BUS INTERFACE

INVENTRA™

THE INTELLIGENT APPROACH TO INTELLECTUAL PROPERTY

MCAN2

CAN 2.0 NETWORK CONTROLLER

Product Specification

SOFT CORES

05/06 PS-40107.003-FC

The document is for informational and instructional purposes. Mentor Graphics reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in the written contracts between Mentor Graphics and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Mentor Graphics whatsoever.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

RESTRICTED RIGHTS LEGEND Use, duplication, or disclosure by the Government is subject to restrictions as set forth in the subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

A complete list of trademark names appears in a separate “Trademark Information” document.

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

This is an unpublished work of Mentor Graphics Corporation.

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

MCAN2

CAN 2.0 NETWORK CONTROLLER

- **Supports full CAN 2.0 – both 2.0A (equivalent to CAN 1.2) and 2.0B**
- **Supports both 11-bit and 29-bit identifiers**
- **Supports bit rates from less than 125Kbaud to more than 1Mbaud**
- **64 byte Receive FIFO**
- **Software-driven bit-rate detection (offering hot plug-in support)**
- **Acceptance filtering**
- **Single-shot transmission option**
- **Listen-only mode**
- **Reception of 'own' messages**
- **Self Test option**
- **Error interrupt generated for each CAN bus error**
- **Arbitration lost interrupt with record of bit position**
- **Read/write error counters**
- **Last error register**
- **Programmable error limit warning**
- **Synchronous PVCI[1]-compatible CPU interface for easy connection to a range of microprocessors**
- **Verified against BOSCH CAN2.0 test suite**

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 but with some exceptions (detailed in Section 8). In particular, the design has a synchronous PVCI[1]-compatible CPU interface for ease of connection to a range of microprocessor buses.

This specification 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.

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

2. FUNCTIONAL DESCRIPTION

2.1. BLOCK DIAGRAM

The following diagram shows the main functional blocks of the MCAN2 design.

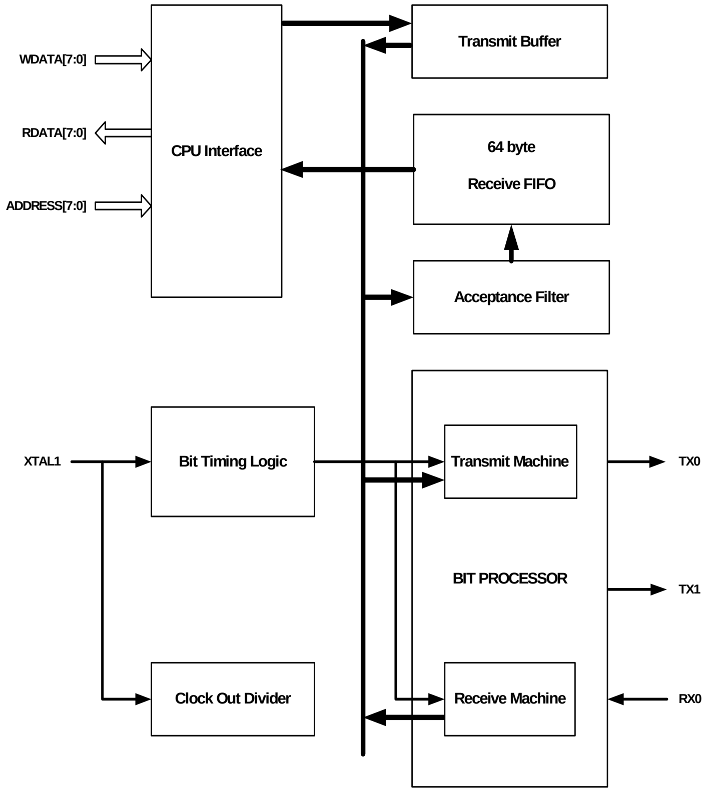


Figure 1. MCAN2 Block Diagram

CPU access to the MCAN2 is via separate address, input data and output data buses. Messages for transmission are placed into the Transmit Buffer by the host device for transmission by the Bit Processor. Messages received by the device are first filtered by an Acceptance Filter, then placed into the Receive FIFO.

The CPU accesses the Receive FIFO through a 13-byte window referred to as the Receive Buffer. The use of the Receive Buffer in conjunction with the Receive FIFO allows the CPU to process one message while other messages are being received. The Receive FIFO has a total length of 64 bytes and is used in circular fashion, giving it the capacity to accommodate up to five Extended Frame Format messages at a time.

The Bit Timing Logic block is responsible for the baud rate of the device and is programmable. The range of baud rates supported depends on the frequency of the main system clock XTAL1 and can readily span a wider range of baud rates than the 125Kbaud – 1Mbaud picked out by the BOSCH specification.

The interface to the CAN bus is provided by the signals TX0, TX1 for transmit and RX0 for receive. TX1 is normally the inverse of TX0 but can be programmed to output the Transmit clock, which is useful for testing purposes.

2.2. INTENDED APPLICATION

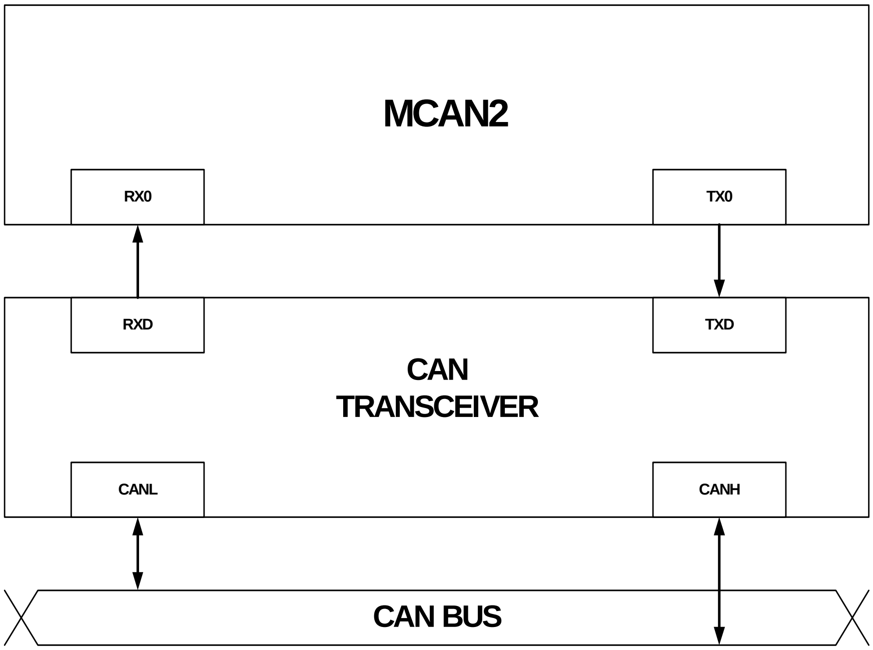
Figure 2 below shows how the MCAN2 core is intended to be used.

The MCAN2 sits between a Host microprocessor and a standard CAN bus transceiver. The transceiver is responsible for putting logical levels from the MCAN2 onto the CAN bus. The CAN bus is usually a twisted pair which is fed into the differential inputs of the CAN transceiver.

The figure below shows TX0 driving the CAN transceiver. This is the usual arrangement. However, the TX1 output, which can be programmed to output either the inverse of TX0 or the Transmit clock, gives the system designer the option of creating their own interface to the CAN bus instead of using a standard CAN transceiver.

Note that the host CPU has not been drawn.

Figure 2. MCAN2 interface to the CAN bus



2.3. OPERATION

2.3.1. 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.

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' (see Section 2.3.8 'Error Handling' below). 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 of the MCAN2 Programmer's Guide).

In Self Test Mode, the MCAN2 sends and receives messages using the MCAN2's Self Reception feature without looking for 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 a copy of the Transmit clock rather than a second (inverted) copy of the transmission data.

For further information about Reset Mode, see Section 5. For further information about the Listen Only and Self-Test Modes, see the description of the Mode register in Section 4.1. For further information about Clock Output Mode, see the description of the Output Control Register in Section 4.7.

Note: The MCAN2 needs to be taken out of Reset Mode to transmit or receive any data.

2.3.2. TRANSMISSION

Data to be transmitted is written to the MCAN2's Transmit Buffer in either the Standard Frame Format (SFF) or Extended Frame Format (EFF). (These formats are defined in Section 4.13.) 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 the data to the buffer, the Transmit Buffer Status (Status Register, bit 2) needs to be checked to ensure that the buffer is 'released' (SR.2 = '1'). Any data written to the buffer when the buffer is locked (SR.2 = '0') is simply lost without any indication.

Transmission of the data that has been written to the Transmit Buffer is initiated by issuing either a Transmit Request through the Command Register (by setting CMR.0 = '1') – or a Self Reception Request (CMR.4 = '1') if Self-Reception of the message is required (see Section 2.3.4). The moment transmission starts, the Transmit Status (SR.5) changes to '1' and the Transmission Request bit is cleared.

The bit sequence is output on TX0. In Normal Output Mode (see Section 4.7), the inverse signal is also output on TX1. If bus arbitration is lost (see Section 2.3.7) or if transmission errors occur (see Section 2.3.8) while the message is being sent, the MCAN2 will automatically try again to send the message.

A 15-bit Cycle Redundancy Check (CRC) is sent with each frame, generated from the start-of-frame, arbitration, control and data fields of the frame to be sent. Full details are given in the CAN 2.0 specification.

Transmission of a message can be aborted by issuing an Abort Transmission command through the Command Register (CMR.1 = '1') – provided transmission has not yet started. It is not possible to abort a message after it has started to be transmitted. *Note:* To see if the original message has been either transmitted successfully or aborted, wait for the Transmit Buffer Status bit (SR.2) to be set to '1' or a Transmit Interrupt to be generated, then check the Transmission Complete Status bit (SR.3). (A Transmit Interrupt is generated even if the message was aborted because the Transmit Buffer Status bit changes to 'released'.)

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

2.3.3. RECEPTION

Data received by the MCAN2 is first filtered by the Acceptance Filter then passed to the Receive FIFO. The Acceptance Filter only passes on those messages with identifier bits that match the ones recorded in the Acceptance Filter registers.

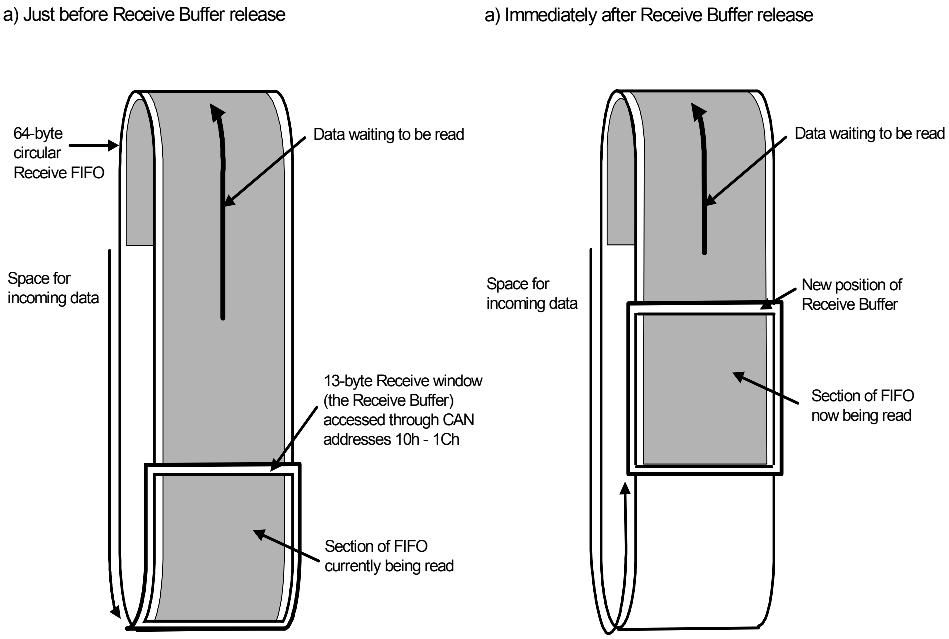
The moment data starts being placed in the Receive FIFO, the Receive Status bit in Status Register (SR.4) goes to '1'. Then once the data has been received, the Receive Buffer Status bit (SR.0) goes to '1' and a Receive Interrupt is generated (if enabled).

The Receive FIFO is 64 bytes deep, allowing space for up to five full Extended Frame Format (EFF) messages, and is used in a circular fashion. If there is not enough space in the FIFO for the data being received, the Data Overrun Status bit in the Status Register (SR.1) is set and the data frame being received is discarded. A Data Overrun Interrupt is also generated (if enabled).

The 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.

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'). 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.

Figure 3. Receive Buffer window on Receive FIFO


a) Just before Receive Buffer release
a) Immediately after Receive Buffer release
64-byte circular Receive FIFO
Data waiting to be read
Space for incoming data
13-byte Receive window (the Receive Buffer) accessed through CAN addresses 10h - 1Ch
Section of FIFO currently being read
Data waiting to be read
Space for incoming data
New position of Receive Buffer
Section of FIFO now being read

2.3.4. 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 4.2. 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. (See Section 4.1).

2.3.5. ACCEPTANCE FILTERING

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 allows pre-filtering of received messages by applying a 4-byte Acceptance Filter to received data. Only the messages with identifier bits that match the filter are passed to the Receive FIFO.

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

The filtering is carried out using four 8-bit Acceptance Code Registers (which record the bit patterns to match), together with four 8-bit Acceptance Mask Registers which mark particular bits of the Acceptance Code bit patterns as 'don't care'. Both sets of registers are applied either as a single 32-bit filter to the first 4 bytes of each received message, or as two separate 16-bit filters to the first 2 bytes of the message. Where two filters are applied, messages are accepted when the identifier bits tested match at least one of the filters.

The precise application of the filters depends on whether the data is in Standard Frame Format or Extended Frame Format and on whether one or two filters are applied. Details are given in Section 6.

2.3.6. BAUD RATE

The baud rate used on the CAN bus is specified through giving the length of the bit period as a number of 'time quanta' TQ, the length of which is defined in terms of periods of the XTAL1 input clock. The point in the bit period at which the incoming bit stream is sampled is also defined as a number of time quanta.

The MCAN2 allows bit periods from 3 to 25 TQ to be defined. 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.

The bit period and sampling point used by the MCAN2 are set through 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 number of time quanta by which the bit period may be shortened or lengthened in attempting to re-synchronize with the current transmission. BTR1 defines the number of time quanta up to and after the point at which the sample is taken, and the number of samples taken (one or three).

Details of the BTR0 and BTR1 registers are given in Section 4.6.

2.3.7. BUS ARBITRATION

Control of the CAN bus is governed by the rules specified in the CAN 2.0 specification.

At any time, the CAN node that has control of the bus is the one with the lowest identifier. A CAN node that loses arbitration must withdraw and not attempt to control the CAN bus again until the CAN bus is idle. When the device containing the MCAN2 loses arbitration, an Arbitration Lost Interrupt will be generated (if enabled in the Interrupt Enable Register (see Section 4.5)) and the bit position at which arbitration was lost will be recorded in the Arbitration Lost Capture Register. For details of the way in which this bit position is recorded, see Section 4.8.

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 register.

2.3.8. ERROR HANDLING

Errors in reception and transmission are handled according to the CAN 2.0B protocol specification.

The MCAN2 includes two error counters – one for reception errors (RXERR) and one for transmission errors (TXERR) – which are automatically incremented in accordance with the CAN 2.0B specification when an error occurs. In addition, the type of error (bit/form/stuff/other) and the location of the error within the message frame are captured in an Error Code Capture Register (described in Section 4.9) where they remain until this register is read. (Successful reception and transmission also decrement the counters in accordance with the CAN 2.0 specification.)

The MCAN2 also includes an Error Warning Limit (EWL) Register, the value of which represents the number of errors in either reception or transmission at which a warning should be generated. The default value for the EWL is 96. 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).

If either counter goes over 127, the MCAN2 goes into 'Error Passive' state (as defined by the CAN protocol) and issues an Error Passive Interrupt (if enabled).

Should the Transmit Error Counter exceed 255 (the limit of the counter), the Bus Status bit (SR.7) will be set to '1' (Bus Off), the MCAN2 will be set into Reset Mode and an Error Warning Interrupt (EI) generated (if enabled). The MCAN2 will then stay in Reset Mode 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 (the minimum time defined by the CAN protocol) before becoming 'Bus On' again.

2.3.9. SLEEP MODE

When there is no bus activity and no interrupts are pending, power can be saved by putting the MCAN2 into Sleep Mode.

Sleep Mode is selected by setting the Sleep Mode bit in the Mode Register (MOD.4) to '1' (see Section 4.1). Following entry into Sleep Mode, the CLKOUT signal continues for 15 bit periods to allow the host CPU time to enter its standby mode. NXTAL1_ENABLE is then taken high. XTAL1_IN, which is used to clock the majority of the MCAN2's internal registers and is produced by gating XTAL1 externally with NXTAL1_ENABLE (see Section 3), is then turned off – disabling the MCAN2.

TX0 and TX1 state in sleep mode depends on the previous mode. If it was Normal Mode, TX0 goes high and TX1 goes low in Sleep Mode. If it was Clock Mode, TX0 goes high. TX1 is the bit clock, and can go high or low, depending on the phase at the time Sleep Mode started.

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

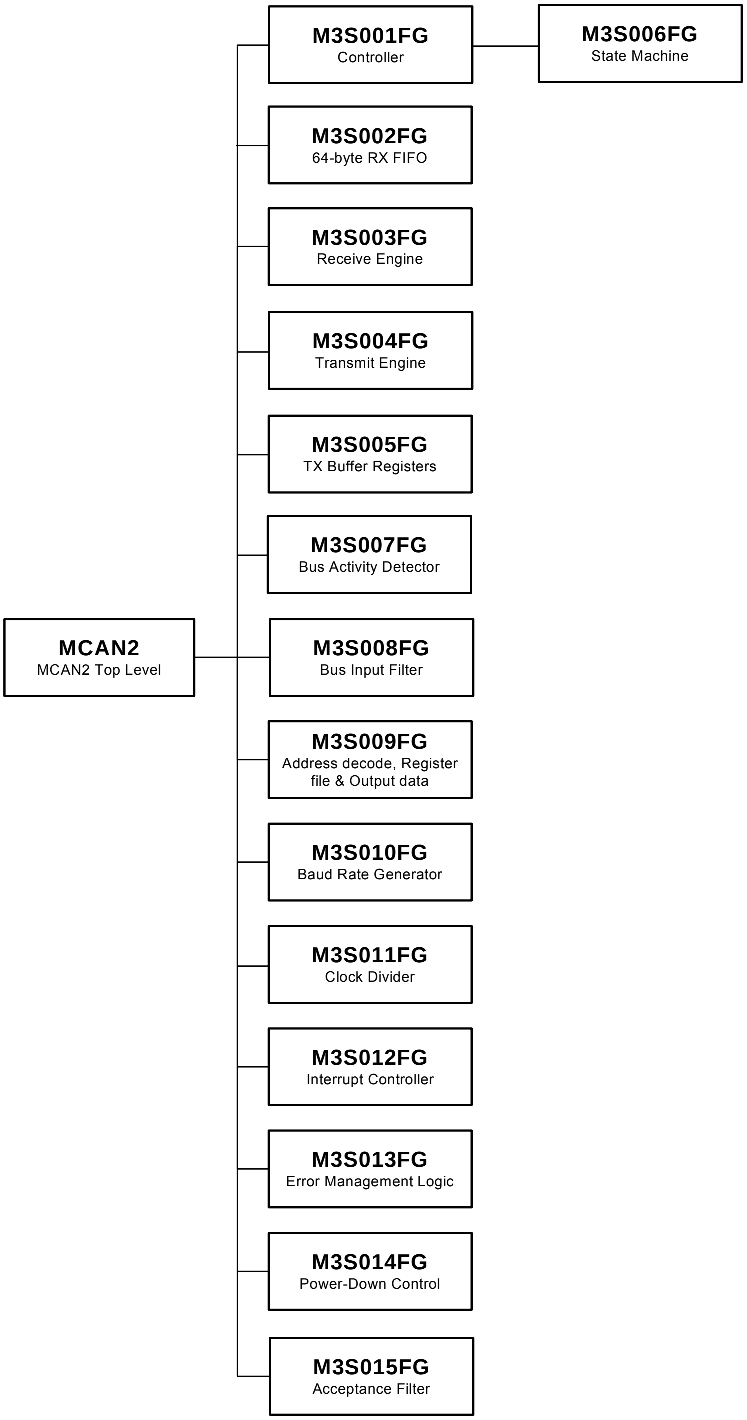
- Setting the Sleep Mode bit to '0'
- Activity on the CAN bus input (RX0)
- A low on NINT_IN

On waking up, the MCAN2 will generate a Wake-Up Interrupt (if enabled in the Interrupt Enable Register (see Section 4.5)).

Note: If the MCAN2 is awakened by bus activity, it will not receive any messages 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.

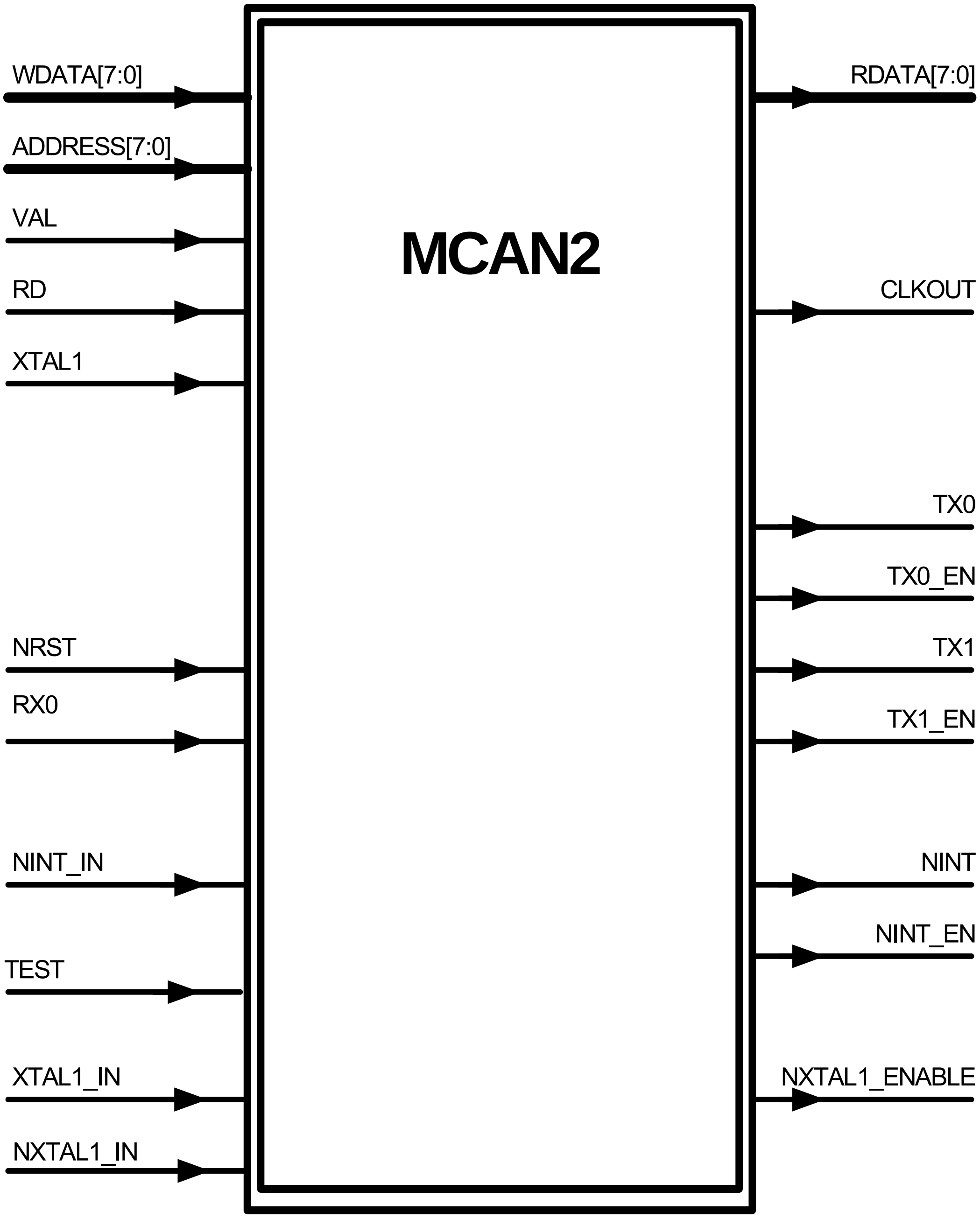
2.4. TREE DIAGRAM

The following tree diagram shows the hierarchical structure of the MCAN2.



3. SIGNAL DESCRIPTION

The MCAN2 has 41 external signals; 25 inputs and 16 outputs.



SYMBOL	TYPE	DESCRIPTION
WDATA[7:0]	Input	Data bus input.
ADDRESS[7:0]	Input	Address bus.
XTAL1	Input	System clock.
RX0	Input	Input from CAN transceiver (CAN bus input).
NRST	Input	Reset, active low.
VAL	Input	CPU Access Validate.
RD	Input	CPU Read signal.
NINT_IN	Input	Input from NINT bi-direct buffer. Used to wake device from Sleep Mode.
TEST	Input	Test pin, included to allow better fault coverage.
RDATA[7:0]	Output	Data bus output.
CLKOUT	Output	Clock output signal derived from divided input clock.
NINT	Output	Interrupt output, active low. Used to interrupt the host microcontroller.
NINT_EN	Output	Enable for Interrupt signal, which is open-drain.
TX0	Output	Serial output for CAN output driver 0.
TX1	Output	Serial output for CAN output driver 1.
TX0_EN	Output	Enable signal for TX0.
TX1_EN	Output	Enable signal for TX1.
XTAL1_IN	Input	Gated system clock, used to drive the parts of the core that are inactive while the MCAN2 is in Sleep Mode.
NXTAL1_IN	Input	Inverted version of XTAL1_IN, used to drive certain synchronization logic at the RX0 line
NXTAL1_ENABLE	Output	Enable signal for XTAL1_IN, active low. Used to disable XTAL1_IN while the MCAN2 is in Sleep Mode.

Notes: (i) The output driver configuration of TX1, TX0 is determined by the Output Control Register (OCR), which is detailed in Section 4.7.

(ii) The purpose of the NXTAL1_ENABLE and XTAL1_IN signals is to allow power to be saved when the MCAN2 is placed in Sleep Mode. XTAL1_IN is produced by OR-ing XTAL1 externally with NXTAL1_ENABLE and is used to clock the majority of the MCAN2's internal registers. When the MCAN2 is placed in Sleep Mode, NXTAL1_ENABLE is taken high, turning XTAL1_IN off. XTAL1 is not switched off while the MCAN2 is in Sleep Mode and is required to wake the MCAN2 synchronously with the surrounding circuitry.

To avoid metastability problems on the return from Sleep Mode, a suitable delay should be built into the XTAL1_IN signal so that the surrounding circuitry is given time to settle before being used by XTAL1_IN. In the testbench provided with the core, this is done by inverting the XTAL1 input to the core. This gives the external signals a full half-cycle to settle. A clock driver should be placed between the gate and XTAL1_IN to minimize the skew between XTAL1 and XTAL1_IN.

4. REGISTER DESCRIPTION

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

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

Note: (i) An 8-bit address bus is provided for software compatibility with the reference device (the Philips SJA1000) and to allow future enhancements of the core.

(ii) Receive data is read from same MCAN2 address to which Transmit data is written (10h – 1Ch). Transmit data may however be read back from 60h – 6Ch.

4.1. 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.
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 [2]	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:

1. 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.
2. Further information about Sleep Mode is given in Section 2.3.9.
3. Further information about Acceptance Filters and the choice of Acceptance Filter Mode is given in Section 6.
4. Further information about Reset Mode is given in Section 5.

4.2. 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 at least one external clock cycle is needed 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.
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.

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 using the self reception feature, 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'.

4.3. 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.

BIT	SYMBOL	NAME	VALUE	FUNCTION
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).

4.4. 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.

Note: The Interrupt Register is read-only. Also, 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]

BIT	SYMBOL	NAME	FUNCTION
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.

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

This read/write register is used to select the events that cause an interrupt to be 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

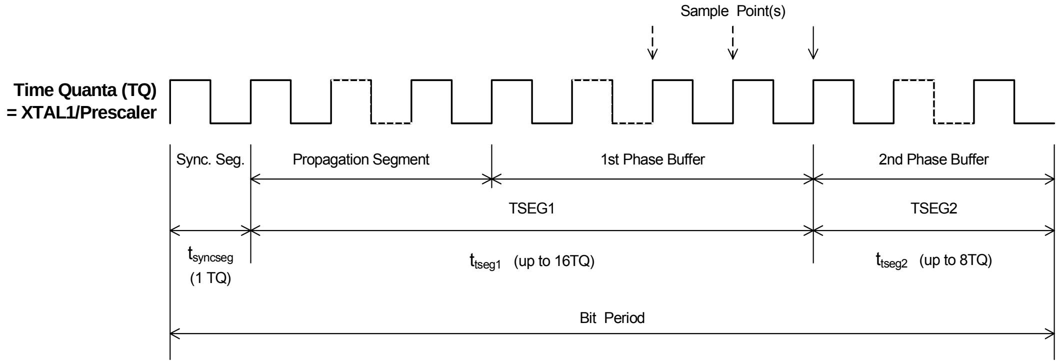
RIE is cleared, NINT will immediately become inactive (high) if no other interrupt is pending.

4.6. BUS TIMING REGISTERS

The MCAN2 has two Bus Timing Registers: BTR0 at Address 06h and BTR1 at 07h. Together they define the structure of the bit period. Both registers can only be written in Reset Mode. In Operating Mode, they are read only.

The CAN specification describes the bit period as being composed of a Synchronization segment, a Propagation segment and two Phase Buffers. For the purposes of BTR1, the bit period is seen as being composed of the Synchronization segment plus TSEG1 and TSEG2, where TSEG1 equals the Propagation segment plus the first Phase Buffer and TSEG2 is the second Phase Buffer as shown in the following diagram.

Figure 4. General Structure of a Bit Period



4.6.1. 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

4.6.1.1. 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.

4.6.1.2. 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 = $1/f_{xtal1}$

4.6.2. 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

4.6.2.1. 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).)

4.6.2.2. 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} in Figure 4 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.

4.7. 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 t_{scl}.

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 zero when read.

4.8. 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 bit 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.
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)

Extended Frame Format messages only

ALC[4:0]	DECIMAL VALUE	FUNCTION
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

Notes

1. RTR Bit in Standard Frame Format messages.

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

This read-only register contains information about the type and location of errors on the bus.

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 4.9.1 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 4.9.2 below

4.9.1. 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

4.9.2. 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

ECC[4:0]	FUNCTION
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
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

4.10. 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 may only be written in Reset Mode. In Operating Mode it is read only. You should also note that changes made within Reset Mode are only put into effect on the return to Operating Mode.

The default value of this register (after hardware reset) is 0110000 (i.e. 96).

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

4.11. 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 be;low), it provides an indication of the quality of transmission being experienced on the CAN bus.

Two levels of the counter trigger specific events.

- When the counter reaches the level set in the Error Warning Limit register (see Section 4.10), 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 hardware reset or when a Bus Off event occurs, 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.

4.12. 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 above), it provides an indication of the quality of transmission being experienced on the CAN bus.

Three levels of the counter trigger specific events.

- When the counter reaches the level set in the Error Warning Limit register (see Section 4.10), 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.

The register is read only in Operating Mode but may be written in Reset Mode.

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).

4.13. 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.

Write-only access to the Transmit Buffer is provided in Operating Mode 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.

4.13.1. 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

4.13.2. 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 4.13.3 – 4.13.7).

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.

4.13.3. 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).

4.13.4. 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. a request for data from another node); '0' indicates a data frame.

4.13.5. DATA LENGTH CODE (DLC)

The DLC [3:0] bits are used to specify the number of data bytes included in 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 start a remote frame transmission with the same Identifier simultaneously.

4.13.6. 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 binary 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.

4.13.7. 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.

4.14. 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.

4.14.1. 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

4.14.2. 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 4.13.3 – 4.13.7 for 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.

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

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

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 6.

The registers are only accessible for Read/Write access in Reset Mode.

4.16. 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 6.

The registers are only accessible for Read / Write access in Reset Mode.

4.17. 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

4.18. 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, a software reset sets the RX FIFO Write Pointer to the value of the RX FIFO Read Pointer, so the data 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.

4.19. 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.

4.19.1. CDR[2:0]

The bits CD.2 to CD.0 are accessible without restrictions in Reset Mode as well as in Operating Mode. These bits are used to define the frequency at the external CLKOUT pin as shown in the following table (f_{osc} is the frequency of the external oscillator (XTAL1)):

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

4.19.2. CLOCK OFF (CDR.3)

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

5. RESET MODE

Setting the Reset Mode bit in the Mode Register (MOD.0: see Section 4.1) causes the current transmission/reception of any message to be aborted and the MCAN2 to enter the Reset Mode. (This happens on the next positive edge of the system clock.) The MCAN2 also goes into Reset Mode following a hardware reset (NRST taken low) and on the Bus Status going to 'Bus Off' (SR.7 set to '1') as happens, for example, if the Transmit Error Counter goes over 255.

The MCAN2 is returned to Operating Mode by clearing the Reset Mode bit. However, it is important to check first that NRST is not active at this point as the Reset Mode bit cannot be set to '0' during an external reset. You should also note that the MCAN2 won't send or receive any data until after it has detected a 'Bus Free' sequence of 11 recessive bits on the bus (if the reset was triggered either by the host CPU or by hardware) or until it has detected 128 such sequences if the reset was caused by the Bus Status going to 'Bus Off' (see Section 4.12).

The following table details the effect of the reset on the MCAN2's registers. X means that the reset has no effect on the value of these registers or bits.

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)

REGISTER	BIT	SYMBOL	NAME	VALUE	
				FOLLOWING HARDWARE RESET	FOLLOWING SOFTWARE RESET (MOD.0) OR BUS OFF
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
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

REGISTER	BIT	SYMBOL	NAME	VALUE	
				FOLLOWING HARDWARE RESET	FOLLOWING SOFTWARE RESET (MOD.0) OR BUS OFF
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 will be cleared and the Transmit Error Counter will be initialized to 127 and used to count-down the CAN-defined bus-off recovery time consisting of 128 occurrences of 11 consecutive recessive bits.

6. ACCEPTANCE FILTERING

The MCAN2 filters the incoming data stream, discarding any message that does not have the required bit pattern in its identifier.

The bit pattern against which the message identifier is matched is recorded in the Acceptance Code Registers ACR0 – 3, masked by the values recorded in the Acceptance Mask Registers AMR0 – 3. ‘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’.

The bit patterns recorded as ACR0 – 3 can either be used as a single 4-byte filter or 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 will be accepted if its identifier matches either filter.

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 follows:

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]

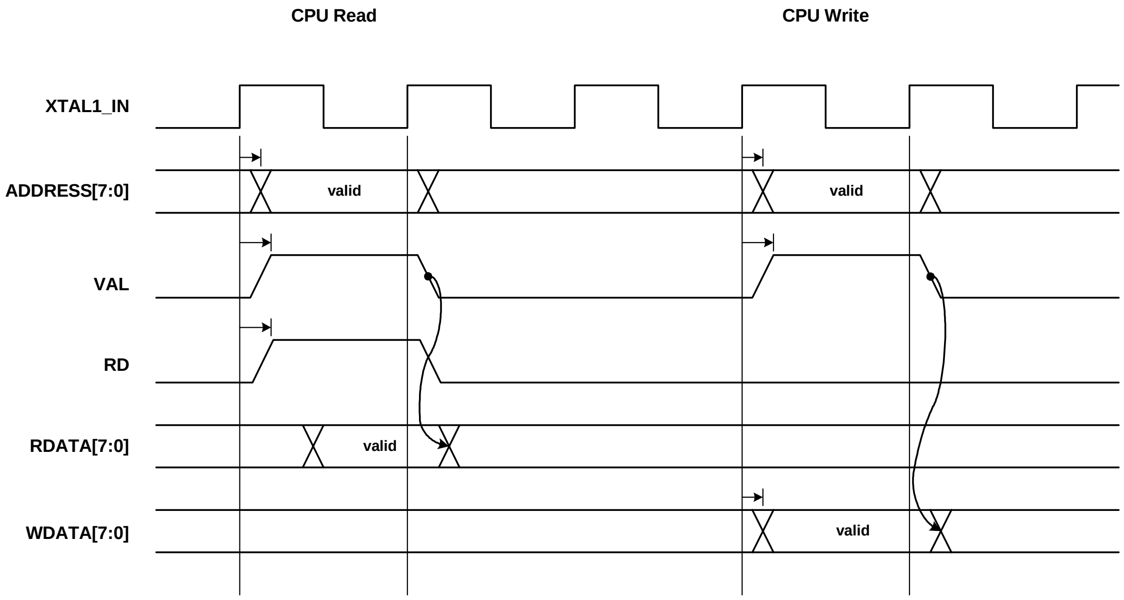
Filter 2:

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

7. TIMING WAVEFORMS

The timing diagram below shows when inputs to the MCAN2 must be valid and when the outputs are valid (e.g. which clock edges sample inputs and which clock edges cause outputs to change). The actual set-up and hold times will depend on the technology and layout used for the MCAN2.

CPU Read/Write Timing



8. SPECIAL FEATURES

This section summarizes the differences between the MCAN2 and the reference Philips SJA1000 device.

8.1. OPERATING MODES

The reference device has two operating modes referred to as BasicCAN and PeliCAN. The MCAN2 has a single operating mode which is broadly compatible with the SJA1000's PeliCAN mode. (The differences between the MCAN2 and the Philips SJA1000's PeliCAN mode are outlined below.)

8.2. HANDLING OF TRANSMITTED MESSAGES

The MCAN2 does not copy transmitted messages to the Receive Buffer as in the reference device. Instead, transmitted messages may be read back from CAN addresses 61h – 6Ch.

8.3. MODE REGISTER

Write access to MOD[3:1] is restricted to Reset Mode in the reference device. In the MCAN2, these bits can be written from either reset Mode or Operating Mode.

8.4. OUTPUT CONTROL REGISTER

The MCAN2 does not support the output voltage level and polarity options selected through OCR Bits 7 – 2 in the reference device. In the MCAN2, these bits are reserved and will return 0 when read.

You should also note that the output modes available for selection through OCR[1:0] are limited to just Normal Output Mode and Clock Output Mode (see Section 4.7).

8.5. CLOCK DIVIDER REGISTER

The options offered through CDR Bits 7 – 5 in the reference device are not supported by the MCAN2. These bits are hard-wired to 1 1 0 in the MCAN2.

8.6. PVCI-COMPATIBLE CPU INTERFACE

The MCAN2's CPU interface has been made to be compatible with the Peripheral Virtual Component Interface defined by the VSI Alliance™ Virtual Component Interface Standard (OCB 2 1.0).

8.7. SLEEP MODE

The reference device uses internal gating to disable the device on entry into Sleep Mode. The MCAN2 instead uses an XTAL1_IN signal to drive the majority of the registers in the core and provides an NXTAL1_ENABLE signal which is intended to be used to turn XTAL1_IN off when the MCAN2 enters Sleep mode.

XTAL1_IN is derived from the system clock XTAL1. XTAL1 is used by several flip flops within the core so that the MCAN2 wakes up in sync with any circuitry external to the core.

9. REVISION HISTORY

9.1. ISSUE 1

3rd October 2001. First issue of document.

9.2. ISSUE 2

31st July 2003. Amendment made to range of bit periods supported (Section 2.3.6).

9.3. ISSUE 3

7th May 2006. Amendment made for CDR R/W description (Section 4.19). Addition made for NXTAL1_IN port (Section 3). Amendment made for TX1 behavior on Sleep mode (Section 2.3.9).

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

Mentor Graphics®

05/06 PS-40107.003-FC