BUS INTERFACE

INVENTRA™

THE INTELLIGENT APPROACH TO INTELLECTUAL PROPERTY

MCAN2

CAN 2.0 NETWORK CONTROLLER

Programmer's Guide

05/06 PG-40107.003-FC

Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070.

For Customer Support on this product:

- Call up the Inventra Customer Inquiry Service at **http://www.mentor.com/supportnet**
- Email **support_net@mentor.com**
- Phone **1-877-763-8470** (toll-free in US, Mexico and Canada)
  (Customers in other parts of the world can access this service via the AT&T USA Direct service in their country.)

Full details are given in the Customer Support Handbook, provided in Adobe Acrobat format as **custhb.pdf** in the **/databook** directory on the Inventra™ Soft Cores CD. Please note the checklists of actions to take and information to have to hand when contacting Inventra Customer Support that are given in the Customer Support Handbook.

TABLE OF CONTENTS

1. INTRODUCTION

The Inventra™ MCAN2 is a stand-alone controller for the CAN Controller Area Network used in the automotive industry and in a number of other industrial environments. It provides an interface between a microprocessor and a CAN bus which carries out all the actions of data encoding/decoding (including serialisation/deserialisation of data, bit stuffing/unstuffing), message management (acceptance filtering, acknowledgement, error detection and signaling, and re-transmission), bit timing and synchronization involved in transmitting and receiving information over a CAN network.

The MCAN2 controller implements the BOSCH CAN message transfer protocols 2.0A and 2.0B. Specification 2.0A (which is equivalent to CAN 1.2) covers standard message formats (11-bit identifier); Specification 2.0B covers both standard and extended message formats (both 11-bit and 29-bit identifiers).

The MCAN2 is broadly compatible with a Philips SJA1000 working in its PeliCAN mode (with some exceptions, detailed in Section 8 of the MCAN2 Product Specification.).

The design has a synchronous PVCI[1]-compatible CPU interface for ease of connection to a range of microprocessor buses.

This document describes the software interface to the MCAN2 and describes how a typical application should drive the core. It should be read in conjunction with the BOSCH CAN specification version 2.0 (hereafter referred to as the CAN 2.0 specification).

Note: The CAN 2.0 specification uses the convention that, on the CAN bus, Recessive bits are logic '1' while Dominant bits are logic '0'. This convention is also used in the MCAN2 design.

2. MODES OF OPERATION

The MCAN2 has two main modes of operation: an Operating Mode in which data may be transmitted and received, and a Reset Mode in which bus timing parameters and message acceptance filters can be set. Reset Mode also allows the Receive and Transmit Error Counters and the Error Warning Limit to be changed.

Reset Mode is selected either by executing a hardware reset or by setting the Reset Mode bit in the Mode Register (MOD.0) to '1'. The MCAN2 is returned to Operating Mode by clearing the MOD.0 bit. *Note: Details of the Mode register may be found alongside the details of the other MCAN2 registers in Section 10.*

The MCAN2 also supports a Listen Only Mode and a Self Test Mode, selectable through the Mode Register in either Operating Mode or Reset Mode.

In Listen Only Mode, the MCAN2 is only able to receive data: no transmission is possible. The MCAN2 does not even transmit any acknowledgement of data being successfully received. It is also forced to be 'error passive'. *Note:* This mode allows software-driven bit-rate detection, which in turn allows the MCAN2 to support 'hot plug-in' of the MCAN2 device – see Section 9.

In Self Test Mode, the MCAN2 sends and receives a message using the MCAN2's Self Reception feature without looking for any acknowledgement from any remote node.

The device also offers a Clock Output Mode, only selectable within Reset Mode, in which TX1 is used to output the Transmit clock rather than a second copy of the transmission data.

[1] Peripheral Virtual Component Interface, as defined by the VSI Alliance™ Virtual Component Interface Standard (OCB 2 1.0).

3. CONFIGURATION

The way in which the MCAN2 is configured to operate is set within its Reset Mode, into which it is placed both immediately following a hardware reset (NRST taken low) and as a result of setting the Reset Mode bit in the Mode register (MOD.0) to 1 (software reset).

The details of the register settings following both these events are given in an Appendix at the end of this document.

While in Reset Mode, you may wish to set the following aspects of the MCAN2's operation:

- The bus timing parameters to be applied (These select the baud rate used on the CAN bus)
- The acceptance filters to be applied to received messages
- The required interrupts
- The desired error warning limit
- The required output mode – with either a copy of the transmission bit stream or the Transmit clock on TX1
- The relationship of the CLKOUT signal to the input clock.

Further details are given in the following sections.

Note: The MCAN2 needs to be taken out of Reset Mode before any data can be sent or received.

3.1. BUS TIMING PARAMETERS

The bus timing parameters configure the MCAN2 for the bit rate used on the CAN bus and set the point within each bit period at which the received bit stream is to be sampled. They also specify the degree to which the MCAN2 may compensate for variations in the bit rates generated by other nodes by re-synchronizing to the bit stream.

To cater for variations in the bit rate generated by other nodes and for physical delay times both on the bus and within the CAN nodes, the bit period is seen as being composed of a Synchronization segment, a Propagation segment and two Phase Buffers. The Synchronization segment represents the part of the bit period in which the bit edge is expected to arrive. The Propagation segment represents the part of the bit time that is allowed to compensate for physical delay times. The two Phase Buffers surround the sampling point and are shortened or lengthened as necessary to re-synchronize to the incoming bit stream when the bit edge arrives outside of the Synchronization segment.

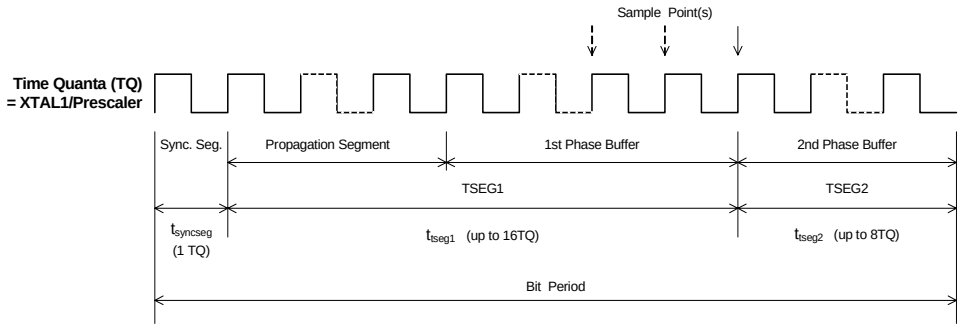
The length of each of these segments is defined as a number of 'Time Quanta' (TQ). The Synchronization segment is always 1 TQ, the Propagation segment may be 1 – 8 TQ, and the two Phase Buffers may be 1 – 8 TQ. The maximum amount by which the Phase Buffers can be lengthened or shortened is also defined – as the Synchronization Jump Width. This is limited to 1 – 4 TQ and is also required to not be longer than either of the two Phase Buffers.

The timing parameters used on the MCAN2 are selected through the two Bus Timing Registers: BTR0 and BRT1.

BTR0 defines the time quantum to be used in terms of periods of the XTAL1 input clock, together with the Synchronization Jump Width (in time quanta). Time quanta between 2 x the XTAL1 clock period and 128 x the XTAL1 clock period are supported.

BTR1 defines the lengths in time quanta of two time segments, TSEG1 and TSEG2, and the number of samples made of each bit period – 1 or 3 (1 is recommended for high-speed (class C) buses; 3 is recommended for low/medium (class A or B) buses). TSEG1 represents the time between the Synchronization segment and the sample point (i.e. the Propagation segment plus the first Phase Buffer). TSEG2 represents the time between the sample point and the end of the bit period (i.e. the second Phase Buffer).

General Structure of a Bit Period



TSEG1 can be between 1 and 16 TQ long while TSEG2 can be between 1 and 8 TQ long. In theory, bit periods can therefore be defined between 3 and 25 TQ in length. In practice, however, they are required to be in the range 8 and 25 TQ picked out by the BOSCH standard.

Details of the BTR0 and BTR1 registers are given in Sections 10.4 and 10.5.

3.2. ACCEPTANCE FILTERS

Within a CAN network, all nodes receive all messages transmitted on the bus.

To allow a node to ignore messages that are not relevant to it, the MCAN2 provides a 4-byte Acceptance Filter, which can be used to pick out only those messages with an appropriate identifier. Any message that does not pass though this filter can be discarded as not applicable to the receiving CAN node.

Normally message filtering is based upon the whole identifier, which can be 11 or 29 bits long depending if the received message is a standard or extended frame format. However, in the MCAN2, optional mask registers allow groups of identifiers to be received and placed into the Receive FIFO by setting particular identifier bits to be 'don't care'.

The filter can be applied either as a single 4-byte filter or as two shorter filters. The selection is made through the AFM bit of the Mode register (bit 3). If AFM = '1', a single filter will be applied; if AFM = '0', two filters will be applied. Where two filters are used, the incoming message is accepted if its identifier matches either filter.

The filters applied are defined in a set of Acceptance Code Registers ACR0 – 3, used in conjunction with a corresponding set of Acceptance Mask Registers AMR0 – 3 (see Sections 10.1 and 10.2). The bit pattern against which the message identifier is matched is recorded in the ACR registers, masked by the values recorded in the AMR registers. '0's in AMR0 – 3 identify the bits at the corresponding positions in ACR0 – 3 which must be matched in the message identifier, '1's identify the corresponding bits as 'don't care'. Both groups of registers are set to zero by a hardware reset (i.e. set to accept only messages with a zero identifier) but are left unchanged by a software reset.

The way in which the bit patterns defined by ACR0 – 3 are applied further depend on whether the incoming message is in Standard Frame Format (SFF) or Extended Frame Format (EFF) as shown on the following page:

Standard Frame Format, Single Filter

Receive Buffer: Address 11h	*12h*		*13h*	*14h*	
ID.28 .... ID.21	ID.20 ... ID18	RTR	X X X X *(not matched)*	Data Byte 1	Data Byte 2

Filter:

ACR0[7:0]	ACR1[7:4]	*(ACR1[3:0] unused)*	ACR2[7:0]	ACR3[7:0]
AMR0[7:0]	AMR1[7:4]	*(AMR1[3:0] unused)*	AMR2[7:0]	AMR3[7:0]

Note: If matching of the data bytes is not required, AMR2 and AMR3 should be set to FFh.

Standard Frame Format, Dual Filters

Receive Buffer: Address 11h	*12h*		*13h*		*14h*
ID.28 .... ID.21	ID.20 ... ID18	RTR	X X X X *(not matched)*	Data Byte 1 [7:4]	Data Byte 1 [3:0]	Data Byte 2 *(not matched)*

Filter 1:

ACR0[7:0]	ACR1[7:4]
AMR0[7:0]	AMR1[7:4]

ACR1[3:0]	ACR3[3:0]
AMR1[3:0]	AMR3[3:0]

Filter 2:

ACR2[7:0]	ACR3[7:4]
AMR2[7:0]	AMR3[7:4]

Extended Frame Format, Single Filter

Receive Buffer: Address 11h	*12h*	*13h*	*14h*		
ID.28 .... ID.21	ID.20... ID.13	ID.12 ... ID.5	ID.4 ... ID0	RTR	X X *(not matched)*

Filter:

ACR0[7:0]	ACR1[7:0]	ACR2[7:0]	ACR3[7:2]	*(ACR3[1:0] unused)*
AMR0[7:0]	AMR1[7:0]	AMR2[7:0]	AMR3[7:2]	*(AMR3[1:0] unused)*

Extended Frame Format, Dual Filters

Receive Buffer: Address 11h	*12h*	*13h*	*14h*		
ID.28 .... ID.21	ID.20... ID.13	ID.12 ... ID.5 *(not matched)*	ID.4 ... ID0 *(not matched)*	RTR *(not matched)*	X X *(not matched)*

Filter 1:

ACR0[7:0]	ACR1[7:0]
AMR0[7:0]	AMR1[7:0]

Filter 2:

ACR2[7:0]	ACR3[7:0]
AMR2[7:0]	AMR3[7:0]

3.3. INTERRRUPTS

The MCAN2 supports the generation of an interrupt for any of the following conditions:

- Bus activity while the MCAN2 is in Sleep Mode (Wake-Up Interrupt)
- Receipt of a message (Receive Interrupt)
- Completion of the current transmission (Transmit Interrupt)
- Loss of Received data through the FIFO being full (Data Overrun Interrupt)
- Loss of Arbitration on the CAN bus* (Arbitration Loss Interrupt)
- Error on the CAN bus* (Bus Error Interrupt)
- MCAN2 coming out of 'Error Passive' state (Error Passive Interrupt)
- The number of errors either exceeding the Error Warning Limit or causing the device to go into Bus Off state (Error Warning Interrupt)

Following a hardware reset, these interrupts are disabled. The user therefore needs to enable the ones they require in the Interrupt Enable Register (described in Section 10.11). The selection of interrupts that are enabled is not however affected by a software reset.

* **Note:** There is not normally any need for the CPU to monitor Loss of Arbitration events or individual bus errors as the MCAN2 will automatically retry transmission when these happen.

3.4. ERROR WARNING LIMIT

The Error Warning Limit (EWL) represents the number of errors in either reception or transmission at which a warning should be generated. When either the Transmit Error Counter or the Receive Error Counter passes this value, the Error Status bit in the Status Register (SR.6) is set and an Error Warning Interrupt is generated (if enabled).

The value for the EWL is recorded in the Error Warning Limit Register (described in Section 10.9). The value selected following a hardware reset is 96 which, if reached, would indicate a seriously disturbed bus.

The current setting is left unchanged by a software reset.

Note: The way in which the Transmit Error and Receive Error Counters operate is described in Sections 10.16 and 10.20.

3.5. OUTPUT MODE

The MCAN2 supports two possible output driver configurations: 'Normal Output' and 'Clock Output'. **Note:** The additional driver configurations available in the SJA1000 through this register are not supported by the MCAN2.

In Normal Output Mode, the bit sequence (TXD) is sent to TX0 with an inverse copy sent to TX1. In Clock Output Mode, the bit sequence is output on the TX0 signal as in Normal Output Mode but the data stream to TX1 is replaced by a copy of the Transmit clock (TXCLK), the rising edge of which marks the beginning of a bit period.

Normal Output Mode is automatically selected following a hardware reset. If Clock Output Mode is required, it may be selected through the Output Control Register (described in Section 10.13).

The selected mode is left unchanged by a software reset.

3.6. CLKOUT SIGNAL

The CLKOUT signal is derived from the XTAL1 input clock.

The relationship between the CLKOUT signal and the XTAL1 clock is defined by the Clock Divider Register (described in Section 10.6). The bottom three bits of this register specify a divisor for the XTAL1 clock between 2 and 14, while bit 3 of the register enables or disables the CLKOUT signal as required.

After a hardware reset, the Clock Divider Register is set so that the CLKOUT signal is enabled and equal to XTAL1 divided by 2.

The current setting is left unchanged by a software reset.

3.7. EXAMPLE CONFIGURATION STEPS

Address	Register	Action
1Fh	Clock Divider	Set CLKOUT frequency, derived from XTAL1 input.
08h	Output Control Register	Configure the output driver for outputs TX0 and TX1
04h	Interrupt Enable	Enable required interrupts
00h	Mode	Select message filter type
10h-13h	Acceptance Code Registers	Set to select required range of identifiers
14h-17h	Acceptance Mask Registers	
06h	Bus Timing Register:0	Set SJW and clocks per Time Quanta
07h	Bus Timing Register:1	Set Tseg1 and Tseg2, and therefore sample point of bit period
00'h	Mode	Release Reset Mode

4. TRANSMISSION

To transmit a message, the MCAN2 must be in its Operating Mode (MOD.0 = 0).

Enabling the Transmit and the Error Warning interrupts is recommended but it is usually unnecessary to enable the Arbitration Loss and Bus Error interrupts as the MCAN2 will automatically try again to send the message if bus arbitration is lost or if transmission errors occur while the message is being sent.

The data to be transmitted should be written to the MCAN2's Transmit Buffer. The Transmit Buffer comprises the 13 bytes between CAN addresses 10h and 1Ch, giving space for one message frame containing up to eight bytes of data. *Note:* Before writing to the buffer, the Transmit Buffer Status (Status Register, bit 2 – see Section 10.18) should be checked to ensure that the buffer is 'released' (SR.2 = '1'). Any byte written to the buffer when the buffer is locked (SR.2 = '0') will be lost without any indication.

The format of the data may be either Standard Frame Format (SFF) or Extended Frame Format (EFF) as required. The details of these message formats are given in Section 10.19.

Transmission of the data is then initiated by issuing a Transmit Request through the Command Register (by setting CMR.0 = '1'). The MCAN2 then starts a sequence of steps in which it prepares the data for transmission over the bus (including generating the appropriate 15-bit CRC), waits for the CAN bus to become idle then starts transmission. The moment transmission starts, the Transmit Status (SR.5) changes to '1' and the Transmission Request bit (CMR.0) is cleared.

Once transmission is in progress, the CPU simply has to wait for a Transmit Interrupt to occur (if enabled) or for the Transmit Buffer to be released (SR.2 = '1') to indicate that transmission has finished. If bus arbitration is lost or bus errors occur, the MCAN2 will automatically retry transmission.

At any time prior to its transmission being started, a message can be aborted by issuing an Abort Transmission command through the Command Register (CMR.1 = '1' – see Section 10.7). The message cannot however be aborted after transmission is started, nor can it be aborted by clearing the Transmission Request bit.

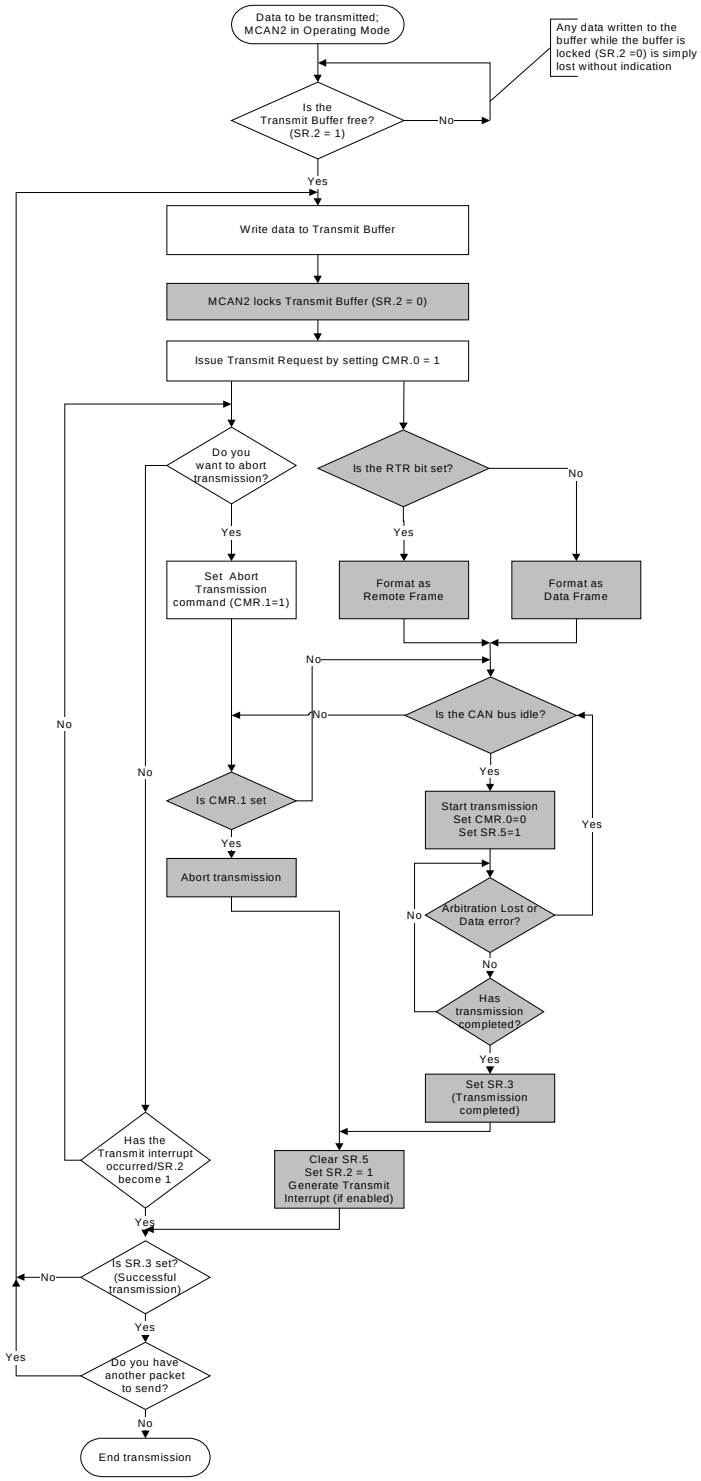
After aborting the current message, the MCAN2 releases the Transmit Buffer (SR.2 goes to '1') and generates a Transmit Interrupt (if enabled), as after completing transmission of the message. To check whether the original message was transmitted or aborted, the CPU should check the Transmission Complete Status bit (SR.3).

A flow diagram showing the different actions that may be taken in transmitting a message is shown on the following page.

Note: If a 'Single Shot' transmission of the current message is required – without any retry in the event of either bus errors or loss of bus arbitration – issue an Abort Transmission command (CMR.1 = '1') at the same time as the Transmit Request (CMR.0 = '1').

Transmission Flow

Note: Actions that the MCAN2 carries out are shown against a gray background.



5. RECEPTION

To receive messages, the MCAN2 must be in its Operating Mode (MOD.0 = 0).

Reception is initiated by the MCAN2 detecting a Start of Frame.

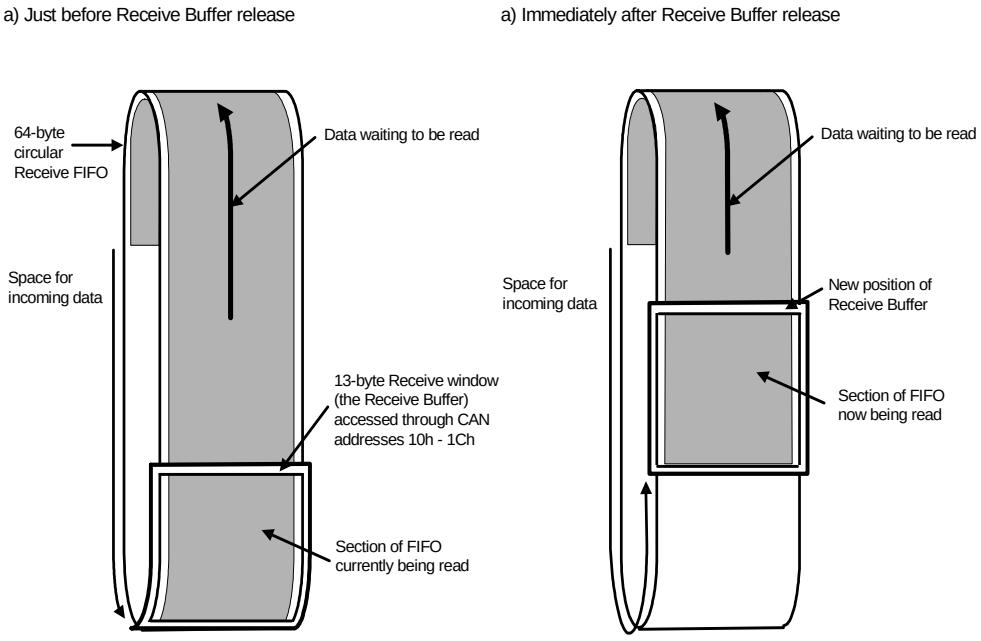
The data received by the MCAN2 is first filtered by the Acceptance Filter (see Section 3.2). If the message identifier bits match the ones stored in the Acceptance Filter registers, the message is passed to the Receive FIFO.

The Receive FIFO is 64 bytes deep and is used in a circular fashion, allowing space for up to five full Extended Frame Format (EFF) messages – one being read and four waiting to be read.

Data placed in the Receive FIFO is read through a 13-byte window referred to as the Receive Buffer. This window is located at CAN addresses 10h – 1Ch i.e. it occupies the same address space as the Transmit Buffer. Like the Transmit Buffer, it is wide enough to accommodate one message containing up to eight bytes of data.

A count of the number of messages currently available in the Receive FIFO is given by the Receive Message Counter register (see Section 10.17).

Receive Buffer window on Receive FIFO.



The moment data starts being placed in the Receive FIFO, the Receive Status bit in Status Register (SR.4 – see Section 10.18) goes to '1'. Then once the data has been received, the Receive Buffer Status bit (SR.0) goes to '1' and a Receive Interrupt is generated (if enabled). Either the generation of the Receive Interrupt or the Receive Buffer Status bit becoming set should be used to trigger the CPU to read from the Receive Buffer.

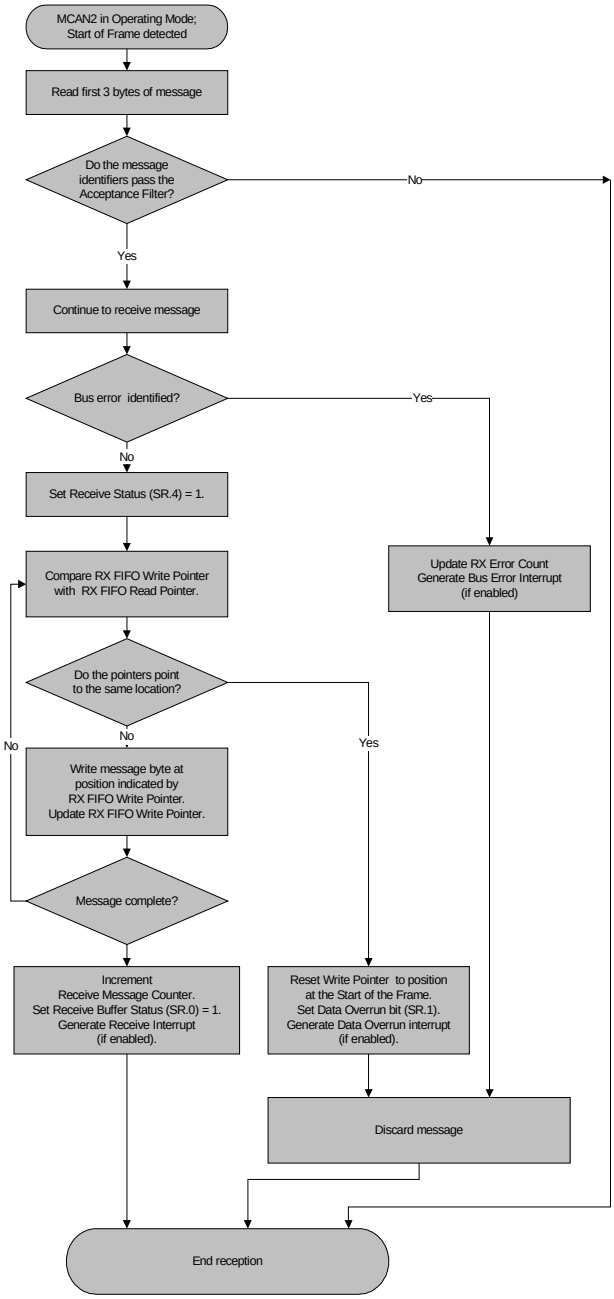
As each message is read from the Receive FIFO, the host CPU needs to release the window it currently has on the FIFO by issuing the Release Receive Buffer command through the Command Register (CMR.2 = '1' – see Section 10.7). If another message is waiting to be read in the Receive FIFO, this will immediately become available through the Receive Buffer. If no message is waiting, the Receive Status bit, the Receive Interrupt bit and the Receive Buffer Status bit will all be cleared.

The data received is written at the position indicated by the RX FIFO Write Pointer with the pointer updated as each byte is written. The position at which data is currently being read is given by the RX FIFO Read Pointer. When the Write Pointer coincides with the Read Pointer, there is no longer space in the FIFO for the data being received. The data frame being received is discarded, the Data Overrun Status bit in the Status Register (SR.1) is set and a Data Overrun Interrupt is generated (if enabled). It is up to the CPU to recover from this loss of data.

A flow diagram showing the different actions in receiving a message is shown below. The actions involved in reading the received message are described in Section 7.1 below.

Reception Flow

Note: All these actions are carried out by the MCAN2.



6. SELF-RECEPTION

A feature of the MCAN2 is that it allows the message it is transmitting to another CAN node to be simultaneously received by the MCAN2.

Self-reception of the current transmit message is selected by issuing a Self Reception Request through the Command Register (CMR.4) – see Section 10.7. The MCAN2 automatically generates the Transmit and Receive Interrupts required for correct operation.

Note: (i) If self-reception is requested at the same time as normal transmission (i.e. CMR.0 and CMR.4 set simultaneously), the Self Reception Request is ignored.

(ii) A special version of the Self-Reception feature is offered by the MCAN2's Self-Test Mode in which a test message is both sent and received, but without requiring an acknowledgement from a remote node. This therefore allows a full test of the node containing the MCAN2 without needing any other active node on the bus.

7. INTERRUPT HANDLING

When the CPU is interrupted (NINT going low), it needs to read the interrupt register to determine which type of event caused the interrupt.

The possible interrupts are (in the order they appear in the Interrupt register):

- Receive Interrupt (IR.0 set)
- Transmit Interrupt (IR.1 set)
- Error Warning Interrupt (IR.2 set)
- Data Overrun Interrupt (IR.3 set)
- Wake-Up Interrupt (IR.4 set)
- Error Passive Interrupt (IR.5 set)
- Arbitration Loss Interrupt (IR.6 set)
- Bus Error Interrupt (IR.7 set)

The following sections describe the actions to be taken in response to each type of interrupt.

Note: Reading the Interrupt Register (address 03h) automatically clears all the interrupts in the register with the exception of the Receive Interrupt.

7.1. RECEIVE INTERRUPT

The generation of a Receive Interrupt indicates the availability of a message to be read in the Receive FIFO.

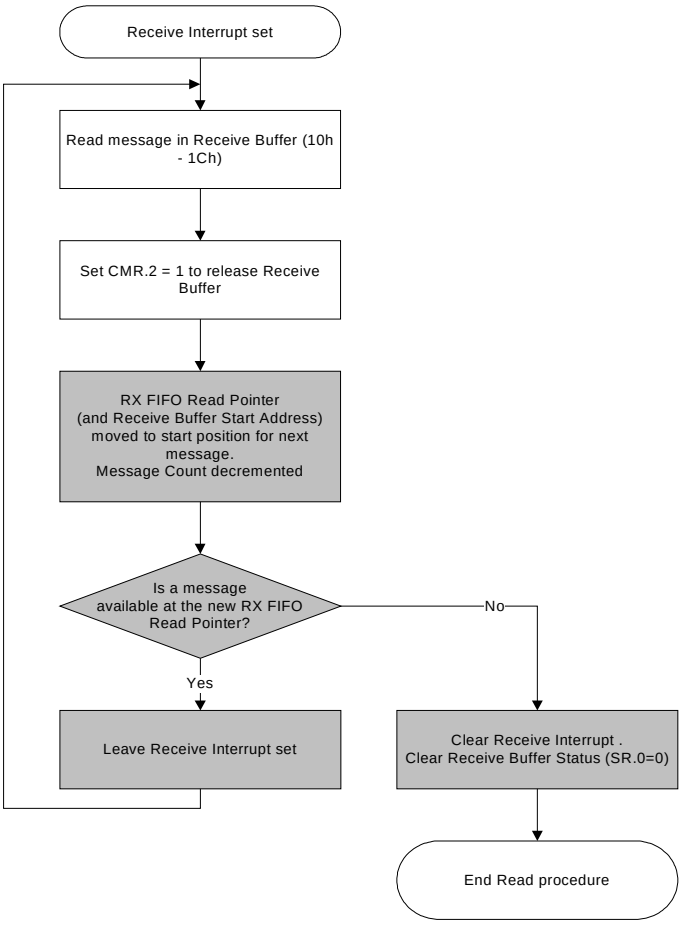
The message is read through a 13-byte window onto the Receive FIFO referred the Receive Buffer, which is located at CAN addresses 10h – 1Ch.

Once the message currently accessible through the Receive Buffer has been read, the CPU needs to release the window it currently has on the FIFO by issuing a Release Receive Buffer command (CMR.2 = '1'). The RX FIFO Read Pointer (and hence the Receive Buffer Start Address) then moves to the position in the Receive FIFO at which the next message will start.

If there is an unread message at this position, this becomes immediately available to read through the Receive Buffer. If no message is available, the Receive Interrupt (IR.0) and Receive Buffer Status (SR.0) bits will be cleared.

Read Flow

Note: Actions that the MCAN2 carries out are shown against a gray background.



7.2. TRANSMIT INTERRUPT

The generation of a Transmit Interrupt indicates the readiness of the Transmit Buffer to receive another message for transmission. The response made to this interrupt simply depends on whether there is further data to be sent. If there is, the transmission procedure outlined in Section 4 needs to be repeated. If not, the interrupt may be ignored.

7.3. ERROR WARNING INTERRUPT

The generation of an Error Warning interrupt indicates either that the count of transmission errors or the count of reception errors has passed the EWL value recorded in the Error Warning Limit register, or that the MCAN2 has been put into Bus Off state because the number of transmission errors has exceeded 255.

The count of reception errors is recorded in the RXERR register, the count of transmission errors is recorded in the TXERR

register. (These registers are described in Sections 10.14 and 10.19.)

If the MCAN2 has been placed in Bus Off state, the Bus Status bit (SR.7) will be set to '1' (Bus Off). In addition, the Reset Mode bit (MOD.0) will have been set, causing a software reset and placing the MCAN2 in Reset Mode where it will then stay until the host CPU clears the Reset Mode bit in the Mode Register (MOD.0).

Furthermore, on its return to Operating Mode, the MCAN2 will wait for 128 occurrences of the Bus Free sequence of 11 successive recessive bits (the minimum time defined by the CAN protocol) before becoming 'Bus On' again. *Note:* During this period, the progress that is being made towards Bus On can be monitored by reading the TXERR register. On leaving Reset Mode, this is initially set to 127. It then counts down through the required number of Bus Free sequences to become zero at the point when the device is allowed to become Bus On again.

If the interrupt has been generated as a result of the EWL value being exceeded, it is up to the programmer what action is taken in response to the generation of this interrupt.

7.4. DATA OVERRUN INTERRUPT

A Data Overrun Interrupt is only generated when the required storage space for the received message is greater than the number of free bytes in the Receive FIFO. The Data Overrun Status bit (SR.1) will also be set.

The required storage space is determined from the RTR, FF and DLC bits of the received message which respectively define whether the message is a Remote Transmission Request, whether it is a standard frame format or extended frame format message, and the number of bytes included in the message.

The assessment of the space required is made after the message has been received. If insufficient space is available to store the message, the message will be lost.

The recovery that can be made when messages are lost will depend on the system design. However, experiencing significant numbers of Data Overrun events would suggest that the volume of data traffic has been under-estimated and that the system would benefit from a larger memory buffer for incoming messages.

Note: Any interrupt routine that handles Data Overrun events should finish by issuing a Clear Data Overrun command (CMR.3 = 1) to clear the Data Overrun Status bit, because no further Data Overrun interrupt will be generated while the Data Overrun Status bit remains set.

7.5. WAKE-UP INTERRUPT

A Wake-Up Interrupt is generated when the MCAN2 is awakened from Sleep Mode.

Any of the following events will cause the MCAN2 to 'wake up' from Sleep Mode:

- Clearing the Sleep Mode bit (MOD.4)
- A low on NINT_IN
- Activity on the CAN bus input (RX0)

It is up to the CPU to identify why the device has been awoken, for example by first reading the Mode register then testing the level of NINT_IN.

There is more about Sleep Mode in Section 8.

7.6. ERROR PASSIVE INTERRUPT

The Receive Error (RXERR) and Transmit Error (TXERR) counters are respectively automatically incremented by one each time a Receive error or Transmit error occurs, and decremented by one by each successful reception or transmission.

If the accumulated total of either Receive or Transmit Errors goes over 127, the MCAN2 goes into state in which further errors continue to be counted but individual interrupts are no longer generated. This state is described as 'Error Passive' and an Error Passive Interrupt is generated (if enabled) to signal that the Error Passive state has been entered.

The MCAN2 remains in Error Passive state while either error count remains over 127. The Transmission Error count continues to be incremented and decremented while it remains over 127. The Receive Error count, however, is automatically reduced to a value between 119 and 127 by each message that is successfully received, potentially taking the MCAN2 out of Error Passive state.

A further Error Passive Interrupt is generated when the MCAN2 leaves Error Passive state.

7.7. ARBITRATION LOSS INTERRUPT

The generation of an Arbitration Loss Interrupt indicates that the MCAN2 has lost control of the CAN bus while it was in the process of transmitting a message.

Normally, there is no need for any special action to be taken as the MCAN2 will automatically try again to transmit the current message. The fact that arbitration has been lost may however be of importance if the option of a One-Shot transmission has been taken (see Section 4).

The bit position at which arbitration was lost will be recorded in the Arbitration Lost Capture (ALC) Register. For details of the way in which this bit position is recorded, see Section 10.3.

Note: The details recorded in the Arbitration Lost Capture Register are not cleared until this register has been read. As a result, no further information about arbitration loss can be recorded until the previously recorded details have been read from the ALC Register.

7.8. BUS ERROR INTERRUPT

The generation of a Bus Error Interrupt indicates the occurrence of a transmission error on the CAN bus.

Normally, there is no need for any special action to be taken as the MCAN2 will automatically discard any incoming message in which bus errors have occurred and it will automatically try to send again any transmit message that experienced bus errors. However, should additional information on a bus error be required, the type of error (bit/form/stuff/other) and the location of the each error are captured in an Error Code Capture Register (described in Section 10.8) where they remain until this register is read.

Experiencing significant numbers of such errors may however indicate that corrective action should be taken, so the MCAN2 maintains two error counters – one for reception errors (RXERR) and one for transmission errors (TXERR) – which are automatically incremented whenever an error occurs. Should either counter exceed the value recorded in the Error Warning Limit register, an Error Warning interrupt is generated (if enabled) while if either counter exceeds a count of 127, an Error Passive interrupt is generated (if enabled). An Error Warning interrupt will also be generated if the MCAN2 goes into Bus Off state as a result of the count of transmission errors exceeding 255.

Both these additional interrupts are discussed above.

8. SLEEP MODE

When there is no bus activity and no interrupts are pending, power can be saved by putting the MCAN2 into a Sleep Mode in which XTAL1_IN is turned off. This is selected by setting the Sleep Mode bit in the Mode Register (MOD.4) to '1'.

Any of the following events will cause the MCAN2 to 'wake up' from Sleep Mode:

- Setting the Sleep Mode bit to '0'
- Activity on the CAN bus input (RX0)
- A low on NINT_IN

On waking up, the MCAN2 will generate a Wake-Up Interrupt.

Note: If the MCAN2 is awakened by bus activity, it cannot receive any message until after it has detected a Bus-Free sequence of 11 recessive bits on the bus. You should also note that it is not possible to select Sleep Mode while the MCAN2 is in Reset Mode.

9. HOT PLUG-IN SUPPORT

The bit-rate that is being used on the CAN bus can be determined automatically by using the MCAN2 in its Listen Only Mode.

The procedure is to sample messages on the CAN bus at different bit rates until a Receive Interrupt is generated, indicating that a message has been received without any bus errors.

Carrying out this procedure requires there to be at least one node on the bus sending CAN messages and another node receiving these messages. The second node is required in order for transmissions to be acknowledged: if no acknowledge is detected, a bus error will be reported even if the rest of the message was received correctly.

It also requires a pre-defined table within the software that lists the settings of BTR0 and BTR1 corresponding to the different bit rates that you intend to test.

The routine used needs first to go into Reset Mode to enable Bus Error and Receive Interrupts by setting IER.0 and IER.7, and to set the Acceptance Masks to all 1s (so that any message transmitted on the CAN bus may be received). It then needs to select the BTR0 and BTR1 settings corresponding to the first bit rate in the table, select Listen Only mode (from within Reset Mode) and wait for an interrupt to be received.

When the interrupt is received, the routine needs to test if it is a Receive Interrupt (IR.0 = 1).

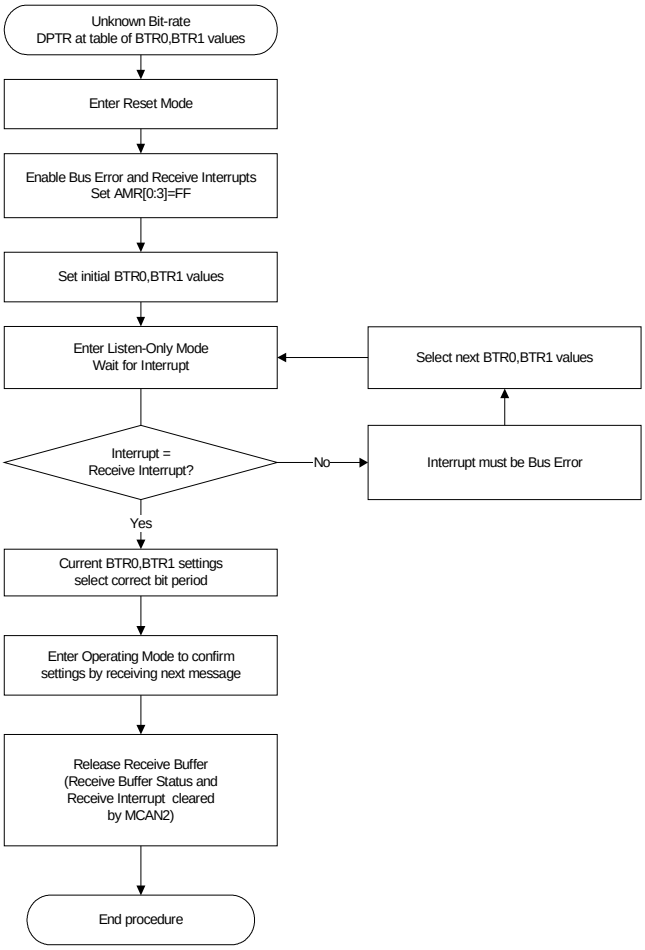
If the interrupt received is a Receive Interrupt, the message has been received correctly – which in turn means that the bit rate selected matches the bit rate being used on the CAN bus.

If the interrupt received is not a Receive Interrupt, it must be a Bus Error Interrupt because that is the only other interrupt enabled. Bus errors indicate that the bit rate currently selected does not match the bit rate being used on the CAN bus. The routine needs to select the BTR0 and BTR1 settings corresponding to the next bit rate in the table then repeat the process.

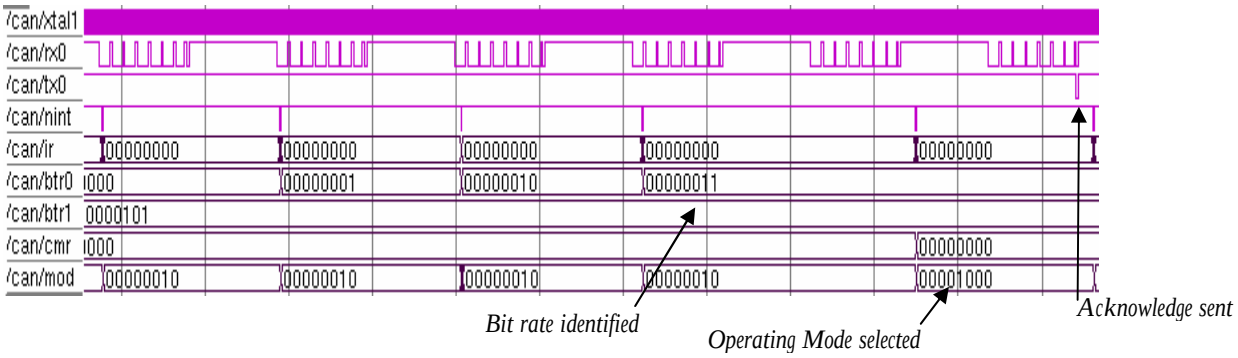
As a last step, the routine could switch to Operating mode to further confirm that the correct bit rate has been selected by responding to the next message delivered over the CAN bus.

The flow of this example routine is illustrated on the following page together with an example trace showing the additional steps taken when the bit rate has been identified.

OUTLINE OF PROCEDURE



Example section of trace



'rx0' = Receive data
'tx0' = Transmit data

'nint' = Interrupt signal
'ir' = Interrupt register
'btr0' and 'btr1' = Timing registers

'cmr' = Command register
'mod' = Mode register

10. REGISTER DESCRIPTION

The registers used in the MCAN2 are listed below, with detailed information about the individual registers given in the following sections (referenced in the table). Note: Different Read/Write permissions apply depending on whether the MCAN2 is in Operating Mode or Reset Mode.

Address	Register Name		See Section	Operating Mode	Reset Mode	Comment
00h	MOD	Mode	10.12	Read/Write	Read/Write	
01h	CMR	Command	10.7	Write only	Write only	Returns 00h when read.
02h	SR	Status	10.18	Read only	Read only	
03h	IR	Interrupt	10.10	Read only	Read only	
04h	IER	Interrupt enable	10.11	Read/Write	Read/Write	
05h		*Reserved*		N/A	N/A	Returns 00h when read.
06h	BTR0	Bus Timing 0	10.4	Read only	Read/Write	
07h	BTR1	Bus Timing 1	10.5	Read only	Read/Write	
08h	OCR	Output Control Register	10.13	Read only	Read/Write	
09h		*Reserved*		N/A	N/A	
0Ah		*Reserved*		N/A	N/A	Returns 00h when read.
0Bh	ALC	Arbitration Lost Capture	10.3	Read only	Read only	
0Ch	ECC	Error Code Capture	10.8	Read only	Read only	
0Dh	EWLR	Error Warning Limit	10.9	Read only	Read/Write	
0Eh	RXERR	Receive Error Counter	10.16	Read only	Read/Write	
0Fh	TXERR	Transmit Error Counter	10.20	Read only	Read/Write	
10h	Transmit Buffer	{ Transmit Frame Information	10.19	Write	(see below)	Read back from 60h.
11h – 1Ch		{ Transmit Data Information	10.19	Write	"	Read back from 61h – 6Ch.
10h	Receive Window	{ Receive Frame Information	10.14	Read	"	
11h – 1Ch		{ Receive Data Information	10.14	Read	"	
10h – 13h	ACR0–3	Acceptance Code Registers 0 – 3	10.1	(see above)	Read/Write	***Note:*** *18h – 1Ch reserved in Reset Mode (return 00h when read)*
14h – 17h	AMR0–3	Acceptance Mask Registers 0 – 3	10.2	"	Read/Write	
1Dh	RMC	Receive Message Counter	10.17	Read only	Read only	
1Eh	RBSA	Receive Buffer Start Address	10.15	Read only	Read/Write	
1Fh	CDR	Clock Divider	0	Read/Write	Read/Write	
20h – 5Fh		Receive FIFO	10.14	Read only	Read/Write	
60h – 6Ch		Transmit Buffer	10.19	Read only	Read only	
6Dh – 7Fh		*Reserved*		N/A	N/A	Return 00h when read.

10.1. ACCEPTANCE CODE REGISTERS (ACR0 – ACR3): ADDRESS 10h – 13h

These 8-bit registers record the bit patterns used by the Acceptance Filter in conjunction with the masks provided by AMR0 – AMR3 in filtering received data.

The way in which these bit patterns are applied depends on whether a single filter or dual filters are being used and on whether the data is in Standard Frame Format (SFF) or Extended Frame Format (EFF). See Section 3.2.

These registers can only be accessed in Reset Mode.

10.2. ACCEPTANCE MASK REGISTERS (AMR0 – AMR3): ADDRESS 14h – 17h

These 8-bit registers record the mask patterns applied by the Acceptance Filter in filtering the data received. '0's in these registers identify the bits of the incoming data bytes that are required to match the bit values in the corresponding Acceptance Code Registers. '1's mark individual bits as 'don't care'.

The bits of the incoming data picked out by these masks depends on whether a single filter or dual filters are being used and on whether the data is in Standard Frame Format (SFF) or Extended Frame Format (EFF). See Section 3.2.

The registers can only be accessed in Reset Mode.

10.3. ARBITRATION LOST CAPTURE REGISTER (ALC): ADDRESS 0Bh

This read-only register records the bit position at which arbitration was lost.

When bus arbitration lost, an Arbitration Lost Interrupt is generated (if enabled) and the current position of the Bit Processor is captured into this Arbitration Lost Capture Register. The contents of this register are then maintained until the register has been read by the user's software. The capture mechanism is then activated again.

BIT	COMMENT
ALC[7:5]	*Reserved. Return zero when read.*
ALC[4:0]	See table below.

ALC[4:0]	DECIMAL VALUE	FUNCTION
0 0 0 0 0	00	Arbitration lost in 1st bit of identifier (ID.28).
0 0 0 0 1	01	Arbitration lost in 2nd bit of identifier (ID.27).
0 0 0 1 0	02	Arbitration lost in 3rd bit of identifier (ID.26).
0 0 0 1 1	03	Arbitration lost in 4th bit of identifier (ID.25).
0 0 1 0 0	04	Arbitration lost in 5th bit of identifier (ID.24).
0 0 1 0 1	05	Arbitration lost in 6th bit of identifier (ID.23).
0 0 1 1 0	06	Arbitration lost in 7th bit of identifier (ID.22).
0 0 1 1 1	07	Arbitration lost in 8th bit of identifier (ID.21).
0 1 0 0 0	08	Arbitration lost in 9th bit of identifier (ID.20).
0 1 0 0 1	09	Arbitration lost in 10th bit of identifier (ID.19).
0 1 0 1 0	10	Arbitration lost in 11th bit of identifier (ID.18).
0 1 0 1 1	11	Arbitration lost in SRTR bit [1].
0 1 1 0 0	12	Arbitration lost in IDE bit.

ALC[4:0]	DECIMAL VALUE	FUNCTION
0 1 1 0 1	13	Arbitration lost in 12th bit of identifier (ID.17)
0 1 1 1 0	14	Arbitration lost in 13th bit of identifier (ID.16)
0 1 1 1 1	15	Arbitration lost in 14th bit of identifier (ID.15)
1 0 0 0 0	16	Arbitration lost in 15th bit of identifier (ID.14)
1 0 0 0 1	17	Arbitration lost in 16th bit of identifier (ID.13)
1 0 0 1 0	18	Arbitration lost in 17th bit of identifier (ID.12)
1 0 0 1 1	19	Arbitration lost in 18th bit of identifier (ID.11)
1 0 1 0 0	20	Arbitration lost in 19th bit of identifier (ID.10)
1 0 1 0 1	21	Arbitration lost in 20th bit of identifier (ID.9)
1 0 1 1 0	22	Arbitration lost in 21st bit of identifier (ID.8)
1 0 1 1 1	23	Arbitration lost in 22nd bit of identifier (ID.7)
1 1 0 0 0	24	Arbitration lost in 23rd bit of identifier (ID.6)
1 1 0 0 1	25	Arbitration lost in 24th bit of identifier (ID.5)
1 1 0 1 0	26	Arbitration lost in 25th bit of identifier (ID.4)
1 1 0 1 1	27	Arbitration lost in 26th bit of identifier (ID.3)
1 1 1 0 0	28	Arbitration lost in 27th bit of identifier (ID.2)
1 1 1 0 1	29	Arbitration lost in 28th bit of identifier (ID.1)
1 1 1 1 0	30	Arbitration lost in 29th bit of identifier (ID.0)
1 1 1 1 1	31	Arbitration lost in RTR bit

Extended Frame Format messages only

Note 1: RTR Bit in Standard Frame Format messages.

10.4. BUS TIMING REGISTER 0 (BTR0): ADDRESS 06h

Bus Timing Register 0 defines the values of the Synchronization Jump Width (SJW) and the Baud Rate Prescaler (BRP).

BTR0.7	BTR0.6	BTR0.5	BTR0.4	BTR0.3	BTR0.2	BTR0.1	BTR0.0
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

10.4.1. SYNCHRONIZATION JUMP WIDTH (SJW): BTR0[7:6].

The Synchronization Jump Width defines the maximum number of time quanta by which a bit period may be shortened or lengthened in attempting to re-synchronize on the relevant signal edge (recessive to dominant) of the current transmission.

10.4.2. BAUD RATE PRESCALER (BRP): BTR0[5:0]

The Baud Rate Prescaler defines the 'time quantum' TQ of the CAN clock as a multiple of the XTAL1 input clock period. The time quantum of the CAN clock is given by:

$TQ = 2 \times t_{clk} \times (32 \times BRP.5 + 16 \times BRP.4 + 8 \times BRP.3 + 4 \times BRP.2 + 2 \times BRP.1 + BRP.0 + 1)$

where t_{clk} = time period of the XTAL1 frequency = $1/f_{xtal1}$

10.5. BUS TIMING REGISTER 1 (BTR1): ADDRESS 07h

Bus Timing Register 1 defines the length of the bit period, the location of the sample point and the number of samples to be taken at each sample point.

BTR1.7	BTR1.6	BTR1.5	BTR1.4	BTR1.3	BTR1.2	BTR1.1	BTR1.0
SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

10.5.1. SAMPLING (SAM): BTR1.7

BIT	VALUE	FUNCTION
SAM	1	The bus will be sampled three times. (This is recommended for low/medium speed buses (class A or B).)
	0	The bus will be sampled once. (This is recommended for high speed buses (SAE class C).)

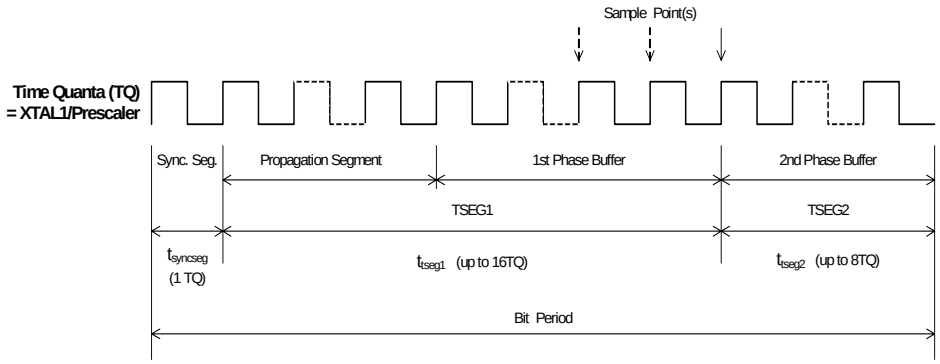
10.5.2. TSEG1 AND TSEG2: BTR1[3:0], BTR1[6:4]

TSEG1 and TSEG2 define the length of the bit period by giving the number of time quanta up to and after the point(s) at which incoming data will be sampled. In terms of TSEG1 and TSEG2, the parameters $t_{syncseg}$, t_{tseg1} and t_{tseg2} shown in the diagram are:

$t_{syncseg}$ =1 x TQ

t_{tseg1} = TQ x (8 x TSEG1.3 + 4 x TSEG1.2 + 2 x TSEG1.1 + TSEG1.0 + 1)

t_{tseg2}= TQ x (4 x TSEG2.2 + 2 x TSEG2.1 + TSEG2.0 + 1)



Note: In theory, it is possible to define bit periods of between 3 and 25 TQ through these register settings. However the bit periods used in practice are required to follow the BOSCH standard, which defines bit periods between 8 and 25 TQ in length.

10.6. CLOCK DIVIDER REGISTER (CDR): ADDRESS 1Fh

The Clock Divider Register controls the CLKOUT signal. The default state of the register after a hardware reset is 11000000 (divide by 2 and CLKOUT signal enabled). The register is not changed by a software reset.

CDR.7	CDR.6	CDR.5	CDR.4	CDR.3	CDR.2	CDR.1	CDR.0
1	1	0	0	Clock Off	CDR.2	CDR.1	CDR.0

Note: The additional options available in the SJA1000 through bits 7 – 5 of this register are not supported by the MCAN2.

10.6.1. CDR[2:0]

The bits CD.2 to CD.0 define the frequency at the external CLKOUT pin as shown in the following table (f_{osc} is the frequency of the external oscillator (XTAL1)). These bits may be accessed from either Reset Mode or Operating Mode.

CD[2:0]	CLKOUT FREQUENCY	CD[2:0]	CLKOUT FREQUENCY
0 0 0	$f_{osc}/2$	1 0 0	$f_{osc}/10$
0 0 1	$f_{osc}/4$	1 0 1	$f_{osc}/12$
0 1 0	$f_{osc}/6$	1 1 0	$f_{osc}/14$
0 1 1	$f_{osc}/8$	1 1 1	f_{osc}

10.6.2. CLOCK OFF (CDR.3)

Setting this bit allows the external CLKOUT signal to be disabled.

10.7. COMMAND REGISTER (CMR): ADDRESS 01h

Setting one or more bits within the Command Register initiates an action within the transfer layer of the CAN controller.

Note: This register is write only. When read, all bits return '0'. You should also note that there must be at least one external clock cycle between consecutive commands.

BIT	SYMBOL	NAME	FUNCTION
CMR[7:5]	–	–	*Reserved*
CMR.4	SRR	Self Reception Request	Set to '1' when a message is to be transmitted and received simultaneously.
CMR.3	CDO	Clear Data Overrun	Set to '1' to clear the data overrun condition signaled by the Data Overrun Status bit (SR.1). *Note:* No further Data Overrun Interrupt will be generated while the Data Overrun Status bit remains set.
CMR.2	RRB	Release Receive Buffer	Set to '1' to release the Receive Buffer – see Section 5.
CMR.1	AT	Abort Transmission	Set to '1' to cancel the next transmission request, provided this is not already in progress.
CMR.0	TR	Transmission Request	Set to '1' when a message is to be transmitted – see Section 4.

Notes

1. Setting the command bits CMR.0 and CMR.1 simultaneously results in a Single-Shot transmission of the Transmit message without re-transmission in the event of an error or loss of arbitration.

   Setting the command bits CMR.4 and CMR.1 simultaneously results in a Single-Shot transmission of the Transmit message with self reception, again without re-transmission in the event of an error or arbitration loss.

   If CMR.0 and CMR.4 are set simultaneously, the CMR.4 bit is ignored.

2. A Transmission Request made in a previous command cannot be cancelled by setting the Transmission Request bit to '0'.

The requested transmission can only be cancelled by setting the Abort Transmission bit to '1'.

10.8. ERROR CODE CAPTURE REGISTER (ECC): ADDRESS 0CH

This read-only register may be used to obtain detailed information about the type and location of bus errors.

When a bus error occurs, a Bus Error Interrupt is generated (if enabled) and the current bit position of the Bit Processor is captured into this Error Code Capture Register. The contents of this register are then maintained until the register has been read by the user's software. The capture mechanism is then activated again.

BIT	NAME	COMMENT
ECC[7:6]	Error Code	See Section 10.8.1 below
ECC.5	Direction	If '1', the error occurred during reception. If '0', the error occurred during transmission.
ECC[4:0]	Segment Code	See Section 10.8.2 below

10.8.1. ERROR CODE (ECC[7:6])

ECC[7:6]	FUNCTION
0 0	Bit error
0 1	Form error
1 0	Stuff error
1 1	Some other type of error

10.8.2. SEGMENT CODE (ECC[4:0])

ECC[4:0]	FUNCTION
0 0 0 1 1	Start of frame
0 0 0 1 0	ID.28 to ID.21
0 0 1 1 0	ID.20 to ID.18
0 0 1 0 0	SRTR bit
0 0 1 0 1	IDE bit
0 0 1 1 1	ID.17 to ID.13
0 1 1 1 1	ID.12 to ID.5

ECC[4:0]	FUNCTION
0 1 1 1 0	ID.4 to ID.0
0 1 1 0 0	RTR bit
0 1 1 0 1	Reserved bit 1
0 1 0 0 1	Reserved bit 0
0 1 0 1 1	Data Length Code
0 1 0 1 0	Data Field
0 1 0 0 0	CRC sequence
1 1 0 0 0	CRC delimiter
1 1 0 0 1	Acknowledge
1 1 0 1 1	Acknowledge delimiter
1 1 0 1 0	End of frame
1 0 0 1 0	Intermission
1 0 0 0 1	Active error flag
1 0 1 1 0	Passive error flag
1 0 0 1 1	Tolerate dominant bits
1 0 1 1 1	Error delimiter
1 1 1 0 0	Overload flag

10.9. ERROR WARNING LIMIT REGISTER (EWLR): ADDRESS 0Dh

This register defines the number of errors after which an Error Warning Interrupt should be generated (if enabled).

This register is read only in Operating Mode but may be written in Reset Mode. You should note that changes made within Reset Mode are only put into effect on return to Operating Mode.

The default value of this register (after hardware reset) is 0110000 (i.e. 96). An error count of this level suggests a significantly disturbed bus, the causes of which should be investigated.

EWLR.7	EWLR.6	EWLR.5	EWLR.4	EWLR.3	EWLR.2	EWLR.1	EWLR.0
EWL.7	EWL.6	EWL.5	EWL.4	EWL.3	EWL.2	EWL.1	EWL.0

10.10. INTERRUPT REGISTER (IR): ADDRESS 03h

The Interrupt Register allows the source of an interrupt to be identified. When one or more bits of this register are set, the MCAN2 sends an interrupt to the CPU. The way the different interrupts should be handled is discussed in Section 7.

Note: The Interrupt Register is read-only. After the register has been read by the CPU, all except the Receive Interrupt bit are reset.

BIT	SYMBOL	NAME	FUNCTION
IR.7	BEI	Bus Error Interrupt	Set when the MCAN2 detects an error on the CAN-bus – provided the BEIE bit (IER.7) is set within the Interrupt Enable Register.
IR.6	ALI	Arbitration Lost Interrupt	Set when the MCAN2 loses arbitration and becomes a receiver – provided the ALIE bit (IER.6) is set within the Interrupt Enable Register.
IR.5	EPI	Error Passive Interrupt	Set when the MCAN2 re-enters error active state after being in error passive state or when at least one error counter exceeds the protocol-defined level of 127 – provided the EPIE bit (IER.5) is set within the Interrupt Enable Register.
IR.4	WUI	Wake-Up Interrupt	Set when bus activity is detected while the CAN controller is sleeping – provided the WUIE bit (IER.4) is set within the Interrupt Enable Register. [1]
IR.3	DOI	Data Overrun Interrupt	Set on a '0-to-1' transition of the Data Overrun Status bit (SR.1) – provided the DOIE bit (IER.3) is set within the Interrupt Enable Register.
IR.2	EI	Error Warning Interrupt	Set on every change (set or clear) of either the Bus Status or Error Status bits (SR.7,SR.6) – provided the EIE bit (IER.2) is set within the Interrupt Enable Register.
IR.1	TI	Transmit Interrupt	Set whenever the Transmit Buffer Status (SR.2) changes from '0-to-1' (released) – provided the TIE bit (IER.1) is set within the Interrupt Enable Register.
IR.0	RI	Receive Interrupt [2]	Set whenever the Receive Buffer contains one or more messages – provided the RIE bit (IER.0) is set within the Interrupt Enable Register. Cleared when the release Receive Buffer command (CMR. 2) is issued provided there is no further data to read in the Receive Buffer.

Notes

1. A wake-up interrupt is also generated if the CPU tries to set the Sleep Mode bit while the MCAN2 is involved in bus activities or a CAN interrupt is pending.
2. The RI bit (when enabled) mirrors the Receive Buffer Status bit (SR.0), which is why it is not automatically cleared when the Interrupt Register is read.

10.11. INTERRUPT ENABLE REGISTER (IER): ADDRESS 04h

This read/write register is used to select the events that are indicated to the CPU through an interrupt being generated.

BIT	SYMBOL	NAME	FUNCTION
IER.7	BEIE	Bus Error Interrupt Enable	When set to '1', an interrupt will be generated when a bus error has been detected. When set to '0', the interrupt is disabled.
IER.6	ALIE	Arbitration Lost Interrupt Enable	When set to '1', an interrupt will be generated when the MCAN2 loses arbitration. When set to '0', the interrupt is disabled.
IER.5	EPIE	Error Passive Interrupt Enable	When set to '1', an interrupt will be generated when the error status of the MCAN2 changes from error active to error passive or vice versa. When set to '0', the interrupt is disabled.
IER.4	WUIE	Wake-Up Interrupt Enable	When set to '1', an interrupt will be generated when the sleeping MCAN2 wakes up. When set to '0', the interrupt is disabled.
IER.3	DOIE	Data Overrun Interrupt Enable	When set to '1', an interrupt will be generated when the Data Overrun Status bit (SR.1) is set. When set to '0', the interrupt is disabled.
IER.2	EIE	Error Warning Interrupt Enable	When set to '1', an interrupt will be generated when the bus status or error status bits (SR.7, SR.6) change. When set to '0', the interrupt is disabled.
IER.1	TIE	Transmit Interrupt Enable	When set to '1', an interrupt will be generated when a message has been successfully transmitted or the Transmit Buffer is accessible again. When set to '0', the interrupt is disabled.
IER.0	RIE	Receive Interrupt Enable[1]	When set to '1', an interrupt will be generated when the Receive Buffer Status (SR.0) goes from '0' to '1' ('full'). When set to '0', the interrupt is disabled.

Note

1. The Receive Interrupt Enable bit has direct influence on the Receive Interrupt bit and the external interrupt output NINT. If RIE is cleared, NINT will immediately become inactive (high) if no other interrupt is pending.

10.12. MODE REGISTER (MOD): ADDRESS 00h

This read/write register is used to set the behavior of the CAN controller.

BIT	SYMBOL	NAME	VALUE	FUNCTION
MOD[7:5]	–	–	–	*Reserved. Return zero when read.*
MOD.4	SM	Sleep Mode *(Can only be written in Operating Mode)*	1	Sleep. The MCAN2 enters its Sleep mode provided no CAN interrupt is pending and there is no bus activity. (If there is bus activity or an interrupt is pending, the Wake-Up procedure is executed.)
			0	Wake-up (normal operation). If sleeping, the MCAN2 wakes up.
MOD.3	AFM	Acceptance Filter Mode[1]	1	Single Filter. Receive data filtered using one 4-byte filter.
			0	Dual Filter. Receive data filtered using two shorter filters.

BIT	SYMBOL	NAME	VALUE	FUNCTION
MOD.2	STM	Self Test Mode[1]	1	Self Test enabled. In this mode, a full node test is possible without any other active node on the bus using the self reception request command. The MCAN2 will perform a successful transmission, even if no acknowledge is received.
			0	Normal operation. An acknowledge is required for successful transmission.
MOD.1	LOM	Listen Only Mode[1]	1	Listen Only enabled. In this mode, the MCAN2 does not send an acknowledge to the CAN bus, even when a message is received successfully.
			0	Normal operation. The error counters are stopped at the current value.
MOD.0	RM	Reset Mode	1	Reset Mode selected. Any message currently being transmitted or received is aborted and Reset Mode is entered.
			0	Normal operation. The MCAN2 returns to Operating Mode on the '1-to-0' transition of this bit.

Note: In the MCAN2, bits MOD[3:1] can be written in both Operating Mode and Reset Mode. In the reference device, they can only be written in Reset Mode.

10.13. OUTPUT CONTROL REGISTER (OCR): ADDRESS 08h

The Output Control Register allows the selection of two possible output driver configurations: 'Normal Output' and 'Clock Output'.

In Normal Output Mode, the bit sequence (TXD) is sent to TX0 with the inverse sent to TX1.

In Clock Output Mode, the bit sequence is output on the TX0 signal as in normal output mode but the data stream on TX1 is replaced by a copy of the Transmit clock (TXCLK), the rising edge of which marks the beginning of a bit period. The pulse width of this clock is one Time Quantum (TQ).

Note: The additional driver configurations available in the SJA1000 through this register are not supported by the MCAN2.

OCR.7	OCR.6	OCR.5	OCR.4	OCR.3	OCR.2	OCR.1	OCR.0
Reserved	*Reserved*	*Reserved*	*Reserved*	*Reserved*	*Reserved*	OCMODE1	OCMODE0

Interpretation of OCMODE bits

OCMODE1	OCMODE0	DESCRIPTION
0	X	*Reserved*
1	0	Normal Output Mode
1	1	Clock Output Mode

Note: The Output Control Register may only be written in Reset Mode. In Operating Mode, this register is read only. The Reserved bits return '0' when read.

10.14. RECEIVE BUFFER (10h – 1Ch)

The Receive Buffer provides the window through which the CPU accesses the Receive FIFO. Like the Transmit Buffer, the Receive Buffer has a length of 13 bytes (enough to accommodate one Receive message of up to eight data bytes).

Read-only access to the Receive Buffer is provided in Operating Mode using CAN addresses 10h – 1Ch

The layout of the Receive Buffer is similar to the Transmit Buffer described in the previous section. Indeed, the configuration used was chosen specifically to be compatible with the layout of the Transmit Buffer. Again, it is important to distinguish between Standard Frame Format (SFF) messages and the Extended Frame Format (EFF) messages.

10.14.1. RECEIVE BUFFER LAYOUT

The Receive Buffer is subdivided into descriptor and data fields. The first byte of the descriptor field holds frame information. It describes the frame format (SFF or EFF), specifies remote or data frame and gives the data length. This is then followed by either two identifier bytes for SFF or four bytes for EFF messages. The data field contains up to eight data bytes.

Standard Frame Format (SFF)		Extended Frame Format (EFF)	
CAN Address	**Field**	**CAN Address**	**Field**
10h	RX Frame Information	10h	RX Frame Information
11h	RX Identifier 1	11h	RX Identifier 1
12h	RX Identifier 2	12h	RX Identifier 2
13h	RX Data Byte 1	13h	RX Identifier 3
14h	RX Data Byte 2	14h	RX Identifier 4
15h	RX Data Byte 3	15h	RX Data Byte 1
16h	RX Data Byte 4	16h	RX Data Byte 2
17h	RX Data Byte 5	17h	RX Data Byte 3
18h	RX Data Byte 6	18h	RX Data Byte 4
19h	RX Data Byte 7	19h	RX Data Byte 5
1Ah	RX Data Byte 8	1Ah	RX Data Byte 6
1Bh	(Unused)	1Bh	RX Data Byte 7
1Ch	(Unused)	1Ch	RX Data Byte 8

10.14.2. DESCRIPTOR FIELD OF THE RECEIVE BUFFER

The bit layout of the Descriptor Field of the Receive Buffer is shown below, first for SFF then for EFF. The different elements are all as explained in Sections 10.19.3 – 10.19.7for the Transmit Buffer.

Receive Frame (SFF)

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	RTR	0	0	0	0

Receive Frame (EFF)

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	0	0	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13
13h	ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5
14h	ID.4	ID.3	ID.2	ID.1	ID.0	RTR	0	0

Note: The received data length code in the frame information byte (CAN Address 10h) represents the length of the data sent, which may be greater than eight bytes. However, the maximum number of data bytes received will be eight.

10.15. RECEIVE BUFFER START ADDRESS (RBSA): ADDRESS 1Eh

The Receive Buffer Start Address register records the current location of the RX FIFO Read Pointer within the 64-byte Receive FIFO as a value between 0 and 63. Location 0 maps to CAN address 20h: Location 63 maps to CAN address 5Fh.

This register is reset to 00h by a hardware reset but is left unchanged by a software reset (which also does not change the FIFO contents). However, the software reset sets the RX FIFO Write Pointer to the value of the RX FIFO Read Pointer with the result that the data currently accessed by the Receive Buffer following a software reset will be overwritten by the next message to be recorded in the Receive FIFO.

Note: It is only possible to write to this register in Reset Mode.

RBSA.7	RBSA.6	RBSA.5	RBSA.4	RBSA.3	RBSA.2	RBSA.1	RBSA.0
–	–	RBSA.5	RBSA.4	RBSA.3	RBSA.2	RBSA.1	RBSA.0

Note: RBSA[7:6] cannot be written and always return zero when read.

10.16. RECEIVE ERROR COUNTER REGISTER (RXERR): ADDRESS 0Eh

The Receive Error Counter Register records the current value of the Receive Error Counter. This counter is incremented when errors are experienced in the Receive bit stream and decremented when messages are received without error, in line with the rules given in the CAN 2.0 specification. Together with the associated Transmit Error Counter (see Section 10.20), it provides an indication of the quality of transmission being experienced on the CAN bus.

An outline of the rules by which the counter is incremented and decremented is given in the table below. For full details, you should refer to the CAN 2.0 specification.

Two levels of the counter trigger specific events.

- When the counter reaches the level set in the Error Warning Limit register (see Section 10.9), an Error Warning Interrupt is generated (if enabled) unless this has previously been triggered by the Transmit Error Counter.
- When the counter goes over 127, the device is put into Error Passive state in accordance with the CAN 2.0 specification

(unless previously triggered by the Transmit Error Counter) and an Active error is sent. An Error Passive Interrupt is also generated (if enabled).

After a hardware reset or when a Bus Off event occurs (see Transmit Error Counter – see Section 10.20), the counter is automatically set to '0'.

The register is read only in Operating Mode but may be written in Reset Mode. You should note, however, that writing to this register has no effect when the MCAN2 is in Bus Off state and that any change made within Reset Mode will in any case only come into effect on return to Operating Mode.

ERROR EVENT	ACTION TAKEN
Receiver detects an error	RXERR incremented by 1
Receiver detects dominant bit as the first bit after sending an Error flag	RXERR incremented by 8
Receiver detects a bit error while sending an Active error or an Overload error	RXERR incremented by 8
Message successfully received	RXERR decremented by 1
Message is received successfully when the count had previously been above the Error Passive trigger level of 127	RXERR automatically set to a value between 119 and 127
14th consecutive dominant bit received after sending an Active error or an Overload error	RXERR incremented by 8 and TXERR incremented by 8 both at this point and after each additional sequence of 8 consecutive dominant bits
8th consecutive dominant bit received after sending a Passive error	
Transmitter sends an error	TXERR incremented by 8
Transmitter detects a bit error while sending an Active error or an Overload error	TXERR incremented by 8
Transmitter successfully transmits message	TXERR decremented by 1

10.17. RECEIVE MESSAGE COUNTER (RMC): ADDRESS 1Dh

The Receive Message Counter register records the number of messages currently available in the Receive FIFO. It is automatically incremented by each Receive event and decremented by each Release Receive Buffer command. It is available for Read only access in both Operating Mode and Reset Mode.

The register is reset to 00h by either a hardware or a software reset.

RMC.7	RMC.6	RMC.5	RMC.4	RMC.3	RMC.2	RMC.1	RMC.0
0	0	0	RMC.4	RMC.3	RMC.2	RMC.1	RMC.0

10.18. STATUS REGISTER (SR): ADDRESS 02h

This read-only register reflects the status of the MCAN2 controller.

BIT	SYMBOL	NAME	VALUE	FUNCTION
SR.7	BS	Bus Status	1	The MCAN2 is in 'Bus Off' state and is not involved in bus activities.
			0	The MCAN2 is involved in bus activities.
SR.6	ES	Error Status	1	At least one of the error counters has reached or exceeded the CPU warning limit defined by the Error Warning Limit Register (EWL).
			0	Both error counters are below the warning limit.
SR.5	TS	Transmit Status [1]	1	The MCAN2 is in the process of transmitting a message.
			0	No message is being transmitted.
SR.4	RS	Receive Status [1]	1	The MCAN2 is in the process of receiving a message.
			0	Nothing is currently being received.
SR.3	TCS	Transmission Complete Status	1	The last requested transmission has been successfully completed.
			0	The last requested transmission has not been completed yet.
SR.2	TBS	Transmit Buffer Status	1	Transmit Buffer Released. The CPU may write a message to the Transmit Buffer.
			0	Transmit Buffer Locked. The CPU cannot access the Transmit Buffer because a message is either waiting for transmission or is in the process of being transmitted.
SR.1	DOS	Data Overrun Status	1	Data Overrun. A message has been lost because there was not enough space for that message in the Receive FIFO. [2]
			0	No data overrun has occurred since the last Clear Data Overrun command was given.
SR.0	RBS	Receive Buffer Status	1	Receive Buffer Full. One or more complete messages are available to be read from the Receive FIFO via the Receive Buffer.
			0	Receive Buffer Empty. No message currently available to be read.

Notes

1. If both the Receive Status and the Transmit Status bits are '0', the CAN bus is idle. If both bits are '1', the controller is waiting to become idle again. After a hardware reset, idle state is entered once the Bus-Free sequence (11 consecutive recessive bits) has been detected. After a Bus Off event, 128 Bus-Free sequences must be received before idle state is entered.
2. The overrun condition is only indicated if the entire message was received. No overrun condition is shown if the message did not complete (e.g. due to an error).

10.19. TRANSMIT BUFFER (Write: 10h – 1Ch; Read: 60h – 6Ch)

The Transmit Buffer has a length of 13 bytes. It accommodates one Transmit message of up to eight data bytes.

Access to the Transmit Buffer in Operating Mode is write-only and is provided using CAN addresses 10h – 1Ch.

The global layout of the Transmit Buffer is shown below. It is important to distinguish between Standard Frame Format (SFF) messages and the Extended Frame Format (EFF) messages.

Note: Read access to the Transmit Buffer is possible using CAN addresses 60h – 6Ch.

10.19.1. TRANSMIT BUFFER LAYOUT

The Transmit Buffer is subdivided into descriptor and data fields. The first byte of the descriptor field holds frame information. It describes the frame format (SFF or EFF), remote or data frame and the data length. This is then followed by either two identifier bytes for SFF or four bytes for EFF messages. The data field contains up to eight data bytes.

Standard Frame Format (SFF)		Extended Frame Format (EFF)	
CAN Address	**Field**	**CAN Address**	**Field**
10h	TX Frame Information	10h	TX Frame Information
11h	TX Identifier 1	11h	TX Identifier 1
12h	TX Identifier 2	12h	TX Identifier 2
13h	TX Data Byte 1	13h	TX Identifier 3
14h	TX Data Byte 2	14h	TX Identifier 4
15h	TX Data Byte 3	15h	TX Data Byte 1
16h	TX Data Byte 4	16h	TX Data Byte 2
17h	TX Data Byte 5	17h	TX Data Byte 3
18h	TX Data Byte 6	18h	TX Data Byte 4
19h	TX Data Byte 7	19h	TX Data Byte 5
1Ah	TX Data Byte 8	1Ah	TX Data Byte 6
1Bh	(Unused)	1Bh	TX Data Byte 7
1Ch	(Unused)	1Ch	TX Data Byte 8

10.19.2. DESCRIPTOR FIELD OF THE TRANSMIT BUFFER

The bit layout of the Descriptor Field of the Transmit Buffer is shown below, first for SFF then for EFF.

The different elements of the Descriptor Field are explained in the following sections (Sections 10.19.3 – 10.19.7).

Transmit Frame (SFF)

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	X (1)	X (1)	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	X (2)	X (1)	X (1)	X (1)	X (1)

Transmit Frame (EFF)

CAN Address	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
10h	FF	RTR	X (1)	X (1)	DLC.3	DLC.2	DLC.1	DLC.0
11h	ID.28	ID.27	ID.26	ID.25	ID.24	ID.23	ID.22	ID.21
12h	ID.20	ID.19	ID.18	ID.17	ID.16	ID.15	ID.14	ID.13
13h	ID.12	ID.11	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5
14h	ID.4	ID.3	ID.2	ID.1	ID.0	X (2)	X (1)	X (1)

Notes

1. Don't care but recommend '0' to be compatible with Receive Buffer in case the Self Reception or the Self Test option is used.
2. Don't care but recommend matching the RTR bit used in the Receive Buffer in case the Self Reception or the Self Test option is used.

10.19.3. FRAME FORMAT (FF)

The FF bit selects the type of frame format to be transmitted. '1' selects Extended Frame Format (EFF); '0' selects Standard Frame Format (SFF).

10.19.4. REMOTE TRANSMISSION REQUEST (RTR)

The RTR bit is used to identify the frame as either a remote frame or a data frame (as defined in the CAN protocol). '1' indicates a remote frame (i.e. data requested from another node); '0' indicates a data frame.

10.19.5. DATA LENGTH CODE (DLC)

The DLC [3:0] bits are used to specify the number of data bytes included in the message being sent.

The maximum number of data bytes that can be included in a frame is eight so values of DLC [3:0] greater than eight are automatically interpreted as eight.

You should also note that, although no data bytes are transmitted from the local host in the case of a remote frame transmission, the data length of the remote frame should still be specified to avoid bus errors if two CAN controllers simultaneously start a remote frame transmission with the same identifier.

10.19.6. IDENTIFIER (ID)

The identifier acts as the message's name, used in a receiver for acceptance filtering, and also determines the bus access priority. The lower the value of the identifier the higher the priority.

In Standard Frame Format (SFF) the identifier consists of 11 bits (ID.28 to ID.18). In Extended Frame Format (EFF) messages the identifier consists of 29 bits (ID.28 to ID.0). ID.28 is the most significant bit and is transmitted first on the bus.

10.19.7. DATA FIELD

The data field should comprise the number of data bytes defined by the data length code. The most significant bit of data byte 1 at CAN address 19 (SFF) or CAN address 21 (EFF) is transmitted first.

10.20. TRANSMIT ERROR COUNTER REGISTER (TXERR): ADDRESS 0Fh

The Transmit Error Counter Register records the current value of the Transmit Error Counter. This counter is incremented when Transmission errors are experienced and decremented when messages are transmitted without error, in line with the rules given in the CAN 2.0 specification. Together with the associated Receive Error Counter (see Section 10.16), it provides an indication of the quality of transmission being experienced on the CAN bus.

An outline of the rules by which the counter is incremented and decremented is given in the table in Section 10.16. For full details, you should refer to the CAN 2.0 specification.

Three levels of the counter trigger specific events.

- When the counter reaches the level set in the Error Warning Limit register (see Section 10.9), an Error Warning Interrupt is generated (if enabled) unless this has previously been triggered by the Receive Error Counter.
- When the counter goes over 127, the device is put into Error Passive state in accordance with the CAN 2.0 specification (unless previously triggered by the Receive Error Counter), an Active error is sent and an Error Passive Interrupt is generated (if enabled).
- When the counter goes over 255, the device is put into Bus Off state in accordance with the CAN 2.0 specification and is automatically put into Reset mode (except during start-up when there is only one node on the CAN bus). An Error Warning Interrupt is also generated (if enabled).

After a hardware reset, the Transmit Error Counter is automatically set to '0'.

After a 'Bus Off' event, the register is initialized to 127 in order to count the minimum protocol-defined time before the MCAN2 can take part in further transmission on the CAN bus (128 occurrences of the 'Bus-Free' sequence of 11 consecutive recessive bits). Reading the Transmit Error Counter during this time will give the status of the Bus Off recovery. *Note:* If the Reset Mode is re-entered before the Bus Off recovery has been completed (TXERR > 0), Bus Off will stay active with TXERR frozen until the MCAN2 is taken back into Operating Mode.

It is possible to write to this register but only in Reset Mode. In Operating Mode, this register appears as read only memory to the CPU.

While in Bus Off state, writing a value in the range from 0 to 254 to TXERR clears the Bus Off flag. The MCAN2 will then wait for just one Bus Free sequence after the Reset Mode has been cleared.

Writing 255 to TXERR in Reset Mode initiates a CPU-driven Bus Off event. No error or bus status change happens in response to the new TXERR value until the MCAN2 is taken back into Operating Mode when a Bus Off event will be performed exactly as if it had been forced by a bus error. This means Reset Mode is entered again, the Transmit Error Counter is initialized to 127, the Receive counter is cleared and the relevant Status and Interrupt register bits are set. Clearing Reset Mode now will perform the protocol-defined Bus Off recovery sequence (waiting for 128 occurrences of the bus-free signal).

11. APPENDIX: RESET VALUES

REGISTER	BIT	SYMBOL	NAME	VALUE	
				FOLLOWING HARDWARE RESET	FOLLOWING SOFTWARE RESET (MOD.0) OR BUS OFF
Mode	MOD[7:5]	-	*Reserved*	0 (reserved)	0 (reserved)
	MOD.4	SM	Sleep Mode	0 (wake-up)	0 (wake-up)
	MOD.3	AFM	Acceptance Filter Mode	0 (dual filters)	X
	MOD.2	STM	Self Test Mode	0 (normal)	X
	MOD.1	LOM	Listen Only Mode	0 (normal)	X
	MOD.0	RM	Reset Mode	1 (present)	1 (present)
Command	CMR[7:5]	-	*Reserved*	0 (reserved)	0 (reserved)
	CMR.4	SRR	Self Reception Request	0 (absent)	0 (absent)
	CMR.3	CDO	Clear Data Overrun	0 (no action)	0 (no action)
	CMR.2	RRB	Release Receive Buffer	0 (no action)	0 (no action)
	CMR.1	AT	Abort Transmission	0 (absent)	0 (absent)
	CMR.0	TR	Transmission Request	0 (absent)	0 (absent)
Status	SR.7	BS	Bus Status	0 (bus-on)	X
	SR.6	ES	Error Status	0 (ok)	X
	SR.5	TS	Transmit Status	1 (wait idle)	1 (wait idle)
	SR.4	RS	Receive Status	1 (wait idle)	1 (wait idle)
	SR.3	TCS	Transmission Complete Status	1 (complete)	X
	SR.2	TBS	Transmit Buffer Status	1 (released)	1 (released)
	SR.1	DOS	Data Overrun Status	0 (absent)	0 (absent)
	SR.0	RBS	Receive Buffer Status	0 (empty)	0 (empty)
Interrupt	IR.7	BEI	Bus Error Interrupt	0 (reset)	0 (reset)
	IR.6	ALI	Arbitration Lost Interrupt	0 (reset)	0 (reset)
	IR.5	EPI	Error Passive Interrupt	0 (reset)	0 (reset)
	IR.4	WUI	Wake-Up Interrupt	0 (reset)	0 (reset)
	IR.3	DOI	Data Overrun Interrupt	0 (reset)	0 (reset)
	IR.2	EI	Error Warning Interrupt	0 (reset)	X; see Note 1 below
	IR.1	TI	Transmit Interrupt	0 (reset)	0 (reset)
	IR.0	RI	Receive Interrupt	0 (reset)	0 (reset)
Interrupt Enable	IER.7	BEIE	Bus Error Interrupt Enable	0	X
	IER.6	ALIE	Arbitration Lost Interrupt Enable	0	X
	IER.5	EPIE	Error Passive Interrupt Enable	0	X
	IER.4	WUIE	Wake-Up Interrupt Enable	0	X
	IER.3	DOIE	Data Overrun Interrupt Enable	0	X
	IER.2	EIE	Error Warning Interrupt Enable	0	X
	IER.1	TIE	Transmit Interrupt Enable	0	X
	IER.0	RIE	Receive Interrupt Enable	0	X

REGISTER	BIT	SYMBOL	NAME	VALUE	
				FOLLOWING HARDWARE RESET	FOLLOWING SOFTWARE RESET (MOD.0) OR BUS OFF
Bus Timing 0	BTR0.7	SJW.1	Synchronization Jump Width 1	0	X
	BTR0.6	SJW.0	Synchronization Jump Width 0	0	X
	BTR0.5	BRP.5	Baud Rate Prescaler 5	0	X
	BTR0.4	BRP.4	Baud Rate Prescaler 4	0	X
	BTR0.3	BRP.3	Baud Rate Prescaler 3	0	X
	BTR0.2	BRP.2	Baud Rate Prescaler 2	0	X
	BTR0.1	BRP.1	Baud Rate Prescaler 1	0	X
	BTR0.0	BRP.0	Baud Rate Prescaler 0	0	X
Bus Timing 1	BTR1.7	SAM	Sampling	0	X
	BTR1.6	TSEG2.2	Time Segment 2.2	0	X
	BTR1.5	TSEG2.1	Time Segment 2.1	0	X
	BTR1.4	TSEG2.0	Time Segment 2.0	0	X
	BTR1.3	TSEG1.3	Time Segment 1.3	0	X
	BTR1.2	TSEG1.2	Time Segment 1.2	0	X
	BTR1.1	TSEG1.1	Time Segment 1.1	0	X
	BTR1.0	TSEG1.0	Time Segment 1.0	0	X
Output Control	OCR[7:2]	-	*Reserved*	0	X
	OCR.1	OCMODE1	Output Control Mode 1	0	X
	OCR.0	OCMODE0	Output Control Mode 0	0	X
Arbitration Lost Capture	-	ALC	Arbitration Lost Capture	00h	X
Error Code Capture	-	ECC	Error Code Capture	00h	X
Error Warning Limit	-	EWLR	Error Warning Limit Register	96 (decimal)	X
Receive Error Counter	-	RXERR	Receive Error Counter	0 (reset)	X; see Note 2
Transmit Error Counter	-	TXERR	Transmit Error Counter	0 (reset)	X; see Note 2
Transmit Buffer	-	TXB	Transmit Buffer	X	X
Receive Buffer	-	RXB	Receive Buffer	X	X
Acceptance Code Registers 0-3	-	ACR0-3	Acceptance Code Registers 0-3	00h	X
Acceptance Mask Registers 0-3	-	AMR0-3	Acceptance Mask Registers 0-3	00h	X
Receive Message Count	-	RMC	Receive Message Count	0	0
Receive Buffer Start Address	-	RBSA	Receive Buffer Start Address	00h	X
Receive FIFO	-	-	Receive FIFO	X	X
Clock Divider	-	CDR	Clock Divider Register	00h	X

Notes

1. If the Reset Mode was entered due to a Bus Off condition, the Error Warning Interrupt will be set (if enabled).
2. If the Reset Mode was entered due to a Bus Off condition, the Receive Error Counter is cleared and the Transmit Error Counter is initialized to 127 to count-down the CAN-defined Bus Off recovery time consisting of 128 occurrences of 11 consecutive recessive bits.

12. REVISION HISTORY

12.1. ISSUE 1

3rd October 2001. First issue of document.

12.2. ISSUE 2

31st July 2003. Amendment made to range of bit periods supported (Section 3.1).

12.3. ISSUE 3

7th May 2006. Amendment made for CDR R/W description (Section 10.6).

http://www.mentor.com/inventra

Corporate Headquarters
Mentor Graphics Corporation
8005 S.W. Boeckman Road
Wilsonville, OR 97070 USA
Phone: 503-685-7000
North American Support Center
Phone: 800-547-4303
Fax: 800-684-1795

Silicon Valley Headquarters
Mentor Graphics Corporation
1001 Ridder Park Drive
San Jose, California 95131 USA
Phone: 408-436-1500
Fax: 408-436-1501

Europe Headquarters
Mentor Graphics Corporation
Immeuble le Pasteur
13/15, rue Jeanne Braconnier
92360 Meudon La Foret
France
Phone: 33-1-40-94-74-74
Fax: 33-1-46-01-91-73

Pacific Rim Headquarters
Mentor Graphics (Taiwan)
Room 1603, 16F,
International Trade Building
No.333, Section 1, Keelung Road
Taipei, Taiwan, ROC
Phone: 886-2-87252000
Fax: 886-2-27576027

Japan Headquarters
Mentor Graphics Japan Co., Ltd.
Gotenyama Hills
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140
Japan
Phone: 81-3-5488-3033
Fax: 81-3-5488-3021

05/06 PG-40107.003-FC