32-bit RISC Microcontroller

TXZ+ Family

Reference Manual
CAN Controller
(CAN-B)

Revision 1.0

2020-10

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related document

Document Name
Clock Control and Operation Mode
Input/Output ports
Product Information

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> =0x01 (hexadecimal), ***[XYZn]***<VW> =1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value.
  In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

CAN	Controller Area Network

1. Outlines

The CAN can operate as a transmission/ reception circuit of 1channel per unit.
The following shows the List of Function.

Function classification	Function	Operation explanation
Protocol	Compliant with CAN version 2.0B active	Standard and extended formats supported
	CAN bus bit rate	Max 1Mbps (fsys=48MHz(Min)) Baud rate prescaler built in
	Bit timing parameter	Equivalent to Intel 82527
Message Box	32 Mailboxes	31 reception and transmission 1 reception only
Transmission	Frame support	Data Frame / Remote Frame supported
	Arbitration	The mailbox with the lower number will be sent first The mailbox with the higher priority identifier will be sent first
Reception	Frame support	Data Frame / Remote Frame supported
	Masking function	Programmable global receive mask (common to mailboxes 0 to 30)
		Programmable local receive mask (for mailbox 31 only)
	ID format	Standard / Extended ID selection Receive Mask bit function
Time stamp	For Transmission/Reception	16-bit time stamp counter
Operation Mode	Configuration mode	Mode for setting CAN
	Sleep mode	Low power consumption mode(Bus Sleep Mode)
	Suspend mode	Inactive state on the CAN bus
	Test loop-back mode	Self-acknowledge
	Test error mode	Writable error counters
Interrupt	CANx transmission completion (INTCANxTXD)	Interrupt when a message transmission is terminated in normal
	CANx reception completion (INTCANxRXD)	Interrupt when a message reception is terminated in normal
	CANx global (INTCANxGLB) (8 factors)	Warning level / Error passive / Bus off / Time stamp overflow / Transmit abort / Receive message lost / Wake-up / Remote frame receive

2. Block Diagram

Figure 2.1 shows the Block Diagram of CAN controller

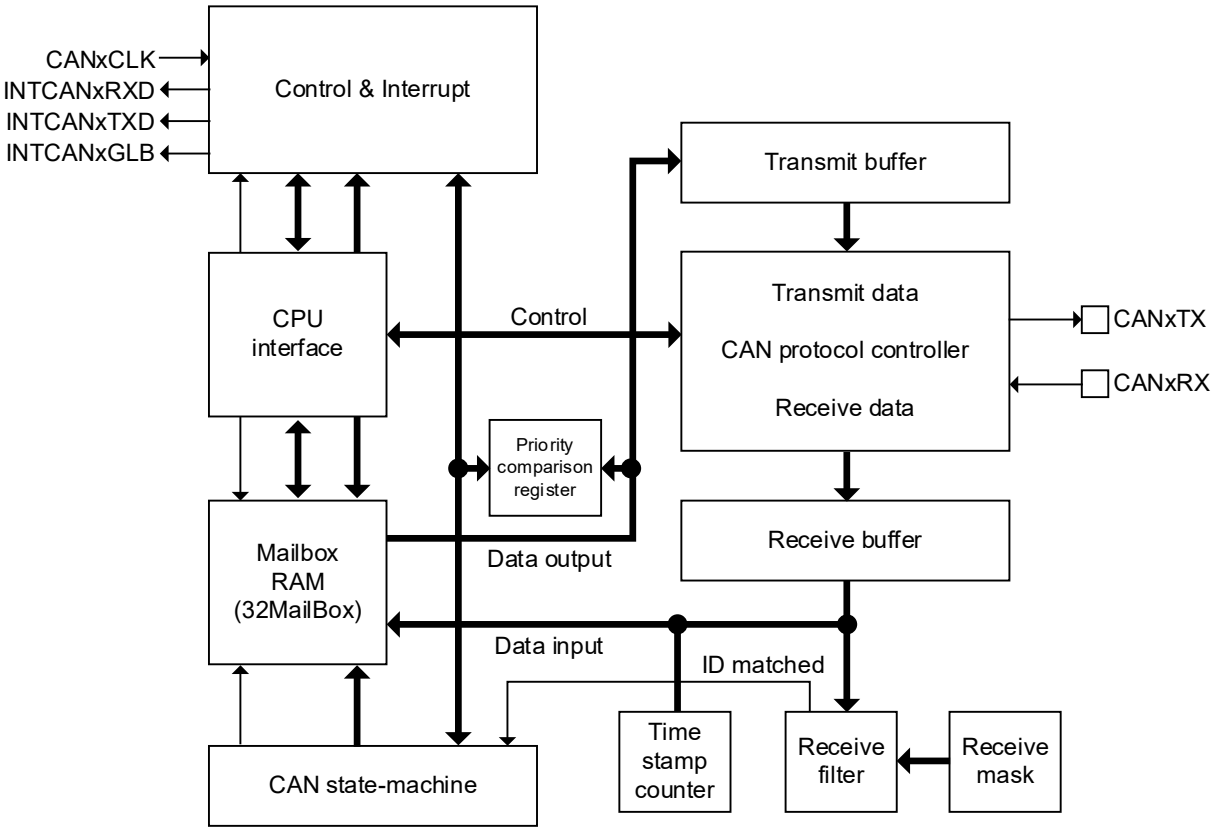

Figure 2.1 Block Diagram of CAN controller

Table 2.1 List of Signals

No	Symbol	Signal name	I/O	Related Reference Manual
1	INTCANxRXD	CANx reception completion interrupt	Output	Exception
2	INTCANxTXD	CANx transmission completion interrupt	Output	Exception
3	INTCANxGLB	CANx global interrupt	Output	Exception
4	CANxTX	CANx TX signal	Output	Input/Output ports, Product information
5	CANxRX	CANx RX signal	Input	Input/Output ports, Product information
6	CANxCLK	CANx clock(f_{sys}/4)	Input	Clock Control and Operation Mode

3. Function and Operation

3.1. Clock Supply

When you use CAN, please set an applicable clock enable bit to "1" (clock supply) in fsys supply stop register A (***[CGFSYSENA]***, ***[CGFSYSMENA]***), fsys supply stop register B (***[CGFSYSENB]***, ***[CGFSYSMENB]***), fsys supply stop register C(***[CGFSYSMENC]***), and fc supply stop registers (***[CGFCEN]***).

An applicable register and the bit position vary according to a product. Therefore, the register may not exist with the product. Please refer to "Clock Control and Operation Mode" of reference manual for the details.

When attempting to stop supplying the clock, make sure to check whether the CAN is stopped. Note that when the MCU enters STOP mode, make sure to check whether the CAN is stopped as well.

When fsys clocks of CAN stop, CANxCLK stops.

3.2. CAN Interface

The interface to the CAN bus are an input pin CANxRX and an output pin CANxTX. Connect these pins via the CAN bus transceiver (ISO / DIS 11898 compliant).

High speed and low speed transceivers are differentiated. Care must be taken to ensure that the electrical characteristics of these pins at the chip level (eg, 3.3 V to 5 V) satisfy the requirements of the transceiver.

3.3. Function

3.3.1. Mailbox

The mailboxes consist of a single port RAM (accessible from the internal CAN core and the CPU). The CPU controls the CAN controller by changing the settings of the mailboxes and control registers. The settings of the mailboxes and control registers are used for such processes as reception filtering, message transmission, and interrupt processing.

To start transmission, set the transmission request bit corresponding to the mailbox to be sent. After the bit has been set, all transmission procedures and error processes (when errors occur) are executed without CPU involvement. When the mailbox is set to receive, the CPU reads the mailbox data using read instructions.

The user can also set it so that an interrupt will be issued to the CPU every time a message has been successfully received or transmitted

In total, 32 mailboxes are provided, each of which consists of 8 byte data, 29 bit IDs, and several control bits. The mailboxes (except mailbox 31) can be set to either transmission or reception. Mailbox 31 is the receive only mailbox. Mailbox 31 is designed so that it can receive different message ID groups using other receive masks than mailboxes 0 to 30.

"Figure 3.1" shows the configuration of the mailboxes.

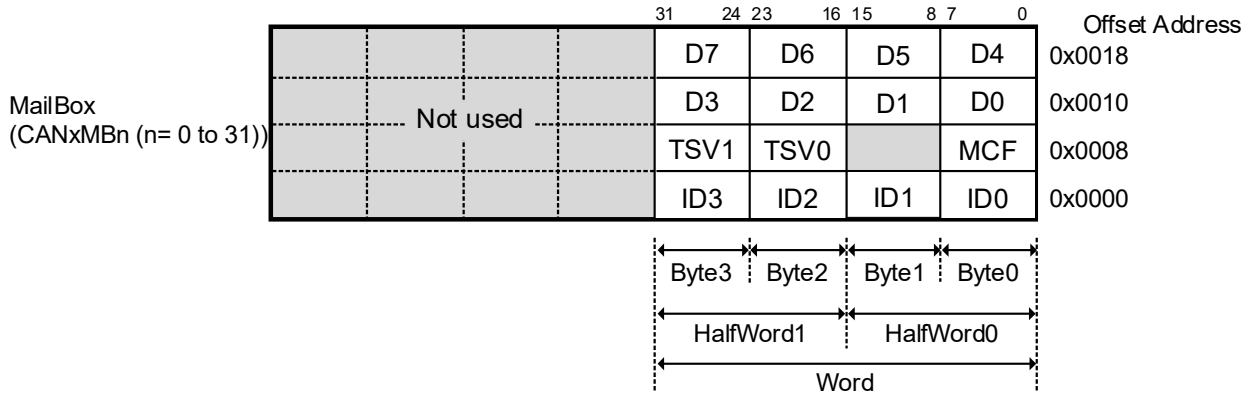

Figure 3.1 Configuration of Mailboxes

(1) Message ID field (ID3 to ID0)
- ID extension bit <IDE>
- Global / local acceptance mask enable bit <GAME_LAME>
- Remote frame handling bit <RFH>
- 29-bit message ID<ID[28:0]>

(2) Message control field (MCF)
- Remote frame transmission request bit <RTR>
- Data length of 4 bits <DLC[3:0]>

(3) Time stamp value (TSV1, TSV0)
- Stores time stamp counter value during receiving / transmitting message <TSV[15:0]>

(4) Data field(D7 to D0)
- Data of 8 bytes <D7[7:0]> to <D0[7:0]>

3.3.2. Transmit Control Register

Transmission control consists of two registers. One is the transmission request set register ***[CANxTRS]***, and the other is the transmission request reset register ***[CANxTRR]***. Therefore it is possible to clear the transmission request without generating a conflict in the handling of the transmission mailboxes in the state-machine.

This mechanism also prevents clearing the transmission request of a mailbox to which transmission is already in progress.

When a write of data and the ID to mailbox n configured as a transmission mailbox (***[CANxMD]***<MDn>=0) is performed and access to mailbox n is enabled (***[CANxMC]***<MCn>=1), setting the ***[CANxTRS]***<TRSn> bit to "1" causes the messages in mailbox n to be transmitted.

If there is more than one mailbox configured as a transmission mailbox and more than one corresponding TRS bit is set, then the messages will be sent in the selected order. The order of transmission depends on the <MTOS> bit in the master control register ***[CANxMCR]***.

If the ***[CANxMCR]***<MTOS> bit is "0", the mailbox with the lower number has the higher priority. For example, if the mailboxes CANxMB0, CANxMB2, and CANxMB5 are configured as transmission mailboxes and the corresponding ***[CANxTRS]***<TRSn> bits are set to "1", then the messages will be transmitted in the following order: CANxMB0, CANxMB2, and CANxMB5. If a new transmission request is set for CANxMB0 during processing of the CANxMB2 message, then in the next internal arbitration-run, CANxMB0 is selected for the next transmission message and transmission of the CANxMB0 message starts after CANxMB2 transmission is completed.

This also happens if an arbitration lost error occurs while the CANxMB2 message is being transmitted.

The CANxMB0 message will be sent instead of the CANxMB2 lost in arbitration.

If the ***[CANxMCR]***<MTOS> bit is "1", the mailbox with the highest priority ID among those mailboxes for which transmission is requested will be transmitted. In a transmission after an arbitration lost error occurred also, the message in the mailbox with the highest priority ID among those mailboxes for which transmission is requested at the time will be transmitted.

3.3.3. Receive Control Register

The ID of a received message is compared to the ID of the mailbox set as the receive mailbox. The comparison of the IDs depends on the ***[CANxMBnID]***<GAME_LAME> values of the global/local acceptance mask enable bit and the data held in the global/local acceptance mask registers ***[CANxGAM]***/***[CANxLAM]***.

When a match is detected, the ID of the received message, the control bits, and data bytes are written in the matching mailbox. At the same time, when the corresponding receive message pending bit ***[CANxRMP]***<RMPn> is set to "1" and the mailbox interrupt is enabled (***[CANxMBIM]***<MBIMn>=1), the CANx reception completion interrupt (INTCANxRXD) occurs. After a match is detected, no further ID comparison takes place.

If the ID of the received message does not match with any of the mailboxes 0 to 30, the ID is compared to the ID of the receive-only mailbox 31. When a match is detected, the settings of the received message are written in receive-only mailbox 31.

If no match is detected, the received message is not stored in the mailbox and no change occurs in the mailbox.

The <RMPn> bit must be cleared by the CPU after data is read. With the <RMPn> bit set to "1", if the next message to this mailbox n is received, the corresponding receive message lost bit <RMLn> is set to "1".

In this case, mailbox n is overwritten with the new message.

Figure 3.2 shows timing when a receive message lost occurs.

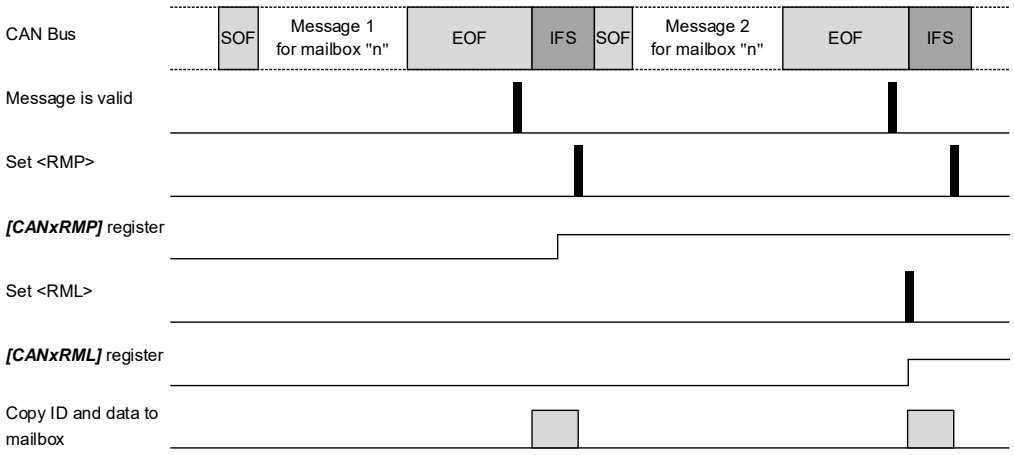

Figure 3.2 Timing when a Receive Message Lost Occurs

3.3.4. Remote Frame Control Register

After a remote frame is received, the remote frame ID is compared to the mailbox ID. The comparison of the IDs depends on the value of the global/local acceptance mask enable bit ***[CANxMBnID]***<GAME_LAME> in the mailbox and the data held in the global/local acceptance mask registers ***[CANxGAM]***/***[CANxLAM]***.

After an ID match is detected, no further comparison takes place.

When the remote frame ID matches with the ID of transmission mailbox n where the remote frame handling bit ***[CANxMBnID]***<RFH> is set to "1", the ***[CANxTRS]***<TRSn> bit is set to "1" so that the message is transmitted responding to the remote frame. For a transmission mailbox where the ***[CANxMBxID]***<RFH> bit is set to "0", the mailbox does not respond to the remote frame even when the ID matches.

If the ID matches with the ID of receive mailbox n, the received message is handled same as a data frame, and this sets the ***[CANxRMP]***<RMPn> bit and the ***[CANxRMP]***<RFPn> bit to "1".

When the remote frame ID matches with the ID of mailbox n where both the ***[CANxMBxID]***<RFH> bit and the global bit are set to "1", the ID of mailbox n is overwritten with the remote frame ID, and the mailbox will automatically respond to the remote frame using the ID (The <TRSn> bit is set to transmit a data frame). Therefore, when the global acceptance mask register ***[CANxGAM]*** is used, one mailbox n may respond to multiple remote frame IDs depending on the mask value.

3.3.5. Receive Filtering

For mailboxes 0 to 30, the global acceptance mask register ***[CANxGAM]*** will be used if the global bit in the mailbox is set. The receiving message will be stored in the first mailbox with a matching ID. Only if there is no matching ID in the mailboxes 0 to 30, the receiving message will be compared to the receive-only mailbox (mailbox 31). If the local bit in mailbox 31 is set, the local acceptance mask register ***[CANxLAM]*** will be used.

Figure 3.3 shows receive filtering.

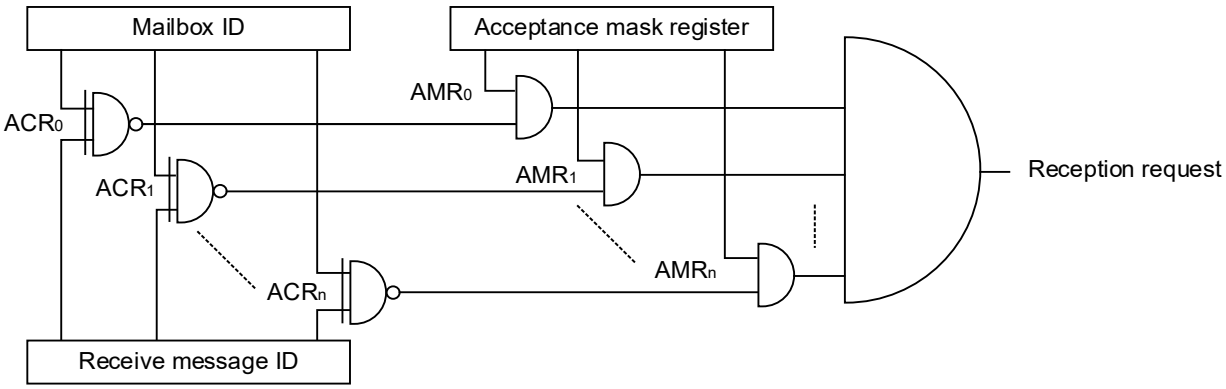

Figure 3.3 Receive filtering

3.3.6. Time Stamp Function

There is a free-running 16-bit time stamp counter ***[CANxTSC]*** implemented in the CAN controller to show the time of message reception and transmission. The content of the ***[CANxTSC]*** is written into the time stamp value (TSV) of the corresponding mailbox when a received message has been stored or a message has been transmitted.

The ***[CANxTSC]*** is driven by the bit clock of the CAN bus line. When the operation mode of the CAN is in configuration mode or in sleep mode, the ***[CANxTSC]*** is stopped. After power-up reset, a write to the time stamp counter prescaler register ***[CANxTSP]*** clears the ***[CANxTSC]*** to "0". The ***[CANxTSC]*** is readable and writable from the CPU both in configuration mode and normal operation mode.

Figure 3.4 shows the structure of the time stamp counter.

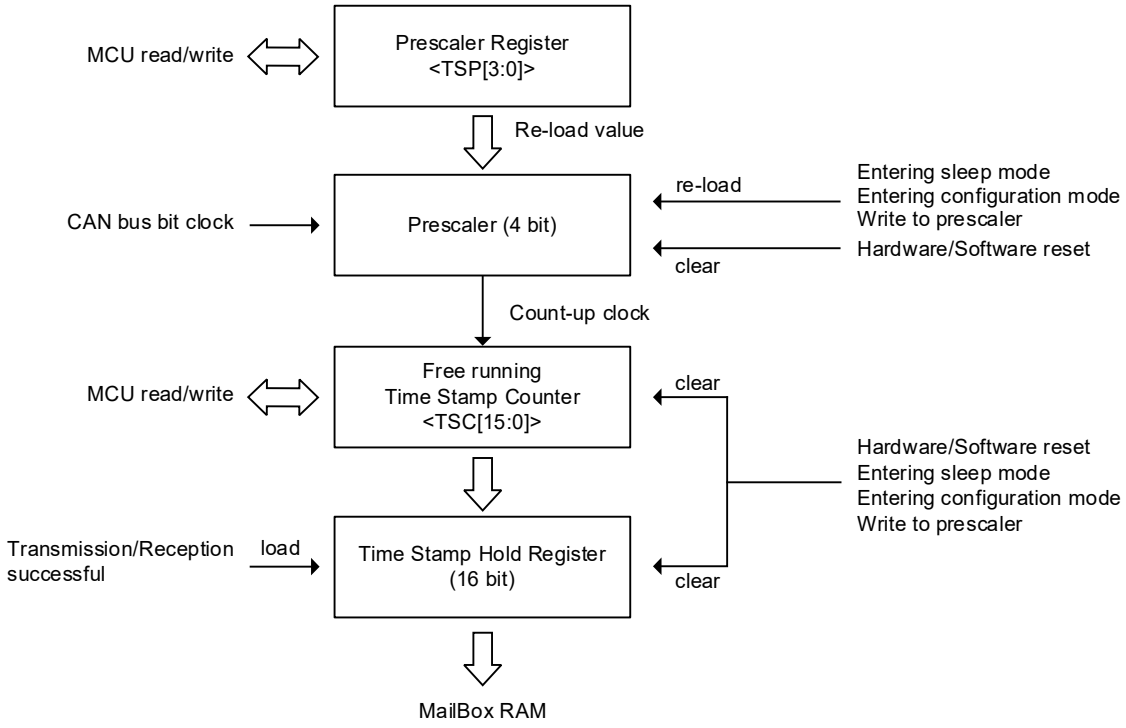

Figure 3.4 Timer Stamp Counter

The free running time stamp counter and the time stamp hold register will be cleared in the following cases:

- After reset (Power on reset or software reset).
- When the controller enters into configuration mode.
- When the controller enters into sleep mode.
- When a write access is performed to the ***[CANxTSP]*** register.

3.3.7. Interrupt Control

The CAN controller has the following interrupt factors. And these interrupt factors are divided into three groups and each group has one interrupt output.

- CANx transmission completion interrupt (INTCANxTXD)
  It occurs at the completion of transmission.
- CANx reception completion interrupt (INTCANxRXD)
  It occurs at the completion of reception.
- CANx global interrupt (INTCANxGLB)
  It occurs by eight factors other than those above.

Table 3.1 List of Interrupt factors

Factors		Group
Transmit interrupt	A message has been transmitted successfully.	INTCANxTXD
Receive interrupt	A message has been received successfully.	INTCANxRXD
Warning level interrupt	At least one of the two error counters is greater than or equal to 97.	INTCANxGLB
Error passive interrupt	CANx enters the error passive mode.	
Bus off interrupt	CANx enters the bus off mode.	
Time stamp overflow interrupt	-	
Transmit abort interrupt	-	
Receive message lost interrupt	-	
Wake-up interrupt	After wake-up from sleep mode, this interrupt will be generated.	
Remote frame receive interrupt	-	

For mailbox interrupts, there are two interrupt output lines separated from global interrupts. These are the mailbox reception completion interrupt (INTCANxRXD) and the mailbox transmission completion interrupt (INTCANxTXD), which are dependent on mailbox settings.

There are two interrupt flag registers and one interrupt mask register. One interrupt flag register is for the mailbox receive interrupt flag register ***[CANxMBRIF]*** and one for the mailbox transmission interrupt flag register ***[CANxMBTIF]***. In addition, there is the mailbox interrupt mask register ***[CANxMBIM]*** for setting whether to enable or disable each mailbox interrupt. The ***[CANxMBIM]*** register is used both for transmission and receive mailboxes.

Figure 3.5 shows the block diagram of CAN interrupt signal.

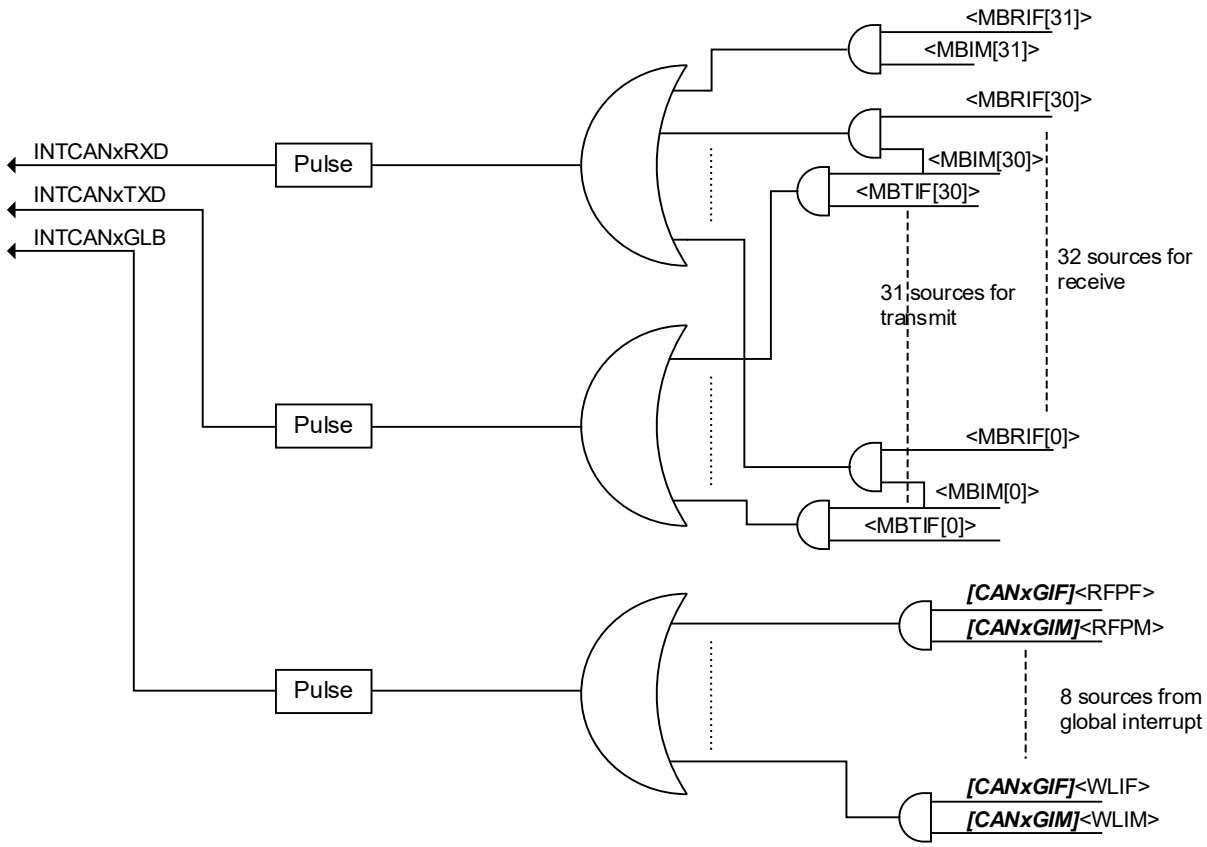

Figure 3.5 Block Diagram of CANx interrupt signals

The CANx reception completion interrupt signal INTCANxRXD is the OR of the signal for which the 32 factors issued by the mailbox receive interrupt flag register ***[CANxMBRIF]*** that are ANDed with each bit of the mailbox interrupt mask register ***[CANxMBIM]***.

The CANx transmission completion interrupt signal INTCANxTXD is the OR of the signal for which the 31 factors issued by the mailbox transmission interrupt flag register ***[CANxMBTIF]*** that are ANDed with each bit of the mailbox interrupt mask register ***[CANxMBIM]***.

The CAN global interrupt signal INTCANxGLB is the OR of the signal for which the 8 factors issued by the global interrupt flag register ***[CANxGIF]*** that are ANDed with each bit of the global interrupt mask register ***[CANxGIM]***.

3.4. Operation Mode

3.4.1. Configuration Mode

The CAN controller needs initial setup before starting operation (setting of the bit configuration registers, ***[CANxBCR1]*** and ***[CANxBCR2]***). Writes to the ***[CANxBCR1]*** and ***[CANxBCR2]*** are possible only when the CAN controller is in configuration mode.

After reset, the ***[CANxMCR]***<CCR> and the ***[CANxGSR]***<CCE> are set to "1" and the configuration mode is set. A write of "0" to the ***[CANxMCR]***<CCR> bit sets the CAN controller to the normal operation mode. After leaving configuration mode, the ***[CANxGSR]***<CCE> bit is cleared to "0" and the power-up sequence starts. The power-up sequence detects 11 consecutive recessive bits on the CAN bus line. After detection, the CAN controller is a bus on and ready for operation.

A write of "1" to the ***[CANxMCR]***<CCR> bit sets the CAN controller to enter configuration mode from normal operation mode. After the CAN controller has entered configuration mode, the ***[CANxGSR]***<CCE> bit is set to "1".

Figure 3.6 shows the flowchart of the initial setup of the CAN controller.

When the CAN controller enters into configuration mode, the CAN error counter ***[CANxCEC]***, the time stamp counter ***[CANxTSC]***, and the time stamp hold registers will be cleared.

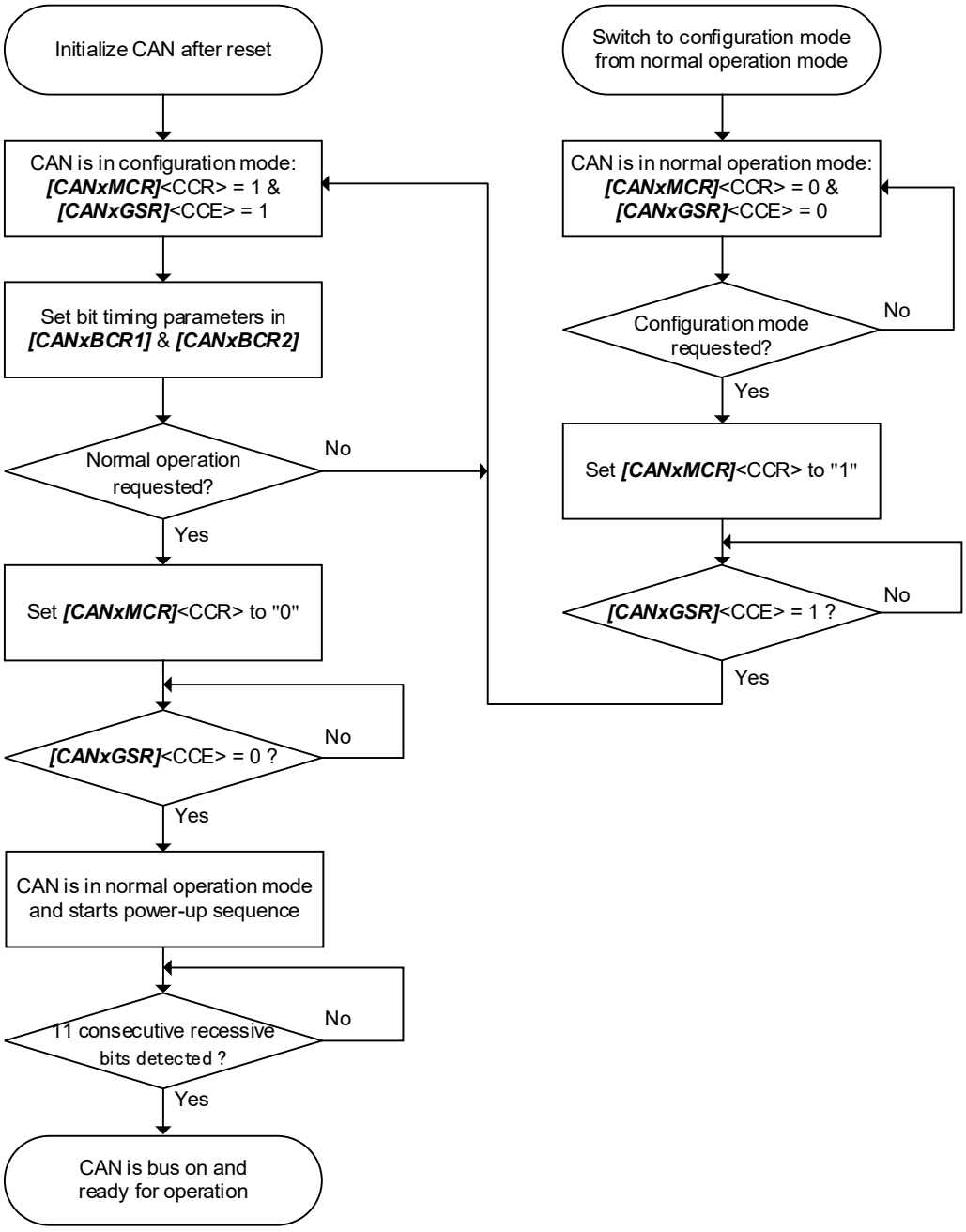

Figure 3.6 Flowchart of Initial Setup of CAN Controller

3.4.2. Sleep Mode

Sleep mode is requested by a write of "1" to the ***[CANxMCR]***<SMR> bit. After the CAN controller has entered into sleep mode, the ***[CANxGSR]***<SMA> bit is set to "1".

The read value of the ***[CANxGSR]*** register is "0x0001F040". This means that there is no message in the transmission buffer and sleep mode is active where the <SMA> bit is "1". Read values to all other registers deliver the value "0". Write accesses to all registers, except the ***[CANxMCR]*** register, will be denied.

The CAN controller cancels sleep mode (wakes up) and starts the power-up sequence if a write access to the ***[CANxMCR]*** register is detected, or there is any bus activity detected on the CAN bus with the ***[CANxMCR]***<WUBA> bit set to "1". The CAN controller waits until detecting 11 consecutive recessive bits on the CANxRX input terminal, after it goes into bus active state. The wake-up message is invalid.

In sleep mode, the CAN error counters and all transmission request set ***[CANxTRS]***<TRSn> bits and transmission request reset ***[CANxTRR]***<TRRn> bits are cleared. The ***[CANxMCR]***<SMR> bit and the ***[CANxGSR]***<SMA> bit are cleared after the CAN controller leaves sleep mode.

If sleep mode is requested while the CAN controller is transmitting a message (***[CANxMCR]***<SMR>=1), the CAN controller enters sleep mode after any of the following occurs:

- The message has been successfully transmitted.
- The message has been successfully transmitted after an arbitration lost error.
- The message has been successfully received after an arbitration lost error.

3.4.3. Suspend Mode

The suspend mode is requested by writing "1" to the ***[CANxMCR]***<SUR> bit. If the CAN bus line is not idle, the current message transmission/reception is completed before suspend mode is activated. After the CAN controller has entered suspend mode, the ***[CANxGSR]***<SUA> bit is set to "1".

In suspend mode, the CAN controller is not active on the CAN bus line. That means neither error frames nor acknowledgment will be sent. The error counters and the ***[CANxGSR]***<EP> bit will not be cleared either.

If suspend mode is requested during the bus off recovery sequence execution, the CAN controller enters suspend mode after the bus off recovery sequence is finished.

To restart the CAN controller, the ***[CANxMCR]***<SUR> bit needs to be programmed to "0". After leaving the bus off state or the inactive state, the CAN controller restarts the bus off recovery sequence.

The CAN controller cancels suspend mode with a write of "0" to the ***[CANxMCR]***<SUR> bit.

3.4.4. Test Loop Back Mode

In test loop back mode, the CAN controller can receive its own transmitted message and generates its own acknowledge bit. No other CAN node is necessary for the operation.

The test loop back mode can be enabled or disabled only when the CAN controller is in suspend mode. In test loop back mode, the CAN controller can transmit a message from a mailbox and receive it in another mailbox.

The setup for mailbox is the same as in normal operation mode.

3.4.5. Test Error Mode

In test error mode, writes to the CAN error counter register ***[CANxCEC]*** are possible. The values of the lower 8 bits are concurrently written to both the transmission error counter (TEC) and the reception error counter (REC). The maximum value that can be written into the error counters is "255". The error counter value of "256" which forces the CAN controller into bus off mode cannot be written.

The test error mode can be enabled or disabled only when the CAN controller is in suspend mode.

Figure 3.7 shows the flowchart of the setup of test loop-back mode and test error mode.

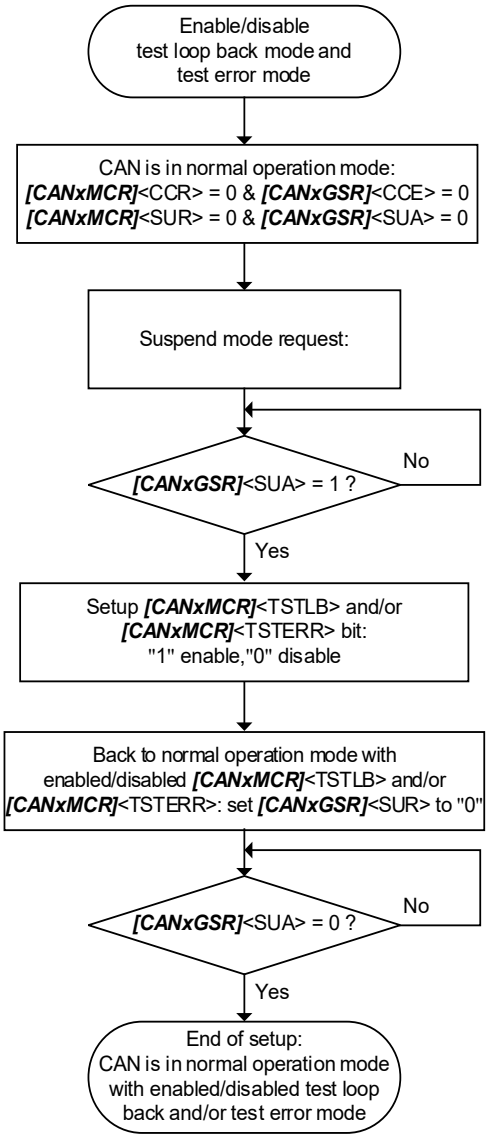

Figure 3.7 Flowchart of Setup of Test Loop Back Mode and Test Error Mode

3.5. Bit Configuration

The bit length is determined by the parameters ***[CANxBCR2]***<TSEG1>, ***[CANxBCR2]***<TSEG2>, and ***[CANxBCR1]***<BRP>. All controllers on the CAN bus must have the same baud rate and bit length. At different clock frequencies of the individual controllers, the baud rate has to be adjusted by the above-mentioned parameters. In the bit timing logic, the conversion of the parameters to the required bit timing is implemented. The configuration registers ***[CANxBCR1]*** and ***[CANxBCR2]*** contain the data about bit timing. Its definition corresponds to the CAN specification 2 (equivalent to Intel 82527).

Figure 3.8 shows CAN bit timing.

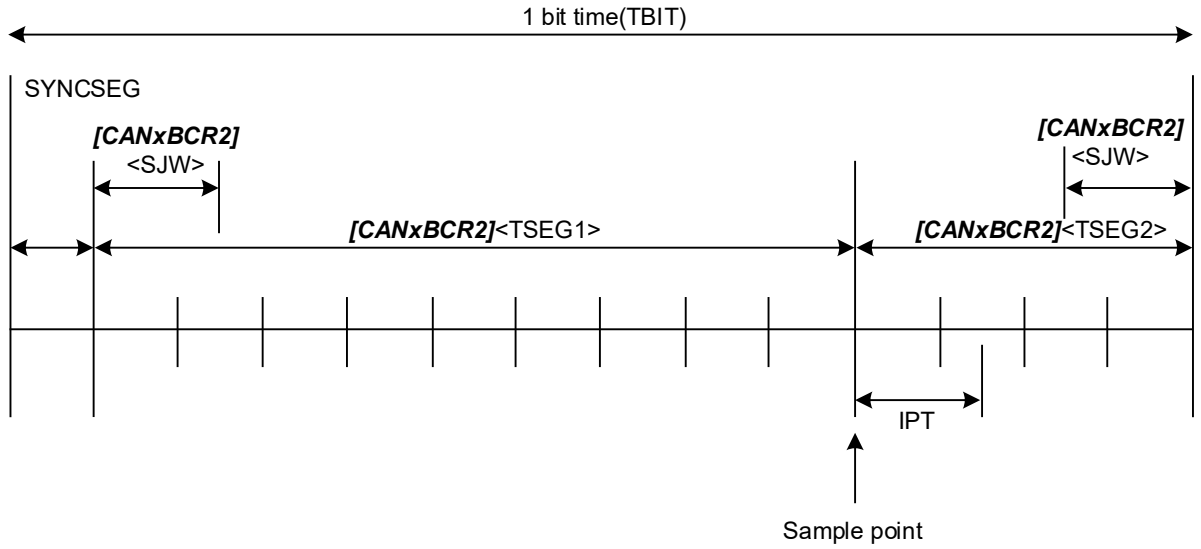

Figure 3.8 CAN Bit Timing

T_{SCL}(CAN system clock) is defined by:

$$T_{SCL} = \frac{\textbf{\textit{[CANxBCR1]}}\text{<BRP[9:0]>}+1}{f_{CANOSC}}$$

$1 \times T_{SCL} = 1 \times T_Q$ (T_Q: time quantum)

f_{CANOSC} is the clock for CAN baud rate generation. The clock obtained by dividing the system clock fsys by 4 is supplied as the clock for CAN baud rate generation. If fsys = 40MHz then f_{CANOSC} = 10MHz.

The synchronization segment SYNCSEG always has the length of one T_Q.

The baud rate is defined by:

$$BR = \frac{1}{((\textbf{\textit{[CANxBCR2]}}\text{<TSEG1[3:0]>}+1)+(\textbf{\textit{[CANxBCR2]}}\text{<TSEG2[2:0]>}+1)+1)\times T_{SCL}}$$

Note: It is not T_Q unit value.

Information processing time (IPT) is the time segment starting with the sample point reserved for processing of the sampled bit level. The information processing time is equal to three CAN system clock cycles.

[CANxBCR2]<SJW[1:0]> indicates how much the time quantum (T_Q) value in bit length is allowed to be lengthened or shortened when resynchronizing. Values between "1" (***[CANxBCR2]***<SJW[1:0]> = 00) and "4" (<SJW[1:0]> = 11) are adjustable.
The bus line is sampled and synchronization is performed at each falling edge of the bus signal within a bit grid.
Set ***[CANxBCR2]***<SJW[1:0]> to a value equal to or smaller than ***[CANxBCR2]***<TSEG2[2:0]>.

Setting the ***[CANxBCR2]***<SAM> bit enables the multiple sampling of the bus line. The level is determined by the result from the majority decision of three sampling values. Sampling is taken at the sample point and the previous last two CAN system clock points. When ***[CANxBCR1]***<BRP[9:0]> is smaller than 4, the sampling performed is always once regardless of the value set in the ***[CANxBCR2]***<SAM> bit.

Table 3.2 shows the restrictions when the baud rate is set.

Table 3.2 Restrictions when Setting the Baud Rate

[CANxBCR1] <BRP[9:0]>	T_Q length (number of CAN clock cycles)	IPT length (number of CAN clock cycles)	Minimum ***[CANxBCR2]***<TSEG2> length (T_Q unit)
0	1	3	3
1	2	3	2
> 1	<BRP[9:0]>+1	3	2

- Restrictions for ***[CANxBCR2]***<TSEG1>
  [CANxBCR2]<TSEG1> ≥ ***[CANxBCR2]***<TSEG2>: The length of <TSEG1> should be equal to or greater than the length of <TSEG2>.
- Restrictions for ***[CANxBCR2]***<SJW>
  [CANxBCR2]<SJW> ≤ ***[CANxBCR2]***<TSEG2>: Set the synchronization jump width to a value equal to or smaller than <TSEG2>.
- Restrictions for ***[CANxBCR2]***<SAM>
  The three-time sampling is not allowed under the condition that ***[CANxBCR1]***<BRP[9:0]> is smaller than 4.
  For the condition that ***[CANxBCR1]***<BRP[9:0]> < 4, a one-time sampling will always be performed regardless of the value of <SAM>.

Example: For 500 kbps
A bit has a length of 2µs. If f_{CANOSC} = 12 MHz, the baud rate prescaler is set to "1". That means a bit for this data transmission rate has to be programmed with a length of $12T_Q$. According to the above formula, the values to be programmed always are smaller by one than the calculated values:

[CANxBCR1]<BRP[9:0]> = 0000000001
[CANxBCR2]<TSEG1[3:0]> = 0110 ($7T_Q$)
[CANxBCR2]<TSEG2[2:0]> = 011 ($4T_Q$)

In this case, the sample point is 8 / 12 = 66%
Other combinations for ***[CANxBCR2]***<TSEG1>, <TSEG2> are possible; with <TSEG2> = 3 the full range for <SJW>.
<SJW> should always be set to the highest value possible. <SJW> is not allowed to be greater than > <TSEG2>.
The three-time sampling of the bus cannot be set because of the condition that ***[CANxBCR1]***<BRP[9:0]> is smaller than "4". Thus, <SAM>=0 should be set.

4. Register

4.1. Register list

The control registers and their addresses are shown as follows:

Function		Channel/Unit	Base address
			TYPE 1
CAN controller	CAN	unit A	0x40005000
		unit B	0x40006000

Register Name		Address(+Base)
CAN Mailbox	Refer to "4.1.1"	0x0000 to 0x03E0
Mailbox Configuration Register	***[CANxMC]***	0x0400
Mailbox Direction Register	***[CANxMD]***	0x0408
Transmission Request Set Register	***[CANxTRS]***	0x0410
Transmission Request Reset Register	***[CANxTRR]***	0x0418
Transmission Acknowledge Register	***[CANxTA]***	0x0420
Abort Acknowledge Register	***[CANxAA]***	0x0428
Receive Message Pending Register	***[CANxRMP]***	0x0430
Receive Message Lost Register	***[CANxRML]***	0x0438
Local Acceptance Mask Register	***[CANxLAM]***	0x0440
Global Acceptance Mask Register	***[CANxGAM]***	0x0448
Master Control Register	***[CANxMCR]***	0x0450
Global Status Register	***[CANxGSR]***	0x0458
Bit Configuration Register1	***[CANxBCR1]***	0x0460
Bit Configuration Register2	***[CANxBCR2]***	0x0468
Global Interrupt Flag Register	***[CANxGIF]***	0x0470
Global Interrupt Mask Register	***[CANxGIM]***	0x0478
Mailbox Transmit Interrupt Flag register	***[CANxMBTIF]***	0x0480
Mailbox Receive Interrupt Flag register	***[CANxMBRIF]***	0x0488
Mailbox Interrupt Mask Register	***[CANxMBIM]***	0x0490
Change Data Request Register	***[CANxCDR]***	0x0498
Remote Frame Pending Register	***[CANxRFP]***	0x04A0
CAN Error Counter Register	***[CANxCEC]***	0x04A8
Time Stamp Counter Prescaler Register	***[CANxTSP]***	0x04B0
Time Stamp Counter Register	***[CANxTSC]***	0x04B8

4.1.1. CAN Mailbox

The address of each mailbox is as follows.

Function name		Mailbox No.	Address(+Base)
CAN Mailbox	CANxMBn	No.0	0x0000
		No.1	0x0020
		No.2	0x0040
		No.3	0x0060
		No.4	0x0080
		No.5	0x00A0
		No.6	0x00C0
		No.7	0x00E0
		No.8	0x0100
		No.9	0x0120
		No.10	0x0140
		No.11	0x0160
		No.12	0x0180
		No.13	0x01A0
		No.14	0x01C0
		No.15	0x01E0
		No.16	0x0200
		No.17	0x0220
		No.18	0x0240
		No.19	0x0260
		No.20	0x0280
		No.21	0x02A0
		No.22	0x02C0
		No.23	0x02E0
		No.24	0x0300
		No.25	0x0320
		No.26	0x0340
		No.27	0x0360
		No.28	0x0380
		No.29	0x03A0
		No.30	0x03C0
		No.31	0x03E0

Each of the 32 mailboxes consists of the following field registers.

Register Name (Mailbox No. n = 0 to 31)		Offset Address
Message ID Field Register	***[CANxMBnID]***	0x0000
Time Stamp Values/ Message Control Field Register	***[CANxMBnTSVMCF]***	0x0008
Data Field Register	***[CANxMBnDL]***	0x0010
Data Field Register	***[CANxMBnDH]***	0x0018

4.2. Details of Registers

4.2.1. *[CANxMBnID]*(Message ID Field Register)

Bit	Bit Symbol	After Reset	Type	Function
31	IDE	-	R/W	ID Extension bit 0: Standard format (11-bit ID) from <ID28> to <ID18> used 1: Extended format (29-bit ID) from <ID28> to <ID0> used Sets the mailbox by selecting whether to receive or transmit the extended format (<IDE>=1) or the standard format (<IDE>=0).
30	GAME_LAME	-	R/W	Global (GAME) / Local (LAME) acceptance mask enable bit 0: Receive mask is not used for receive filtering. 1: Receive mask is used for receive filtering. <GAME> is the enable bit for the global acceptance mask GAM shared in mailboxes 0 to 30, and <LAME> is the enable bit for the local acceptance mask LAM used only for mailbox 31. When <GAME_LAME>=0, the received message are stored in the mailbox only when the receive message ID is the same as the mailbox ID. For transmission mailboxes, the acceptance mask function is not applied. In such case, always set <GAME> to "0".
29	RFH	-	R/W	Remote frame handling bit (only for transmission mailboxes) 0: Transmit mailboxes do not respond to remote frames. Software must handle remote frames 1: Transmit mailboxes respond to remote frames. (The <TRS> bit is set.) <RFH> determines whether a mailbox configured as a transmission mailbox will automatically respond to remote frame reception. When the ID of the received remote frame matches the ID of the transmission mailbox where <RFH>=1 and <GAME_LAME>=1, this mailbox ID is overwritten with the remote ID, and the mailbox automatically responds the remote frame using the overwritten ID. Handled as data frames in the case of receive mailboxes. (The <RMP> bit and the <RFP> bit are set.)
28:0	ID[28:0]	-	R/W	Message ID Standard format (11-bit ID): From <ID28> to <ID18> are used. Extended format (29-bit ID): From <ID28> to <ID0> are used. For the priority of message IDs, the message ID having most "0"s consecutively starting from the ID's highest bit (<ID28> bit) has the higher priority.

Register the mailbox IDs at the time of initial setup. To change the message ID field in the mailbox after the mailbox is enabled, clear the ***[CANxMC]***<MCn> bit corresponding to the mailbox to "0", and then disable the mailbox for the CAN controller before writing a new ID.

4.2.2. *[CANxMBnTSVMCF]*(Time Stamp Values Message Control Field Register)

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>After Reset</th><th>Type</th><th>Function</th></tr>
<tr><td>31:16</td><td>TSV[15:0]</td><td>-</td><td>R/W</td><td>Time stamp counter value
When message have been successfully received or transmitted, the 16-bit time stamp counter values are stored. No value is set when message reception or transmission fails.

For the details of the entire time stamp counter function, Refer to "3.3.6Time Stamp Function".</td></tr>
<tr><td>15:5</td><td>-</td><td>-</td><td>R</td><td>Read as undefined.</td></tr>
<tr><td>4</td><td>RTR</td><td>-</td><td>R/W</td><td>Remote frame transmission request bit.
0: Data frame
1: Remote frame</td></tr>
<tr><td>3:0</td><td>DLC[3:0]</td><td>-</td><td>R/W</td><td>Data length code
Sets the data length (number of bytes) of messages
<table>
<tr><th><DLC[3:0]></th><th>Number of bytes</th><th>Corresponding data</th></tr>
<tr><td>0000</td><td>0byte</td><td>None</td></tr>
<tr><td>0001</td><td>1byte</td><td>D0</td></tr>
<tr><td>0010</td><td>2bytes</td><td>D0,D1</td></tr>
<tr><td>0011</td><td>3bytes</td><td>D0,D1,D2</td></tr>
<tr><td>0100</td><td>4bytes</td><td>D0,D1,D2,D3</td></tr>
<tr><td>0101</td><td>5bytes</td><td>D0,D1,D2,D3,D4</td></tr>
<tr><td>0110</td><td>6bytes</td><td>D0,D1,D2,D3,D4,D5</td></tr>
<tr><td>0111</td><td>7bytes</td><td>D0,D1,D2,D3,D4,D5,D6</td></tr>
<tr><td>1000</td><td>8bytes</td><td>D0,D1,D2,D3,D4,D5,D6,D7</td></tr>
</table>
When <DLC[3:0]> = 1001 or more is set, data length is processed as 8 bytes.</td></tr>
</table>

The time stamp values do not need to be initially set.

The message control field needs no initial programming in the case of receive mailboxes. When a received message is stored in the mailbox, <RTR> and <DLC[3:0]> are also stored in the message control field at the same time. The transmission mailboxes need initial setting.

To change the message control field of a transmission mailbox (which is set to ***[CANxMBnID]***<RFH>=1) after enabling the mailbox, clear the ***[CANxMC]***<MCn> bit to "0" and then disable the mailbox for the CAN controller before writing a new <RTR> and <DLC[3:0]>. The message control field of the transmission mailbox set to <RFH>=0 can be changed irrespective of the ***[CANxMC]***<MCn> bit setting, but the user needs to check that the ***[CANxTRS]***<TRSn> bit is "0" before writing a new <RTR> and <DLC[3:0]>.

4.2.3. *[CANxMBnDL]*(Data fields Register)

Bit	Bit Symbol	After Reset	Type	Function
31:24	D3[7:0]	-	R/W	Transmitted or received data is stored.
23:16	D2[7:0]	-	R/W	Transmitted or received data is stored.
15:8	D1[7:0]	-	R/W	Transmitted or received data is stored.
7:0	D0[7:0]	-	R/W	Transmitted or received data is stored.

For transmission, data is transmitted according to the data byte count set in the ***[CANxMBnTSVMCF]***<DLC[3:0]> of the mailbox.

For reception, the data length code in the received message is copied to the ***[CANxMBnTSVMCF]***<DLC[3:0]> of the mailbox, and the data byte count only set in the ***[CANxMBnTSVMCF]***<DLC[3:0]> is made valid.

Mailboxes are readable and writable, but do not write data fields for receive mailboxes. If data fields are written, a mismatch may occur in received data.

To update the data field of a transmission mailbox set to ***[CANxMBnID]***<RFH>=1, set ***[CANxCDR]***<CDRn> to "1" and suspend transmission requests temporarily before writing new data. To update the data field of a transmission mailbox set to ***[CANxMBnID]***<RFH>=0, check that the ***[CANxTRS]***<TRSn> bit is "0" before writing new data.

4.2.4. *[CANxMBnDH]*(Data fields Register)

Bit	Bit Symbol	After Reset	Type	Function
31:24	D7[7:0]	-	R/W	Transmitted or received data is stored.
23:16	D6[7:0]	-	R/W	Transmitted or received data is stored.
15:8	D5[7:0]	-	R/W	Transmitted or received data is stored.
7:0	D4[7:0]	-	R/W	Transmitted or received data is stored.

4.2.5. *[CANxMC]*(Mailbox Configuration Register)

<table>
<tr><th>Bit</th><th>Bit Symbol</th><th>After Reset</th><th>Type</th><th>Function</th></tr>
<tr><td>31:0</td><td>MC[31:0]</td><td>0</td><td>R/W</td><td>Access configuration to the mailbox (Each bit corresponds to mailboxes 31 to 0)
0: The corresponding mailbox MBn is disabled for the CAN controller.
1: The corresponding mailbox MBn is enabled for the CAN controller.

Write access from CPU
<table>
<tr><th></th><th>ID field</th><th>Transmit mailbox with <RFH>=1</th><th>Data field</th><th>Control field</th></tr>
<tr><td><MCn>=0</td><td>Enabled</td><td>Enabled</td><td>Enabled</td><td>Enabled</td></tr>
<tr><td><MCn>=1</td><td>Disabled</td><td>Disabled</td><td>Enabled</td><td>Enabled</td></tr>
</table>
</td></tr>
</table>

Note: the Following care is required during reprogramming of a ***[CANxMC]*** in operation.

Receive: For a receive mailbox it needs to be ensured that the mailbox is not being disabled while a reception for this mailbox is ongoing. If a mailbox is disabled or reconfigured during an ongoing reception, the current frame might be received.

Transmit: When the CAN controller is transmitting data (***[CANxTRS]***<TRSn>=1), Clear <MCn> to "0" after the transmission is completed (***[CANxTRS]***<TRSn>=0).

4.2.6. *[CANxMD]*(Mailbox Direction Register)

Bit	Bit Symbol	After Reset	Type	Function
31	MD31	1	R	Mailbox direction: (for Mailbox 31) Mailbox 31 is the receive-only mailbox. This is always set to "1" and cannot be changed.
30:0	MD[30:0]	0	R/W	Mailbox direction: Mailboxes 30 to 0 (Each bit corresponds to mailboxes 30 to 0.) 0: Set as a transmission mailbox. 1: Set as a receive mailbox. Each mailbox can be set as a transmission or receive mailbox.

Set the ***[CANxMD]*** register at the initial setup. The directions of mailboxes cannot be changed when the operation is ongoing. To change ***[CANxMD]*** register settings, set the corresponding ***[CANxMC]***<MCn> bit to "0" before making changes.

4.2.7. *[CANxTRS]*(Transmission Request Set Register)

Bit	Bit Symbol	After Reset	Type	Function
31	-	0	R	Read as "0"
30:0	TRS[30:0]	0	R/W	Transmit request set (Each bit corresponds to mailboxes 30 to 0.) Set <TRSn> requests the message transmission of corresponding mailbox n. When the transmission is requested for multiple mailboxes, the message is transmitted in accordance with the priority corresponding to the ***[CANxMCR]***<MTOS> bit. A write of "1" from the CPU to mailbox configured as transmission mailbox can set the bit. A write of "0" from the CPU is invalid.

Note: Mailbox 31 is receive only mailbox.

The transmission request set register can be set by a write of "1" from the CPU to only the ***[CANxTRS]***<TRSn> bits of the mailboxes configured for transmission. The ***[CANxTRS]***<TRSn> bits of the mailboxes configured for reception cannot be set.

The ***[CANxTRS]***<TRSn> bit is cleared to "0" when the message has been successfully transmitted or the transmission request is reset by setting the ***[CANxTRR]***<TRRn> bit to "1".

When the transmission fails, the transmission process is repeated until it succeeds or the transmission request is reset by setting the ***[CANxTRR]***<TRRn> bit to "1".

When the ***[CANxTRS]***<TRSn> bit is "1", do not write to mailbox n.

4.2.8. *[CANxTRR]*(Transmission Request Reset Register)

Bit	Bit Symbol	After Reset	Type	Function
31	-	0	R	Read as "0"
30:0	TRR[30:0]	0	R/W	Transmit request reset (Each bit corresponds to mailboxes 30 to 0.) Setting <TRRn> cancels the message transmission of corresponding mailbox n. A write of "1" from the CPU to mailbox n configured as transmission mailbox can set the bit. Write of "0" from the CPU is invalid.

Note: Mailbox 31 is receive only mailbox.

The transmission request reset register can be set by a write of "1" from the CPU to only the ***[CANxTRR]***<TRRn> bits of the mailboxes configured for transmission. The ***[CANxTRR]***<TRRn> bits of the mailboxes configured for reception cannot be set.

The ***[CANxTRR]***<TRRn> bit is cleared to "0" by the internal logic when the message has been successfully transmitted or the transmission is aborted. A write of "0" from the CPU is invalid.

When the ***[CANxTRR]***<TRRn> bit is "1", do not write to mailbox n.

Setting the ***[CANxTRR]***<TRRn> bit cancels the message transmission of mailbox n set by the ***[CANxTRS]***<TRSn> bit, where the operation executed will be any of the following three sequences:

(a) A transmission request of a message has not yet been transmitted.

A transmission request of a message will be cleared immediately.
(***[CANxTRS]***<TRSn> = 0, ***[CANxTRR]***<TRRn> = 0, ***[CANxAA]***<AAn> = 1)

(b) A transmission request of a message is currently being transmitted and an arbitration lost error occurs or an error is detected on the CAN bus.

A transmission request of a message will be cleared and the transmission will be canceled.
(***[CANxTRS]***<TRSn> = 0, ***[CANxTRR]***<TRRn> = 0, ***[CANxAA]***<AAn> = 1)

(c) A transmission request of a message is currently being transmitted and no arbitration lost error occurs and no error is detected on the CAN bus.

A transmission request of a message will not be cleared and the transmission will be completed
(***[CANxTRS]***<TRSn> = 0, ***[CANxTRR]***<TRRn> = 0, ***[CANxTA]***<TAn> = 1)

4.2.9. *[CANxTA]*(Transmission Acknowledge Register)

Bit	Bit Symbol	After Reset	Type	Function
31	-	0	R	Read as "0".
30:0	TA[30:0]	0	R/W	Transmission acknowledge (Each bit corresponds to mailboxes 30 to 0) When the message in mailbox n has been successfully transmitted, the <TAn> bit is set to "1". The <TAn> bit can be cleared by a write of "1" from the CPU to the <TAn> bit or the ***[CANxTRS]***<TRSn> bit.

Note: Mailbox 31 is receive only mailbox

The ***[CANxTA]***<TAn> bit is set to "1" when a message in mailbox n has been successfully transmitted.
When the mailbox interrupt is enabled by setting the corresponding <MBIMn> bit in the mailbox interrupt mask register ***[CANxMBIM]*** to "1", the <MBTIFn> bit of the mailbox transmission interrupt flag register ***[CANxMBTIF]*** is set to "1" and the CANx transmission completion interrupt INTCANxTXD occurs.

A write of "1" to the <TAn> bit or the ***[CANxTRS]***<TRSn> bit from the CPU can clear the <TAn> bit. A write of "0" to the <TAn> bit or the ***[CANxTRS]***<TRSn> bit from the CPU is invalid.

4.2.10. *[CANxAA]*(Abort Acknowledge Register)

Bit	Bit Symbol	After Reset	Type	Function
31	-	0	R	Read as "0".
30:0	AA[30:0]	0	R/W	Abort acknowledge (Each bit corresponds to mailboxes 30 to 0.) When the message in mailbox n has not been successfully transmitted, the <AAn> bit is set to "1". The <AAn> bit can be cleared by a write of "1" from CPU to the <AAn> bit or the ***[CANxTRS]***<TRSn> bit.

Note: Mailbox 31 is receive only mailbox.

The ***[CANxAA]***<AAn> bit is set to "1" when a message in mailbox n has not been successfully transmitted.
When ***[CANxGIF]***<TRMABF> bit in the global interrupt flag register is also set to "1", and the transmission abort interrupt is enabled by setting the ***[CANxGIM]***<TRAMABM> bit in the global interrupt mask register to "1", the CAN global interrupt(INTCANxGLB) occurs.

A write of "1" to the <AAn> bit or the ***[CANxTRS]***<TRSn> bit from the CPU can clear the <AAn> bit. A write of "0" to the <AAn> bit or the ***[CANxTRS]***<TRSn> bit from the CPU is invalid.

4.2.11. *[CANxRMP]*(Receive Message Pending Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	RMP[31:0]	0	R/W	Receive message pending (Each bit corresponds to mailboxes 31 to 0.) After a message is received and the content of the received message is written in mailbox n, the <RMPn> bit is set to "1". After received data is read, a write of "1" to the <RMPn> bit can clear the <RMPn> bit.

Note: This register cannot use read-modify-write instruction.

The ***[CANxRMP]***<RMPn> bit is set to "1" when a message in mailbox n has been successfully received.

When the mailbox interrupt is enabled by setting the corresponding <MBIMn> bit in the mailbox interrupt mask register ***[CANxMBIM]*** to "1", the <MBRIFn> bit of the mailbox receive interrupt flag register ***[CANxMBRIF]*** is set to "1" and the CANx reception completion interrupt INTCANxRXD occurs.

To clear the ***[CANxRMP]*** <RMPn> bit, write "1" to the ***[CANxRMP]***<RMPn> bit from the CPU. A write of "0" to the ***[CANxRMP]***<RMPn> bit from the CPU is invalid.

4.2.12. *[CANxRML]*(Receive Message Lost Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	RML[31:0]	0	R/W	Receive message lost (Each bit corresponds to mailboxes 31 to 0.) When mailbox n for which the <RMPn> bit is set to "1" receives the next message, the content of the received message is overwritten to the mailbox n, and the <RMLn> bit is set to "1". A write of "1" to the <RMPn> bit can clear the <RMLn> bit.

The ***[CANxRML]***<RMLn> bit is set by the internal logic and can be cleared with a write of "1" to the ***[CANxRMP]***<RMPn> bit from the CPU. The <RMPn> bit is also cleared at the same time. A write of "1" or "0" to the <RMLn> bit from the CPU is invalid.

With the ***[CANxRMP]***<RMPn> bit set to "1", if mailbox n receives the next message, the corresponding <RMLn> bit in the receive message lost register ***[CANxRML]*** is set to "1". In this case, mailbox n is overwritten with the new received message.

When the <TRMABF> bit in the global interrupt flag register ***[CANxGIF]*** is also set to "1", and the transmission abort interrupt is enabled by setting the <TRMABM> bit in the global interrupt mask register ***[CANxGIM]*** to "1", the CAN global interrupt INTCANxGLB occurs.

When the receive message lost interrupt is enabled by setting the <RMLIM> bit in the global interrupt mask register ***[CANxGIM]*** to "1", the CAN global interrupt INTCANxGLB occurs.

Table 4.1 shows the changes of the ***[CANxRMP]*** and ***[CANxRML]*** registers before and after a message is received.

Table 4.1 Change of *[CANxRMP]* and *[CANxRML]* Registers before/ after a Message is received

ID	Before Reception		After Reception		Operation
	<RMPn>	<RMLn>	<RMPn>	<RMLn>	
No match	Don't care	Don't care	Don't care	Don't care	The received message is not stored in any mailboxes.
Match	0	0	1	0	The received message is stored in mailbox n with a matching ID.
	1	0	1	1	The received message is overwritten in mailbox n with a matching ID. This shows that the previous message was lost.
	1	1	1	1	

4.2.13. *[CANxLAM]*(Local Acceptance Mask Register)

Bit	Bit Symbol	After Reset	Type	Function
31	LAMI	0	R/W	Mask of the ID extension bit<IDE> (mailbox 31) 0: Not masked 1: Masked In case of <LAMI>=0, the message in the standard or the extended format is received, according to the <IDE> bit of the mailbox 31. In case of <LAMI>=1, the message in the standard and the extended format is received, regardless of the <IDE> bit of the mailbox 31.
30:29	-	0	R	Read as "0"
28:0	LAM[28:0]	0	R/W	Mask of receive message ID 0: Not masked The reception message is received when the corresponding bit of reception message ID is the same as mailbox ID. 1: Masked The reception message is received regardless of the value of the corresponding bit of reception message.

The local acceptance mask register ***[CANxLAM]*** will only be used for filtering of the receiving message ID for mailbox 31. This feature allows locally masking to any ID bits of the receiving message for mailbox 31.

In the extended format, ***[CANxMBnID]***< ID[28:0] > and < LAM[28:0] > are used to filtering.
In the standard format, ***[CANxMBnID]***< ID[28:18] > and < LAM[28:18] > are used to filtering.

When the message in a standard format is received, the part of the extended ID (<ID[17:0]>) will become an undefined value. Therefore, receiving the standard and the extended format alternately in the same mailbox is not recommended.
Please set ***[CANxLAM]*** when initialization (At the configuration mode) and do not change the setting while operating.
When the setting is changed while receiving the message, the ***[CANxLAM]*** value on the way of the setting change is used to filtering of reception message ID.

4.2.14. *[CANxGAM]*(Global Acceptance Mask Register)

Bit	Bit Symbol	After Reset	Type	Function
31	GAMI	0	R/W	Mask of the ID extension bit<IDE> (mailboxes 0 to 30) 0: Not masked 1: masked In case if <GAMI> =0, the message of the standard or the extended format is received, according to the <IDE> bit of the mailboxes 0 to 30. In case of <GAMI> = 1, the message of the standard and the extended format is received, regardless of the <IDE> bit of the mailboxes 0 to 30.
30:29	-	0	R	Read as "0".
28:0	GAM[28:0]	0	R/W	Mask of receive message ID 0: Not masked The reception message is received when the corresponding bit of reception message ID is the same as mailbox ID. 1: Masked The reception message is received regardless of the value of the corresponding bit of reception message.

The global acceptance mask register ***[CANxGAM]*** will be used for filtering of the receiving message ID for mailbox 0 to 30. This feature allows to globally masking any ID bits of the receiving message for mailbox 0 to 30.

In the extended format, ***[CANxMBnID]***<ID[28:0]> and <GAM[28:0]> are used for filtering.
In the standard format, ***[CANxMNnID]***<ID[28:18]> and <GAM[28:18]> are used for filtering.

When the message in the standard format is received, the part of the extended ID (<ID[17:0]>) will become an undefined value. Therefore, receiving the standard and the extended format alternately in the same mailbox is not recommended.
Please set ***[CANxGAM]*** during the initialization (At the configuration mode) and do not change the setting during the operation. When the setting is changed while receiving the message, the ***[CANxGAM]*** value on the way of the setting change is used for filtering of reception message ID.

4.2.15. *[CANxMCR]*(Master Control Register)

Bit	Bit Symbol	After Reset	Type	Function
31:12	-	0	R	Read as "0".
11	SUR	0	R/W	Suspend mode request 0: Cancels suspend mode (normal operation) 1: Requests suspend mode
10	-	0	R	Read as "0".
9	TSTLB	0	R/W	Test loop back 0: Cancels test loop back mode (normal operation) 1: Requests test loop back mode (This mode supports stand-alone operation.)
8	TSTERR	0	R/W	Test error 0: Cancels test error mode (normal operation) 1: Requests test error mode (In this mode, it is possible to write the CAN error counter register ***[CANxCEC]***.
7	CCR	1	R/W	Change configuration request 0: Cancels configuration mode (normal operation) 1: Requests configuration mode (In this mode, it possible to write the bit configuration registers, ***[CANxBCR1]*** and ***[CANxBCR2]***.)
6	SMR	0	R/W	Sleep mode request (Note3) 0: Cancels sleep mode (normal operation) 1: Requests sleep mode (In this mode, the clock of the CAN controller stops and the error counters and transmission requests are reset.)
5	-	0	R	Read as "0".
4	WUBA	0	R/W	Wake-up on bus activity 0: Wakes up only by a write access to the ***[CANxMCR]*** register. 1: Wakes up by detecting a bus active state or a write access to the ***[CANxMCR].***
3	MTOS	0	R/W	Mailbox transmission order select 0: Messages are transmitted in ascending order of mailbox number. 1: Messages in mailboxes are transmitted in descending order of message ID priority.
2	-	0	R	Read as "0".
1	TSCC	0	W	Time stamp counter clear 0: Disable 1: Clears the time stamp counter to "0". (Note1)
			R	Read as always "0".
0	SRES	0	W	Software reset (Note2) 0: Disable 1: Resets the CAN controller by software.
			R	Read as always "0".

Note1: The time stamp counter is also cleared by a write to the ***[CANxTSP]*** register and a write of "0" to the ***[CANxTSC]*** register.

Note2: After software reset, all registers in the CAN must be accessed after the following time.
(1) When communication by CAN bus is not performed, please wait for more than 16 clocks based on fsys.
(2) When communication by CAN bus is performed, please wait for more than 88 clocks based on fsys.

Note 3: To cancel sleep mode of the CAN operation, check that the ***[CANxGSR]***<SMA> bit is "1" before setting ***[CANxMCR]***<SMR> to "0".

4.2.16. *[CANxGSR]*(Global Status Register)

Bit	Bit Symbol	After Reset	Type	Function
31:17	-	0	R	Read as "0".
16:12	MIS[4:0]	11111	R	Message in slot Indicates the mailbox number of a message located in the transmission buffer. 00000: Message for mailbox 0 01011: Message for mailbox 11 10110: Message for mailbox 22 00001: Message for mailbox 1 01100: Message for mailbox 12 10111: Message for mailbox 23 00010: Message for mailbox 2 01101: Message for mailbox 13 11000: Message for mailbox 24 00011: Message for mailbox 3 01110: Message for mailbox 14 11001: Message for mailbox 25 00100: Message for mailbox 4 01111: Message for mailbox 15 11010: Message for mailbox 26 00101: Message for mailbox 5 10000: Message for mailbox 16 11011: Message for mailbox 27 00110: Message for mailbox 6 10001: Message for mailbox 17 11100: Message for mailbox 28 00111: Message for mailbox 7 10010: Message for mailbox 18 11101: Message for mailbox 29 01000: Message for mailbox 8 10011: Message for mailbox 19 11110: Message for mailbox 30 01001: Message for mailbox 9 10100: Message for mailbox 20 11111: No message 01010: Message for mailbox 10 10101: Message for mailbox 21
11	RM	0	R	Receive mode 0: The CAN controller is not receiving a message. 1: The CAN controller is receiving a message.
10	TM	0	R	Transmit mode 0: The CAN controller is not transmitting a message. 1: The CAN controller is transmitting a message.
9	-	0	R	Read as "0".
8	SUA	0	R	Suspend mode acknowledge 0: The CAN controller is not in suspend mode. 1: The CAN controller is in suspend mode.
7	CCE	1	R	Change configuration enable 0: The CAN controller is not in configuration mode. 1: The CAN controller is in configuration mode. In this mode, it is possible to write the bit configuration registers, ***[CANxBCR1]*** and ***[CANxBCR2]***.
6	SMA	0	R	Sleep mode acknowledge 0: The CAN controller is not in sleep mode. 1: The CAN controller is in sleep mode. In this mode, the clock of the CAN controller stops and the error counters and transmission request are reset.
5:4	-	0	R	Read as "0".
3	TSO	0	R	Time stamp overflow 0: The time stamp counter does not overflow. 1: The time stamp counter has overflowed at least once after this bit was last cleared to "0". To clear this bit, a clear the ***[CANxGIF]***<TSOIF> bit to "0".
2	BO	0	R	Bus off status 0: In the bus on state (normal operation) 1: In bus off state When CAN bus errors occur abnormally often and the transmission error counter <TEC> reaches its limit of 256, the CAN controller enters bus off state. No messages can be transmitted or received. The error counter is undefined. After the bus off recovery sequence, the CAN controller automatically enters bus on state.
1	EP	0	R	Error passive status 0: The CAN controller is not in error passive mode. 1: The CAN controller is in error passive mode.
0	EW	0	R	Warning status 0: Both <TEC> and <REC> values are 96 or less. 1: At least one of the <TEC> and <REC> values is greater than 96 and has reached the warning level.

4.2.17. *[CANxBCR1]*(Bit Configuration Register1)

Bit	Bit Symbol	After Reset	Type	Function
31:10	-	0	R	Read as "0".
9:0	BRP[9:0]	0	R/W	Set the value of Baud rate prescaler value: 0 to 1023

4.2.18. *[CANxBCR2]*(Bit Configuration Register2)

Bit	Bit Symbol	After Reset	Type	Function
31:10	-	0	R	Read as "0".
9:8	SJW[1:0]	0	R/W	Resynchronization jump width 00: 1 × TQ 01: 2 × TQ 10: 3 × TQ 11: 4 × TQ
7	SAM	0	R/W	Setting of sampling count 0: Single sampling 1: Triple sampling
6:4	TSEG2[2:0]	0	R/W	Setting of bit time after sample point 000: reserved 100: 5 × TQ 001: 2 × TQ 101: 6 × TQ 010: 3 × TQ 110: 7 × TQ 011: 4 × TQ 111: 8 × TQ
3:0	TSEG1[3:0]	0	R/W	Setting of bit time before sample point (except SYNCSEG). 0000: reserved 1000: 9 × TQ 0001: 2 × TQ 1001: 10 × TQ 0010: 3 × TQ 1010: 11 × TQ 0011: 4 × TQ 1011: 12 × TQ 0100: 5 × TQ 1100: 13 × TQ 0101: 6 × TQ 1101: 14 × TQ 0110: 7 × TQ 1110: 15 × TQ 0111: 8 × TQ 1111: 16 × TQ

For detail, please refer to "3.5 Bit Configuration ".

4.2.19. *[CANxGIF]*(Global Interrupt Flag Register)

Bit	Bit Symbol	After Reset	Type	Function
31:8	-	0	R	Read as "0".
7	RFPF	0	R/W	Remote frame pending flag 0: No remote frame has been received. 1: Remote frames have been received. (in the receive mailbox) This bit will not be set when matching with the transmission mailbox for which the <RFH> bit is "1".
6	WUIF	0	R/W	Wake-up interrupt flag 0: In sleep mode or normal operation mode 1: Sleep mode has been canceled.
5	RMLIF	0	R/W	Receive message lost interrupt flag 0: No receive message lost error has occurred. 1: A receive message lost error has occurred in at least one mailbox configured as a receive mailbox.
4	TRMABF	0	R/W	Transmission abort flag 0: No transport abort has occurred. 1: Transport abort has occurred. (At least one bit in the ***[CANxAA]*** register is set.)
3	TSOIF	0	R/W	Time stamp counter overflow interrupt flag 0: No overflow has occurred in the time stamp counter after this bit was last cleared. 1: There was at least one overflow of the time stamp counter after this bit was last cleared.
2	BOIF	0	R/W	Bus off interrupt flag 0: The CAN controller is in bus on mode. 1: The CAN controller is in bus off mode.
1	EPIF	0	R/W	Error passive interrupt flag 0: The CAN controller is in error active mode. 1: The CAN controller is in error passive mode.
0	WLIF	0	R/W	Warning level interrupt flag 0: None of the error counters have reached the warning level. 1: At least one of the error counters has reached the warning level.

Each interrupt flag of the global interrupt flag register ***[CANxGIF]*** will be set to "1" if the corresponding global interrupt condition has met. When the global interrupt flag is set to "1", if the corresponding bit in the global interrupt mask register ***[CANxGIM]*** is "1" (interrupt enabled), the CAN global interrupt (INTCANxGLB) will be "High".

The ***[CANxGIF]*** register (Lower 8bits) can be cleared by writing "1" to the corresponding bit in the ***[CANxGIF]*** register. A write of "0" is invalid.

4.2.20. *[CANxGIM]*(Global Interrupt Mask Register)

Bit	Bit Symbol	After Reset	Type	Function
31:8	-	0	R	Read as "0".
7	RFPM	0	R/W	Remote frame pending interrupt mask 0: Interrupt disable 1: Interrupt enable
6	WUIM	0	R/W	Wake-up interrupt mask 0: Interrupt disable 1: Interrupt enable
5	RMLIM	0	R/W	Receive message lost interrupt mask 0: Interrupt disable 1: Interrupt enable
4	TRMABF	0	R/W	Transmit abort interrupt mask 0: Interrupt disable 1: Interrupt enable
3	TSOIM	0	R/W	Time stamp counter overflow interrupt mask 0: Interrupt disable 1: Interrupt enable
2	BOIM	0	R/W	Bus off interrupt mask 0: Interrupt disable 1: Interrupt enable
1	EPIM	0	R/W	Error passive interrupt mask 0: Interrupt disable 1: Interrupt enable
0	WLIM	0	R/W	Warning level interrupt mask 0: Interrupt disable 1: Interrupt enable

The global interrupt mask register ***[CANxGIM]*** controls whether to enable or disable a global interrupt correspondingly to each interrupt condition of the ***[CANxGIF]*** register. When the bit in the ***[CANxGIF]*** register is "0", the corresponding CAN global interrupt is disabled. When the bit in the ***[CANxGIF]*** register is "1", the corresponding CAN global interrupt is enabled.

Reset operation clears all bits in the ***[CANxGIM]*** register to "0", disabling global interrupts.

4.2.21. *[CANxMBTIF]*(Mailbox Transmit Interrupt Flag Register)

Bit	Bit Symbol	After Reset	Type	Function
31	-	0	R	Read as "0".
30:0	MBTIF[30:0]	0	R/W	Mailbox transmission interrupt flag (Each bit corresponds to mailboxes 30 to 0.) When the message in mailbox n has been successfully transmitted and the interrupt mask of the ***[CANxMBIM]*** register is enabled (<MBIMn>=1), the <MBTIFn> bit is set to "1" and the transmission completion interrupt (INTCANxTXD) becomes the "High" level. When ***[CANxMBIM]***<MBIMn> bit is "0", the <MBTIFn> bit is not set and INTCANxTXD stays at the "Low" level. Transmission completion is checked by reading the ***[CANxTA]*** register. If even one bit in the ***[CANxMBTIF]*** register is "1", INTCANxTXD is the "High" level. The <MBTIFn> bit is cleared by a write of "1" to the <MBTIFn> bit from the CPU. A write of "0" is invalid.

When the mailbox is set to receive, the corresponding bit in the ***[CANxMBTIF]*** register is read as "0". When the mailbox is set to transmit, the corresponding bit in the ***[CANxMBRIF]*** register is read as "0".

4.2.22. *[CANxMBRIF]*(Mailbox Receive Interrupt Flag Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	MBRIF[31:0]	0	R/W	Mailbox receive interrupt flag (Each bit corresponds to mailboxes 31 to 0.) When mailbox n has successfully received the message and the interrupt mask of the ***[CANxMBIM]*** register is enabled (<MBIMn> = 1). When the <MBIMn> bit in the, ***[CANxMBIM]*** register is "0", the <MBRIFn> bit is not set and INTCANxRXD stays at the "Low" level. Receive completion is checked by reading the ***[CANxRMP]*** register. If even one bit in the ***[CANxMBRIF]*** register is "1", INTCANxRXD is the "High" level. The <MBRIFn> bit is cleared by a write of "1" to the <MBRIFn> bit from the CPU. A write of "0" is invalid.

4.2.23. *[CANxMBIM]*(Mailbox Interrupt Mask Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	MBIM[31:0]	0	R/W	Mailbox interrupt mask 0: Interrupt disabled for corresponding mailbox 1: Interrupt enabled for corresponding mailbox

The settings in ***[CANxMBIM]*** determine, for which mailbox the interrupt generation is enabled or disabled.

If a bit in ***[CANxMBIM]*** is "0", the interrupt generation for the corresponding mailbox is disabled and if it is "1", the interrupt generation is enabled. Reset value of ***[CANxMBIM]*** is "0".

4.2.24. *[CANxCDR]*(Change Data Request Register)

Bit	Bit Symbol	After Reset	Type	Function
31	-	0	R	Read as "0".
30:0	CDR[30:0]	0	R/W	Change data request (Each bit corresponds to mailboxes 30 to 0.) When the <CDRn> bit of transmission mailbox n is set to "1", the transmission request of this mailbox n is ignored. It means mailbox n for which the ***[CANxTRS]***<TRSn> bit and the <CDRn> bit are set will be excluded from the internal arbitration range and will not be transmitted if transmission has not started. After the <CDRn> bit is cleared to "0", mailbox n is back to be included in the internal arbitration range.

Note: Mailbox 31 is receive only mailbox.

The change data request register ***[CANxCDR]*** is effective when updating the data field of transmission mailbox n where auto acknowledgment of remote frames is enabled (***[CANxMBnID]***<RFH>=1). Mailbox n enabling automatic acknowledgment starts message transmission automatically responding to received remote frames and so may update the data field during message transmission (In such cases, updated data is output midway through transmission). The update of the data field can be avoided by setting the <CDRn> bit to "1" and temporarily suspending data transmission.

4.2.25. *[CANxRFP]*(Remote Frame Pending Register)

Bit	Bit Symbol	After Reset	Type	Function
31:0	RFP[31:0]	0	R/W	Remote frame pending (Each bit corresponds to mailboxes 31 to 0.) When mailbox n configured as receive mailbox receives a remote frame, the <RFPn> bit and ***[CANxRMP]***<RMPn> bit are set to "1". The <RFPn> bit can be cleared by a write of "1" to the ***[CANxRMP]***<RMPn> bit.

The ***[CANxRFP]***<RFPn> bit is set by the internal logic and can be cleared with a write of "1" to the ***[CANxRMP]***<RMPn> bit from the CPU. The ***[CANxRMP]***<RMPn> bit is also cleared at the same time. A write of "0" to the ***[CANxRMP]***<RMPn> bit and a write of "1" or "0" to the ***[CANxRFP]***<RFPn> bit from the CPU are invalid.

Even when mailbox n with ***[CANxRFP]***<RFPn>=1 is overwritten by data frame reception, the ***[CANxRFP]***<RFPn> bit is cleared.

When the remote frame pending interrupt is enabled by setting the <RFPM> bit in the global interrupt mask register ***[CANxGIM]*** to "1", the CAN global interrupt (INTCANxGLB) occurs.

4.2.26. *[CANxCEC]*(CANx Error Counter Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0".
15:8	TEC[7:0]	0	R	8-bit transmission error counter (After reset release)
		-	W	8-bit transmission error counter (***[CANxMCR]***<TSTERR>=1)
7:0	REC[7:0]	0	R	8-bit reception error counter (After reset release)
		-	W	8-bit reception error counter (***[CANxMCR]***<TSTERR>=1)

The CAN controller contains two error counters: the reception error counter <REC[7:0]> and the transmission error counter <TEC[7:0]>. The value of both counters can be read from the CPU. A write access to the error counters is only possible in test error mode (The ***[CANxMCR]***<TSTERR> bit is "1"). In the case of a write to the ***[CANxCEC]*** register, the write data to the lower 8 bits <REC[7:0]> is written also to the higher 8 bits <TEC[7:0]>.

The CAN error counters count up or down according to the CAN Specification 2.0B.

The ***[CANxCEC]***<REC[7:0]> is not increased after exceeding the error passive limit (128). When ***[CANxCEC]***<REC[7:0]>=128, after the correct reception of a message, the ***[CANxCEC]***<REC[7:0]> is set to a value between 119 and 127. After reaching the "bus off" status, the error counters are undefined.

If the status "bus off" is reached, the reception error counter is incremented after 11 consecutive recessive bits on the bus. If the counter reaches the count 128, the module changes automatically to the status error active.

All internal flags are reset and the error counters will be cleared to "0". The configuration registers keep the programmed values. The values of the counters are undefined during "bus off" status.

When CAN enters configuration mode, the error counters will be cleared.

4.2.27. *[CANxTSP]*(Time Stamp Counter Prescaler Register)

Bit	Bit Symbol	After Reset	Type	Function
31:4	-	0	R	Read as "0".
3:0	TSP[3:0]	0	R/W	Time stamp counter prescaler Sets the value to be loaded to the prescaler for the 4-bit TSC.

To ensure that the value of the ***[CANxTSC]*** will not change during the write cycle to the mailbox, a hold register is implemented. The value of the ***[CANxTSC]*** will be copied to the hold register and then written to the mailbox from the hold register if a message has been received or transmitted successfully. The reception is successful for the receiver, if there is no error but the last one bit of End-of-frame. Transmission is successful for the transmitter if there is no error until the last bit of End-of-frame. (Refer to the CAN specification 2.0B).

4.2.28. *[CANxTSC]*(Time Stamp Counter Register)

Bit	Bit Symbol	After Reset	Type	Function
31:16	-	0	R	Read as "0".
15:0	TSC[15:0]	0	R	Time stamp counter Free running 16-bit counter

The overflow of the ***[CANxTSC]*** can be detected by the time stamp counter overflow interrupt flag <TSOIF> of the global interrupt flag register ***[CANxGIF]***, and the time stamp counter overflow flag <TSO> of the global status register ***[CANxGSR]***. Both flags can be cleared by writing "1" to <TSOIF> in the ***[CANxGIF]*** register.

There is a 4-bit prescaler for the ***[CANxTSC]***. After power-up the time stamp counter is driven directly from the bit clock (***[CANxTSP]***<TSP[3:0]>=0). The period T_{TSC} for the time stamp counter will be calculated by the following formula:

T_{TSC} = TBIT × (***[CANxTSP]***<TSP[3:0]> + 1)

5. Usage

5.1. Receive Messages

Figure 5.1 shows an example flowchart of message reception using the CANx reception completion interrupt (INTCANxRXD).

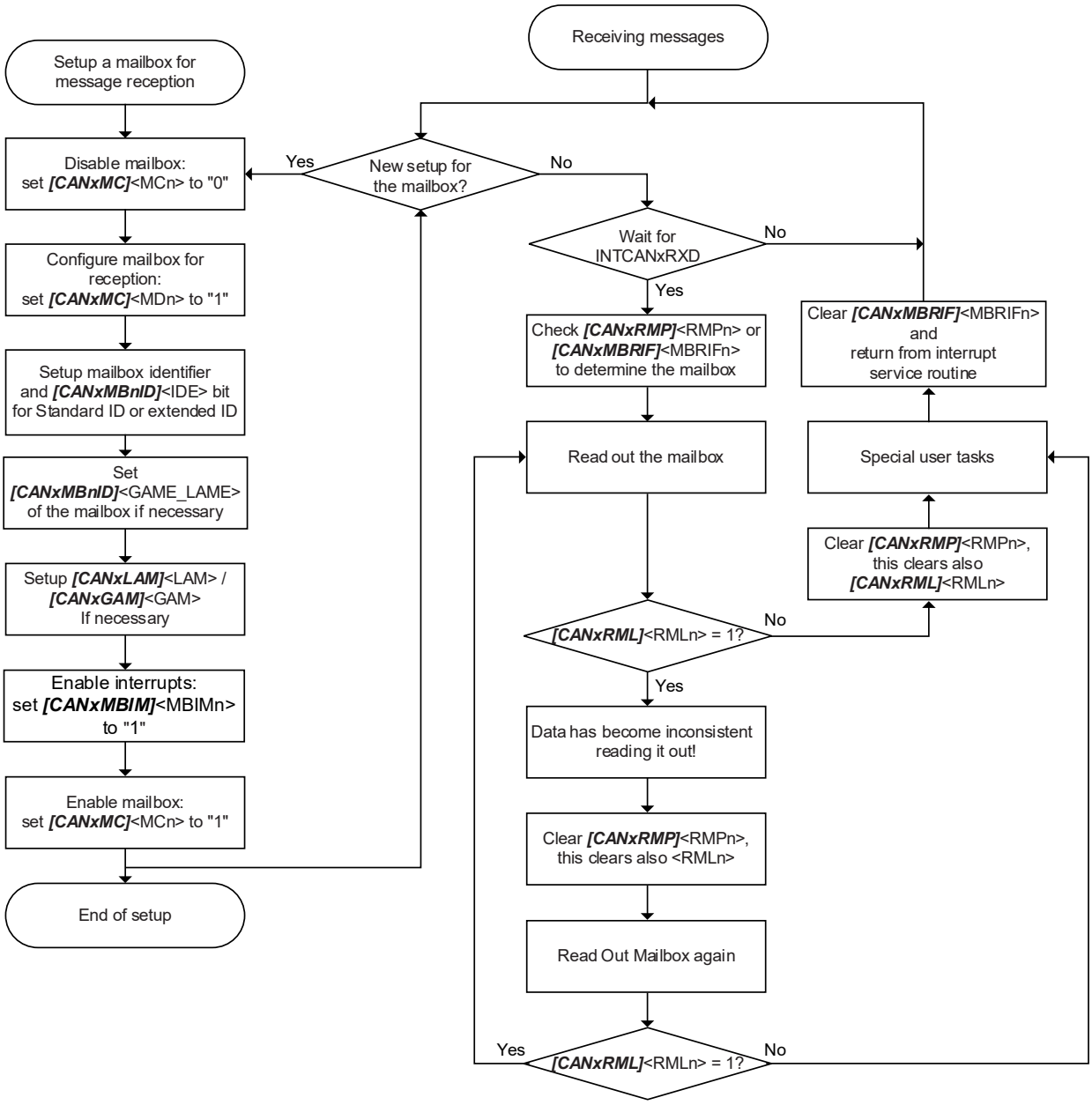

Figure 5.1 Flowchart of Message Reception

It is also possible to use polling instead of receive interrupts. In this case, the "waiting for INTCANxRXD" in above flowchart must be replaced by polling ***[CANxRMP]***. Further, enabling interrupts and clearing ***[CANxMBRIF]*** must be removed from the flow.

5.2. Transmitting Message

Figure 5.2 shows an example flowchart of message transmission using the CANx transmission completion interrupt (INTCANxTXD).

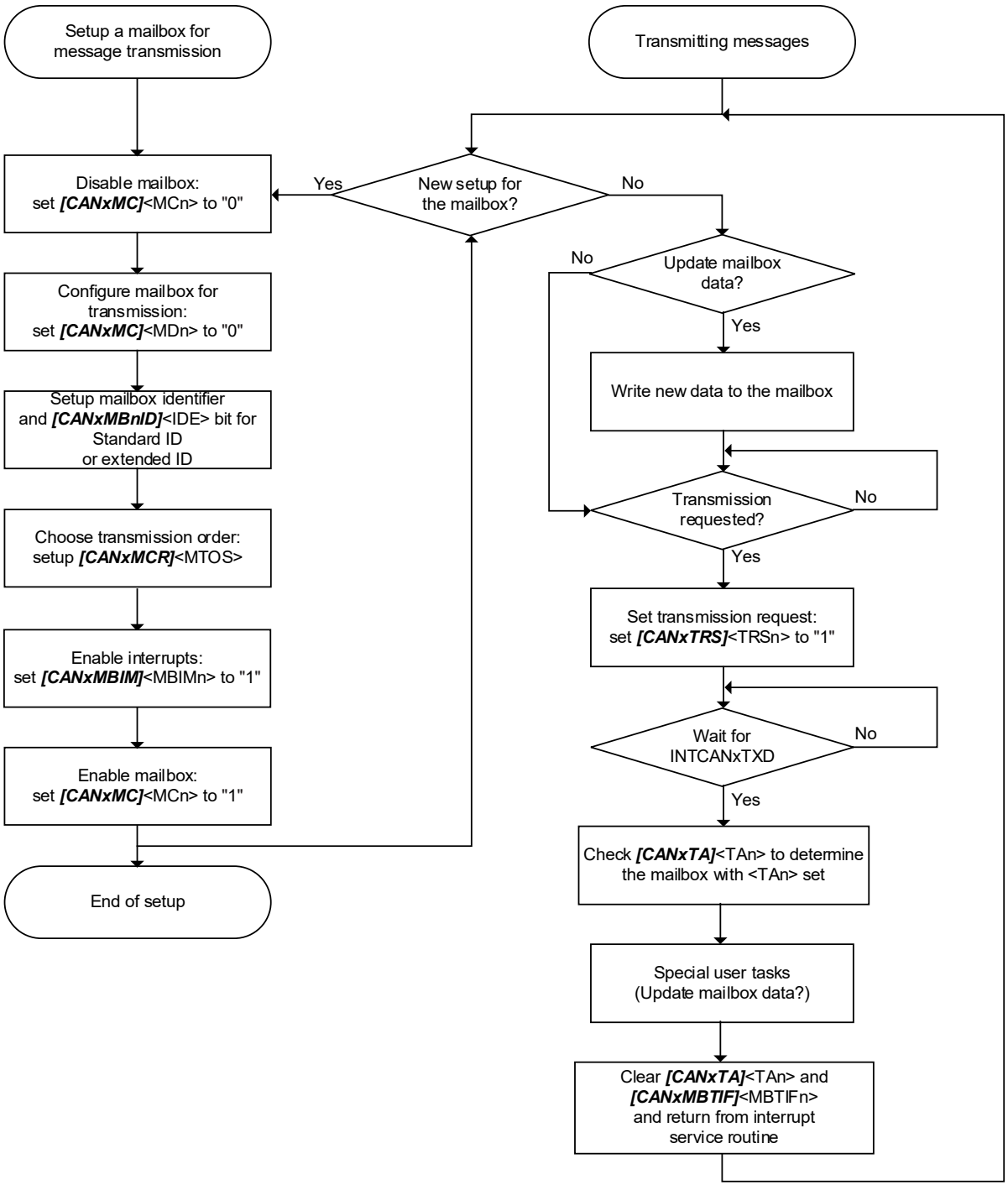

Figure 5.2 Flowchart of Message Transmission

It is also possible to use polling instead of transmission interrupts. In this case, the "waiting for INTCANxTXD" in above flowchart must be replaced by polling ***[CANxTA]***. Further, enabling interrupts and clearing ***[CANxMBTIF]*** must be removed from the flow.

5.3. Remote Frame Handling

Figure 5.3 shows an example flowchart of remote frame handling by using the automatic reply feature.

This feature is available when the ***[CANxMBnID]***<RFH> bit of the transmission mailbox is set to "1". To avoid data inconsistency when updating the mailbox data, the ***[CANxCDR]*** register controls transmission during data update of the mailbox.

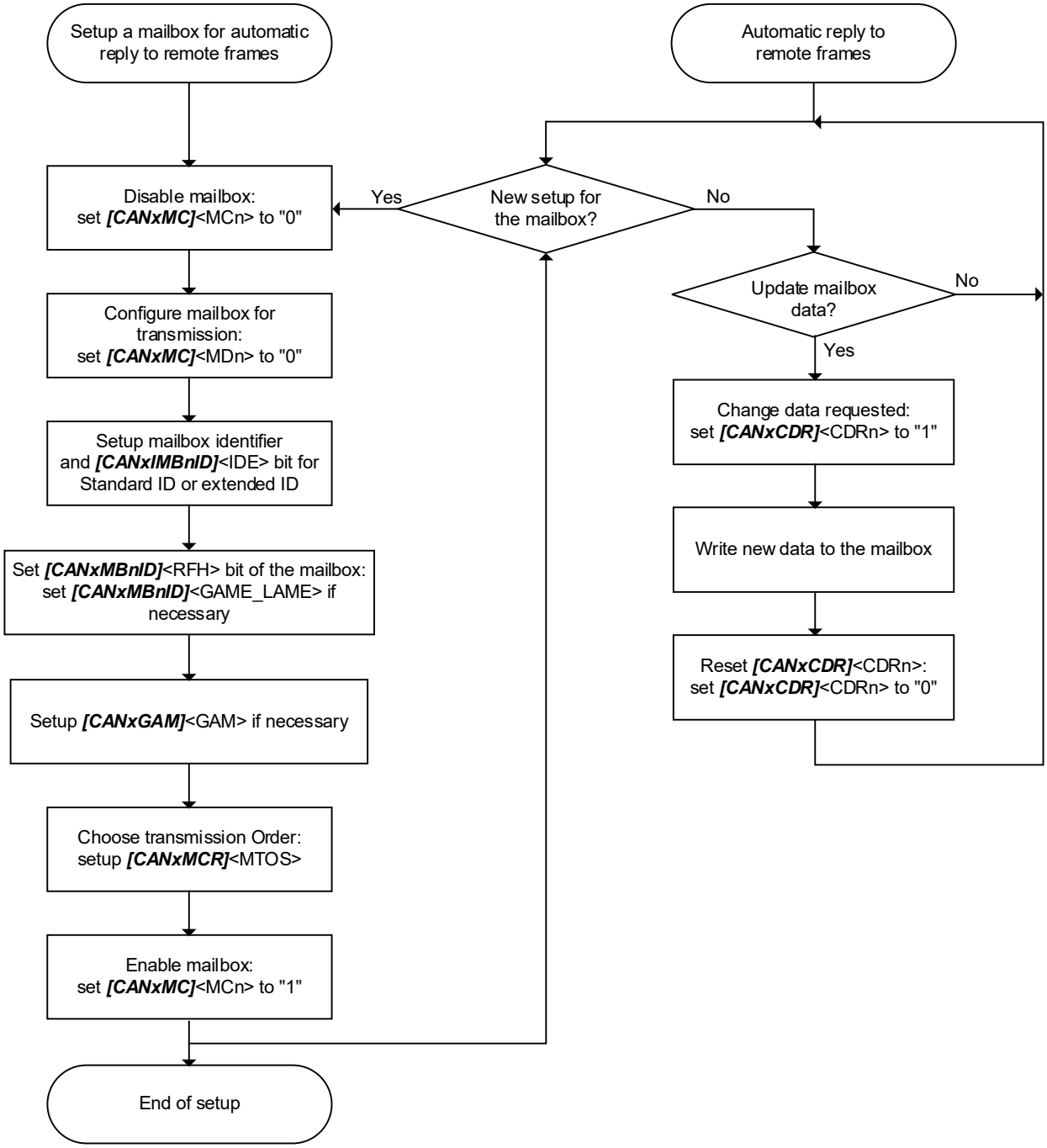

Figure 5.3 Flowchart of Remote Frame Handling Using the Automatic Reply Feature

6. Revision History

Table 6.1 Revision History

Revision	Date	Description
1.0	2020-10-01	New Release

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**