only be written in Freeze mode as it is blocked by hardware in other modes  Phase Buffer Segment 2 = (PSEG2 + 1) x Time-Quanta.
15 BOFF_MSK	This bit provides a mask for the Bus Off Interrupt.  1 Bus Off interrupt enabled 0 Bus Off interrupt disabled
14 ERR_MSK	This bit provides a mask for the Error Interrupt.  1 Error interrupt enabled 0 Error interrupt disabled
13 -	This field is reserved. Reserved
12 LPB	This bit configures FlexCAN to operate in Loop-Back Mode. In this mode, FlexCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The FLEXCAN_RX input pin is ignored and the FLEXCAN_TX output goes to the recessive state (logic '1'). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.  1 Loop Back enabled 0 Loop Back disabled
11 TWRN_MSK	This bit provides a mask for the Tx Warning Interrupt associated with the TWRN_INT flag in the Error and Status Register. This bit is read as zero when MCR[WRN_EN] bit is negated. This bit can only be written if MCR[WRN_EN] bit is asserted.  1 Tx Warning Interrupt enabled 0 Tx Warning Interrupt disabled
10 RWRN_MSK	This bit provides a mask for the Rx Warning Interrupt associated with the RWRN_INT flag in the Error and Status Register. This bit is read as zero when MCR[WRN_EN] bit is negated. This bit can only be written if MCR[WRN_EN] bit is asserted.  1 Rx Warning Interrupt enabled 0 Rx Warning Interrupt disabled
9–8 -	This field is reserved. Reserved
7 SMP	This bit defines the sampling mode of CAN bits at the FLEXCAN_RX. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.

*Table continues on the next page...*

**FLEXCANx\_CTRL1 field descriptions (continued)**

Field	Description
	<p>1 Three samples are used to determine the value of the received bit: the regular one (sample point) and 2 preceding samples, a majority rule is used</p> <p>0 Just one sample is used to determine the bit value</p>
6 BOFF_REC	<p>This bit defines how FLEXCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFF_REC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FLEXCAN will re-synchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, the BOFF_REC bit can be re-asserted again during Bus Off, but it will only be effective the next time the module enters Bus Off. If BOFF_REC was negated when the module entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery.</p> <p>1 Automatic recovering from Bus Off state disabled</p> <p>0 Automatic recovering from Bus Off state enabled, according to CAN Spec 2.0 part B</p>
5 TSYN	<p>This bit enables a mechanism that resets the free-running timer each time a message is received in Message Buffer 0. This feature provides means to synchronize multiple FLEXCAN stations with a special "SYNC" message (i.e., global network time). If the RFEN bit in MCR is set (FIFO enabled), the first available Mailbox, according to CTRL2[RFFN] setting, is used for timer synchronization instead of MB0. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Timer Sync feature enabled</p> <p>0 Timer Sync feature disabled</p>
4 LBUF	<p>This bit defines the ordering mechanism for Message Buffer transmission. When asserted, the LPRIO_EN bit does not affect the priority arbitration. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Lowest number buffer is transmitted first</p> <p>0 Buffer with highest priority is transmitted first</p>
3 LOM	<p>This bit configures FLEXCAN to operate in Listen Only Mode. In this mode, transmission is disabled, all error counters are frozen and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FLEXCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.</p> <p>Listen-Only Mode acknowledgement can be obtained by the state of ESR1[FLT_CONF] field which is Passive Error when Listen-Only Mode is entered. There can be some delay between the Listen-Only Mode request and acknowledge.</p> <p>This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 FLEXCAN module operates in Listen Only Mode</p> <p>0 Listen Only Mode is deactivated</p>
PROP_SEG	<p>This 3-bit field defines the length of the Propagation Segment in the bit time. The valid programmable values are 0-7. This field can only be written in Freeze mode as it is blocked by hardware in other modes</p> <p>Propagation Segment Time = (PROPSEG + 1) * Time-Quanta.</p> <p>Time-Quantum = one Sclck period.</p>

1. One time quantum is equal to the Sclck period.

### 26.8.3 Free Running Timer Register (FLEXCANx\_TIMER)

This register represents a 16-bit free running counter that can be read and written by the ARM. The timer starts from \$0000 after Reset, counts linearly to \$FFFF, and wraps around.

The timer is clocked by the FLEXCAN bit-clock (which defines the baud rate on the CAN bus). During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. During Freeze Mode, disable, and stop mode, the timer is not incremented.

The timer value is captured at the beginning of the identifier field of any frame on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.

If bit CTRL1[TSYN] is asserted the Timer is reset whenever a message is received in the first available Mailbox, according to CTRL2[RFFN] setting.

ARM can write to this register anytime. However, if the write occurs at the same time that the Timer is being reset by a reception in the first Mailbox, then the write value is discarded.

Reading this register affects the Mailbox Unlocking procedure. For additional details, refer to [Message Buffer Lock Mechanism](#).

Address: Base address + 8h offset

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

FLEXCANx\_TIMER field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
TIMER	TIMER

### 26.8.4 Rx Mailboxes Global Mask Register (FLEXCANx\_RXMGMASK)

RXMGMASK is provided for legacy support. Asserting the MCR[IRMQ] bit causes the RXMGMASK Register to have no effect on the module operation.

RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14-15, which have individual mask registers.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

**Table 26-49. Rx Mailboxes Global Mask usage**

SMB[RTR] <sup>1</sup>	CTRL2[RRS]	CTRL2[EACEN]	Mailbox filter fields			
			MB[RTR]	MB[IDE]	MB[ID]	reserved
0	-	0	- Note <sup>2</sup>	- Note <sup>3</sup>	MG[28:0]	MG[31:29]
0	-	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	-	-	-	-	MG[31:0]
1	1	0	-	-	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If CTRL2[EACEN] bit is negated the RTR bit of Mailbox is never compared with the RTR bit of the Incoming Frame (Rx SMB[RTR]).
3. If CTRL2[EACEN] bit is negated the IDE bit of Mailbox is always compared with the IDE bit of the Incoming Frame (Rx SMB[IDE]).

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MG31_MG0																															
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

### FLEXCANx\_RXMGMASK field descriptions

Field	Description
MG31_MG0	<p>These bits mask the Mailbox filter bits as shown in the figure above. Note that the alignment with the ID word of the Mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE which are located in the Control and Status word of the Mailbox. <a href="#">Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK)</a> shows in detail which MG bits mask each Mailbox filter field.</p> <p>1 The corresponding bit in the filter is checked against the one received  0 the corresponding bit in the filter is "don't care"</p>

## 26.8.5 Rx Buffer 14 Mask Register (FLEXCANx\_RX14MASK)

RX14MASK is provided for legacy support, asserting the MCR[IRMQ] bit causes the RX14MASK to have no effect on the module operation.

RX14MASK is used to mask the filter fields of Message Buffer 14.

This register can only be programmed while the module is in Freeze Mode as it is blocked by hardware in other modes.

## FLEXCAN Memory Map/Register Definition

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX14M31_RX14M0																															
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

### FLEXCANx\_RX14MASK field descriptions

Field	Description
RX14M31_ RX14M0	These bits mask Mailbox 14 filter bits in the same fashion as RXMGMASK masks other Mailboxes filters (see <a href="#">Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK)</a> )  1 The corresponding bit in the filter is checked 0 the corresponding bit in the filter is "don't care"

## 26.8.6 Rx Buffer 15 Mask Register (FLEXCANx\_RX15MASK)

RX15MASK is provided for legacy support, asserting the MCR[IRMQ] bit causes the RX15MASK Register to have no effect on the module operation.

RX15MASK is used to mask the filter fields of Message Buffer 15.

This register can only be programmed while the module is in Freeze Mode as it is blocked by hardware in other modes.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX15M31_RX15M0																															
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

### FLEXCANx\_RX15MASK field descriptions

Field	Description
RX15M31_ RX15M0	These bits mask Mailbox 15 filter bits in the same fashion as RXMGMASK masks other Mailboxes filters (see <a href="#">Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK)</a> ).  1 The corresponding bit in the filter is checked 0 the corresponding bit in the filter is "don't care"

### 26.8.7 Error Counter Register (FLEXCANx\_ECR)

This register has 2 8-bit fields reflecting the value of two FLEXCAN error counters: Transmit Error Counter (Tx\_Err\_Counter field) and Receive Error Counter (Rx\_Err\_Counter field). The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FLEXCAN module. Both counters are read only except in Freeze Mode, where they can be written by the ARM.

FLEXCAN responds to any bus state as described in the protocol, e.g. transmit 'Error Active' or 'Error Passive' flag, delay its transmission start time ('Error Passive') and avoid any influence on the bus when in 'Bus Off' state. The following are the basic rules for FLEXCAN bus state transitions.

- If the value of Tx\_Err\_Counter or Rx\_Err\_Counter increases to be greater than or equal to 128, the FLT\_CONF field in the Error and Status Register is updated to reflect 'Error Passive' state.
- If the FLEXCAN state is 'Error Passive', and either Tx\_Err\_Counter or Rx\_Err\_Counter decrements to a value less than or equal to 127 while the other already satisfies this condition, the FLT\_CONF field in the Error and Status Register is updated to reflect 'Error Active' state.
- If the value of Tx\_Err\_Counter increases to be greater than 255, the FLT\_CONF field in the Error and Status Register is updated to reflect 'Bus Off' state, and an interrupt may be issued. The value of Tx\_Err\_Counter is then reset to zero.
- If FLEXCAN is in 'Bus Off' state, then Tx\_Err\_Counter is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, Tx\_Err\_Counter is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around while incrementing the Tx\_Err\_Counter. When Tx\_Err\_Counter reaches the value of 128, the FLT\_CONF field in the Error and Status Register is updated to be 'Error Active' and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the Tx\_Err\_Counter value.
- If during system start-up, only one node is operating, then its Tx\_Err\_Counter increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the ACK\_ERR bit in the Error and Status Register). After the transition to 'Error Passive' state, the Tx\_Err\_Counter does not increment anymore by acknowledge errors. Therefore the device never goes to the 'Bus Off' state.
- If the Rx\_Err\_Counter increases to a value greater than 127, it is not incremented further, even if more errors are detected while being a receiver. At the next

successful message reception, the counter is set to a value between 119 and 127 to resume to 'Error Active' state.

Address: Base address + 1Ch offset

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

### FLEXCANx\_ECR field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 Rx_Err_Counter	Rx_Err_Counter
Tx_Err_Counter	Tx_Err_Counter

## 26.8.8 Error and Status 1 Register (FLEXCANx\_ESR1)

This register reflects various error conditions, some general status of the device and it is the source of four interrupts to the ARM.

The ARM read action clears bits 15-10, therefore the reported *error conditions*(bits 15-10) are those that occurred since the last time the ARM read this register. Bits 9-3 are status bits.

Some bits in this register are read-only and some are not.

**Table 26-54. FlexCAN State**

SYNCH	IDLE	TX	RX	FlexCAN state
0	0	0	0	Not synchronized to CAN bus
1	1	x	x	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving
other combinations				Reserved

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved													SYNCH	TWRN_INT	RWRN_INT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BIT1_ERR	BIT0_ERR	ACK_ERR	CRC_ERR	FRM_ERR	STF_ERR	TX_WRN	RX_WRN	IDLE	TX	FLT_CONF	RX		BOFF_INT	ERR_INT	WAK_INT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**FLEXCANx\_ESR1 field descriptions**

Field	Description
31–19 -	This field is reserved. Reserved
18 SYNCH	This read-only flag indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. Refer to <a href="#">Table 26-29</a>  1 FlexCAN is synchronized to the CAN bus 0 FlexCAN is not synchronized to the CAN bus
17 TWRN_INT	If the WRN_EN bit in MCR is asserted, the TWRN_INT bit is set when the TX_WRN flag transition from '0' to '1', meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control Register (TWRN_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to '1'. When WRN_EN is negated, this flag is masked. ARM must clear this flag before disabling the bit. Otherwise it will be set when the WRN_EN is set again. Writing '0' has no effect. This flag is not generated during "Bus Off" state. This bit is not updated during Freeze mode.  1 The Tx error counter transition from < 96 to >= 96 0 No such occurrence

*Table continues on the next page...*



**FLEXCANx\_ESR1 field descriptions (continued)**

Field	Description
16 RWRN_INT	<p>If the WRN_EN bit in MCR is asserted, the RWRN_INT bit is set when the RX_WRN flag transition from '0' to '1', meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control Register (RWRN_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to '1'. When WRN_EN is negated, this flag is masked. ARM must clear this flag before disabling the bit. Otherwise it will be set when the WRN_EN is set again. Writing '0' has no effect. This bit is not updated during Freeze mode.</p> <p>1 The Rx error counter transition from &lt; 96 to &gt;= 96 0 No such occurrence</p>
15 BIT1_ERR	<p>This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message.</p> <p>This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits.</p> <p>1 At least one bit sent as recessive is received as dominant 0 No such occurrence</p>
14 BIT0_ERR	<p>This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message.</p> <p>1 At least one bit sent as dominant is received as recessive 0 No such occurrence</p>
13 ACK_ERR	<p>This bit indicates that an Acknowledge Error has been detected by the transmitter node, i.e., a dominant bit has not been detected during the ACK SLOT.</p> <p>1 An ACK error occurred since last read of this register 0 No such occurrence</p>
12 CRC_ERR	<p>This bit indicates that a CRC Error has been detected by the receiver node, i.e., the calculated CRC is different from the received.</p> <p>1 A CRC error occurred since last read of this register. 0 No such occurrence</p>
11 FRM_ERR	<p>This bit indicates that a Form Error has been detected by the receiver node, i.e., a fixed-form bit field contains at least one illegal bit.</p> <p>1 A Form Error occurred since last read of this register 0 No such occurrence</p>
10 STF_ERR	<p>This bit indicates that a Stuffing Error has been detected.</p> <p>1 A Stuffing Error occurred since last read of this register. 0 No such occurrence.</p>
9 TX_WRN	<p>This bit indicates when repetitive errors are occurring during message transmission.</p> <p>1 TX_Err_Counter ≥ 96 0 No such occurrence</p>
8 RX_WRN	<p>This bit indicates when repetitive errors are occurring during message reception.</p> <p>1 Rx_Err_Counter ≥ 96 0 No such occurrence</p>
7 IDLE	<p>This bit indicates when CAN bus is in IDLE state. Refer to <a href="#">Table 26-29</a>.</p>

Table continues on the next page...

**FLEXCANx\_ESR1 field descriptions (continued)**

Field	Description
	1 CAN bus is now IDLE 0 No such occurrence
6 TX	This bit indicates if FLEXCAN is transmitting a message. Refer to <a href="#">Table 26-29</a> .  1 FLEXCAN is transmitting a message 0 FLEXCAN is receiving a message
5–4 FLT_CONF	If the LOM bit in the Control Register is asserted, after some delay that depends on the CAN bit timing the FLT_CONF field will indicate "Error Passive". The very same delay affects the way how FLT_CONF reflects an update to ECR register by the ARM. It may be necessary up to one CAN bit time to get them coherent again.  Since the Control Register is not affected by soft reset, the FLT_CONF field will not be affected by soft reset if the LOM bit is asserted.  This 2-bit field indicates the Confinement State of the FLEXCAN module, as shown in below:  00 Error Active 01 Error Passive 1x Bus off
3 RX	This bit indicates if FlexCAN is receiving a message. Refer to <a href="#">Table 26-29</a> .  1 FLEXCAN is transmitting a message 0 FLEXCAN is receiving a message
2 BOFF_INT	This bit is set when FLEXCAN enters 'Bus Off' state. If the corresponding mask bit in the Control Register (BOFF_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to '1'. Writing '0' has no effect.  1 FLEXCAN module entered 'Bus Off' state 0 No such occurrence
1 ERR_INT	This bit indicates that at least one of the Error Bits (bits 15-10) is set. If the corresponding mask bit in the Control Register (ERR_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to '1'. Writing '0' has no effect.  1 Indicates setting of any Error Bit in the Error and Status Register 0 No such occurrence
0 WAK_INT	When FLEXCAN is Stop Mode and a recessive to dominant transition is detected on the CAN bus and if the WAK_MSK bit in the MCR Register is set, an interrupt is generated to the ARM. This bit is cleared by writing it to '1'. When SLF_WAK is negated, this flag is masked. ARM must clear this flag before disabling the bit. Otherwise it will be set when the SLF_WAK is set again. Writing '0' has no effect  1 Indicates a recessive to dominant transition received on the CAN bus when the FLEXCAN module is in Stop Mode 0 No such occurrence

## 26.8.9 Interrupt Masks 2 Register (FLEXCANx\_IMASK2)

This register allows any number of a range of 32 Message Buffer Interrupts to be enabled or disabled. It contains one interrupt mask bit per buffer, enabling the ARM to determine which buffer generates an interrupt after a successful transmission or reception (i.e. when the corresponding IFLAG2 bit is set).

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF63M_BUF32M																															
W																																
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

### FLEXCANx\_IMASK2 field descriptions

Field	Description
BUF63M_BUF32M	Each bit enables or disables the respective FLEXCAN Message Buffer (MB32 to MB63) Interrupt. Setting or clearing a bit in the IMASK2 Register can assert or negate an interrupt request, if the corresponding IFLAG2 bit is set.  1 The corresponding buffer Interrupt is enabled 0 The corresponding buffer Interrupt is disabled

## 26.8.10 Interrupt Masks 1 Register (FLEXCANx\_IMASK1)

This register allows to enable or disable any number of a range of 32 Message Buffer Interrupts. It contains one interrupt mask bit per buffer, enabling the ARM to determine which buffer generates an interrupt after a successful transmission or reception (i.e., when the corresponding IFLAG1 bit is set).

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF31M_BUF0M																															
W																																
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

### FLEXCANx\_IMASK1 field descriptions

Field	Description
BUF31M_BUF0M	Each bit enables or disables the respective FLEXCAN Message Buffer (MB0 to MB31) Interrupt. Setting or clearing a bit in the IMASK1 Register can assert or negate an interrupt request, if the corresponding IFLAG1 bit is set

**FLEXCANx\_IMASK1 field descriptions (continued)**

Field	Description
1	The corresponding buffer Interrupt is enabled
0	The corresponding buffer Interrupt is disabled

**26.8.11 Interrupt Flags 2 Register (FLEXCANx\_IFLAG2)**

This register defines the flags for 32 Message Buffer interrupts. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG2 bit. If the corresponding IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing it to '1'. Writing '0' has no effect. Before updating MCR[MAXMB] field, ARM must treat the IFLAG2 bits which MB value is greater than the MCR[MAXMB] to be updated, otherwise they will keep set and be inconsistent with the amount of MBs available.

Address: Base address + 2Ch offset

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

**FLEXCANx\_IFLAG2 field descriptions**

Field	Description
BUF63I_BUF32I	Each bit flags the respective FLEXCAN Message Buffer (MB32 to MB63) interrupt.
1	The corresponding buffer has successfully completed transmission or reception
0	No such occurrence

**26.8.12 Interrupt Flags 1 Register (FLEXCANx\_IFLAG1)**

This register defines the flags for 32 Message Buffer interrupts and FIFO interrupts. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit. If the corresponding IMASK1 bit is set, an interrupt will be generated. The Interrupt flag must be cleared by writing it to '1'. Writing '0' has no effect.

When the RFEN bit in the MCR is set (Rx FIFO enabled), the function of the 8 least significant interrupt flags (BUF7I - BUF0I) is changed to support the FIFO operation. BUF7I, BUF6I and BUF5I indicate operating conditions of the FIFO, while BUF4I to BUF0I are not used. Before enabling the RFEN, ARM must service the IFLAG1 asserted

in the Rx FIFO region (see [Rx FIFO](#)). Otherwise, these IFLAGS will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When the RFEN is negated, the FIFO flags must be cleared. The same care must be taken when a RFFN value is selected extending Rx FIFO filters beyond MB7 (see [Control 2 Register \(FLEXCAN\\_CTRL2\)](#)). For example, when RFFN is 0x8, the MB0-23 range is occupied by Rx FIFO filters and related IFLAGS must be cleared.

Before updating MCR[MAXMB] field, ARM must service the IFLAG1 which MB value is greater than the MCR[MAXMB] to be updated, otherwise they will keep set and be inconsistent with the amount of MBs available.

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF31I_BUF8I															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF31I_BUF8I								BUF7I	BUF6I	BUF5I	BUF4I_BUF0I				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

### FLEXCANx\_IFLAG1 field descriptions

Field	Description
31–8 BUF31I_BUF8I	Each bit flags the respective FLEXCAN Message Buffer (MB8 to MB31) interrupt.  1 The corresponding MB has successfully completed transmission or reception 0 No such occurrence
7 BUF7I	If the Rx FIFO is not enabled, this bit flags the interrupt for MB7.  If the MCR[RFEN] bit is asserted, this flag indicates that a message was lost because Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox.  This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by ARM writes.  1 MB7 completed transmission/reception or FIFO overflow 0 No such occurrence
6 BUF6I	If the Rx FIFO is not enabled, this bit flags the interrupt for MB6.  If the MCR[RFEN] bit is asserted, this flag indicates when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. Note that if the flag is cleared while the number of unread messages is greater than 4 it will not assert again until the number of unread messages within the Rx FIFO is decreased to equal or less than 4.  This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by ARM writes.

Table continues on the next page...

**FLEXCANx\_IFLAG1 field descriptions (continued)**

Field	Description
	1 MB6 completed transmission/reception or FIFO almost full 0 No such occurrence
5 BUF5I	If the Rx FIFO is not enabled, this bit flags the interrupt for MB5. If the Rx FIFO is enabled, this flag indicates that at least one frame is available to be read from the Rx FIFO.  This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by ARM writes.  1 MB5 completed transmission/reception or frames available in the FIFO 0 No such occurrence
BUF4I_BUF0I	If the Rx FIFO is not enabled, these bits flag the interrupts for MB0 to MB4. If the Rx FIFO is enabled, these flags are not used and must be considered as reserved locations.  These flags are cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by ARM writes.  1 Corresponding MB completed transmission/reception 0 No such occurrence

**26.8.13 Control 2 Register (FLEXCANx\_CTRL2)**

This register contains control bits for CAN errors, FIFO features and mode selection.

**Table 26-60. Rx FIFO Filters**

RFFN[3:0]	Number of Rx FIFO filters	Message Buffers occupied by Rx FIFO and ID Filter Table	Remaining Available Mailboxes <sup>1</sup>	Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks <sup>2</sup>	Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask <sup>2</sup>
0x0	8	MB 0-7	MB 8-63	Elements 0-7	none
0x1	16	MB 0-9	MB 10-63	Elements 0-9	Elements 10-15
0x2	24	MB 0-11	MB 12-63	Elements 0-11	Elements 12-23
0x3	32	MB 0-13	MB 14-63	Elements 0-13	Elements 14-31
0x4	40	MB 0-15	MB 16-63	Elements 0-15	Elements 16-39
0x5	48	MB 0-17	MB 18-63	Elements 0-17	Elements 18-47
0x6	56	MB 0-19	MB 20-63	Elements 0-19	Elements 20-55
0x7	64	MB 0-21	MB 22-63	Elements 0-21	Elements 22-63
0x8	72	MB 0-23	MB 24-63	Elements 0-23	Elements 24-71
0x9	80	MB 0-25	MB 26-63	Elements 0-25	Elements 26-79
0xA	88	MB 0-27	MB 28-63	Elements 0-27	Elements 28-87
0xB	96	MB 0-29	MB 30-63	Elements 0-29	Elements 30-95
0xC	104	MB 0-31	MB 32-63	Elements 0-31	Elements 32-103
0xD	112	MB 0-33	MB 34-63	Elements 0-31	Elements 32-111
0xE	120	MB 0-35	MB 36-63	Elements 0-31	Elements 32-119
0xF	128	MB 0-37	MB 38-63	Elements 0-31	Elements 32-127

1. The number of the last remaining available mailboxes is defined by the MCR[MAXMB] field.

2. If Rx Individual Mask Registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.

Each group of eight filters occupies a memory space equivalent to two Message Buffers which means that the more filters are implemented the less Mailboxes will be available.

Considering that the Rx FIFO occupies the memory space originally reserved for MB0-5, RFFN should be programmed with a value corresponding to a number of filters not greater than the number of available memory words which can be calculated as follows:

$$(\text{SETUP\_MB} - 6) \times 4$$

where SETUP\_MB is MAXMB.

The number of remaining Mailboxes available will be:

$$\text{SETUP\_MB} - 8 - (\text{RFFN} \times 2)$$

If the Number of Rx FIFO Filters programmed through RFFN exceeds the SETUP\_MB value, the exceeding ones will not be functional. Unshaded regions in [Table 26-36](#) indicate the valid combinations of MAXMB, RFEN and RFFN, shaded regions are not functional.

**Table 26-61. Valid Combinations of MAXMB, RFEN and RFFN**

RFF N	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RFE N	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MAX MB																	
0 - 6																	
7 - 8																	
9 - 10																	
11 - 12																	
13 - 14																	
15 - 16																	
17 - 18																	
19 - 20																	
21 - 22																	

*Table continues on the next page...*

**Table 26-61. Valid Combinations of MAXMB, RFEN and RFFN (continued)**

RFFN	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RFFEN	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
MAXMB																	
23 - 24																	
25 - 26																	
27 - 28																	
29 - 30																	
31 - 32																	
33 - 34																	
35 - 36																	
37 - 63																	

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R				WRMFRZ	RFEN				TASD					MRP	RRS	EACEN
W	0	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**FLEXCANx\_CTRL2 field descriptions**

Field	Description
31 -	must be written as 0
30-29 -	This field is reserved. Reserved

*Table continues on the next page...*



**FLEXCANx\_CTRL2 field descriptions (continued)**

Field	Description
28 WRMFRZ	<p>Enable unrestricted write access to FlexCAN memory in Freeze mode. This bit can only be written in Freeze mode and has no effect out of Freeze mode.</p> <p>1 Enable unrestricted write access to FlexCAN memory 0 Keep the write access restricted in some regions of FlexCAN memory</p>
27–24 RFEN	<p>This 4-bit field defines the number of Rx FIFO filters according to <a href="#">Table 26-35</a>. The maximum selectable number of filters is determined by the ARM. This field can only be written in Freeze mode as it is blocked by hardware in other modes. RFFN defines a number of Message Buffers occupied by Rx FIFO and ID Filter (see <a href="#">Table 26-35</a>) that <b>may not exceed</b> the number of available Mailboxes present in module, defined by MCR[MAXMB]. Default RFFN value is 0x0, which leads to a total of 8 Rx FIFO filters, occupies the first 8 Message Buffers (MB 0-7) and makes available the next Message Buffers (MB 8-63) for Mailboxes. As a second example, when RFFN is set to 0xD, there will be 112 Rx FIFO filters, located in MB 0-33, and MB 34-63 are available for Mailboxes. Notice that, in this case, individual masks (RXIMR) will just cover Rx FIFO filters in 0-31 range, and filters 32-111 will use RXFGMASK. In case of reducing the number of last Message Buffers, MCR[MAXMB] (see <a href="#">Module Configuration Register (FLEXCAN_MCR)</a>) can be adjusted by the application to minimum of 33, in order to give room to the Rx FIFO and its ID Filter Table defined by RFFN. On the contrary, if the application sets MCR[MAXMB] to 16, for instance, the maximum RFFN is limited to 0x4. RFFN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in <a href="#">Table 26-18</a> (see <a href="#">Arbitration and Matching Timing</a>).</p>
23–19 TASD	<p>This 5-bit field indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus. This field can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>This field is useful to optimize the transmit performance based on factors such as: peripheral/serial clock ratio, CAN bit timing and number of MBs. The duration of an arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and CAN baud rate and inversely proportional to the peripheral clock frequency.</p> <p>The optimal arbitration timing is that in which the last MB is scanned right before the first bit of the Intermission field of a CAN frame. Therefore, if there are few MBs and the system/serial clock ratio is high and the CAN baud rate is low then the arbitration can be delayed and vice-versa.</p> <p>If TASD is 0 then the arbitration start is not delayed, thus ARM has less time to configure a Tx MB for the next arbitration, but more time is reserved for arbitration. In the other hand, if TASD is 24 then ARM can configure a Tx MB later and less time is reserved for arbitration.</p> <p>If too little time is reserved for arbitration the FlexCAN may be not able to find winner MBs in time to compete with other nodes for the CAN bus. If the arbitration ends too much time before the first bit of Intermission field then there is a chance that ARM reconfigure some Tx MBs and the winner MB is not the best to be transmitted.</p> <p>The reset value is different on various platforms, according to their peripheral clock frequency, number of MBs and target CAN baud rate.</p> <p>The optimal configuration for TASD can be calculated as:</p>

*Table continues on the next page...*

**FLEXCANx\_CTRL2 field descriptions (continued)**

Field	Description
	$TASD = 25 - \left\{ \frac{f_{CANCLK} \times [MAXMB + 3 - (RFEN \times 8) - (RFEN \times RFFN \times 2)] \times 2}{f_{SYS} \times [1 + (PSEG1 + 1) + (PSEG2 + 1)] \times (PRES DIV + 1)} \right\}$ <p>where:</p> <p><math>f_{CANCLK}</math> is the Protocol Engine (PE) Clock in Hz; PE clock is derived from CAN_CLK_ROOT in CCM. See <a href="#">Clock Root Generator</a></p> <p><math>f_{SYS}</math> is the peripheral clock in Hz;</p> <p>MAXMB is the value in CTRL1[MAXMB] field;</p> <p>RFEN is the value in CTRL1[RFEN] bit;</p> <p>RFFN is the value in CTRL2[RFFN] field;</p> <p>PSEG1 is the value in CTRL1[PSEG1] field;</p> <p>PSEG2 is the value in CTRL1[PSEG2] field;</p> <p>PROPSEG is the value in CTRL1[PROPSEG] field;</p> <p>PRES DIV is the value in CTRL1[PRES DIV] field.</p> <p>Please refer to <a href="#">Arbitration process</a> and <a href="#">Protocol Timing</a> for more details.</p>
18 MRP	<p>If this bit is set the matching process starts from the Mailboxes and if no match occurs the matching continues on the Rx FIFO. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Matching starts from Mailboxes and continues on Rx FIFO 0 Matching starts from Rx FIFO and continues on Mailboxes</p>
17 RRS	<p>If this bit is asserted Remote Request Frame is submitted to a matching process and stored in the corresponding Message Buffer in the same fashion of a Data Frame. No automatic Remote Response Frame will be generated.</p> <p>If this bit is negated the Remote Request Frame is submitted to a matching process and an automatic Remote Response Frame is generated if a Message Buffer with CODE=0b1010 is found with the same ID.</p> <p>This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Remote Request Frame is stored 0 Remote Response Frame is generated</p>
16 EACEN	<p>This bit controls the comparison of IDE and RTR bits within Rx Mailboxes filters with their corresponding bits in the incoming frame by the matching process. This bit does not affect matching for Rx FIFO. This bit can only be written in Freeze mode as it is blocked by hardware in other modes.</p> <p>1 Enables the comparison of both Rx Mailbox filter's IDE and RTR bit with their corresponding bits within the incoming frame. Mask bits do apply. 0 Rx Mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits.</p>
-	This field is reserved.

*Table continues on the next page...*

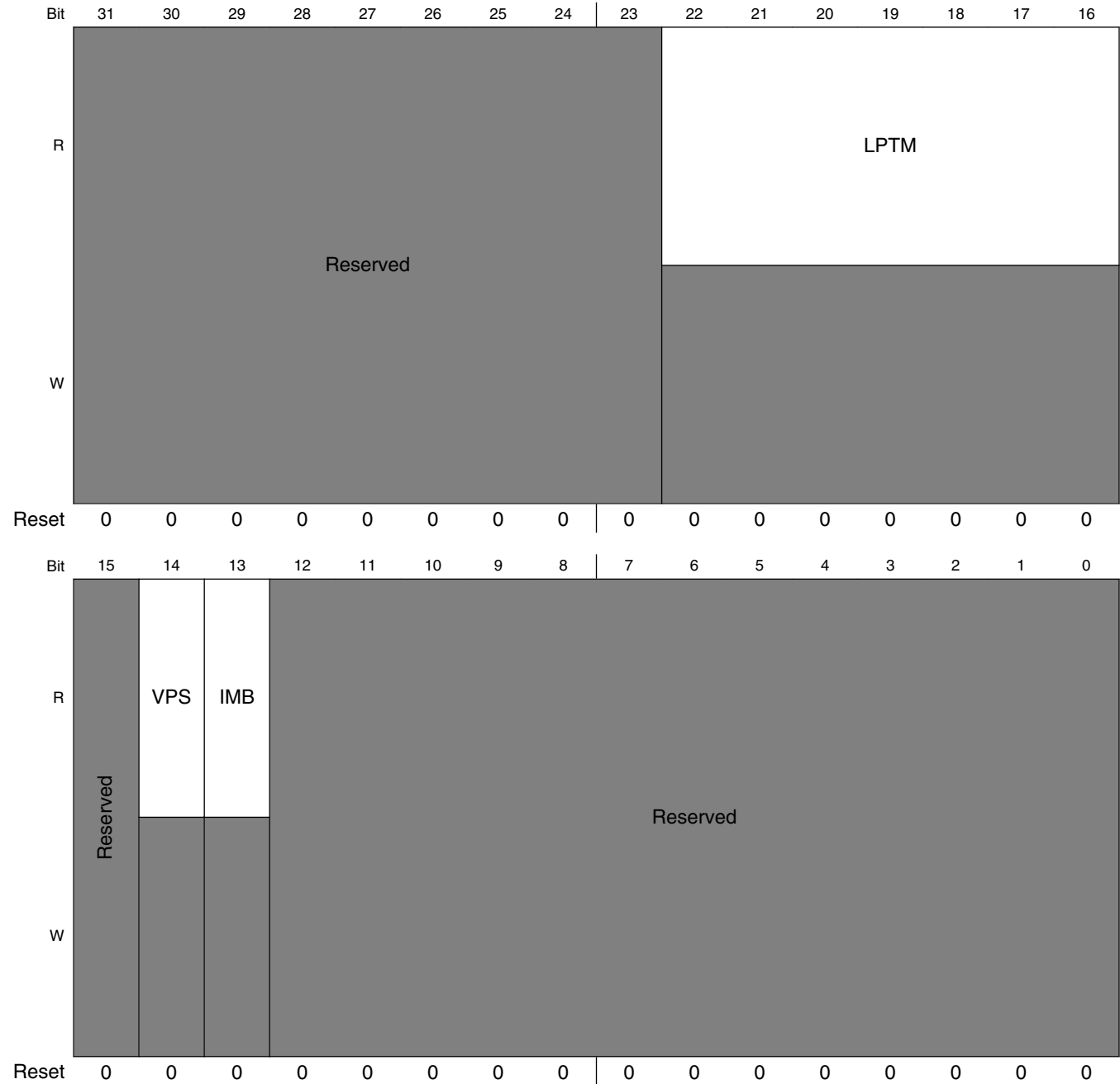
**FLEXCANx\_CTRL2 field descriptions (continued)**

Field	Description
	Reserved

## 26.8.14 Error and Status 2 Register (FLEXCANx\_ESR2)

This register reflects various interrupt flags and some general status.

Address: Base address + 38h offset



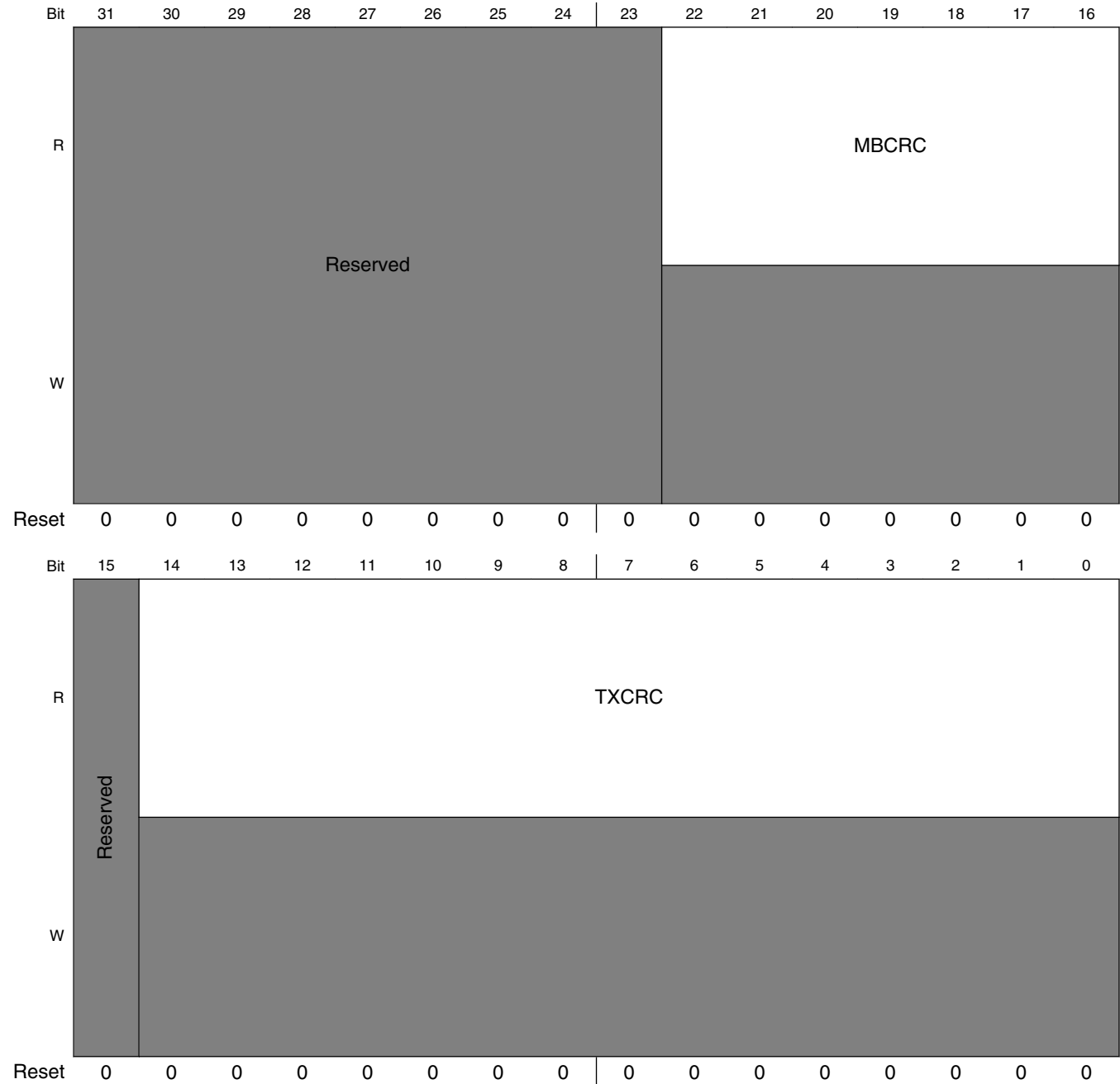
## FLEXCANx\_ESR2 field descriptions

Field	Description
31–23 -	This field is reserved. Reserved
22–16 LPTM	If ESR2[VPS] is asserted, this 7-bit field indicates the lowest number inactive Mailbox (refer to IMB bit description). If there is no inactive Mailbox then the Mailbox indicated depends on CTRL1[LBUF] bit value. If CTRL1[LBUF] bit is negated then the Mailbox indicated is the one which has the greatest arbitration value (see <a href="#">Highest Mailbox priority first</a> ). If CTRL1[LBUF] bit is asserted then the Mailbox indicated is the highest number active Tx Mailbox. If a Tx Mailbox is being transmitted it is not considered in LPTM calculation. If ESR2[IMB] is not asserted and a frame is transmitted successfully, LPTM is updated with its Mailbox number.
15 -	This field is reserved. Reserved
14 VPS	This bit indicates whether IMB and LPTM contents are currently valid or not. VPS is asserted upon every complete Tx arbitration process unless the ARM writes to Control and Status word of a Mailbox that has already been scanned (i.e. it is behind Tx Arbitration Pointer) during the Tx arbitration process. If there is no inactive Mailbox and only one Tx Mailbox which is being transmitted then VPS is not asserted. VPS is negated upon the start of every Tx arbitration process or upon a write to Control and Status word of any Mailbox. ESR2[VPS] is not affected by any ARM write into Control Status (C/S) of a MB which is blocked by abort mechanism. When MCR[AEN] is asserted, the abort code write in C/S of a MB that is been transmitted (pending abort), or any write attempt into a Tx MB with IFLAG set is blocked.  1 Contents of IMB and LPTM are valid 0 Contents of IMB and LPTM are invalid
13 IMB	If ESR2[VPS] is asserted, this bit indicates whether there is any inactive Mailbox (CODE field is either 0b1000 or 0b0000).  This bit is asserted in the following cases: (1) During arbitration, if a LPTM is found and it is inactive. (2) If IMB is not asserted and a frame is transmitted successfully. (3) This bit is cleared in all start of arbitration (see <a href="#">Arbitration process</a> ).  LPTM mechanism have the following behavior: if a MB is successfully transmitted and ESR2[IMB]=0 (no inactive Mailbox), then ESR2[VPS] and ESR2[IMB] are asserted and the index related to the MB just transmitted is loaded into ESR2[LPTM].  1 If ESR2[VPS] is asserted, there is at least one inactive Mailbox. LPTM content is the number of the first one. 0 If ESR2[VPS] is asserted, the ESR2[LPTM] is not an inactive Mailbox.
-	This field is reserved. Reserved

## 26.8.15 CRC Register (FLEXCANx\_CRCR)

This register provides information about the CRC of transmitted messages

Address: Base address + 44h offset



**FLEXCANx\_CRCCR field descriptions**

Field	Description
31–23 -	This field is reserved. Reserved
22–16 MBCRC	This field indicates the number of the Mailbox corresponding to the value in TXCRC field.
15 -	This field is reserved. Reserved
TXCRC	This field indicates the CRC value of the last message transmitted. This field is updated at the same time the Tx Interrupt Flag is asserted.

**26.8.16 Rx FIFO Global Mask Register (FLEXCANx\_RXFGMASK)**

If Rx FIFO is enabled RXFGMASK is used to mask the Rx FIFO ID Filter Table elements that do not have a corresponding RXIMR according to CTRL2[RFFN] field setting.

This register can only be written in Freeze Mode as it is blocked by hardware in other modes.

**Table 26-65. Rx FIFO Global Mask usage**

Rx FIFO ID Filter Table Elements Format (MCR[IDAM])	Identifier Acceptance Filter fields					
	RTR	IDE	RXIDA	RXIDB	RXIDC	reserved
A	FGM[31]	FGM[30]	FGM[29:1]	-	-	FGM[0]
B	FGM[31] FGM[15]	FGM[30] FGM[14]	-	FGM[29:16] FGM[13:0] <sup>1</sup>	FGM[31:24] FGM[23:16] FGM[15:8] FGM[7:0] <sup>2</sup>	-
C	-	-		-		

1. If MCR[IDAM] field is equivalent to the format B only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.
2. If MCR[IDAM] field is equivalent to the format C only the eight most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FGM31_FGM0																															
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

**FLEXCANx\_RXFGMASK field descriptions**

Field	Description
FGM31_FGM0	These bits mask the ID Filter Table elements bits in a perfect alignment. <a href="#">Rx FIFO Global Mask Register (FLEXCAN_RXFGMASK)</a> shows in detail which FGM bits mask each IDAF field. Clear this register has the effect of disabling the ID Filter.  1 The corresponding bit in the filter is checked 0 The corresponding bit in the filter is "don't care"

**26.8.17 Rx FIFO Information Register (FLEXCANx\_RXFIR)**

RXFIR provides information on Rx FIFO.

This register is the port through which ARM accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Rx FIFO as well as its output is updated whenever the output of the Rx FIFO is updated with the next message. Refer to [Rx FIFO](#) to find instructions on reading this register.

Address: Base address + 4Ch offset

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

**FLEXCANx\_RXFIR field descriptions**

Field	Description
31–9 -	This field is reserved. Reserved
IDHIT	This 9-bit field indicates which Identifier Acceptance Filter (see <a href="#">Rx FIFO Structure</a> ) was hit by the received message that is in the output of the Rx FIFO. (refer to <a href="#">Rx FIFO</a> for details) If multiple filters match the incoming message ID then the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only while the IFLAG[BUF5I] is asserted.



## 26.8.18 Rx Individual Mask Registers (FLEXCANx\_RXIMR0\_RXIMR63)

RXIMR are used as acceptance masks for ID filtering in Rx MBs and the Rx FIFO. If the Rx FIFO is not enabled, one mask register is provided for each available Mailbox, providing ID masking capability on a per Mailbox basis.

When the Rx FIFO is enabled (MCR[RFEN] bit is asserted), up to 32 Rx Individual Mask Registers can apply to the Rx FIFO ID Filter Table elements on a one-to-one correspondence depending on CTRL2[RFFN] setting. Refer to [Control 2 Register \(FLEXCAN\\_CTRL2\)](#) for details.

RXIMR can only be written by the ARM while the module is in Freeze Mode, otherwise they are blocked by hardware.

The Individual Rx Mask Registers are not affected by reset and must be explicitly initialized prior to any reception.

Address: Base address + 880h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																	MI31_MIO															
W																																
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

### FLEXCANx\_RXIMR0\_RXIMR63 field descriptions

Field	Description
MI31_MIO	<p>These bits mask both Mailbox filter and Rx FIFO ID Filter Table element in distinct ways.</p> <p>For Mailbox filter refer to <a href="#">Rx Mailboxes Global Mask Register (FLEXCAN_RXMGMASK)</a>.</p> <p>For Rx FIFO ID Filter Table element refer to <a href="#">Rx FIFO Global Mask Register (FLEXCAN_RXFGMASK)</a>.</p> <p>1 The corresponding bit in the filter is checked 0 the corresponding bit in the filter is "don't care"</p>

## 26.8.19 Glitch Filter Width Registers (FLEXCANx\_GFWR)

The Glitch Filter just takes effects when FLEXCAN enters the STOP mode.

Address: Base address + 9E0h offset

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

**FLEXCANx\_GFWR field descriptions**

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
GFWR	It determines the Glitch Filter Width. The width will be divided from Oscillator clock by GFWR values. By default, it is 5.33us when the oscillator is 24MHz.

