MCP25625

CAN Controller with Integrated Transceiver

General Features

- Stand-Alone CAN 2.0B Controller with Integrated CAN Transceiver and Serial Peripheral Interface (SPI)
- Up to 1 Mb/s Operation
- Very Low Standby Current (10 µA, typical)
- Up to 10 MHz SPI Clock Speed
- Interfaces Directly with Microcontrollers with 2.7V to 5.5V I/Os
- Available in SSOP-28L and 6x6 QFN-28L
- Temperature Ranges:
  - Extended (E): -40°C to +125°C

CAN Controller Features

- V_{DD}: 2.7 to 5.5V
- Implements CAN 2.0B (ISO11898-1)
- Three Transmit Buffers with Prioritization and Abort Features
- Two Receive Buffers
- Six Filters and Two Masks with Optional Filtering on the First Two Data Bytes
- Supports SPI Modes 0,0 and 1,1
- Specific SPI Commands to Reduce SPI Overhead
- Buffer Full and Request-to-Send Pins are Configurable as General Purpose I/Os
- One Interrupt Output Pin

CAN Transceiver Features

- V_{DDA}: 4.5V to 5.5V
- Implements ISO-11898-2 and ISO-11898-5 Standard Physical Layer Requirements
- CAN Bus Pins are Disconnected when Device is Unpowered:
  - An unpowered node or brown-out event will not load the CAN bus
- Detection of Ground Fault:
  - Permanent Dominant detection on T_{XD}
  - Permanent Dominant detection on bus
- Power-on Reset and Voltage Brown-Out Protection on V_{DDA} Pin
- Protection Against Damage Due to Short-Circuit Conditions (Positive or Negative Battery Voltage)
- Protection Against High-Voltage Transients in Automotive Environments
- Automatic Thermal Shutdown Protection
- Suitable for 12V and 24V Systems
- Meets or Exceeds Stringent Automotive Design Requirements, Including *"Hardware Requirements for LIN, CAN and FlexRay Interfaces in Automotive Applications"*, Version 1.3, May 2012
- High Noise Immunity Due to Differential Bus Implementation
- High-ESD Protection on CANH and CANL, Meets IEC61000-4-2 up to ±8 kV

Description

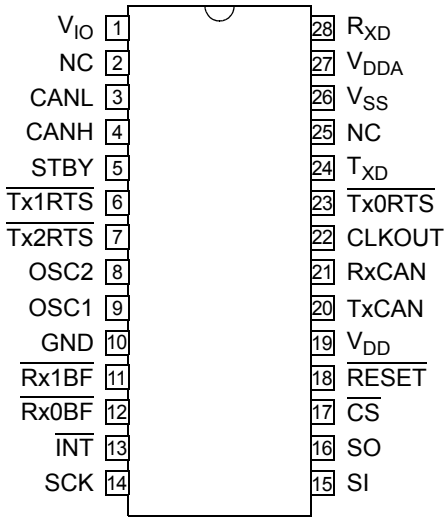
The MCP25625 is a complete, cost-effective and small footprint CAN solution that can be easily added to a microcontroller with an available SPI interface.

The MCP25625 interfaces directly with microcontrollers operating at 2.7V to 5.5V; there are no external level shifters required. In addition, the MCP25625 connects directly to the physical CAN bus, supporting all requirements for CAN high-speed transceivers.

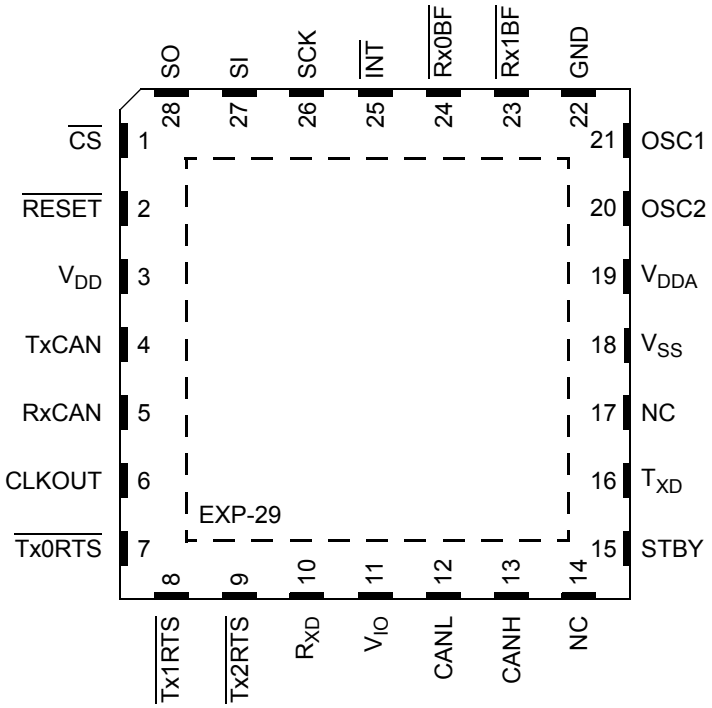
The MCP25625 meets the automotive requirements for high-speed (up to 1 Mb/s), low quiescent current, Electromagnetic Compatibility (EMC) and Electrostatic Discharge (ESD).

Package Types

MCP25625
SSOP



MCP25625
6x6 QFN*



* Includes Exposed Thermal Pad (EP); see Table 1-1.

1.0 DEVICE OVERVIEW

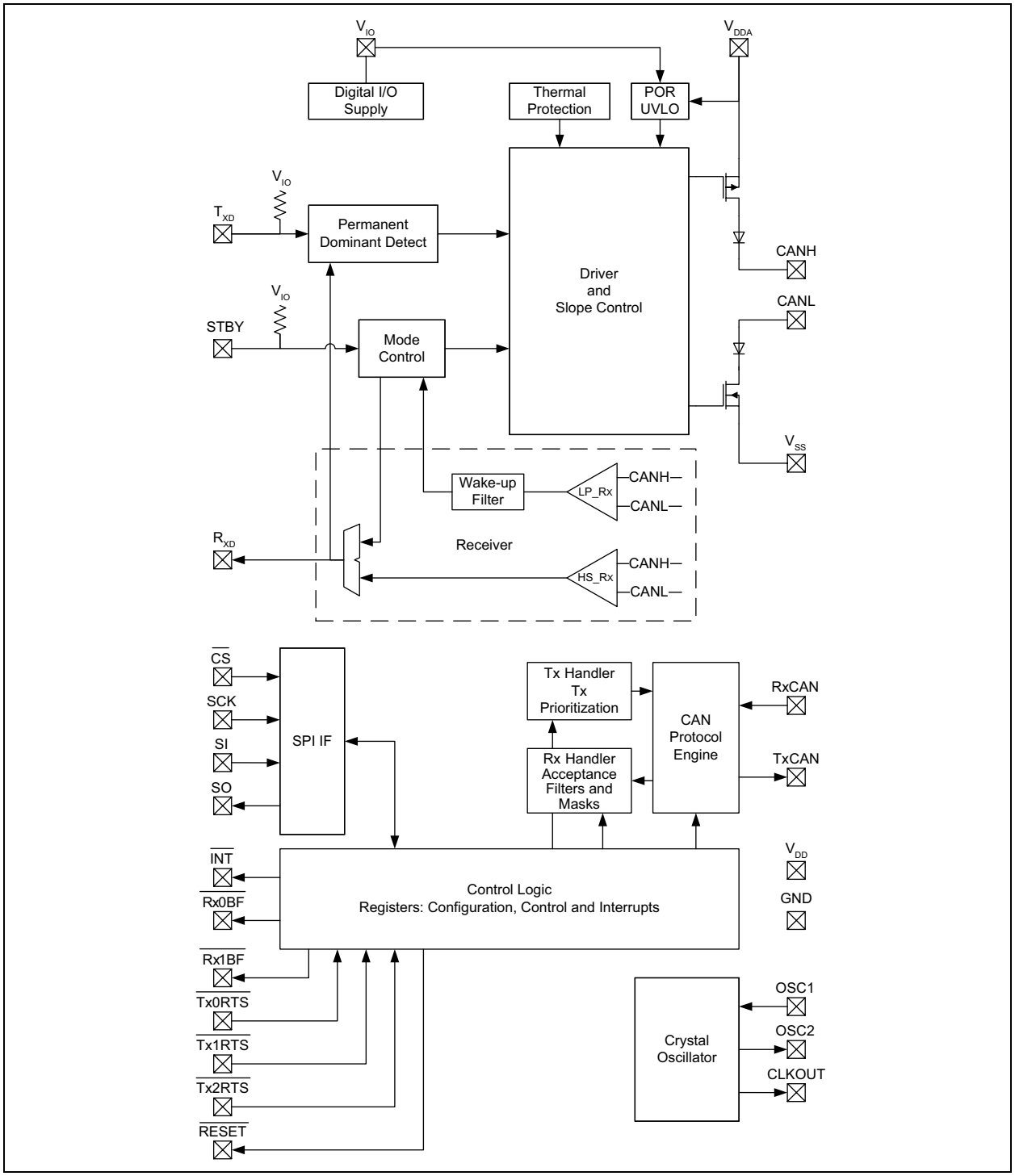
A typical CAN solution consists of a CAN controller that implements the CAN protocol, and a CAN transceiver that serves as the interface to the physical CAN bus. The MCP25625 integrates both the CAN controller and the CAN transceiver. Therefore, it is a complete CAN solution that can be easily added to a microcontroller with an SPI interface.

1.1 Block Diagram

Figure 1-1 shows the block diagram of the MCP25625. The CAN transceiver is illustrated in the top half of the block diagram, see **Section 6.0 "CAN Transceiver"** for more details.

The CAN controller is depicted at the bottom half of the block diagram, and described in more detail in **Section 3.0 "CAN Controller"**.

FIGURE 1-1: MCP25625 BLOCK DIAGRAM



1.2 Pin Out Description

The descriptions of the pins are listed in Table 1-1.

TABLE 1-1: MCP25625 PIN DESCRIPTION

Pin Name	6x6 QFN	SSOP	Block(1)	Pin Type	Description
V_{IO}	11	1	CAN Transceiver	P	Digital I/O Supply Pin for CAN Transceiver
NC	14	2	—	—	No Connection
CANL	12	3	CAN Transceiver	HV I/O	CAN Low-Level Voltage I/O
CANH	13	4	CAN Transceiver	HV I/O	CAN High-Level Voltage I/O
STBY	15	5	CAN Transceiver	I	Standby Mode Input
$\overline{Tx1RTS}$	8	6	CAN Controller	I	TXB1 Request-to-Send
$\overline{Tx2RTS}$	9	7	CAN Controller	I	TXB2 Request-to-Send
OSC2	20	8	CAN Controller	O	External Oscillator Output
OSC1	21	9	CAN Controller	I	External Oscillator Input
GND	22	10	CAN Controller	P	Ground
$\overline{Rx1BF}$	23	11	CAN Controller	O	RxB1 Interrupt
$\overline{Rx0BF}$	24	12	CAN Controller	O	RxB0 Interrupt
$\overline{INT}$	25	13	CAN Controller	O	Interrupt Output
SCK	26	14	CAN Controller	I	SPI Clock Input
SI	27	15	CAN Controller	I	SPI Data Input
SO	28	16	CAN Controller	O	SPI Data Output
$\overline{CS}$	1	17	CAN Controller	I	SPI Chip Select Input
$\overline{RESET}$	2	18	CAN Controller	I	Reset Input
V_{DD}	3	19	CAN Controller	P	Power for CAN Controller
TxCAN	4	20	CAN Controller	O	Transmit Output to CAN Transceiver
RxCAN	5	21	CAN Controller	I	Receive Input from CAN Transceiver
CLKOUT	6	22	CAN Controller	O	Clock Output/SOF
$\overline{Tx0RTS}$	7	23	CAN Controller	I	TXB0 Request-to-Send
T_{XD}	16	24	CAN Transceiver	I	Transmit Data Input from CAN Controller
NC	17	25	—	—	No Connection
V_{SS}	18	26	CAN Transceiver	P	Ground
V_{DDA}	19	27	CAN Transceiver	P	Power for CAN Transceiver
R_{XD}	10	28	CAN Transceiver	O	Receive Data Output to CAN Controller
EP	29	—	—	—	Exposed Thermal Pad

Legend: P = Power, I = Input, O = Output, HV = High Voltage.

Note 1: See **Section 3.0 "CAN Controller"** and **Section 6.0 "CAN Transceiver"** for further information.

1.3 Typical Application

Figure 1-2 shows an example of a typical application of the MCP25625. In this example, the microcontroller operates at 3.3V.

V_{DDA} supplies the CAN transceiver and must be connected to 5V.

V_{DD}, V_{IO} of the MCP25625 are connected to the V_{DD} of the microcontroller. The digital supply can range from 2.7V to 5.5V. Therefore, the I/O of the MCP25625 is connected directly to the microcontroller, no level shifters are required.

The T_{XD} and R_{XD} pins of the CAN transceiver must be externally connected to the TxCAN and RxCAN pins of the CAN controller.

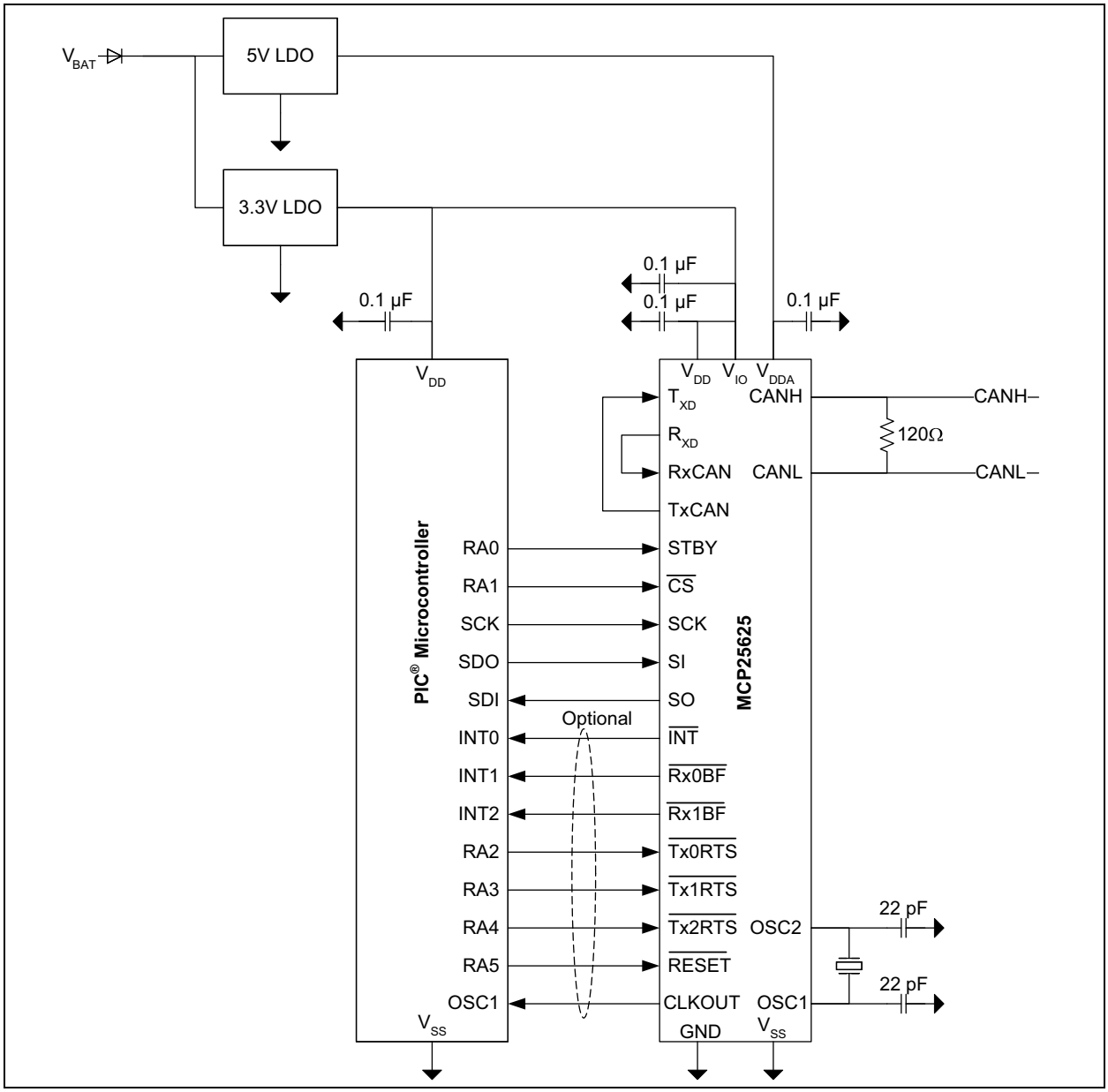
The SPI interface is used to configure and control the CAN controller.

The $\overline{INT}$ pin of the MCP25625 signals an interrupt to the microcontroller. Interrupts need to be cleared by the microcontroller through SPI.

The usage of $\overline{RxnBF}$ and $\overline{TxnRTS}$ is optional, since the functions of these pins can be accessed through SPI. The $\overline{RESET}$ pin can optionally be pulled up to the V_{DD} of the MCP25625 using a 10 kΩ resistor.

The CLKOUT pin provides the clock to the microcontroller.

FIGURE 1-2: MCP25625 INTERFACING WITH A 3.3V MICROCONTROLLER



NOTES:

2.0 MODES OF OPERATION

2.1 CAN Controller Modes of Operation

The CAN controller has five modes of operation:

- Configuration mode
- Normal mode
- Sleep mode
- Listen-Only mode
- Loopback mode

The operational mode is selected via the REQOP[2:0] bits in the CANCTRL register (see Register 4-34).

When changing modes, the mode will not actually change until all pending message transmissions are complete. The requested mode must be verified by reading the OPMOD[2:0] bits in the CANSTAT register (see Register 4-35).

2.2 CAN Transceiver Modes of Operation

The CAN transceiver has two modes of operation:

- Normal mode
- Standby mode

Normal mode is selected by applying a low level to the STBY pin. The driver block is operational and can drive the bus pins. The slopes of the output signals on CANH and CANL are optimized to produce minimal Electromagnetic Emissions (EME). The high-speed differential receiver is active.

Standby mode is selected by applying a high level to the STBY pin. In Standby mode, the transmitter and the high-speed part of the receiver are switched off to minimize power consumption. The low-power receiver and the wake-up filter are enabled in order to monitor the bus for activity. The receive pin (R_{XD}) will show a delayed representation of the CAN bus, due to the wake-up filter.

2.3 Configuration Mode

The MCP25625 must be initialized before activation. This is only possible if the device is in Configuration mode. Configuration mode is automatically selected after power-up, a Reset or can be entered from any other mode by setting the REQOPx bits in the CANCTRL register. When Configuration mode is entered, all error counters are cleared. Configuration mode is the only mode where the following registers are modifiable:

- CNF1, CNF2, CNF3
- TXRTSCTRL
- Acceptance Filter registers

2.4 Normal Mode

Normal mode is the standard operating mode of the MCP25625. In this mode, the device actively monitors all bus messages and generates Acknowledge bits, error frames, etc. This is also the only mode in which the MCP25625 transmits messages over the CAN bus.

Both the CAN controller and the CAN transceiver must be in Normal mode.

2.5 Sleep/Standby Mode

The CAN controller has an internal Sleep mode that is used to minimize the current consumption of the device. The SPI interface remains active for reading even when the MCP25625 is in Sleep mode, allowing access to all registers.

Sleep mode is selected via the REQOPx bits in the CANCTRL register. The OPMODx bits in the CANSTAT register indicate the operation mode. These bits should be read after sending the `SLEEP` command to the MCP25625. The MCP25625 is active and has not yet entered Sleep mode until these bits indicate that Sleep mode has been entered.

When in Sleep mode, the MCP25625 stops its internal oscillator. The MCP25625 will wake-up when bus activity occurs or when the microcontroller sets via the SPI interface. The WAKIF bit in the CANINTF register will “generate” a wake-up attempt (the WAKIE bit in the CANINTE register must also be set in order for the wake-up interrupt to occur).

The CAN transceiver must be in Standby mode in order to take advantage of the low standby current of the transceiver. After a wake-up, the microcontroller must put the transceiver back into Normal mode using the STBY pin.

2.5.1 WAKE-UP FUNCTIONS

The CAN transceiver will monitor the CAN bus for activity. The wake-up filter inside the transceiver is enabled to avoid a wake-up due to noise. In case there is activity on the CAN bus, the R_{XD} pin will go low. The CAN bus wake-up function requires both CAN transceiver supply voltages to be in a valid range: V_{DDA} and V_{IO}.

The CAN controller will detect a falling edge on the RxCAN pin and interrupt the microcontroller if the wake-up interrupt is enabled.

Since the internal oscillator is shut down while in Sleep mode, it will take some amount of time for the oscillator to start-up and the device to enable itself to receive messages. This Oscillator Start-up Timer (OST) is defined as 128 T_{OSC}.

The device will ignore the message that caused the wake-up from Sleep mode, as well as any messages that occur while the device is “waking up”. The device will wake-up in Listen-Only mode.

The microcontroller must set both the CAN controller and CAN transceiver to Normal mode before the MCP25625 will be able to communicate on the bus.

2.6 Listen-Only Mode

Listen-Only mode provides a means for the MCP25625 to receive all messages (including messages with errors) by configuring the RXM[1:0] bits in the RXBxCTRL register. This mode can be used for bus monitor applications or for detecting the baud rate in “hot plugging” situations.

For Auto-Baud Detection (ABD), it is necessary that at least two other nodes are communicating with each other. The baud rate can be detected empirically by testing different values until valid messages are received.

Listen-Only mode is a silent mode, meaning no messages will be transmitted while in this mode (including error flags or Acknowledge signals). In Listen-Only mode, both valid and invalid messages will be received regardless of filters and masks or RXM[1:0] bits in the RXBxCTRL registers. The error counters are reset and deactivated in this state. The Listen-Only mode is activated by setting the REQOPx bits in the CANCTRL register.

2.7 Loopback Mode

Loopback mode will allow internal transmission of messages from the transmit buffers to the receive buffers without actually transmitting messages on the CAN bus. This mode can be used in system development and testing.

In this mode, the ACK bit is ignored and the device will allow incoming messages from itself, just as if they were coming from another node. The Loopback mode is a silent mode, meaning no messages will be transmitted while in this state (including error flags or Acknowledge signals). The TxCAN pin will be in a Recessive state.

The filters and masks can be used to allow only particular messages to be loaded into the receive registers. The masks can be set to all zeros to provide a mode that accepts all messages. The Loopback mode is activated by setting the REQOPx bits in the CANCTRL register.

3.0 CAN CONTROLLER

The CAN controller implements the CAN protocol Version 2.0B. It is compatible with the ISO 11898-1 standard.

Figure 3-1 illustrates the block diagram of the CAN controller. The CAN controller consists of the following major blocks:

- CAN protocol engine
- TX handler
- RX handler
- SPI interface
- Control logic with registers and interrupt logic
- I/O pins
- Crystal oscillator

3.1 CAN Module

The CAN protocol engine, together with the TX and RX handlers, provide all the functions required to receive and transmit messages on the CAN bus. Messages are transmitted by first loading the appropriate message buffers and control registers. Transmission is initiated by using control register bits via the SPI interface or by using the transmit enable pins. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against the user-defined filters to see if it should be moved into one of the two receive buffers.

3.2 Control Logic

The control logic block controls the setup and operation of the MCP25625 and contains the registers.

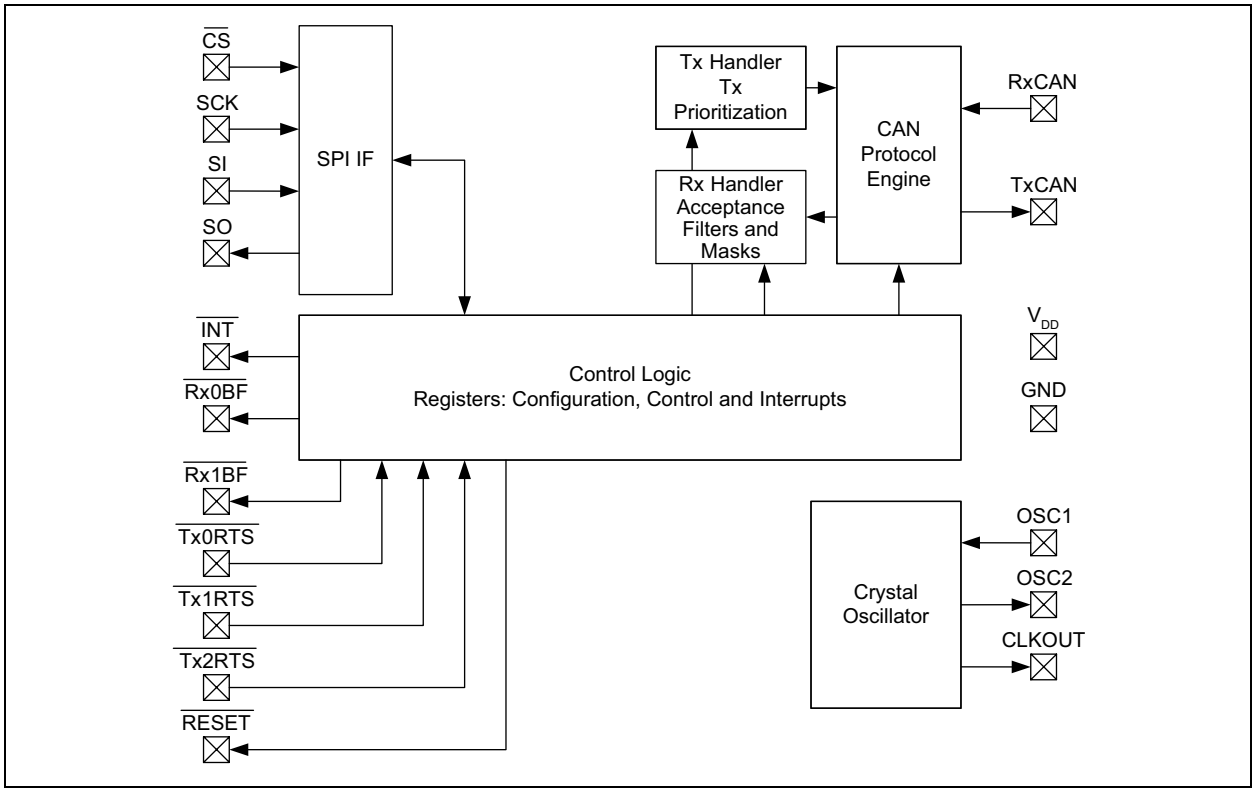
Interrupt pins are provided to allow greater system flexibility. There is one multipurpose interrupt pin (as well as specific interrupt pins) for each of the receive registers that can be used to indicate a valid message has been received and loaded into one of the receive buffers. Use of the specific interrupt pins is optional. The general purpose interrupt pin, as well as Status registers (accessed via the SPI interface), can also be used to determine when a valid message has been received.

Additionally, there are three pins available to initiate immediate transmission of a message that has been loaded into one of the three transmit registers. Use of these pins is optional, as initiating message transmissions can also be accomplished by utilizing control registers accessed via the SPI interface.

3.3 SPI Protocol Block

The microcontroller interfaces to the device via the SPI interface. Registers can be accessed using the SPI `READ` and `WRITE` commands. Specialized SPI commands reduce the SPI overhead.

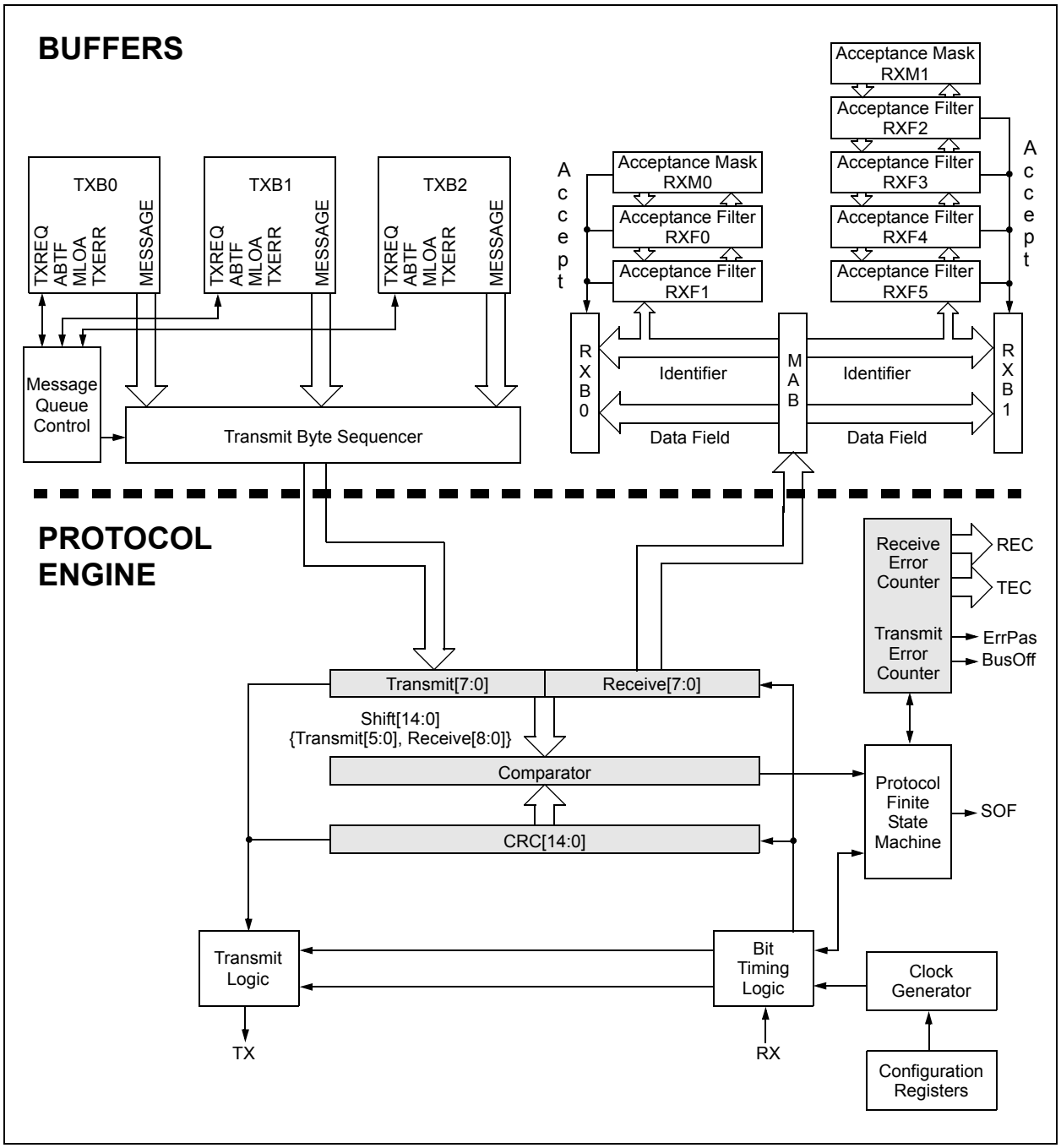
FIGURE 3-1: CAN CONTROLLER BLOCK DIAGRAM



3.4 CAN Buffers and Filters

Figure 3-2 shows the CAN buffers and filters in more detail. The MCP25625 has three transmit and two receive buffers, two acceptance masks (one per receive buffer) and a total of six acceptance filters.

FIGURE 3-2: CAN BUFFERS AND PROTOCOL ENGINE



3.5 CAN Protocol Engine

The CAN protocol engine combines several functional blocks, shown in Figure 3-3 and described below.

3.5.1 PROTOCOL FINITE STATE MACHINE

The heart of the engine is the Finite State Machine (FSM). The FSM is a sequencer that controls the sequential data stream between the TX/RX Shift register, the CRC register and the bus line. The FSM also controls the Error Management Logic (EML) and the parallel data stream between the TX/RX Shift registers and the buffers. The FSM ensures that the processes of reception, arbitration, transmission and error signaling are performed according to the CAN protocol. The automatic retransmission of messages on the bus line is also handled by the FSM.

3.5.2 CYCLIC REDUNDANCY CHECK

The Cyclic Redundancy Check register generates the Cyclic Redundancy Check (CRC) code, which is transmitted after either the control field (for messages with 0 data bytes) or the data field and is used to check the CRC field of incoming messages.

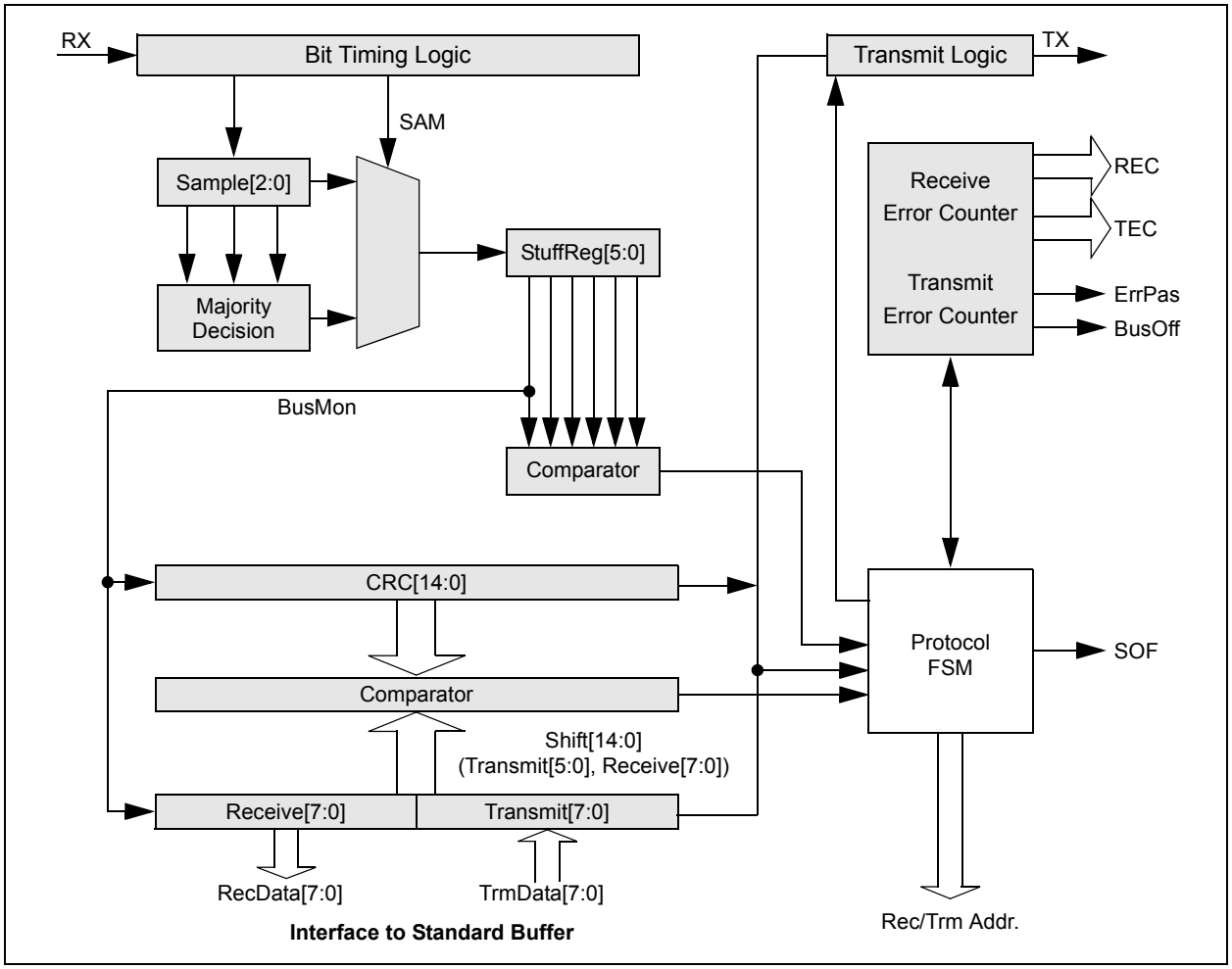
3.5.3 ERROR MANAGEMENT LOGIC

The Error Management Logic (EML) is responsible for the Fault confinement of the CAN device. Its two counters, the Receive Error Counter (REC) and the Transmit Error Counter (TEC), are incremented and decremented by commands from the bit stream processor. Based on the values of the error counters, the CAN controller is set into the states: error-active, error-passive or bus-off.

3.5.4 BIT TIMING LOGIC

The Bit Timing Logic (BTL) monitors the bus line input and handles the bus-related bit timing according to the CAN protocol. The BTL synchronizes on a Recessive-to-Dominant bus transition at Start-of-Frame (hard synchronization) and on any further Recessive-to-Dominant bus line transition if the CAN controller itself does not transmit a Dominant bit (resynchronization). The BTL also provides programmable time segments to compensate for the propagation delay time, phase shifts and to define the position of the sample point within the bit time. The programming of the BTL depends on the baud rate and external physical delay times.

FIGURE 3-3: CAN PROTOCOL ENGINE BLOCK DIAGRAM



3.6 Message Transmission

The transmit registers are described in **Section 4.1 "Message Transmit Registers"**.

3.6.1 TRANSMIT BUFFERS

The MCP25625 implements three transmit buffers. Each of these buffers occupies 14 bytes of SRAM and is mapped into the device memory map.

The first byte, TXBxCTRL, is a control register associated with the message buffer. The information in this register determines the conditions under which the message will be transmitted and indicates the status of the message transmission (see Register 4-1).

Five bytes are used to hold the Standard and Extended Identifiers, as well as other message arbitration information (see Registers 4-3 through 4-7). The last eight bytes are for the eight possible data bytes of the message to be transmitted (see Register 4-8).

At a minimum, the TXBxSIDH, TXBxSIDL and TXBxDLC registers must be loaded. If data bytes are present in the message, the TXBxDn registers must also be loaded. If the message is to use Extended Identifiers, the TXBxEIDn registers must also be loaded and the EXIDE bit in the TXBxSIDL register should be set.

Prior to sending the message, the microcontroller must initialize the TXxIE bit in the CANINTE register to enable or disable the generation of an interrupt when the message is sent.

Note:	The TXREQ bit in the TXBxCTRL register must be clear (indicating the transmit buffer is not pending transmission) before writing to the transmit buffer.

3.6.2 TRANSMIT PRIORITY

Transmit priority is a prioritization within the CAN controller of the pending transmittable messages. This is independent from, and not necessarily related to, any prioritization implicit in the message arbitration scheme built into the CAN protocol.

Prior to sending the Start-of-Frame (SOF), the priority of all buffers that are queued for transmission are compared. The transmit buffer with the highest priority will be sent first. For example, if Transmit Buffer 0 has a higher priority setting than Transmit Buffer 1, Buffer 0 will be sent first.

If two buffers have the same priority setting, the buffer with the highest buffer number will be sent first. For example, if Transmit Buffer 1 has the same priority setting as Transmit Buffer 0, Buffer 1 will be sent first.

The TXP[1:0] bits in the TXBxCTRL register (see Register 4-1) allow the selection of four levels of transmit priority for each transmit buffer individually. A buffer with the TXPx bits equal to '11' has the highest possible priority, while a buffer with the TXPx bits equal to '00' has the lowest possible priority.

3.6.3 INITIATING TRANSMISSION

In order to initiate message transmission, the TXREQ bit in the TXBxCTRL register must be set for each buffer to be transmitted. This can be accomplished by:

- Writing to the register via the SPI `WRITE` command
- Sending the SPI `RTS` command
- Setting the $\overline{\text{TxnRTS}}$ pin low for the particular transmit buffer(s) that is to be transmitted

If transmission is initiated via the SPI interface, the TXREQ bit can be set at the same time as the TXPx priority bits.

When the TXREQ is set, the ABTF, MLOA and TXERR bits in the TXBxCTRL register will be cleared automatically.

Note:	Setting the TXREQ bit in the TXBxCTRL register does not initiate a message transmission. It merely flags a message buffer as being ready for transmission. Transmission will start when the device detects that the bus is available.

Once the transmission has completed successfully, the TXREQ bit will be cleared, the TXxIF bit in the CANINTF register will be set and an interrupt will be generated if the TXxIE bit in the CANINTE register is set.

If the message transmission fails, the TXREQ bit will remain set. This indicates that the message is still pending for transmission and one of the following condition flags will be set:

- If the message started to transmit but encountered an error condition, the TXERR bit in the TXBxCTRL register and the MERRF bit in the CANINTF register will be set, and an interrupt will be generated on the $\overline{\text{INT}}$ pin if the MERRE bit in the CANINTE register is set.
- If arbitration is lost, the MLOA bit in the TXBxCTRL register will be set.

Note:	If One-Shot mode is enabled (OSM bit in the CANCTRL register), the above conditions will still exist. However, the TXREQ bit will be cleared and the message will not attempt transmission a second time.

3.6.4 ONE-SHOT MODE

One-Shot mode ensures that a message will only attempt to transmit one time. Normally, if a CAN message loses arbitration or is destroyed by an error frame, the message is retransmitted. With One-Shot mode enabled, a message will only attempt to transmit one time, regardless of arbitration loss or error frame.

One-Shot mode is required to maintain time slots in deterministic systems, such as TTCAN.

3.6.5 $\overline{\text{TxnRTS}}$ PINS

The $\overline{\text{TxnRTS}}$ pins are input pins that can be configured as:

- Request-to-Send inputs, which provide an alternative means of initiating the transmission of a message from any of the transmit buffers
- Standard digital inputs

Configuration and control of these pins is accomplished using the TXRTSCTRL register (see Register 4-2). The TXRTSCTRL register can only be modified when the CAN controller is in Configuration mode (see **Section 2.0 "Modes of Operation"**). If configured to operate as a Request-to-Send pin, the pin is mapped into the respective TXREQ bit in the TXBxCTRL register for the transmit buffer. The TXREQ bit is latched by the falling edge of the $\overline{\text{TxnRTS}}$ pin. The $\overline{\text{TxnRTS}}$ pins are designed to allow them to be tied directly to the $\overline{\text{RxnBF}}$ pins to automatically initiate a message transmission when the $\overline{\text{RxnBF}}$ pin goes low.

The $\overline{\text{TxnRTS}}$ pins have internal pull-up resistors of 100 kΩ (nominal).

3.6.6 ABORTING TRANSMISSION

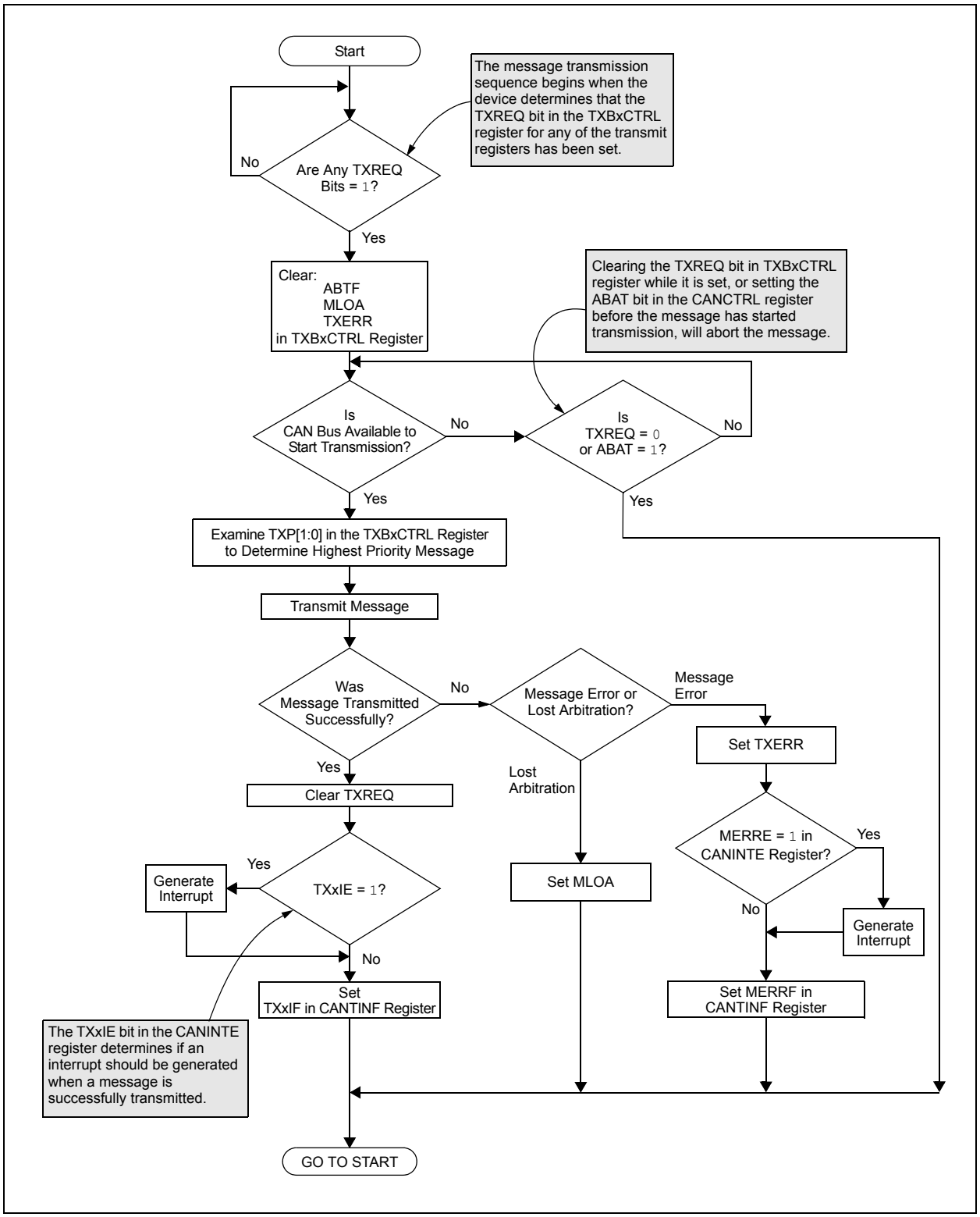
The MCU can request to abort a message in a specific message buffer by clearing the associated TXREQ bit.

In addition, all pending messages can be requested to be aborted by setting the ABAT bit in the CANCTRL register. This bit MUST be reset (typically, after the TXREQ bits have been verified to be cleared) to continue transmitting messages. The ABTF flag in the TXBxCTRL register will only be set if the abort was requested via the ABAT bit in the CANCTRL register. Aborting a message by resetting the TXREQ bit does NOT cause the ABTF bit to be set.

Note 1: Messages that were transmitting when the abort was requested will continue to transmit. If the message does not successfully complete transmission (i.e., lost arbitration or was interrupted by an error frame), it will then be aborted.

2: When One-Shot mode is enabled, if the message is interrupted due to an error frame or loss of arbitration, the ABTF bit in the TXBxCTRL register will be set.

FIGURE 3-4: TRANSMIT MESSAGE FLOWCHART



3.7 Message Reception

The registers required for message reception are described in **Section 4.2 "Message Receive Registers"**.

3.7.1 RECEIVE MESSAGE BUFFERING

The MCP25625 includes two full receive buffers with multiple acceptance filters for each. There is also a separate Message Assembly Buffer (MAB) that acts as a third receive buffer (see Figure 3-6).

3.7.1.1 Message Assembly Buffer

Of the three receive buffers, the MAB is always committed to receiving the next message from the bus. The MAB assembles all messages received. These messages will be transferred to the RXBx buffers (see Registers 4-12 to 4-17) only if the acceptance filter criteria is met.

3.7.1.2 RXB0 and RXB1

The remaining two receive buffers, called RXB0 and RXB1, can receive a complete message from the protocol engine via the MAB. The MCU can access one buffer, while the other buffer is available for message reception, or for holding a previously received message.

Note:	The entire content of the MAB is moved into the receive buffer once a message is accepted. This means that regardless of the type of identifier (Standard or Extended) and the number of data bytes received, the entire receive buffer is overwritten with the MAB contents. Therefore, the contents of all registers in the buffer must be assumed to have been modified when any message is received.

3.7.1.3 Receive Flags/interrupts

When a message is moved into either of the receive buffers, the appropriate RXxIF bit in the CANINTF register is set. This bit must be cleared by the MCU in order to allow a new message to be received into the buffer. This bit provides a positive lockout to ensure that the MCU has finished with the message before the CAN controller attempts to load a new message into the receive buffer.

If the RXxIE bit in the CANINTE register is set, an interrupt will be generated on the $\overline{\text{INT}}$ pin to indicate that a valid message has been received. In addition, the associated $\overline{\text{RxnBF}}$ pin will drive low if configured as a receive buffer full pin. See **Section 3.7.4 "Rx0BF and Rx1BF Pins"** for details.

3.7.2 RECEIVE PRIORITY

RXB0, the higher priority buffer, has one mask and two message acceptance filters associated with it. The received message is applied to the mask and filters for RXB0 first.

RXB1 is the lower priority buffer, with one mask and four acceptance filters associated with it.

In addition to the message being applied to the RB0 mask and filters first, the lower number of acceptance filters makes the match on RXB0 more restrictive and implies a higher priority for that buffer.

When a message is received, the RXBxCTRL[3:0] bits will indicate the acceptance filter number that enabled reception and whether the received message is a remote transfer request.

3.7.2.1 Rollover

Additionally, the RXB0CTRL register can be configured such that, if RXB0 contains a valid message and another valid message is received, an overflow error will not occur and the new message will be moved into RXB1, regardless of the acceptance criteria of RXB1.

3.7.2.2 RXM[1:0] Bits

The RXM[1:0] bits in the RXBxCTRL register set special Receive modes. Normally, these bits are cleared to '00' to enable reception of all valid messages as determined by the appropriate acceptance filters. In this case, the determination of whether or not to receive standard or extended messages is determined by the EXIDE bit in the RFXxSIDL register.

If the RXMx bits are set to '11', the buffer will receive all messages, regardless of the values of the acceptance filters. Also, if a message has an error before the End-of-Frame (EOF), that portion of the message assembled in the MAB before the error frame will be loaded into the buffer. This mode has some value in debugging a CAN system and would not be used in an actual system environment.

Setting the RXMx bits to '01' or '10' is not recommended.

3.7.3 START-OF-FRAME SIGNAL

If enabled, the Start-of-Frame signal is generated on the SOF bit at the beginning of each CAN message detected on the RxCAN pin.

The RxCAN pin monitors an Idle bus for a Recessive-to-Dominant edge. If the Dominant condition remains until the sample point, the MCP25625 interprets this as a SOF and a SOF pulse is generated. If the Dominant condition does not remain until the sample point, the MCP25625 interprets this as a glitch on the bus and no SOF signal is generated. Figure 3-5 illustrates SOF signaling and glitch filtering.

As with One-Shot mode, one use for SOF signaling is for TTCAN-type systems. In addition, by monitoring both the RxCAN pin and the SOF bit, an MCU can detect early physical bus problems by detecting small glitches before they affect the CAN communication.

3.7.4 Rx0BF AND Rx1BF PINS

In addition to the $\overline{\text{INT}}$ pin, which provides an interrupt signal to the MCU for many different conditions, the Receive Buffer Full pins ($\overline{\text{Rx0BF}}$ and $\overline{\text{Rx1BF}}$) can be used to indicate that a valid message has been loaded into RXB0 or RXB1, respectively. The pins have three different configurations (see Table 3-1):

1. Disabled
2. Buffer Full Interrupt
3. Digital Output

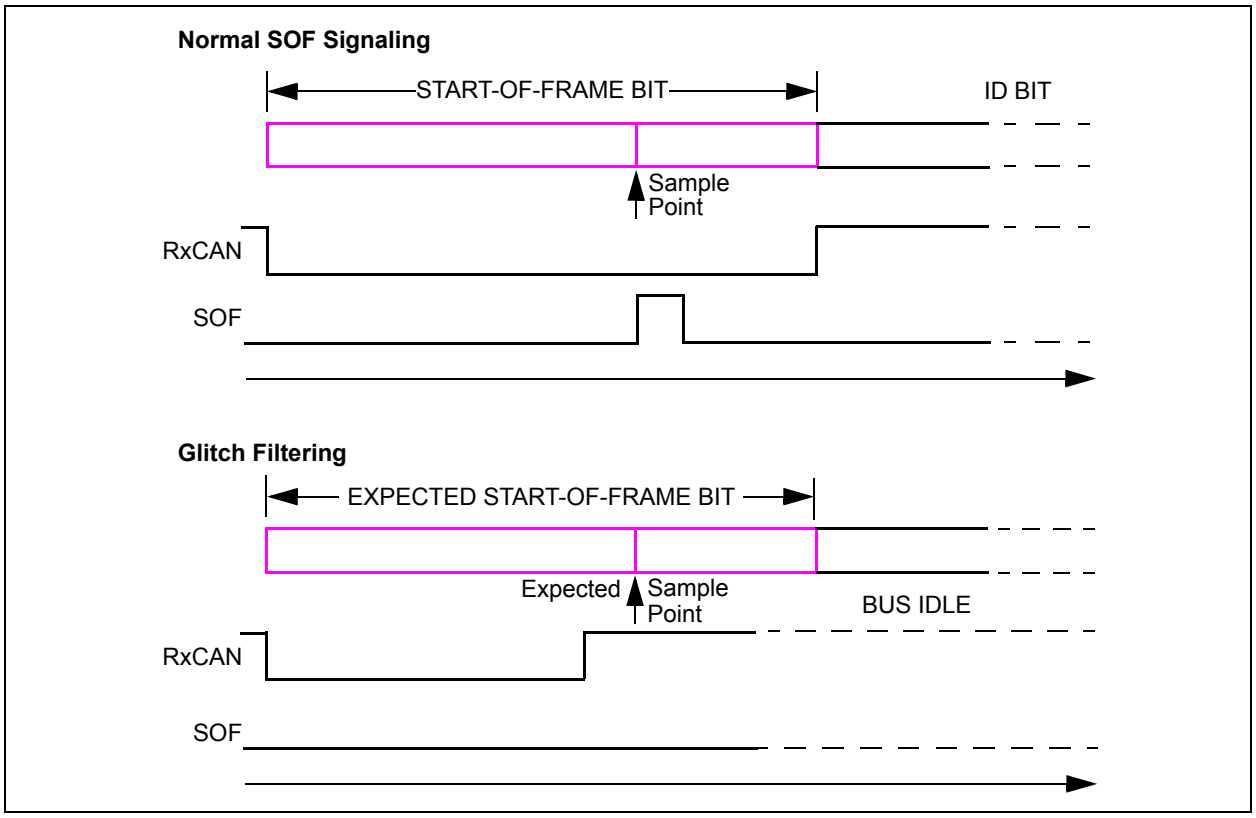
3.7.4.1 Disabled

The $\overline{\text{RxnBF}}$ pins can be disabled to the high-impedance state by clearing the BxBFE bit in the BFPCTRL register.

3.7.4.2 Configured as Buffer Full

The $\overline{\text{RxnBF}}$ pins can be configured to act as either buffer full interrupt pins or as standard digital outputs. Configuration and status of these pins is available via the BFPCTRL register (Register 4-11). When set to operate in Interrupt mode (by setting the BxBFE and BxBFM bits in the BFPCTRL register), these pins are active-low and are mapped to the RXxIF bit in the CANINTF register for each receive buffer. When this bit goes high for one of the receive buffers (indicating that a valid message has been loaded into the buffer), the corresponding $\overline{\text{RxnBF}}$ pin will go low. When the RXxIF bit is cleared by the MCU, the corresponding interrupt pin will go to the logic-high state until the next message is loaded into the receive buffer.

FIGURE 3-5: START-OF-FRAME SIGNALING



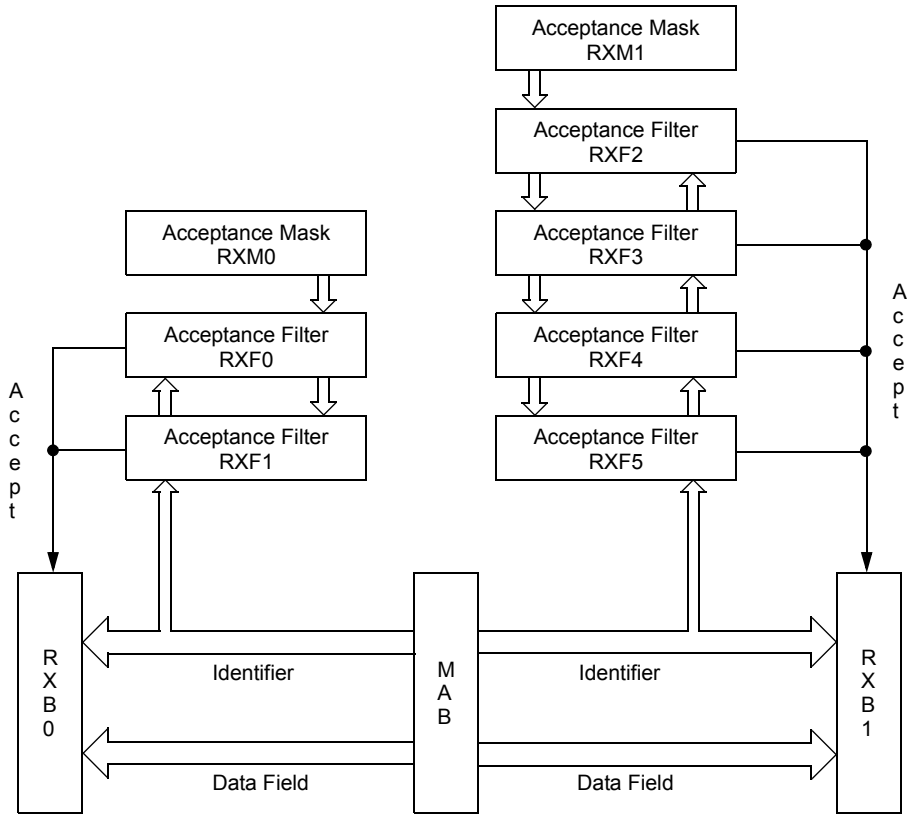
3.7.4.3 Configured as Digital Output

When used as digital outputs, the BxBFM bits in the BFPCTRL register must be cleared and the BxBFE bits must be set for the associated buffer. In this mode, the state of the pin is controlled by the BxBFS bits in the same register. Writing a '1' to the BxBFS bits will cause a high level to be driven on the associated buffer full pin, while a '0' will cause the pin to drive low. When using the pins in this mode, the state of the pin should be modified only by using the SPI `BIT MODIFY` command to prevent glitches from occurring on either of the buffer full pins.

TABLE 3-1: CONFIGURING RxnBF PINS

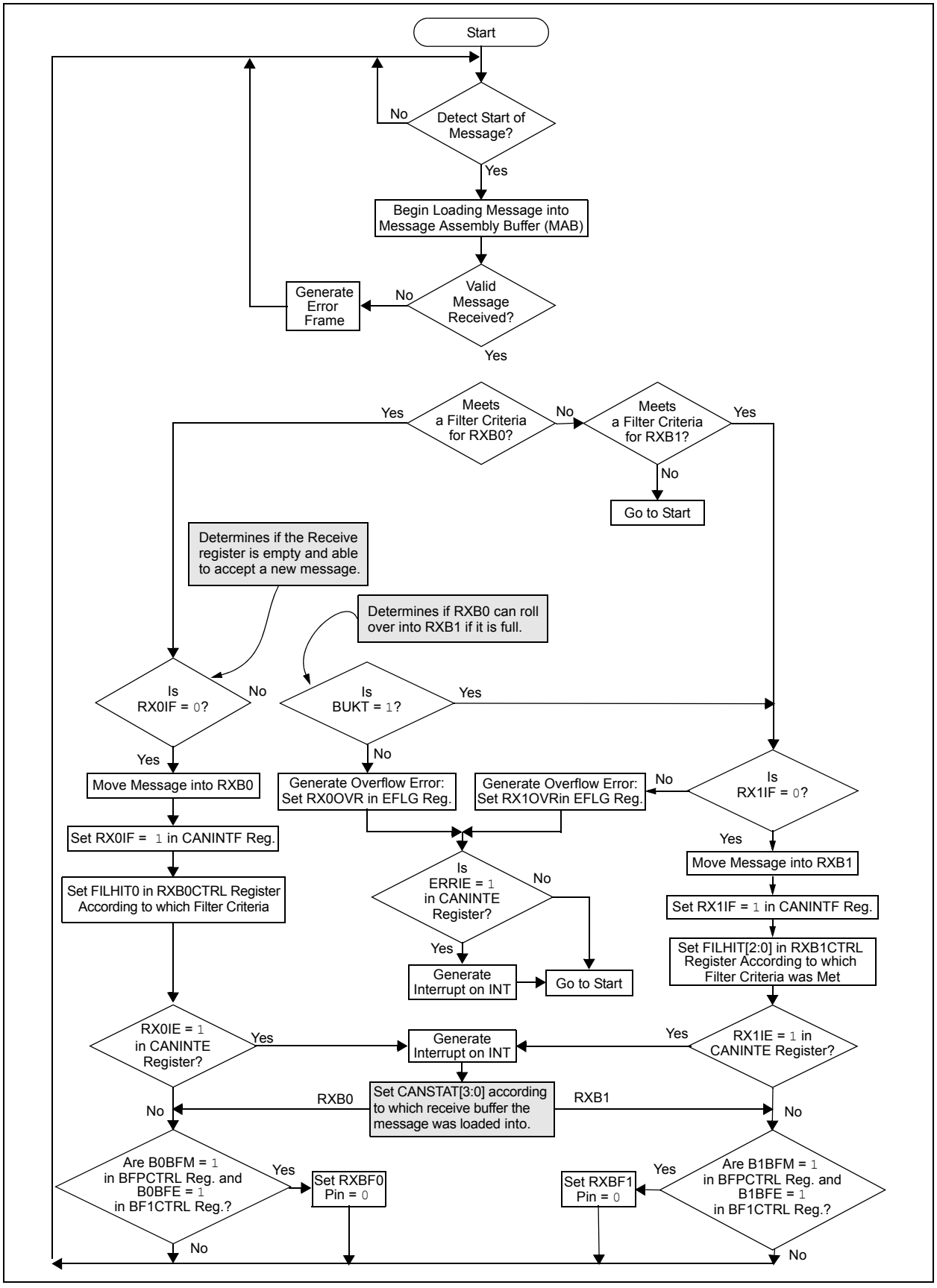
BnBFE	BnBFM	BnBFS	Pin Status
0	X	X	Disabled, high-impedance
1	1	X	Receive buffer interrupt
1	0	0	Digital output = 0
1	0	1	Digital output = 1

FIGURE 3-6: RECEIVE BUFFER BLOCK DIAGRAM



Note: Messages received in the MAB are initially applied to the mask and filters of RXB0. In addition, only one filter match occurs (e.g., if the message matches both RXF0 and RXF2, the match will be for RXF0 and the message will be moved into RXB0).

FIGURE 3-7: RECEIVE FLOW FLOWCHART



3.7.5 MESSAGE ACCEPTANCE FILTERS AND MASKS

The message acceptance filters and masks are used to determine if a message in the Message Assembly Buffer should be loaded into either of the receive buffers (see Figure 3-9). Once a valid message has been received into the MAB, the identifier fields of the message are compared to the filter values. If there is a match, that message will be loaded into the appropriate receive buffer.

The registers required for message filtering are described in **Section 4.3 "Acceptance Filter Registers"**.

3.7.5.1 Data Byte Filtering

When receiving standard data frames (11-bit identifier), the MCP25625 automatically applies 16 bits of masks and filters, normally associated with Extended Identifiers, to the first 16 bits of the data field (data bytes 0 and 1). Figure 3-8 illustrates how masks and filters apply to extended and standard data frames.

Data byte filtering reduces the load on the MCU when implementing Higher Layer Protocols (HLPs) that filter on the first data byte (e.g., DeviceNet™).

3.7.5.2 Filter Matching

The filter masks (see Registers 4-22 through 4-25) are used to determine which bits in the identifier are examined with the filters. A truth table is shown in Table 3-2 that indicates how each bit in the identifier is compared to the masks and filters to determine if the message should be loaded into a receive buffer. The mask essentially determines which bits to apply the acceptance filters to. If any mask bit is set to a zero, that bit will automatically be accepted, regardless of the filter bit.

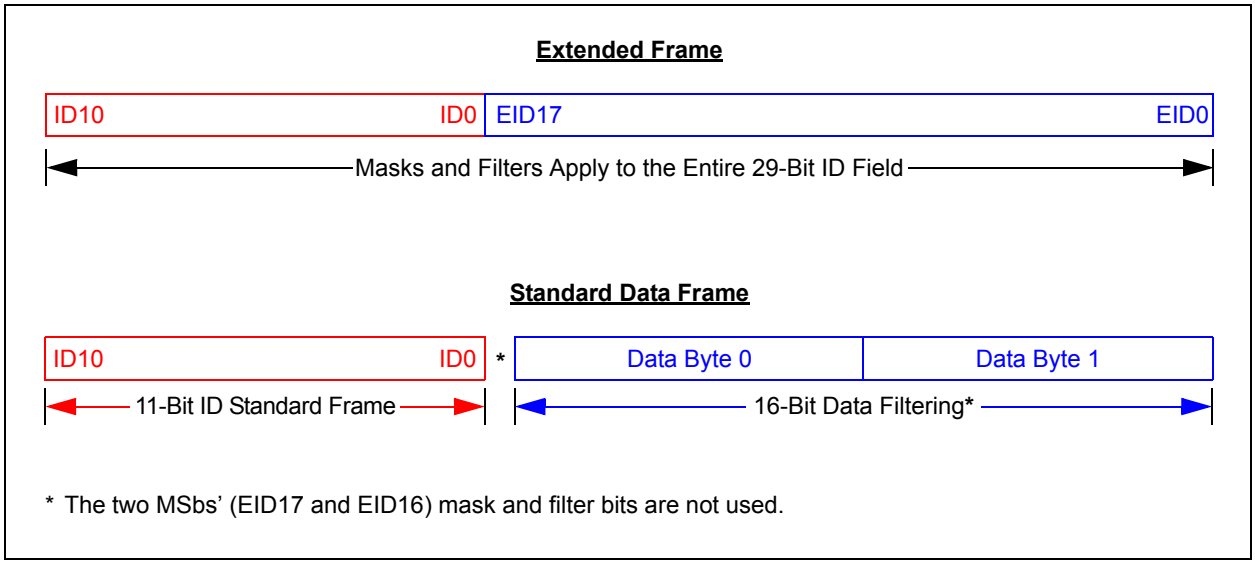
TABLE 3-2: FILTER/MASK TRUTH TABLE

Mask Bit n	Filter Bit n	Message Identifier Bit	Accept or Reject Bit n
0	x	x	Accept
1	0	0	Accept
1	0	1	Reject
1	1	0	Reject
1	1	1	Accept

Note: x = Don't care.

As shown in the Receive Buffer Block Diagram (Figure 3-6), acceptance filters, RXF0 and RXF1 (and filter mask, RXM0), are associated with RXB0. Filters, RXF2, RXF3, RXF4, RXF5 and RXM1 mask, are associated with RXB1.

FIGURE 3-8: MASKS AND FILTERS APPLIED TO CAN FRAMES



3.7.5.3 FILHIT Bits

Filter matches on received messages can be determined by the FILHIT[2:0] bits in the associated RXBxCTRL register. The FILHIT0 bit in the RXB0CTRL register is associated with Buffer 0 and the FILHIT[2:0] bits in the RXB1CTRL register are associated with Buffer 1.

The three FILHITx bits for Receive Buffer 1 (RXB1) are coded as follows:

- `101` = Acceptance Filter 5 (RXF5)
- `100` = Acceptance Filter 4 (RXF4)
- `011` = Acceptance Filter 3 (RXF3)
- `010` = Acceptance Filter 2 (RXF2)
- `001` = Acceptance Filter 1 (RXF1)
- `000` = Acceptance Filter 0 (RXF0)

Note:	'`000`' and '`001`' can only occur if the BUKT bit in RXB0CTRL is set, allowing RXB0 messages to roll over into RXB1.

RXB0CTRL contains two copies of the BUKT bit (BUKT1) and the FILHIT[0] bit.

The coding of the BUKT bit enables these three bits to be used similarly to the FILHITx bits in the RXB1CTRL register and to distinguish a hit on filters, RXF0 and RXF1, in either RXB0 or after a rollover into RXB1.

- `111` = Acceptance Filter 1 (RXB1)
- `110` = Acceptance Filter 0 (RXB0)
- `001` = Acceptance Filter 1 (RXB1)
- `000` = Acceptance Filter 0 (RXB0)

If the BUKT bit is clear, there are six codes corresponding to the six filters. If the BUKT bit is set, there are six codes corresponding to the six filters, plus two additional codes corresponding to the RXF0 and RXF1 filters that roll over into RXB1.

3.7.5.4 Multiple Filter Matches

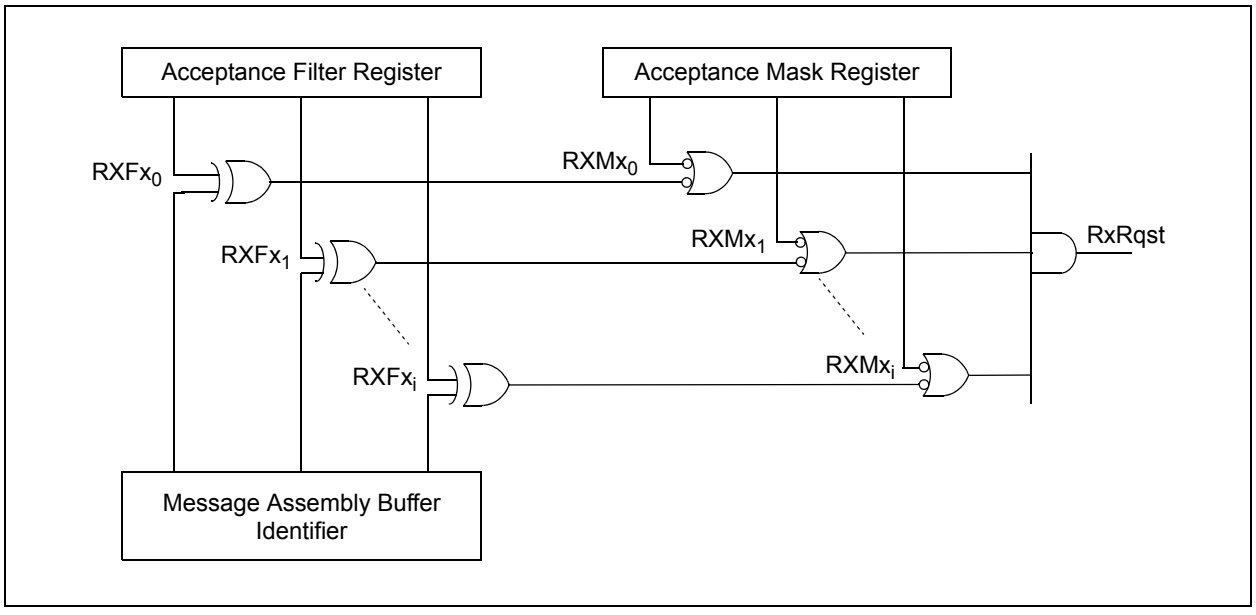
If more than one acceptance filter matches, the FILHITx bits will encode the binary value of the lowest numbered filter that matched. For example, if filters, RXF2 and RXF4, match, FILHITx will be loaded with the value for RXF2. This essentially prioritizes the acceptance filters with a lower numbered filter having higher priority. Messages are compared to filters in ascending order of filter number. This also ensures that the message will only be received into one buffer. This implies that RXB0 has a higher priority than RXB1.

3.7.5.5 Configuring the Masks and Filters

The Mask and Filter registers can only be modified when the MCP25625 is in Configuration mode (see **Section 2.0 "Modes of Operation"**).

Note:	The Mask and Filter registers read all '`0`'s when in any mode except Configuration mode.

FIGURE 3-9: MESSAGE ACCEPTANCE MASK AND FILTER OPERATION



3.8 CAN Bit Time

The Nominal Bit Rate (NBR) is the number of bits per second transmitted on the CAN bus (see Equation 3-1).

EQUATION 3-1: NOMINAL BIT RATE/TIME

$$NBR = \frac{1}{NBT}$$

The Nominal Bit Time (NBT) is made up of four non-overlapping segments. Each of these segments is made up of an integer number of so called Time Quanta (T_Q).

The length of each Time Quantum is based on the oscillator period (T_{OSC}). Equation 3-2 illustrates how the Time Quantum can be programmed using the Baud Rate Prescaler (BRP):

EQUATION 3-2: TIME QUANTA

$$T_Q = 2 \times (BRP<5:0> + 1) \times T_{OSC} = \frac{2 \times (BRP<5:0> + 1)}{F_{OSC}}$$

Figure 3-10 illustrates how the Nominal Bit Time is made up of four segments:

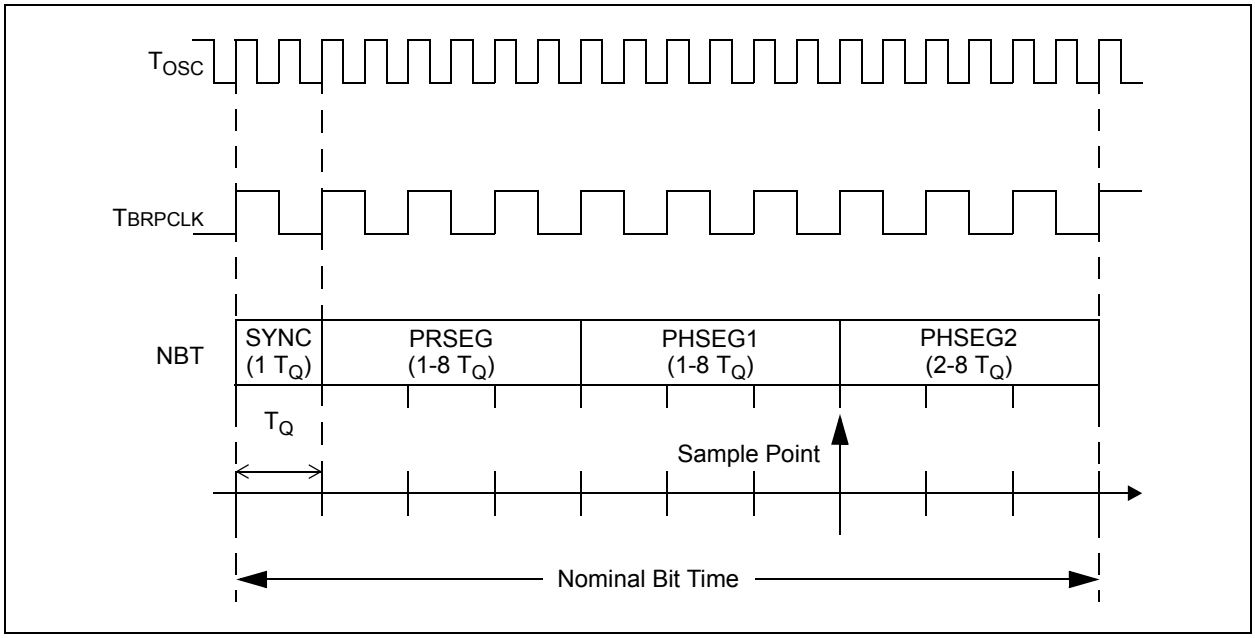
- **Synchronization Segment (SYNC)** – Synchronizes the different nodes connected on the CAN bus. A bit edge is expected to be within this segment. Based on the CAN protocol, the Synchronization Segment is 1 T_Q. See **Section 3.8.3 "Synchronization"** for more details on synchronization.
- **Propagation Segment (PRSEG)** – Compensates for the propagation delay on the bus. It is programmable from 1 to 8 T_Q.
- **Phase Segment 1 (PHSEG1)** – This time segment compensates for errors that may occur due to phase shifts in the edges. The time segment may be automatically lengthened during resynchronization to compensate for the phase shift. It is programmable from 1 to 8 T_Q.
- **Phase Segment 2 (PHSEG2)** – This time segment compensates for errors that may occur due to phase shifts in the edges. The time segment may be automatically shortened during resynchronization to compensate for the phase shift. It is programmable from 2 to 8 T_Q.

The total number of Time Quanta in a Nominal Bit Time is programmable and can be calculated using Equation 3-3.

EQUATION 3-3: T_Q PER NBT

$$\frac{NBT}{T_Q} = SYNC + PRSEG + PHSEG1 + PHSEG2$$

FIGURE 3-10: ELEMENTS OF A NOMINAL BIT TIME



3.8.1 SAMPLE POINT

The sample point is the point in the Nominal Bit Time at which the logic level is read and interpreted. The CAN bus can be sampled once or three times, as configured by the SAM bit in the CNF2 register:

- SAM = 0: The sample point is located between PHSEG1 and PHSEG2.
- SAM = 1: One sample point is located between PHSEG1 and PHSEG2. Additionally, two samples are taken at one-half T_Q intervals prior to the end of PHSEG1, with the value of the bit being determined by a majority decision.

The sample point in percent can be calculated using Equation 3-4.

EQUATION 3-4: SAMPLE POINT

$$SP = \frac{PRSEG + PHSEG1}{\frac{NBT}{T_Q}} \times 100$$

3.8.2 INFORMATION PROCESSING TIME

The Information Processing Time (IPT) is the time required for the CAN controller to determine the bit level of a sampled bit. The IPT for the MCP25625 is 2 T_Q. Therefore, the minimum of PHSEG2 is also 2 T_Q.

3.8.3 SYNCHRONIZATION

To compensate for phase shifts between the oscillator frequencies of the nodes on the bus, each CAN controller must be able to synchronize to the relevant edge of the incoming signal.

The CAN controller expects an edge in the received signal to occur within the SYNC segment. Only Recessive-to-Dominant edges are used for synchronization.

There are two mechanisms used for synchronization:

- **Hard Synchronization** – Forces the edge that has occurred to lie within the SYNC segment. The bit time counter is restarted with SYNC.
- **Resynchronization** – If the edge falls outside the SYNC segment, PHSEG1 and PHSEG2 will be adjusted.

For a more detailed description of the CAN synchronization, please refer to *AN754*, *"Understanding Microchip's CAN Module Bit Timing"* (DS00754) and ISO11898-1.

3.8.4 SYNCHRONIZATION JUMP WIDTH

The Synchronization Jump Width (SJW) is the maximum amount PHSEG1 and PHSEG2 can be adjusted during resynchronization. SJW is programmable from 1 to 4 T_Q.

3.8.5 OSCILLATOR TOLERANCE

According to the CAN specification, the bit timing requirements allow ceramic resonators to be used in applications with transmission rates of up to 125 kbps, as a rule of thumb. For the full bus speed range of the CAN protocol, a quartz oscillator is required. A maximum node-to-node oscillator variation of 1.58% is allowed.

The oscillator tolerance (df), around the nominal frequency of the oscillator ($fnom$), is defined in Equation 3-5.

Equation 3-6 and Equation 3-7 describe the conditions for the maximum tolerance of the oscillator.

EQUATION 3-5: OSCILLATOR TOLERANCE

$$(1 - df) \times (fnom \leq F_{OSC} \leq (1 + df) \times fnom)$$

EQUATION 3-6: CONDITION 1

$$df \leq \frac{SJW}{2 \times 10 \times \frac{NBT}{T_Q}}$$

EQUATION 3-7: CONDITION 2

$$df \leq \frac{min(PHSEG1, PHSEG2)}{2 \times \left(13 \times \frac{NBT}{T_Q} - PHSEG2\right)}$$

3.8.6 PROPAGATION DELAY

Figure 3-11 illustrates the propagation delay between two CAN nodes on the bus. Assuming Node A is transmitting a CAN message, the transmitted bit will propagate from the transmitting CAN Node A, through the transmitting CAN transceiver, over the CAN bus, through the receiving CAN transceiver into the receiving CAN Node B.

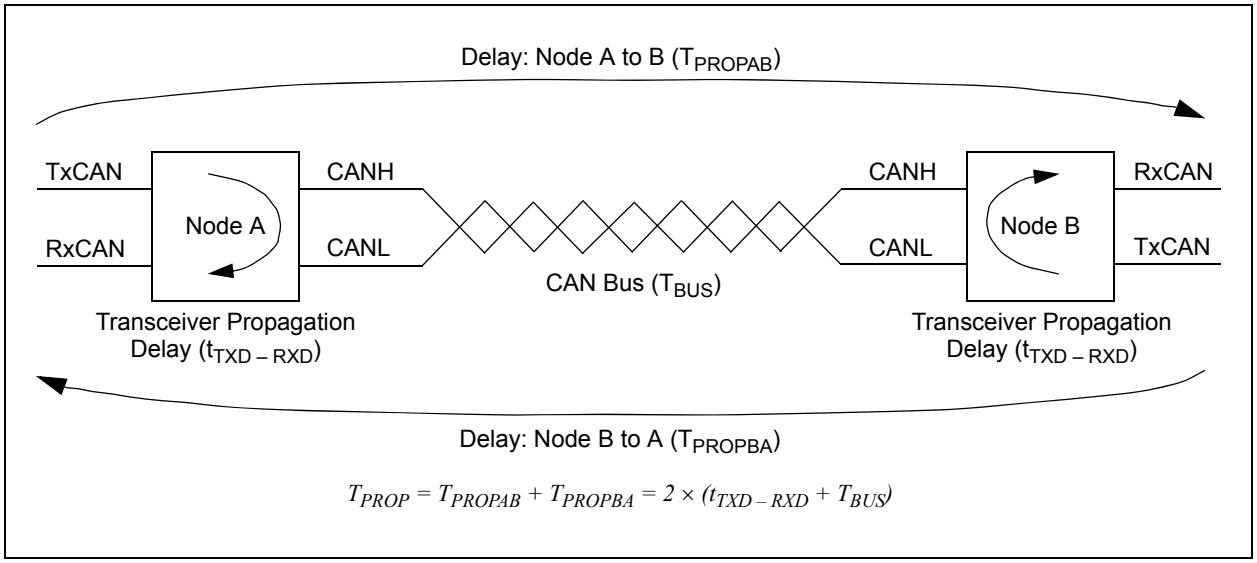
During the arbitration phase of a CAN message, the transmitter samples the bus and checks if the transmitted bit matches the received bit. The transmitting node has to place the sample point after the maximum propagation delay.

Equation 3-8 describes the maximum propagation delay; where $t_{TXD-RXD}$ is the propagation delay of the transceiver, 235 ns for the MCP25625; T_{BUS} is the delay on the CAN bus, approximately 5 ns/m. The factor two comes from the worst case, when Node B starts transmitting exactly when the bit from Node A arrives.

EQUATION 3-8: MAXIMUM PROPAGATION DELAY

$$T_{PROP} = 2 \times (t_{TXD-RXD} + T_{BUS})$$

FIGURE 3-11: PROPAGATION DELAY



3.8.7 BIT TIME CONFIGURATION EXAMPLE

The following example illustrates the configuration of the CAN Bit Time registers. Assuming we want to set up a CAN network in an automobile with the following parameters:

- 500 kbps Nominal Bit Rate (NBR)
- Sample point between 60 and 80% of the Nominal Bit Time (NBT)
- 40 meters minimum bus length

Table 3-3 illustrates how the bit time parameters are calculated. Since the parameters depend on multiple constraints and equations, and are calculated using an iterative process, it is recommended to enter the equations into a spread sheet.

A detailed description of the Bit Time Configuration registers can be found in **Section 4.4 "Bit Time Configuration Registers"**.

TABLE 3-3: STEP-BY-STEP REGISTER CONFIGURATION EXAMPLE

Parameter	Register	Constraint	Value	Unit	Equations and Comments
NBT	—	NBT ≥ 1 µs	2	µs	Equation 3-1
F_{OSC}	—	F_{OSC} ≤ 25 MHz	16	MHz	Select crystal or resonator frequency; usually 16 or 20 MHz work
T_Q/Bit	—	5 to 25	16		The sum of the T_Q of all four segments must be between 5 and 25; selecting 16 T_Q per bit is a good starting point
T_Q	—	NBT, F_{OSC}	125	ns	Equation 3-3
BRP[5:0]	CNF1	0 to 63	0		Equation 3-2
SYNC	—	Fixed	1	T_Q	Defined in ISO 11898-1
PRSEG	CNF2	1 to 8 T_Q; PRSEG > T_{PROP}	7	T_Q	Equation 3-8: T_{PROP} = 870 ns, minimum PRSEG = T_{PROP}/T_Q = 6.96 T_Q; selecting 7 will allow 40m bus length
PHSEG1	CNF2	1 to 8 T_Q; PHSEG1 ≥ SJW[1:0]	4	T_Q	There are 8 T_Q remaining for PHSEG1 + PHSEG2; divide the remaining T_Q in half to maximize SJW[1:0]
PHSEG2	CNF3	2 to 8 T_Q; PHSEG2 ≥ SJW[1:0]	4	T_Q	There are 4 T_Q remaining
SJW[1:0]	CNF1	1 to 4 T_Q; SJW[1:0] ≤ min(PHSEG1, PHSEG2)	4	T_Q	Maximizing SJW[1:0] lessens the requirement for the oscillator tolerance
Sample Point	—	Usually between 60 and 80%	69	%	Use Equation 3-4 to double check the sample point
Oscillator Tolerance Condition 1	—	Double Check	1.25	%	Equation 3-6
Oscillator Tolerance Condition 2	—	Double Check	0.98	%	Equation 3-7; better than 1% crystal oscillator required

3.9 Error Detection

The CAN protocol provides sophisticated error detection mechanisms. The following errors can be detected.

The registers required for error detection are described in **Section 4.5 "Error Detection Registers"**.

3.9.1 CRC ERROR

With the Cyclic Redundancy Check (CRC), the transmitter calculates special check bits for the bit sequence from the Start-of-Frame until the end of the data field. This CRC sequence is transmitted in the CRC field. The receiving node also calculates the CRC sequence using the same formula and performs a comparison to the received sequence. If a mismatch is detected, a CRC error has occurred and an error frame is generated; the message is repeated.

3.9.2 ACKNOWLEDGE ERROR

In the Acknowledge field of a message, the transmitter checks if the Acknowledge slot (which has been sent out as a Recessive bit) contains a Dominant bit. If not, no other node has received the frame correctly. An Acknowledge error has occurred, an error frame is generated and the message will have to be repeated.

3.9.3 FORM ERROR

If a node detects a Dominant bit in one of the four segments (including End-of-Frame, inter-frame space, Acknowledge delimiter or CRC delimiter), a form error has occurred and an error frame is generated. The message is repeated.

3.9.4 BIT ERROR

A bit error occurs if a transmitter detects the opposite bit level to what it transmitted (i.e., transmitted a Dominant and detected a Recessive, or transmitted a Recessive and detected a Dominant).

Exception: In the case where the transmitter sends a Recessive bit, and a Dominant bit is detected during the arbitration field and the Acknowledge slot, no bit error is generated because normal arbitration is occurring.

3.9.5 STUFF ERROR

If, between the Start-of-Frame and the CRC delimiter, six consecutive bits with the same polarity are detected, the bit-stuffing rule has been violated. A stuff error occurs and an error frame is generated; the message is repeated.

3.9.6 ERROR STATES

Detected errors are made known to all other nodes via error frames. The transmission of the erroneous message is aborted and the frame is repeated as soon as possible. Furthermore, each CAN node is in one of the three error states according to the value of the internal error counters:

- Error-active
- Error-passive
- Bus-off (transmitter only)

The error-active state is the usual state where the node can transmit messages and active error frames (made of Dominant bits) without any restrictions.

In the error-passive state, messages and passive error frames (made of Recessive bits) may be transmitted.

The bus-off state makes it temporarily impossible for the station to participate in the bus communication. During this state, messages can neither be received nor transmitted. Only transmitters can go bus-off.

3.10 Error Modes and Error Counters

The MCP25625 contains two error counters: the Receive Error Counter (REC) (see Register 4-30) and the Transmit Error Counter (TEC) (see Register 4-29). The values of both counters can be read by the MCU. These counters are incremented/decremented in accordance with the CAN bus specification.

The MCP25625 is error-active if both error counters are below the error-passive limit of 128.

The device is error-passive if at least one of the error counters equals or exceeds 128.

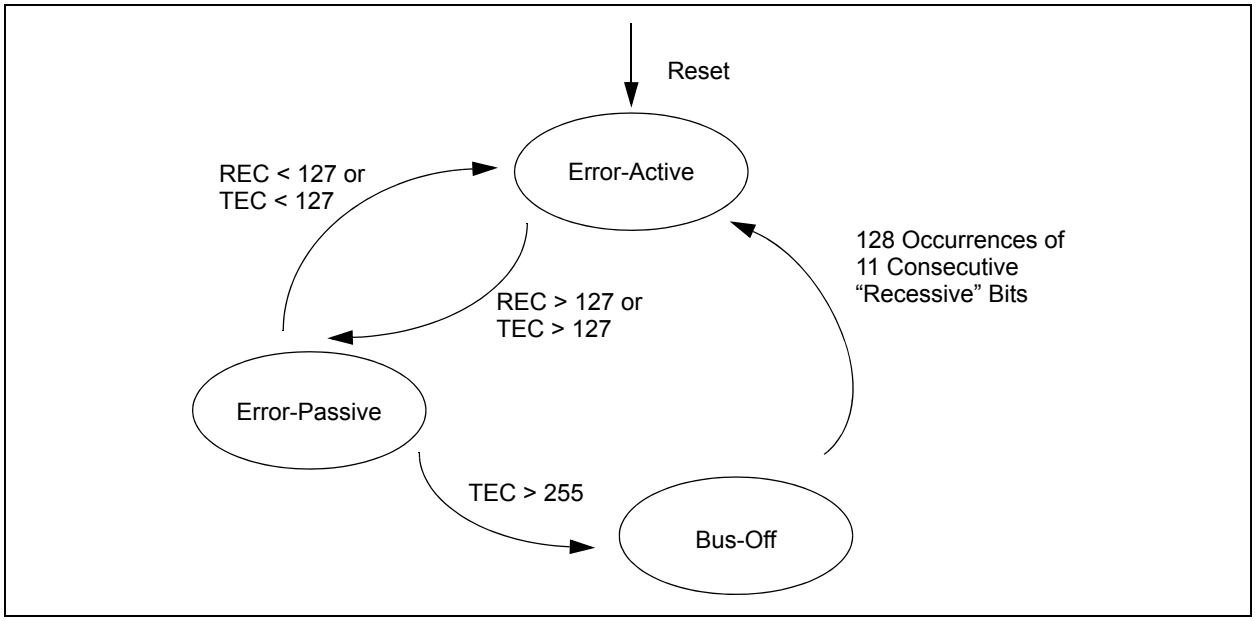
The device goes to bus-off if the TEC exceeds the bus-off limit of 255. The device remains in this state until the bus-off recovery sequence is received. The bus-off recovery sequence consists of 128 occurrences of 11 consecutive Recessive bits (see Figure 3-12).

Note:	The MCP25625, after going bus-off, will recover back to error-active without any intervention by the MCU if the bus remains Idle for 128 x 11 bit times. If this is not desired, the error Interrupt Service Routine (ISR) should address this.

The current Error mode of the MCP25625 can be read by the MCU via the EFLG register (see Register 4-31).

Additionally, there is an error state warning flag bit (EWARN bit in the EFLG register), which is set if at least one of the error counters equals or exceeds the error warning limit of 96. EWARN is reset if both error counters are less than the error warning limit.

FIGURE 3-12: ERROR MODES STATE DIAGRAM



3.11 Interrupts

The MCP25625 has eight sources of interrupts. The CANINTE register contains the individual interrupt enable bits for each interrupt source. The CANINTF register contains the corresponding interrupt flag bit for each interrupt source. When an interrupt occurs, the $\overline{INT}$ pin is driven low by the MCP25625 and will remain low until the interrupt is cleared by the MCU. An interrupt can not be cleared if the respective condition still prevails.

It is recommended that the `BIT MODIFY` command be used to reset the flag bits in the CANINTF register, rather than normal write operations. This is done to prevent unintentionally changing a flag that changes during the `WRITE` command, potentially causing an interrupt to be missed.

It should be noted that the CANINTF flags are read/write and an interrupt can be generated by the microcontroller setting any of these bits, provided the associated CANINTE bit is also set.

The Interrupt registers are described in **Section 4.6 “Interrupt Registers”**.

3.11.1 INTERRUPT CODE BITS

The source of a pending interrupt is indicated in the ICOD[2:0] (Interrupt Code) bits in the CANSTAT register, as indicated in Register 4-35. In the event that multiple interrupts occur, the $\overline{INT}$ pin will remain low until all interrupts have been reset by the MCU. The ICOD bits will reflect the code for the highest priority interrupt that is currently pending. Interrupts are internally prioritized, such that the lower the ICODx value, the higher the interrupt priority. Once the highest priority interrupt condition has been cleared, the code for the next highest priority interrupt that is pending (if any) will be reflected by the ICODx bits (see Table 3-4). Only those interrupt sources that have their associated CANINTE enable bit set will be reflected in the ICODx bits.

TABLE 3-4: ICOD[2:0] DECODE

ICOD[2:0]	Boolean Expression
000	$\overline{ERR}$•$\overline{WAK}$•$\overline{TX0}$•$\overline{TX1}$•$\overline{TX2}$•$\overline{RX0}$•$\overline{RX1}$
001	ERR
010	$\overline{ERR}$•WAK
011	$\overline{ERR}$•$\overline{WAK}$•TX0
100	$\overline{ERR}$•$\overline{WAK}$•$\overline{TX0}$•TX1
101	$\overline{ERR}$•$\overline{WAK}$•$\overline{TX0}$•$\overline{TX1}$•TX2
110	$\overline{ERR}$•$\overline{WAK}$•$\overline{TX0}$•$\overline{TX1}$•$\overline{TX2}$•RX0
111	$\overline{ERR}$•$\overline{WAK}$•$\overline{TX0}$•$\overline{TX1}$•$\overline{TX2}$•$\overline{RX0}$•RX1

Note: $\overline{ERR}$ is associated with the ERRIE bit in the CANINTE register.

3.11.2 TRANSMIT INTERRUPT

When the transmit interrupt is enabled (TXxIE = 1 in the CANINTE register), an interrupt will be generated on the INT pin once the associated transmit buffer becomes empty and is ready to be loaded with a new message. The TXxIF bit in the CANINTF register will be set to indicate the source of the interrupt. The interrupt is cleared by clearing the TXxIF bit.

3.11.3 RECEIVE INTERRUPT

When the receive interrupt is enabled (RXxIE = 1 in the CANINTE register), an interrupt will be generated on the INT pin once a message has been successfully received and loaded into the associated receive buffer. This interrupt is activated immediately after receiving the EOF field. The RXxIF bit in the CANINTF register will be set to indicate the source of the interrupt. The interrupt is cleared by clearing the RXxIF bit.

3.12 Message Error Interrupt

When an error occurs during the transmission or reception of a message, the message error flag (MERRF bit in the CANINTF register) will be set, and if the MERRE bit in the CANINTE register is set, an interrupt will be generated on the INT pin. This is intended to be used to facilitate baud rate determination when used in conjunction with Listen-Only mode.

3.12.1 BUS ACTIVITY WAKE-UP INTERRUPT

When the CAN controller is in Sleep mode and the bus activity wake-up interrupt is enabled (WAKIE = 1 in the CANINTE register), an interrupt will be generated on the INT pin and the WAKIF bit in the CANINTF register will be set when activity is detected on the CAN bus. This interrupt causes the CAN controller to exit Sleep mode. The interrupt is reset by clearing the WAKIF bit.

Note:	The CAN controller wakes up into Listen-Only mode.

3.12.2 ERROR INTERRUPT

When the error interrupt is enabled (ERRIE = 1 in the CANINTE register), an interrupt is generated on the INT pin if an overflow condition occurs, or if the error state of the transmitter or receiver has changed. The Error Flag (EFLG) register will indicate one of the following conditions.

3.12.2.1 Receiver Overflow

An overflow condition occurs when the MAB has assembled a valid receive message (the message meets the criteria of the acceptance filters) and the receive buffer associated with the filter is not available for loading a new message. The associated RXxOVR bit in the EFLG register will be set to indicate the overflow condition. This bit must be cleared by the microcontroller.

3.12.2.2 Receiver Warning

The REC has reached the MCU warning limit of 96.

3.12.2.3 Transmitter Warning

The TEC has reached the MCU warning limit of 96.

3.12.2.4 Receiver Error-Passive

The REC has exceeded the error-passive limit of 127 and the device has gone to error-passive state.

3.12.2.5 Transmitter Error-Passive

The TEC has exceeded the error-passive limit of 127 and the device has gone to error-passive state.

3.12.2.6 Bus-Off

The TEC has exceeded 255 and the device has gone to bus-off state.

3.12.3 INTERRUPT ACKNOWLEDGE

Interrupts are directly associated with one or more status flags in the CANINTF register. Interrupts are pending as long as one of the flags is set. Once an interrupt flag is set by the device, the flag can not be reset by the MCU until the interrupt condition is removed.

3.13 Oscillator

The MCP25625 is designed to be operated with a crystal or ceramic resonator connected to the OSC1 and OSC2 pins. The MCP25625 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications. A typical oscillator circuit is shown in Figure 3-13. The MCP25625 may also be driven by an external clock source connected to the OSC1 pin, as shown in Figure 3-14 and Figure 3-15.

3.13.1 OSCILLATOR START-UP TIMER

The MCP25625 utilizes an Oscillator Start-up Timer (OST) that holds the MCP25625 in Reset to ensure that the oscillator has stabilized before the internal state machine begins to operate. The OST keeps the device in a Reset state for 128 OSC1 clock cycles after the occurrence of a Power-on Reset, SPI Reset, after the assertion of the RESET pin, and after a wake-up from Sleep mode. Note that no SPI protocol operations are to be attempted until after the OST has expired.

3.13.2 CLKOUT PIN

The CLKOUT pin is provided to the system designer for use as the main system clock or as a clock input for other devices in the system. The CLKOUT has an internal prescaler, which can divide F_{OSC} by 1, 2, 4 and 8. The CLKOUT function is enabled and the prescaler is selected via the CANCTRL register (see Register 4-34).

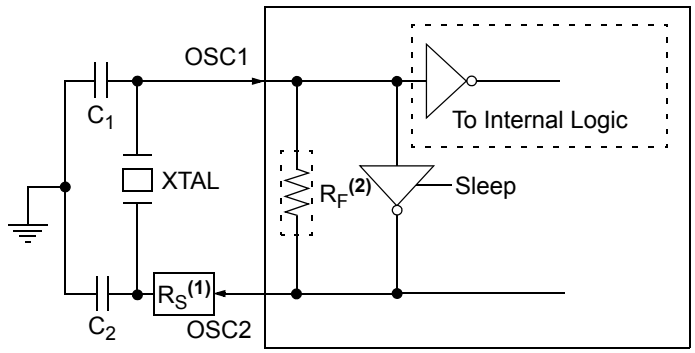
Note:	The maximum frequency on CLKOUT is specified as 25 MHz (see Table 7-5).

The CLKOUT pin will be active upon system Reset and default to the slowest speed (divide-by-8) so that it can be used as the MCU clock.

When Sleep mode is requested, the CAN controller will drive sixteen additional clock cycles on the CLKOUT pin before entering Sleep mode. The Idle state of the CLKOUT pin in Sleep mode is low. When the CLKOUT function is disabled (CLKEN = 0 in the CANCTRL register), the CLKOUT pin is in a high-impedance state.

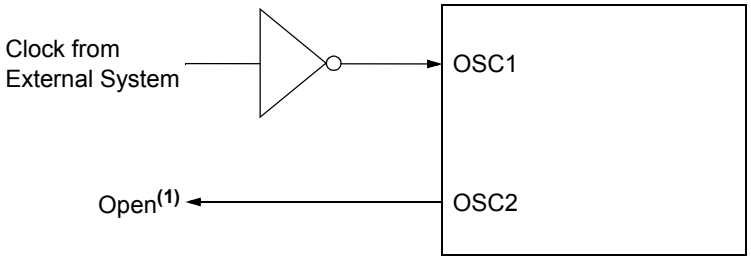
The CLKOUT function is designed to ensure that $t_{HCLKOUT}$ and $t_{LCLKOUT}$ timings are preserved when the CLKOUT pin function is enabled, disabled or the prescaler value is changed.

FIGURE 3-13: CRYSTAL/CERAMIC RESONATOR OPERATION



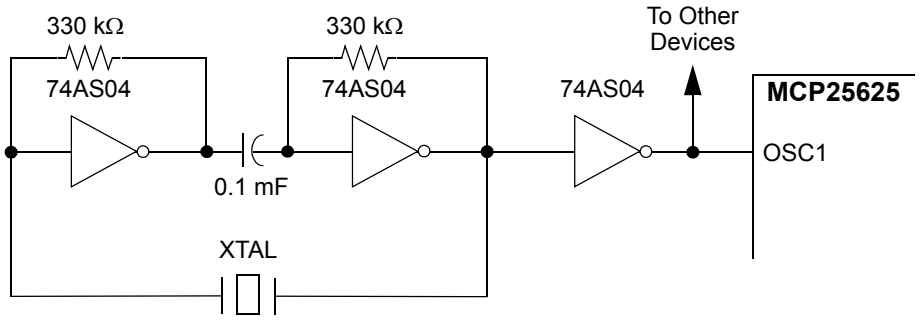
Note 1: A Series Resistor (R_S) may be required for AT strip-cut crystals.
2: The Feedback Resistor (R_F) is typically in the range of 2 to 10 MΩ.

FIGURE 3-14: EXTERNAL CLOCK SOURCE



Note 1: A resistor to ground may be used to reduce system noise. This may increase system current.
2: Duty cycle restrictions must be observed (see Table 7-2).

FIGURE 3-15: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT(1)



Note 1: Duty cycle restrictions must be observed (see Table 7-2).

TABLE 3-5: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:			
Mode	**Freq.**	**OSC1**	**OSC2**
HS	8.0 MHz	27 pF	27 pF
	16.0 MHz	22 pF	22 pF
Capacitor values are for design guidance only: These capacitors were tested with the resonators listed below for basic start-up and operation. **These values are not optimized.** Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected V_{DD} and temperature range for the application. See the notes following Table 3-6 for additional information.			
Resonators Used:			
4.0 MHz			
8.0 MHz			
16.0 MHz			

TABLE 3-6: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type(1)(4)	Crystal Freq.(2)	Typical Capacitor Values Tested:	
		C1	**C2**
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF
Capacitor values are for design guidance only: These capacitors were tested with the crystals listed below for basic start-up and operation. **These values are not optimized.** Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected V_{DD} and temperature range for the application. See the notes following this table for additional information.			
Crystals Used(3):			
4.0 MHz			
8.0 MHz			
20.0 MHz			

Note 1: While higher capacitance increases the stability of the oscillator, it also increases the start-up time.

2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

3: R_S may be required to avoid overdriving crystals with low drive level specification.

4: Always verify oscillator performance over the V_{DD} and temperature range that is expected for the application.

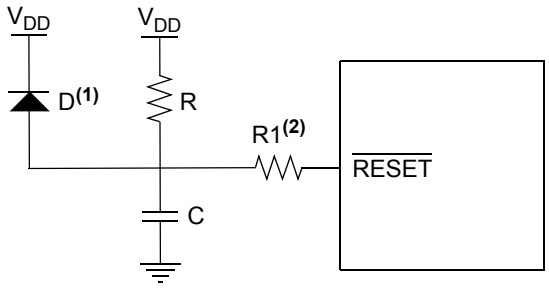
3.14 Reset

The MCP25625 differentiates between two Resets:

1. Hardware Reset – Low on $\overline{\text{RESET}}$ pin
2. SPI Reset – Reset via SPI command (see **Section 5.1 "RESET Instruction"**)

Both of these Resets are functionally equivalent. It is important to provide one of these two Resets after power-up to ensure that the logic and registers are in their default state. A hardware Reset can be achieved automatically by placing an RC on the $\overline{\text{RESET}}$ pin (see Figure 3-16). The values must be such that the device is held in Reset for a minimum of 2 µs after V_{DD} reaches the operating voltage, as indicated in the electrical specification (t_{RL}).

FIGURE 3-16: $\overline{\text{RESET}}$ PIN CONFIGURATION EXAMPLE



Note 1: The diode, D, helps discharge the capacitor quickly when V_{DD} powers down.

2: R1 = 1 kΩ to 10 kΩ will limit any current flowing into $\overline{\text{RESET}}$ from external capacitor, C, in the event of $\overline{\text{RESET}}$ pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

4.0 REGISTER MAP

The register map for the MCP25625 is shown in Table 4-1. Address locations for each register are determined by using the column (higher order four bits) and row (lower order four bits) values. The registers have been arranged to optimize the sequential reading and writing of data. Some specific control and status registers allow individual bit modification using the SPI `BIT MODIFY` command. The registers that allow this command are shown as shaded locations in Table 4-1. A summary of the MCP25625 control registers is shown in Table 4-2.

TABLE 4-1: CAN CONTROLLER REGISTER MAP(1)

Lower Address Bits	Higher Order Address Bits							
	0000 xxxx	0001 xxxx	0010 xxxx	0011 xxxx	0100 xxxx	0101 xxxx	0110 xxxx	0111 xxxx
0000	RXF0SIDH	RXF3SIDH	RXM0SIDH	TXB0CTRL	TXB1CTRL	TXB2CTRL	RXB0CTRL	RXB1CTRL
0001	RXF0SIDL	RXF3SIDL	RXM0SIDL	TXB0SIDH	TXB1SIDH	TXB2SIDH	RXB0SIDH	RXB1SIDH
0010	RXF0EID8	RXF3EID8	RXM0EID8	TXB0SIDL	TXB1SIDL	TXB2SIDL	RXB0SIDL	RXB1SIDL
0011	RXF0EID0	RXF3EID0	RXM0EID0	TXB0EID8	TXB1EID8	TXB2EID8	RXB0EID8	RXB1EID8
0100	RXF1SIDH	RXF4SIDH	RXM1SIDH	TXB0EID0	TXB1EID0	TXB2EID0	RXB0EID0	RXB1EID0
0101	RXF1SIDL	RXF4SIDL	RXM1SIDL	TXB0DLC	TXB1DLC	TXB2DLC	RXB0DLC	RXB1DLC
0110	RXF1EID8	RXF4EID8	RXM1EID8	TXB0D0	TXB1D0	TXB2D0	RXB0D0	RXB1D0
0111	RXF1EID0	RXF4EID0	RXM1EID0	TXB0D1	TXB1D1	TXB2D1	RXB0D1	RXB1D1
1000	RXF2SIDH	RXF5SIDH	CNF3	TXB0D2	TXB1D2	TXB2D2	RXB0D2	RXB1D2
1001	RXF2SIDL	RXF5SIDL	CNF2	TXB0D3	TXB1D3	TXB2D3	RXB0D3	RXB1D3
1010	RXF2EID8	RXF5EID8	CNF1	TXB0D4	TXB1D4	TXB2D4	RXB0D4	RXB1D4
1011	RXF2EID0	RXF5EID0	CANINTE	TXB0D5	TXB1D5	TXB2D5	RXB0D5	RXB1D5
1100	BFPCTRL	TEC	CANINTF	TXB0D6	TXB1D6	TXB2D6	RXB0D6	RXB1D6
1101	TXRTSCTRL	REC	EFLG	TXB0D7	TXB1D7	TXB2D7	RXB0D7	RXB1D7
1110	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT
1111	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL

Note 1: Shaded register locations indicate that these allow the user to manipulate individual bits using the `BIT MODIFY` command.

TABLE 4-2: CONTROL REGISTER SUMMARY

Register Name	Address (Hex)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/RST Value
BFPCTRL	0C	—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM	--00 0000
TXRTSCTRL	0D	—	—	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM	--xx x000
CANSTAT	xE	OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—	100- 000-
CANCTRL	xF	REQOP2	REQOP1	REQOP0	ABAT	OSM	CLKEN	CLKPRE1	CLKPRE0	1000 0111
TEC	1C	Transmit Error Counter (TEC)								0000 0000
REC	1D	Receive Error Counter (REC)								0000 0000
CNF3	28	SOF	WAKFIL	—	—	—	PHSEG2[2:0]			00-- -000
CNF2	29	BTLMODE	SAM	PHSEG1[2:0]			PRSEG2	PRSEG1	PRSEG0	0000 0000
CNF1	2A	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	0000 0000
CANINTE	2B	MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	0000 0000
CANINTF	2C	MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000
EFLG	2D	RX1OVR	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	0000 0000
TXB0CTRL	30	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
TXB1CTRL	40	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
TXB2CTRL	50	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
RXB0CTRL	60	—	RXM1	RXM0	—	RXRTR	BUKT	BUKT1	FILHIT0	-00- 0000
RXB1CTRL	70	—	RXM1	RXM0	—	RXRTR	FILHIT2	FILHIT1	FILHIT0	-00- 0000

4.1 Message Transmit Registers

REGISTER 4-1: TXBxCTRL: TRANSMIT BUFFER x CONTROL REGISTER (ADDRESS: 30h, 40h, 50h)

U-0	R-0	R-0	R-0	R/W-0	U-0	R/W-0	R/W-0
—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **ABTF:** Message Aborted Flag bit
- 1 = Message was aborted
- 0 = Message completed transmission successfully

bit 5 **MLOA:** Message Lost Arbitration bit
- 1 = Message lost arbitration while being sent
- 0 = Message did not lose arbitration while being sent

bit 4 **TXERR:** Transmission Error Detected bit
- 1 = A bus error occurred while the message was being transmitted
- 0 = No bus error occurred while the message was being transmitted

bit 3 **TXREQ:** Message Transmit Request bit
- 1 = Buffer is currently pending transmission (MCU sets this bit to request message be transmitted – bit is automatically cleared when the message is sent)
- 0 = Buffer is not currently pending transmission (MCU can clear this bit to request a message abort)

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **TXP[1:0]:** Transmit Buffer Priority bits
- 11 = Highest message priority
- 10 = High intermediate message priority
- 01 = Low intermediate message priority
- 00 = Lowest message priority

REGISTER 4-2: TXRTSCTRL: $\overline{\text{TxnRTS}}$ PIN CONTROL AND STATUS REGISTER (ADDRESS: 0Dh)

U-0	U-0	R-x	R-x	R-x	R/W-0	R/W-0	R/W-0
—	—	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **B2RTS:** $\overline{\text{Tx2RTS}}$ Pin State bit
- Reads state of $\overline{\text{Tx2RTS}}$ pin when in Digital Input mode
- Reads as '0' when pin is in 'Request-to-Send' mode

bit 4 **B1RTS:** $\overline{\text{Tx1RTX}}$ Pin State bit
- Reads state of $\overline{\text{Tx1RTS}}$ pin when in Digital Input mode
- Reads as '0' when pin is in 'Request-to-Send' mode

bit 3 **B0RTS:** $\overline{\text{Tx0RTS}}$ Pin State bit
- Reads state of $\overline{\text{Tx0RTS}}$ pin when in Digital Input mode
- Reads as '0' when pin is in 'Request-to-Send' mode

bit 2 **B2RTSM:** $\overline{\text{Tx2RTS}}$ Pin mode bit
1 = Pin is used to request message transmission of TXB2 buffer (on falling edge)
0 = Digital input

bit 1 **B1RTSM:** $\overline{\text{Tx1RTS}}$ Pin mode bit
1 = Pin is used to request message transmission of TXB1 buffer (on falling edge)
0 = Digital input

bit 0 **B0RTSM:** $\overline{\text{Tx0RTS}}$ Pin mode bit
1 = Pin is used to request message transmission of TXB0 buffer (on falling edge)
0 = Digital input

REGISTER 4-3: TXBxSIDH: TRANSMIT BUFFER x STANDARD IDENTIFIER HIGH REGISTER (ADDRESS: 31h, 41h, 51h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID[10:3]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **SID[10:3]:** Standard Identifier bits

REGISTER 4-4: TXBxSIDL: TRANSMIT BUFFER x STANDARD IDENTIFIER LOW REGISTER (ADDRESS: 32h, 42h, 52h)

R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **SID[2:0]:** Standard Identifier bits

bit 4 **Unimplemented:** Read as '0'

bit 3 **EXIDE:** Extended Identifier Enable bit

1 = Message will transmit the Extended Identifier
0 = Message will transmit the Standard Identifier

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **EID[17:16]:** Extended Identifier bits

REGISTER 4-5: TXBxEID8: TRANSMIT BUFFER x EXTENDED IDENTIFIER HIGH REGISTER (ADDRESS: 33h, 43h, 53h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID[15:8]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[15:8]:** Extended Identifier bits

REGISTER 4-6: TXBxEID0: TRANSMIT BUFFER x EXTENDED IDENTIFIER LOW REGISTER (ADDRESS: 34h, 44h, 54h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[7:0]:** Extended Identifier bits

REGISTER 4-7: TXBxDLC: TRANSMIT BUFFER x DATA LENGTH CODE REGISTER (ADDRESS: 35h, 45h, 55h)

U-0	R/W-x	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	RTR	—	—	DLC3(1)	DLC2(1)	DLC1(1)	DLC0(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **RTR:** Remote Transmission Request bit
1 = Transmitted message will be a Remote Transmit Request
0 = Transmitted message will be a data frame

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **DLC[3:0]:** Data Length Code bits(1)
Sets the number of data bytes to be transmitted (0 to 8 bytes).

Note 1: It is possible to set the DLC[3:0] bits to a value greater than eight; however, only eight bytes are transmitted.

REGISTER 4-8: TXBxDn: TRANSMIT BUFFER x DATA BYTE n REGISTER (ADDRESS: 36h-3Dh, 46h-4Dh, 56h-5Dh)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
TXBxDn[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **TXBxDn[7:0]:** Transmit Buffer x Data Field Bytes n bits

4.2 Message Receive Registers

REGISTER 4-9: RXB0CTRL: RECEIVE BUFFER 0 CONTROL REGISTER (ADDRESS: 60h)

U-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R-0	R-0
—	RXM1	RXM0	—	RXRTR	BUKT	BUKT1	FILHIT0(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-5 **RXM[1:0]:** Receive Buffer Operating mode bits
- 11 = Turns mask/filters off; receives any message
- 10 = Reserved
- 01 = Reserved
- 00 = Receives all valid messages using either Standard or Extended Identifiers that meet filter criteria; Extended ID Filter registers, RXFxEID8 and RXFxEID0, are applied to the first two bytes of data in the messages with standard IDs.

bit 4 **Unimplemented:** Read as '0'

bit 3 **RXRTR:** Received Remote Transfer Request bit
- 1 = Remote Transfer Request received
- 0 = No Remote Transfer Request received

bit 2 **BUKT:** Rollover Enable bit
- 1 = RXB0 message will roll over and be written to RXB1 if RXB0 is full
- 0 = Rollover is disabled

bit 1 **BUKT1:** Read-Only Copy of BUKT bit (used internally by the MCP25625)

bit 0 **FILHIT0:** Filter Hit bit(1)

Indicates which acceptance filter enabled the reception of a message.
- 1 = Acceptance Filter 1 (RXF1)
- 0 = Acceptance Filter 0 (RXF0)

Note 1: If a rollover from RXB0 to RXB1 occurs, the FILHIT0 bit will reflect the filter that accepted the message that rolled over.

REGISTER 4-10: RXB1CTRL: RECEIVE BUFFER 1 CONTROL REGISTER (ADDRESS: 70h)

U-0	R/W-0	R/W-0	U-0	R-0	R-0	R-0	R-0
—	RXM1	RXM0	—	RXRTR	FILHIT2	FILHIT1	FILHIT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-5 **RXM[1:0]:** Receive Buffer Operating mode bits

11 = Turns mask/filters off; receives any message
10 = Reserved
01 = Reserved
00 = Receives all valid messages using either Standard or Extended Identifiers that meet filter criteria

bit 4 **Unimplemented:** Read as '0'

bit 3 **RXRTR:** Received Remote Transfer Request bit

1 = Remote Transfer Request received
0 = No Remote Transfer Request received

bit 2-0 **FILHIT[2:0]:** Filter Hit bits

Indicates which acceptance filter enabled the reception of a message.

101 = Acceptance Filter 5 (RXF5)
100 = Acceptance Filter 4 (RXF4)
011 = Acceptance Filter 3 (RXF3)
010 = Acceptance Filter 2 (RXF2)
001 = Acceptance Filter 1 (RXF1) (only if the BUKT bit is set in RXB0CTRL)
000 = Acceptance Filter 0 (RXF0) (only if the BUKT bit is set in RXB0CTRL)

REGISTER 4-11: BFPCTRL: $\overline{\text{RxnBF}}$ PIN CONTROL AND STATUS REGISTER (ADDRESS: 0Ch)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **B1BFS:** $\overline{\text{Rx1BF}}$ Pin State bit (Digital Output mode only)
- Reads as '0' when $\overline{\text{Rx1BF}}$ is configured as an interrupt pin

bit 4 **B0BFS:** $\overline{\text{Rx0BF}}$ Pin State bit (Digital Output mode only)
- Reads as '0' when $\overline{\text{Rx0BF}}$ is configured as an interrupt pin

bit 3 **B1BFE:** $\overline{\text{Rx1BF}}$ Pin Function Enable bit
1 = Pin function is enabled, operation mode is determined by the B1BFM bit
0 = Pin function is disabled, pin goes to the high-impedance state

bit 2 **B0BFE:** $\overline{\text{Rx0BF}}$ Pin Function Enable bit
1 = Pin function is enabled, operation mode is determined by the B0BFM bit
0 = Pin function is disabled, pin goes to the high-impedance state

bit 1 **B1BFM:** $\overline{\text{Rx1BF}}$ Pin Operation mode bit
1 = Pin is used as an interrupt when a valid message is loaded into RXB1
0 = Digital Output mode

bit 0 **B0BFM:** $\overline{\text{Rx0BF}}$ Pin Operation mode bit
1 = Pin is used as an interrupt when a valid message is loaded into RXB0
0 = Digital Output mode

REGISTER 4-12: RXBxSIDH: RECEIVE BUFFER x STANDARD IDENTIFIER HIGH REGISTER (ADDRESS: 61h, 71h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
SID[10:3]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **SID[10:3]:** Standard Identifier bits
These bits contain the eight Most Significant bits of the Standard Identifier for the received message.

REGISTER 4-13: RXBxSIDL: RECEIVE BUFFER x STANDARD IDENTIFIER LOW REGISTER (ADDRESS: 62h, 72h)

R-x	R-x	R-x	R-x	R-x	U-0	R-x	R-x
SID2	SID1	SID0	SRR	IDE	—	EID17	EID16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **SID[2:0]:** Standard Identifier bits
These bits contain the three Least Significant bits of the Standard Identifier for the received message.

bit 4 **SRR:** Standard Frame Remote Transmit Request bit (valid only if the IDE bit = 0)
1 = Standard frame Remote Transmit Request received
0 = Standard data frame received

bit 3 **IDE:** Extended Identifier Flag bit
This bit indicates whether the received message was a standard or an extended frame.
1 = Received message was an extended frame
0 = Received message was a standard frame

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **EID[17:16]:** Extended Identifier bits
These bits contain the two Most Significant bits of the Extended Identifier for the received message.

REGISTER 4-14: RXBxEID8: RECEIVE BUFFER x EXTENDED IDENTIFIER HIGH REGISTER (ADDRESS: 63h, 73h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID[15:8]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[15:8]:** Extended Identifier bits
These bits hold bits 15 through 8 of the Extended Identifier for the received message.

REGISTER 4-15: RXBxEID0: RECEIVE BUFFER x EXTENDED IDENTIFIER LOW REGISTER (ADDRESS: 64h, 74h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[7:0]:** Extended Identifier bits
These bits hold the Least Significant eight bits of the Extended Identifier for the received message.

REGISTER 4-16: RXBxDLC: RECEIVE BUFFER x DATA LENGTH CODE REGISTER (ADDRESS: 65h, 75h)

U-0	R-x	R-x	R-x	R-x	R-x	R-x	R-x
—	RTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **RTR:** Extended Frame Remote Transmission Request bit
(valid only when the IDE bit in the RXBxSIDL register is '1')
1 = Extended frame Remote Transmit Request received
0 = Extended data frame received

bit 5 **RB1:** Reserved Bit 1

bit 4 **RB0:** Reserved Bit 0

bit 3-0 **DLC[3:0]:** Data Length Code bits
Indicates the number of data bytes that were received.

REGISTER 4-17: RXBxDn: RECEIVE BUFFER x DATA BYTE n REGISTER (ADDRESS: 66h-6Dh, 76h-7Dh)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
RBxD[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **RBxD[7:0]:** Receive Buffer x Data Field Bytes n bits
Eight bytes containing the data bytes for the received message.

4.3 Acceptance Filter Registers

REGISTER 4-18: RXFxSIDH: FILTER x STANDARD IDENTIFIER HIGH REGISTER (ADDRESS: 00h, 04h, 08h, 10h, 14h, 18h)(1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID[10:3]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **SID[10:3]:** Standard Identifier Filter bits
These bits hold the filter bits to be applied to bits[10:3] of the Standard Identifier portion of a received message.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

REGISTER 4-19: RXFxSIDL: FILTER x STANDARD IDENTIFIER LOW REGISTER (ADDRESS: 01h, 05h, 09h, 11h, 15h, 19h)(1)

R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **SID[2:0]:** Standard Identifier Filter bits
These bits hold the filter bits to be applied to bits[2:0] of the Standard Identifier portion of a received message.

bit 4 **Unimplemented:** Read as '0'

bit 3 **EXIDE:** Extended Identifier Enable bit
1 = Filter is applied only to extended frames
0 = Filter is applied only to standard frames

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **EID[17:16]:** Extended Identifier Filter bits
These bits hold the filter bits to be applied to bits[17:16] of the Extended Identifier portion of a received message.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

REGISTER 4-20: RXFxEID8: FILTER x EXTENDED IDENTIFIER HIGH REGISTER (ADDRESS: 02h, 06h, 0Ah, 12h, 16h, 1Ah)(1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID[15:8]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[15:8]:** Extended Identifier bits
These bits hold the filter bits to be applied to bits[15:8] of the Extended Identifier portion of a received message or to Byte 0 in received data if corresponding with RXM[1:0] = 00 and EXIDE = 0.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

REGISTER 4-21: RXFxEID0: FILTER x EXTENDED IDENTIFIER LOW REGISTER (ADDRESS: 03h, 07h, 0Bh, 13h, 17h, 1Bh)(1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[7:0]:** Extended Identifier bits
These bits hold the filter bits to be applied to bits[7:0] of the Extended Identifier portion of a received message or to Byte 1 in received data if corresponding with RXM[1:0] = 00 and EXIDE = 0.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

REGISTER 4-22: RXMxSIDH: MASK x STANDARD IDENTIFIER HIGH REGISTER (ADDRESS: 20h, 24h)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SID[10:3]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **SID[10:3]:** Standard Identifier Mask bits
These bits hold the mask bits to be applied to bits[10:3] of the Standard Identifier portion of a received message.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

REGISTER 4-23: RXMxSIDL: MASK x STANDARD IDENTIFIER LOW REGISTER (ADDRESS: 21h, 25h)(1)

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
SID2	SID1	SID0	—	—	—	EID17	EID16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **SID[2:0]:** Standard Identifier Mask bits
These bits hold the mask bits to be applied to bits[2:0] of the Standard Identifier portion of a received message.

bit 4-2 **Unimplemented:** Read as '0'

bit 1-0 **EID[17:16]:** Extended Identifier Mask bits
These bits hold the mask bits to be applied to bits[17:16] of the Extended Identifier portion of a received message.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

REGISTER 4-24: RXMxEID8: MASK x EXTENDED IDENTIFIER HIGH REGISTER (ADDRESS: 22h, 26h)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EID[15:8]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[15:8]:** Extended Identifier bits

These bits hold the filter bits to be applied to bits[15:8] of the Extended Identifier portion of a received message. If corresponding with RXM[1:0] = 00 and EXIDE = 0, these bits are applied to Byte 0 in received data.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

REGISTER 4-25: RXMxEID0: MASK x EXTENDED IDENTIFIER LOW REGISTER (ADDRESS: 23h, 27h)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EID[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **EID[7:0]:** Extended Identifier Mask bits

These bits hold the filter bits to be applied to bits[7:0] of the Extended Identifier portion of a received message. If corresponding with RXM[1:0] = 00 and EXIDE = 0, these bits are applied to Byte 1 in received data.

Note 1: The Mask and Filter registers read all '0's when in any mode, except Configuration mode.

4.4 Bit Time Configuration Registers

REGISTER 4-26: CNF1: CONFIGURATION 1 REGISTER (ADDRESS: 2Ah)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **SJW[1:0]:** Synchronization Jump Width Length bits
11 = Length = 4 x T_Q
10 = Length = 3 x T_Q
01 = Length = 2 x T_Q
00 = Length = 1 x T_Q

bit 5-0 **BRP[5:0]:** Baud Rate Prescaler bits
$T_Q = 2 \times (BRP[5:0] + 1)/F_{OSC}$

REGISTER 4-27: CNF2: CONFIGURATION 2 REGISTER (ADDRESS: 29h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTLMODE	SAM	PHSEG1[2:0]			PRSEG2	PRSEG1	PRSEG0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **BTLMODE:** PS2 Bit Time Length bit
1 = Length of PS2 is determined by the PHSEG2[2:0] bits of CNF3
0 = Length of PS2 is the greater of PS1 and IPT (2 T_Q)

bit 6 **SAM**: Sample Point Configuration bit
1 = Bus line is sampled three times at the sample point
0 = Bus line is sampled once at the sample point

bit 5-3 **PHSEG1[2:0]:** PS1 Length bits
(PHSEG1[2:0] + 1) x T_Q

bit 2-0 **PRSEG[2:0]:** Propagation Segment Length bits
(PRSEG[2:0] + 1) x T_Q

REGISTER 4-28: CNF3: CONFIGURATION 3 REGISTER (ADDRESS: 28h)

R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
SOF	WAKFIL	—	—	—	PHSEG2[2:0]		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **SOF:** Start-of-Frame Signal bit
If CLKEN (CANCTRL[2]) = 1:
1 = CLKOUT pin is enabled for SOF signal
0 = CLKOUT pin is enabled for clock out function
If CLKEN (CANCTRL[2]) = 0:
Bit is don't care.

bit 6 **WAKFIL:** Wake-up Filter bit
1 = Wake-up filter is enabled
0 = Wake-up filter is disabled

bit 5-3 **Unimplemented:** Read as '0'

bit 2-0 **PHSEG2[2:0]:** PS2 Length bits
(PHSEG2[2:0] + 1) x T_Q
Minimum valid setting for PS2 is 2 T_Q.

4.5 Error Detection Registers

REGISTER 4-29: TEC: TRANSMIT ERROR COUNTER REGISTER (ADDRESS: 1Ch)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TEC[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **TEC[7:0]:** Transmit Error Count bits

REGISTER 4-30: REC: RECEIVER ERROR COUNTER REGISTER (ADDRESS: 1Dh)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
REC[7:0]							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **REC[7:0]:** Receive Error Count bits

REGISTER 4-31: EFLG: ERROR FLAG REGISTER (ADDRESS: 2Dh)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
RX1OVR	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **RX1OVR:** Receive Buffer 1 Overflow Flag bit
- Sets when a valid message is received for RXB1 and the RX1IF bit in the CANINTF register is '1'
- Must be reset by MCU

bit 6 **RX0OVR:** Receive Buffer 0 Overflow Flag bit
- Sets when a valid message is received for RXB0 and the RX0IF bit in the CANINTF register is '1'
- Must be reset by MCU

bit 5 **TXBO:** Bus-Off Error Flag bit
- Bit sets when TEC reaches 255
- Resets after a successful bus recovery sequence

bit 4 **TXEP:** Transmit Error-Passive Flag bit
- Sets when TEC is equal to or greater than 128
- Resets when TEC is less than 128

bit 3 **RXEP:** Receive Error-Passive Flag bit
- Sets when REC is equal to or greater than 128
- Resets when REC is less than 128

bit 2 **TXWAR:** Transmit Error Warning Flag bit
- Sets when TEC is equal to or greater than 96
- Resets when TEC is less than 96

bit 1 **RXWAR:** Receive Error Warning Flag bit
- Sets when REC is equal to or greater than 96
- Resets when REC is less than 96

bit 0 **EWARN:** Error Warning Flag bit
- Sets when TEC or REC is equal to or greater than 96 (TXWAR or RXWAR = 1)
- Resets when both REC and TEC are less than 96

4.6 Interrupt Registers

REGISTER 4-32: CANINTE: CAN INTERRUPT ENABLE REGISTER (ADDRESS: 2Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **MERRE:** Message Error Interrupt Enable bit
1 = Interrupt on error during message reception or transmission
0 = Disabled

bit 6 **WAKIE:** Wake-up Interrupt Enable bit
1 = Interrupt on CAN bus activity
0 = Disabled

bit 5 **ERRIE:** Error Interrupt Enable bit (multiple sources in the EFLG register)
1 = Interrupt on EFLG error condition change
0 = Disabled

bit 4 **TX2IE:** Transmit Buffer 2 Empty Interrupt Enable bit
1 = Interrupt on TXB2 becoming empty
0 = Disabled

bit 3 **TX1IE:** Transmit Buffer 1 Empty Interrupt Enable bit
1 = Interrupt on TXB1 becoming empty
0 = Disabled

bit 2 **TX0IE:** Transmit Buffer 0 Empty Interrupt Enable bit
1 = Interrupt on TXB0 becoming empty
0 = Disabled

bit 1 **RX1IE:** Receive Buffer 1 Full Interrupt Enable bit
1 = Interrupt when message is received in RXB1
0 = Disabled

bit 0 **RX0IE:** Receive Buffer 0 Full Interrupt Enable bit
1 = Interrupt when message is received in RXB0
0 = Disabled

REGISTER 4-33: CANINTF: CAN INTERRUPT FLAG REGISTER (ADDRESS: 2Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **MERRF:** Message Error Interrupt Flag bit
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

bit 6 **WAKIF:** Wake-up Interrupt Flag bit
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

bit 5 **ERRIF:** Error Interrupt Flag bit (multiple sources in the EFLG register)
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

bit 4 **TX2IF:** Transmit Buffer 2 Empty Interrupt Flag bit
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

bit 3 **TX1IF:** Transmit Buffer 1 Empty Interrupt Flag bit
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

bit 2 **TX0IF:** Transmit Buffer 0 Empty Interrupt Flag bit
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

bit 1 **RX1IF:** Receive Buffer 1 Full Interrupt Flag bit
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

bit 0 **RX0IF:** Receive Buffer 0 Full Interrupt Flag bit
1 = Interrupt pending (must be cleared by MCU to reset interrupt condition)
0 = No interrupt pending

4.7 CAN Control Register

REGISTER 4-34: CANCTRL: CAN CONTROL REGISTER (ADDRESS: XFh)

R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
REQOP2	REQOP1	REQOP0	ABAT	OSM	CLKEN	CLKPRE1	CLKPRE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **REQOP[2:0]:** Request Operation mode bits
- 000 = Sets Normal Operation mode
- 001 = Sets Sleep mode
- 010 = Sets Loopback mode
- 011 = Sets Listen-Only mode
- 100 = Sets Configuration mode

All other values for the REQOPx bits are invalid and should not be used. On power-up, REQOP = b'100.

bit 4 **ABAT:** Abort All Pending Transmissions bit
- 1 = Requests abort of all pending transmit buffers
- 0 = Terminates request to abort all transmissions

bit 3 **OSM:** One-Shot Mode bit
- 1 = Enabled: Message will only attempt to transmit one time
- 0 = Disabled: Messages will reattempt transmission if required

bit 2 **CLKEN:** CLKOUT Pin Enable bit
- 1 = CLKOUT pin is enabled
- 0 = CLKOUT pin is disabled (pin is in a high-impedance state)

bit 1-0 **CLKPRE[1:0]:** CLKOUT Pin Prescaler bits
- 00 = F_{CLKOUT} = System Clock/1
- 01 = F_{CLKOUT} = System Clock/2
- 10 = F_{CLKOUT} = System Clock/4
- 11 = F_{CLKOUT} = System Clock/8

REGISTER 4-35: CANSTAT: CAN STATUS REGISTER (ADDRESS: XEh)

R-1	R-0	R-0	U-0	R-0	R-0	R-0	U-0
OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **OPMOD[2:0]:** Operation Mode bits
000 = Device is in the Normal Operation mode
001 = Device is in Sleep mode
010 = Device is in Loopback mode
011 = Device is in Listen-Only mode
100 = Device is in Configuration mode

bit 4 **Unimplemented:** Read as '0'

bit 3-1 **ICOD[2:0]:** Interrupt Flag Code bits
000 = No Interrupt
001 = Error interrupt
010 = Wake-up interrupt
011 = TXB0 interrupt
100 = TXB1 interrupt
101 = TXB2 interrupt
110 = RXB0 interrupt
111 = RXB1 interrupt

bit 0 **Unimplemented:** Read as '0'

5.0 SPI INTERFACE

The MCP25625 is designed to interface directly with the Serial Peripheral Interface (SPI) port available on many microcontrollers and supports Mode 0,0 and Mode 1,1.

Commands and data are sent to the device via the SI pin, with data being clocked in on the rising edge of SCK. Data is driven out by the MCP25625 (on the SO line) on the falling edge of SCK. The $\overline{CS}$ pin must be held low while any operation is performed.

Table 5-1 shows the instruction bytes for all operations. Refer to Figures 5-10 and 5-11 for detailed input and output timing diagrams for both Mode 0,0 and Mode 1,1 operation.

Note 1: The MCP25625 expects the first byte, after lowering $\overline{CS}$, to be the instruction/command byte. This implies that $\overline{CS}$ must be raised and then lowered again to invoke another command.

TABLE 5-1: SPI INSTRUCTION SET

Instruction Name	Instruction Format	Description
RESET	1100 0000	Resets the internal registers to the default state, sets Configuration mode.
READ	0000 0011	Reads data from the register beginning at the selected address.
READ RX BUFFER	1001 0nm0	When reading a receive buffer, reduces the overhead of a normal READ command by placing the Address Pointer at one of four locations, as indicated by 'nm'.(1)
WRITE	0000 0010	Writes data to the register beginning at the selected address.
LOAD TX BUFFER	0100 0abc	When loading a transmit buffer, reduces the overhead of a normal WRITE command by placing the Address Pointer at one of six locations, as indicated by 'abc'.
RTS (Message Request-to-Send)	1000 0nnn	Instructs the controller to begin the message transmission sequence for any of the transmit buffers. 1000 0nnn Request-to-Send for TXB2 — Request-to-Send for TXBO Request-to-Send for TXB1
READ STATUS	1010 0000	Quick polling command that reads several Status bits for transmit and receive functions.
RX STATUS	1011 0000	Quick polling command that indicates a filter match and message type (standard, extended and/or remote) of the received message.
BIT MODIFY	0000 0101	Allows the user to set or clear individual bits in a particular register.(2)

Note 1: The associated RX flag bit (RXxIF bits in the CANINTF register) will be cleared after bringing $\overline{CS}$ high.

2: Not all registers can be bit modified with this command. Executing this command on registers that are not bit modifiable will force the mask to FFh. See the register map in **Section 4.0 "Register Map"** for a list of the registers that apply.

5.1 RESET Instruction

The RESET instruction can be used to re-initialize the internal registers of the MCP25625 and set Configuration mode. This command provides the same functionality, via the SPI interface, as the RESET pin.

The RESET instruction is a single byte instruction that requires selecting the device by pulling CS low, sending the instruction byte and then raising CS. It is highly recommended that the RESET command be sent (or the RESET pin be lowered) as part of the power-on initialization sequence.

5.2 READ Instruction

The READ instruction is started by lowering the CS pin. The READ instruction is then sent to the MCP25625, followed by the 8-bit address (A7 through A0). Next, the data stored in the register at the selected address will be shifted out on the SO pin.

The internal Address Pointer is automatically incremented to the next address once each byte of data is shifted out. Therefore, it is possible to read the next consecutive register address by continuing to provide clock pulses. Any number of consecutive register locations can be read sequentially using this method. The READ operation is terminated by raising the CS pin (Figure 5-2).

5.3 READ RX BUFFER Instruction

The READ RX BUFFER instruction (Figure 5-3) provides a means to quickly address a receive buffer for reading. This instruction reduces the SPI overhead by one byte: the address byte. The command byte actually has four possible values that determine the Address Pointer location. Once the command byte is sent, the controller clocks out the data at the address location the same as the READ instruction (i.e., sequential reads are possible). This instruction further reduces the SPI overhead by automatically clearing the associated receive flag (RXxIF bit in the CANINTF register) when CS is raised at the end of the command.

5.4 WRITE Instruction

The WRITE instruction is started by lowering the CS pin. The WRITE instruction is then sent to the MCP25625, followed by the address and at least one byte of data.

It is possible to write to sequential registers by continuing to clock in data bytes, as long as CS is held low. Data will actually be written to the register on the rising edge of the SCK line for the D0 bit. If the CS line is brought high before eight bits are loaded, the write will be aborted for that data byte and previous bytes in the command will have been written. Refer to the timing diagram in Figure 5-4 for a more detailed illustration of the byte WRITE sequence.

5.5 LOAD TX BUFFER Instruction

The LOAD TX BUFFER instruction (Figure 5-5) eliminates the 8-bit address required by a normal WRITE command. The 8-bit instruction sets the Address Pointer to one of six addresses to quickly write to a transmit buffer that points to the “ID” or “data” address of any of the three transmit buffers.

5.6 Request-to-Send (RTS) Instruction

The RTS command can be used to initiate message transmission for one or more of the transmit buffers.

The MCP25625 is selected by lowering the CS pin. The RTS command byte is then sent. Shown in Figure 5-6, the last three bits of this command indicate which transmit buffer(s) are enabled to send.

This command will set the TXREQ bit in the TXBxCTRL register for the respective buffer(s). Any or all of the last three bits can be set in a single command. If the RTS command is sent with nnn = 000, the command will be ignored.

5.7 READ STATUS Instruction

The READ STATUS instruction allows single instruction access to some of the often used Status bits for message reception and transmission.

The MCP25625 is selected by lowering the CS pin and the READ STATUS command byte, shown in Figure 5-8, is sent to the MCP25625. Once the command byte is sent, the MCP25625 will return eight bits of data that contain the status.

If additional clocks are sent after the first eight bits are transmitted, the MCP25625 will continue to output the Status bits as long as the CS pin is held low and clocks are provided on SCK.

Each Status bit returned in this command may also be read by using the standard READ command with the appropriate register address.

5.8 RX STATUS Instruction

The RX STATUS instruction (Figure 5-9) is used to quickly determine which filter matched the message and message type (standard, extended, remote). After the command byte is sent, the controller will return eight bits of data that contain the status data. If more clocks are sent after the eight bits are transmitted, the controller will continue to output the same Status bits as long as the CS pin stays low and clocks are provided.

5.9 BIT MODIFY Instruction

The BIT MODIFY instruction provides a means for setting or clearing individual bits in specific status and control registers. This command is not available for all registers. See **Section 4.0 "Register Map"** to determine which registers allow the use of this command.

Note:	Executing the BIT MODIFY command on registers that are not bit-modifiable will force the mask to FFh. This will allow byte writes to the registers, not BIT MODIFY.

The part is selected by lowering the $\overline{CS}$ pin and the BIT MODIFY command byte is then sent to the MCP25625. The command is followed by the address of the register, the mask byte, and finally, the data byte.

The mask byte determines which bits in the register will be allowed to change. A '1' in the mask byte will allow a bit in the register to change, while a '0' will not (see Figure 5-1).

The data byte determines what value the modified bits in the register will be changed to. A '1' in the data byte will set the bit and a '0' will clear the bit, provided that the mask for that bit is set to a '1' (see Figure 5-7).

FIGURE 5-1: BIT MODIFY

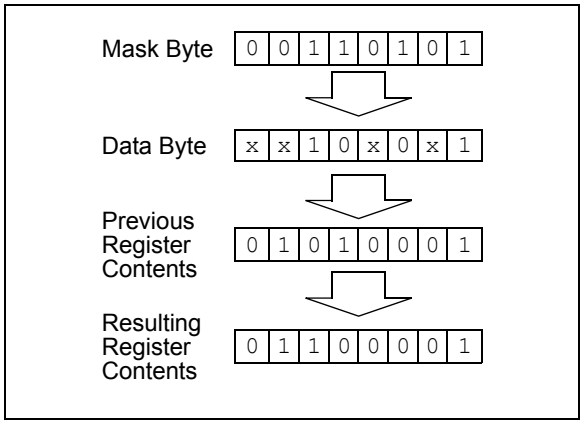


FIGURE 5-2: READ INSTRUCTION

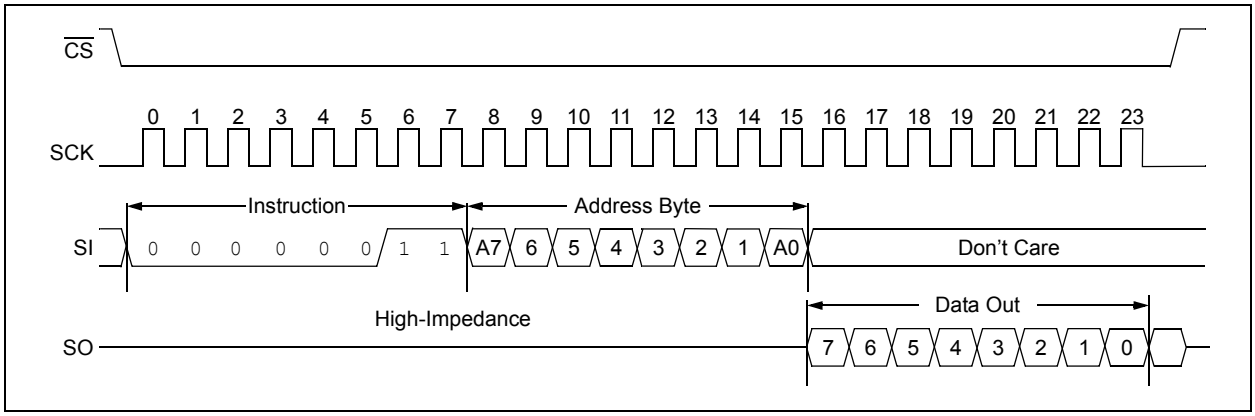
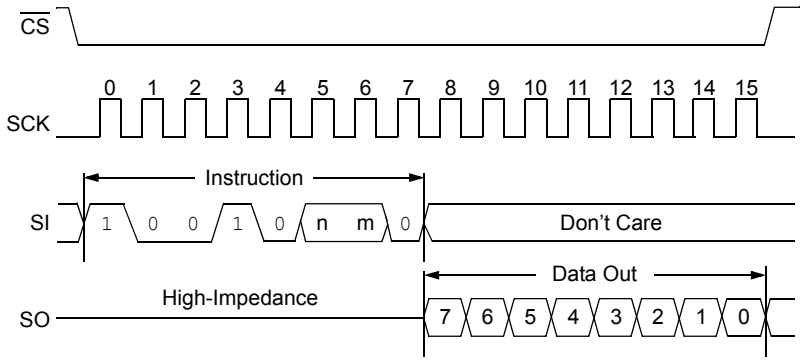


FIGURE 5-3: READ RX BUFFER INSTRUCTION



n	m	Address Points to	Address
0	0	Receive Buffer 0, Start at RXB0SIDH	0x61
0	1	Receive Buffer 0, Start at RXB0D0	0x66
1	0	Receive Buffer 1, Start at RXB1SIDH	0x71
1	1	Receive Buffer 1, Start at RXB1D0	0x76

FIGURE 5-4: BYTE WRITE INSTRUCTION

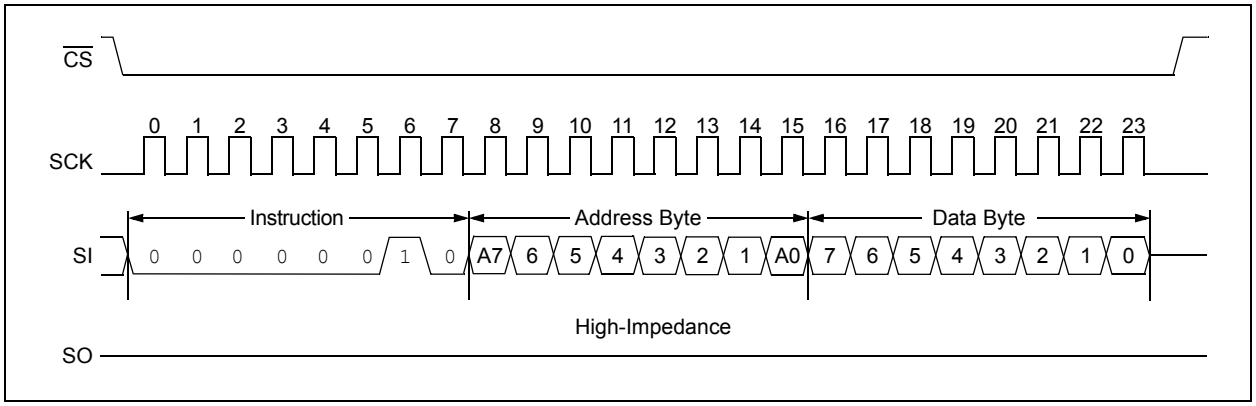
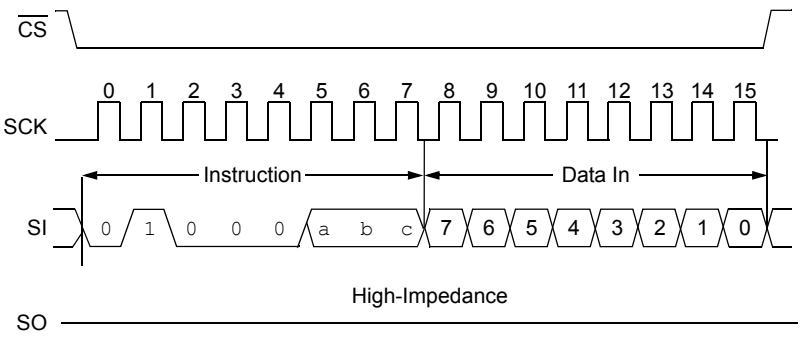


FIGURE 5-5: LOAD TX BUFFER INSTRUCTION



a	b	c	Address Points to	Addr
0	0	0	TX Buffer 0, Start at TXB0SIDH	0x31
0	0	1	TX Buffer 0, Start at TXB0D0	0x36
0	1	0	TX Buffer 1, Start at TXB1SIDH	0x41
0	1	1	TX Buffer 1, Start at TXB1D0	0x46
1	0	0	TX Buffer 2, Start at TXB2SIDH	0x51
1	0	1	TX Buffer 2, Start at TXB2D0	0x56

FIGURE 5-6: REQUEST-TO-SEND (RTS) INSTRUCTION

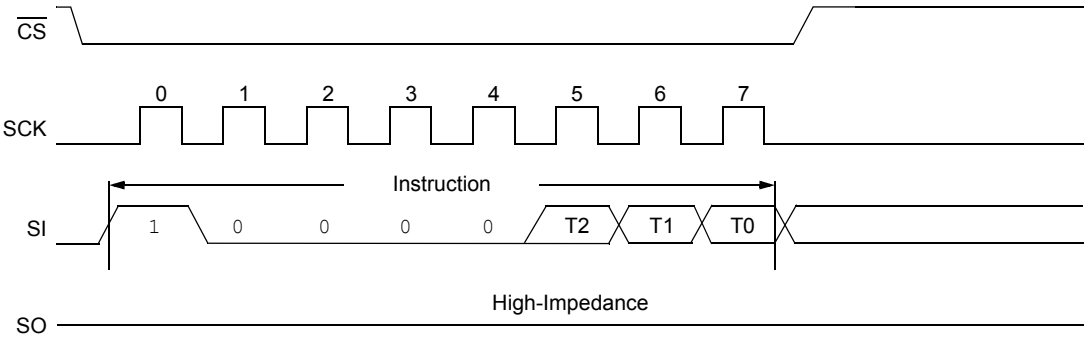
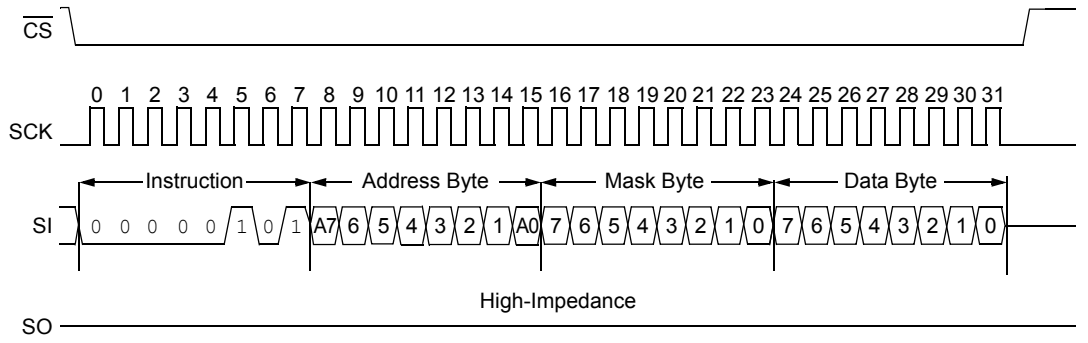


FIGURE 5-7: BIT MODIFY INSTRUCTION



Note: Not all registers can be accessed with this command. See the register map for a list of the registers that apply.

FIGURE 5-8: READ STATUS INSTRUCTION

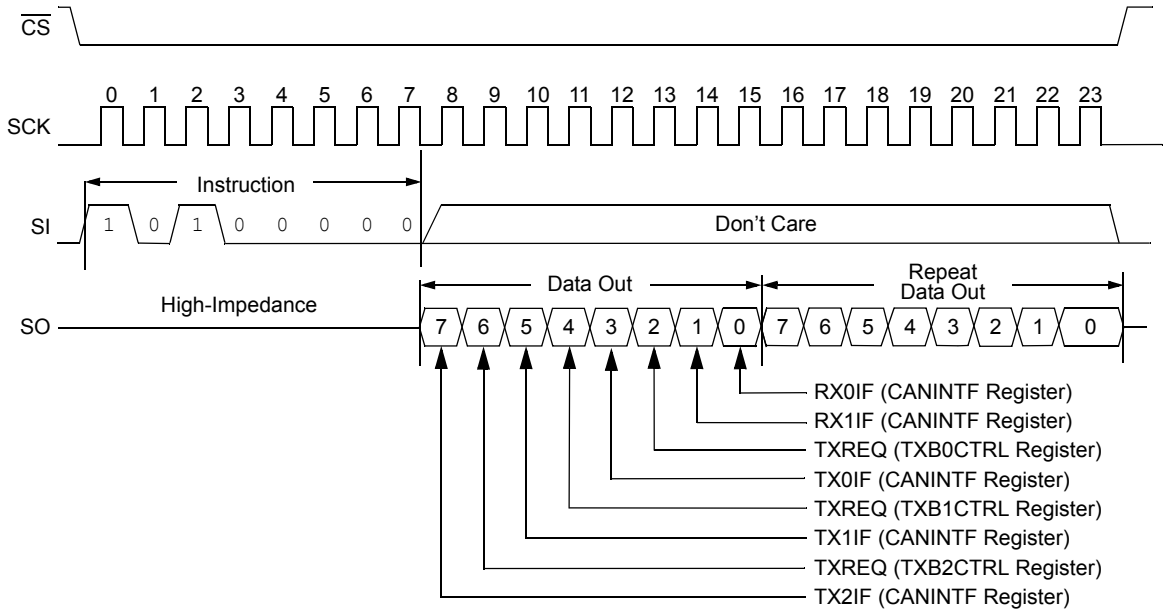
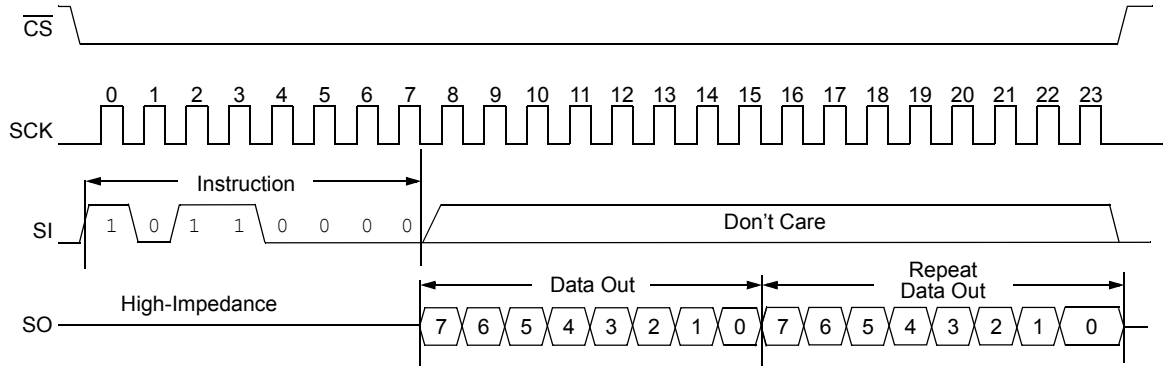


FIGURE 5-9: RX STATUS INSTRUCTION



7	6	Received Message
0	0	No RX Message
0	1	Message in RXB0
1	0	Message in RXB1
1	1	Messages in Both Buffers*

RXxIF (CANINTF) bits are mapped to bits 7 and 6.

4	3	Msg Type Received
0	0	Standard Data Frame
0	1	Standard Remote Frame
1	0	Extended Data Frame
1	1	Extended Remote Frame

The extended ID bit is mapped to bit 4. The RTR bit is mapped to bit 3.

2	1	0	Filter Match
0	0	0	RXF0
0	0	1	RXF1
0	1	0	RXF2
0	1	1	RXF3
1	0	0	RXF4
1	0	1	RXF5
1	1	0	RXF0 (rollover to RXB1)
1	1	1	RXF1 (rollover to RXB1)

* Buffer 0 has higher priority; therefore, RXB0 status is reflected in bits[4:0].

FIGURE 5-10: SPI INPUT TIMING

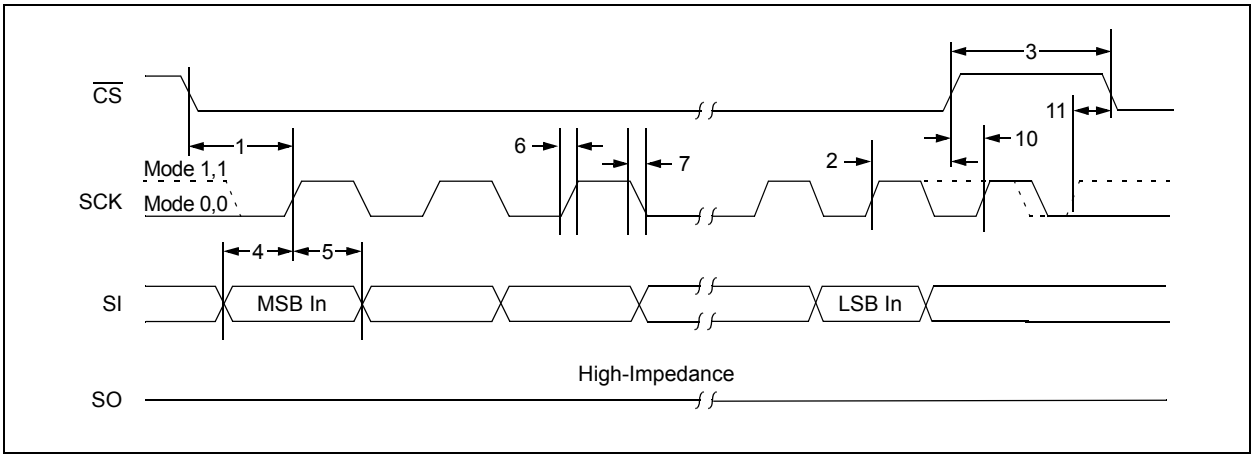
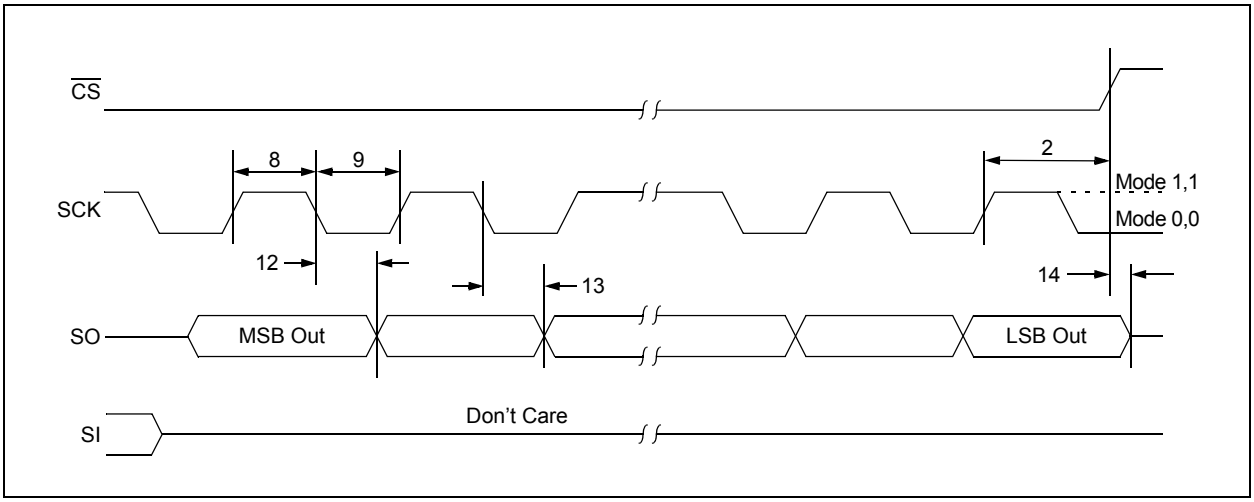


FIGURE 5-11: SPI OUTPUT TIMING



NOTES:

6.0 CAN TRANSCEIVER

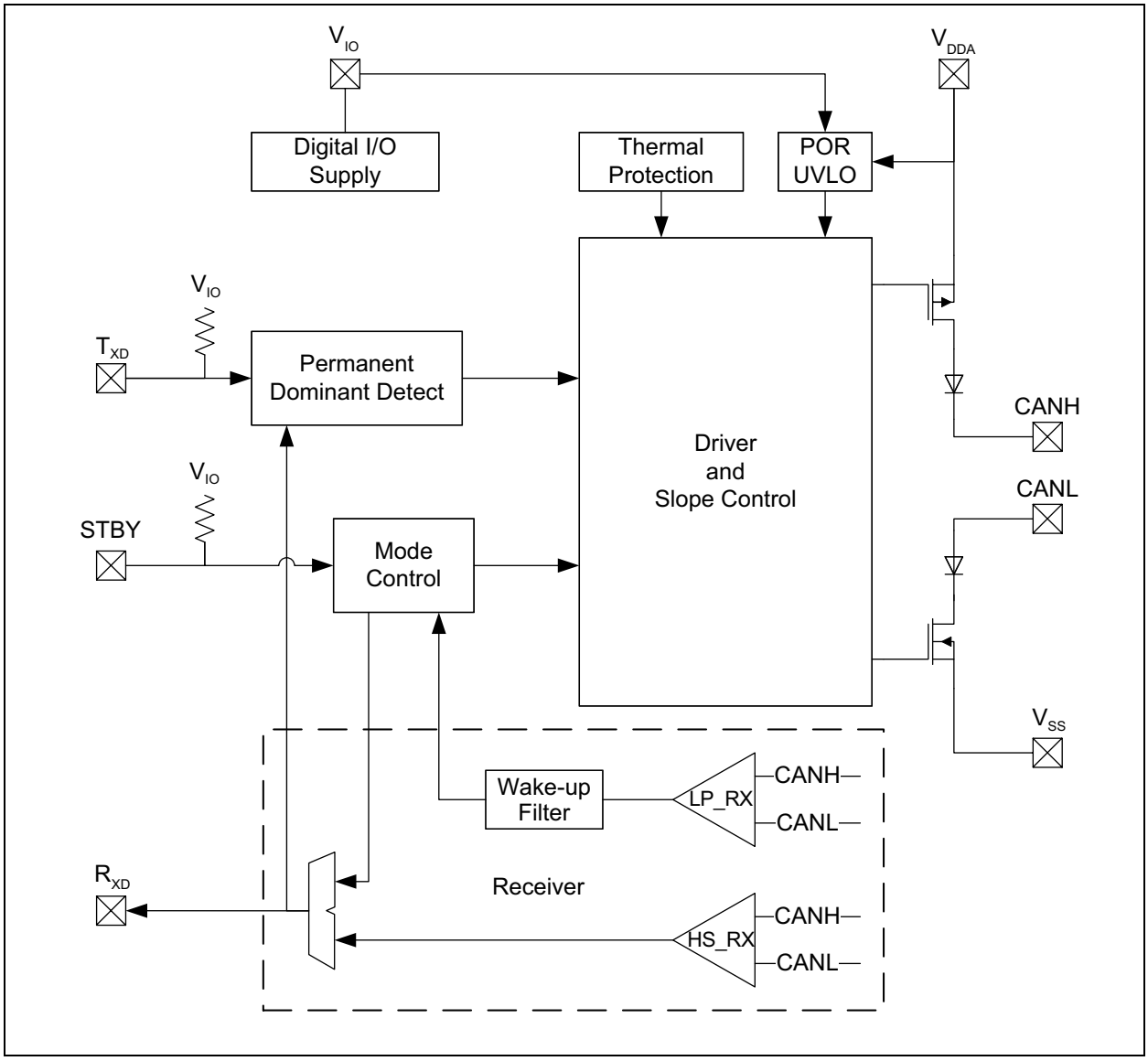
The CAN transceiver is a differential, high-speed, Fault tolerant interface to the CAN physical bus. It is fully compatible with the ISO-11898-2 and ISO-11898-5 standards. It operates at speeds of up to 1 Mb/s.

The CAN transceiver converts the digital TxCAN signal generated by the CAN controller to signals suitable for transmission over the physical CAN bus (differential output). It also translates the differential CAN bus voltage to the RxCAN input signal of the CAN controller.

The CAN transceiver meets the stringent automotive EMC and ESD requirements.

Figure 6-1 illustrates the block diagram of the CAN transceiver.

FIGURE 6-1: CAN TRANSCEIVER BLOCK DIAGRAM



6.1 Transmitter Function

The CAN bus has two states: Dominant and Recessive. A Dominant state occurs when the differential voltage between CANH and CANL is greater than $V_{DIFF(D)(I)}$. A Recessive state occurs when the differential voltage is less than $V_{DIFF(R)(I)}$. The Dominant and Recessive states correspond to the low and high state of the T_{XD} input pin, respectively. However, a Dominant state initiated by another CAN node will override a Recessive state on the CAN bus.

6.2 Receiver Function

In Normal mode, the R_{XD} output pin reflects the differential bus voltage between CANH and CANL. The low and high states of the R_{XD} output pin correspond to the Dominant and Recessive states of the CAN bus, respectively.

6.3 Internal Protection

CANH and CANL are protected against battery short circuits and electrical transients that can occur on the CAN bus. This feature prevents destruction of the transmitter output stage during such a Fault condition.

The device is further protected from excessive current loading by thermal shutdown circuitry that disables the output drivers when the junction temperature exceeds a nominal limit of +175°C. All other parts of the chip remain operational and the chip temperature is lowered due to the decreased power dissipation in the transmitter outputs. This protection is essential to protect against bus line short-circuit induced damage.

6.4 Permanent Dominant Detection

The CAN transceiver device prevents two conditions:

- Permanent Dominant condition on T_{XD}
- Permanent Dominant condition on the bus

In Normal mode, if the CAN transceiver detects an extended low state on the T_{XD} input, it will disable the CANH and CANL output drivers in order to prevent the corruption of data on the CAN bus. The drivers will remain disabled until T_{XD} goes high.

In Standby mode, if the CAN transceiver detects an extended Dominant condition on the bus, it will set the R_{XD} pin to the Recessive state. This allows the attached controller to go to Low-Power mode until the Dominant issue is corrected. R_{XD} is latched high until a Recessive state is detected on the bus and the wake-up function is enabled again.

Both conditions have a time-out of 1.25 ms (typical). This implies a maximum bit time of 69.44 µs (14.4 kHz), allowing up to 18 consecutive Dominant bits on the bus.

6.5 Power-on Reset (POR) and Undervoltage Detection

The MCP25625 has undervoltage detection on both supply pins: V_{DDA} and V_{IO}. Typical undervoltage thresholds are 1.2V for V_{IO} and 4V for V_{DDA}.

When the device is powered on, CANH and CANL remain in a high-impedance state until both V_{DDA} and V_{IO} exceed their undervoltage levels. In addition, CANH and CANL will remain in a high-impedance state if T_{XD} is low when both undervoltage thresholds are reached. CANH and CANL will become active only after T_{XD} is asserted high. Once powered on, CANH and CANL will enter a high-impedance state if the voltage level at V_{DDA} drops below the undervoltage level, providing voltage brown-out protection during normal operation.

In Normal mode, the receiver output is forced to a Recessive state during an undervoltage condition on V_{DDA}. In Standby mode, the low-power receiver is only enabled when both V_{DDA} and V_{IO} supply voltages rise above their respective undervoltage thresholds. Once these threshold voltages are reached, the low-power receiver is no longer controlled by the POR comparator and remains operational down to about 2.5V on the V_{DDA} supply. The CAN transceiver transfers data to the R_{XD} pin down to 1V on the V_{IO} supply.

6.6 Pin Description

6.6.1 TRANSMITTER DATA INPUT PIN (T_{XD})

The CAN transceiver drives the differential output pins, CANH and CANL, according to T_{XD}. The transceiver is connected to the TxCAN pin of the CAN controller. When T_{XD} is low, CANH and CANL are in the Dominant state. When T_{XD} is high, CANH and CANL are in the Recessive state, provided that another CAN node is not driving the CAN bus with a Dominant state. T_{XD} is connected to an internal pull-up resistor (nominal 33 kΩ) to V_{IO}.

6.6.2 GROUND SUPPLY PIN (V_{SS})

Ground supply pin.

6.6.3 SUPPLY VOLTAGE PIN (V_{DDA})

Positive supply voltage pin. Supplies transmitter and receiver, including the wake-up receiver.

6.6.4 RECEIVER DATA OUTPUT PIN (R_{XD})

R_{XD} is a CMOS compatible output that drives high or low depending on the differential signals on the CANH and CANL pins, and is usually connected to the receiver data input of the CAN controller device. R_{XD} is high when the CAN bus is Recessive and low in the Dominant state. R_{XD} is supplied by V_{IO}.

6.6.5 V_{IO} PIN

Supply for digital I/O pins of the CAN transceiver.

6.6.6 CAN LOW PIN (CANL)

The CANL output drives the low side of the CAN differential bus. This pin is also tied internally to the receive input comparator. CANL disconnects from the bus when V_{DDA} is not powered.

6.6.7 CAN HIGH PIN (CANH)

The CANH output drives the high side of the CAN differential bus. This pin is also tied internally to the receive input comparator. CANH disconnects from the bus when V_{DDA} is not powered.

6.6.8 STANDBY MODE INPUT PIN (STBY)

This pin selects between Normal or Standby mode of the CAN transceiver. In Standby mode, the transmitter and the high-speed receiver are turned off, only the low-power receiver and wake-up filter are active. STBY is connected to an internal MOS pull-up resistor to V_{IO}. The value of the MOS pull-up resistor depends on the supply voltage. Typical values are 660 kΩ for 5V, 1.1 MΩ for 3.3V and 4.4 MΩ for 1.8V

6.6.9 EXPOSED THERMAL PAD (EP)

It is recommended to connect this pad to V_{SS} to enhance electromagnetic immunity and thermal resistance.

NOTES:

7.0 ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Ratings†

V_{DD} 7.0V
V_{DDA} 7.0V
V_{IO} 7.0V
DC Voltage at CANH, CANL ..... -58V to +58V
DC Voltage at T_{XD}, R_{XD}, STBY w.r.t V_{SS} ..... -0.3V to V_{IO} + 0.3V
DC Voltage at All Other I/Os w.r.t GND ..... -0.3V to V_{DD} + 0.3V
Transient Voltage on CANH, CANL (ISO-7637) (Figure 7-5) ..... -150V to +100V
Storage temperature ..... -55°C to +150°C
Operating Ambient Temperature ..... -40°C to +125°C
Virtual Junction Temperature, T_{VJ} (IEC60747-1) ..... -40°C to +150°C
Soldering Temperature of Leads (10 seconds) ..... +300°C
ESD Protection on CANH and CANL Pins (IEC 61000-4-2) ..... ±8 kV
ESD Protection on CANH and CANL Pins (IEC 801; Human Body Model) ..... ±8 kV
ESD Protection on All Other Pins (IEC 801; Human Body Model) ..... ±4 kV
ESD Protection on All Pins (IEC 801; Machine Model) ..... ±300V
ESD Protection on All Pins (IEC 801; Charge Device Model) ..... ±750V

† **NOTICE:** Stresses above those listed under “Maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

7.2 CAN Controller Characteristics

TABLE 7-1: DC CHARACTERISTICS

Electrical Characteristics:		Extended (E): T_{AMB} = -40°C to +125°C; V_{DD} = 2.7V to 5.5V			
Sym.	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
V_{DD}	Supply Voltage	2.7	5.5	V	
V_{RET}	Register Retention Voltage	2.4	—	V	
High-Level Input Voltage					
V_{IH}	RxCAN	2	V_{DD} + 1	V	
	SCK, $\overline{CS}$, SI, $\overline{TxnRTS}$ Pins	0.7 V_{DD}	V_{DD} + 1	V	
	OSC1	0.85 V_{DD}	V_{DD}	V	
	$\overline{RESET}$	0.85 V_{DD}	V_{DD}	V	
Low-Level Input Voltage					
V_{IL}	RxCAN, $\overline{TxnRTS}$ Pins	-0.3	0.15 V_{DD}	V	
	SCK, $\overline{CS}$, SI	-0.3	0.4 V_{DD}	V	
	OSC1	V_{SS}	0.3 V_{DD}	V	
	$\overline{RESET}$	V_{SS}	0.15 V_{DD}	V	
Low-Level Output Voltage					
V_{OL}	TxCAN	—	0.6	V	I_{OL} = +6.0 mA, V_{DD} = 4.5V
	$\overline{RxnBF}$ Pins	—	0.6	V	I_{OL} = +8.5 mA, V_{DD} = 4.5V
	SO, CLKOUT	—	0.6	V	I_{OL} = +2.1 mA, V_{DD} = 4.5V
	$\overline{INT}$	—	0.6	V	I_{OL} = +1.6 mA, V_{DD} = 4.5V
High-Level Output Voltage					
V_{OH}	TxCAN, $\overline{RxnBF}$ Pins	V_{DD} – 0.7	—	V	I_{OH} = -3.0 mA, V_{DD} = 4.5V
	SO, CLKOUT	V_{DD} – 0.5	—	V	I_{OH} = -400 µA, V_{DD} = 4.5V
	$\overline{INT}$	V_{DD} – 0.7	—	V	I_{OH} = -1.0 mA, V_{DD} = 4.5V
Input Leakage Current					
I_{LI}	All I/Os except OSC1 and $\overline{TxnRTS}$ Pins	-1	+1	µA	$\overline{CS}$ = $\overline{RESET}$ = V_{DD}, V_{IN} = V_{SS} to V_{DD}
	OSC1 Pin	-5	+5	µA	
C_{INT}	Internal Capacitance (all inputs and outputs)	—	7	pF	T_{AMB} = +25°C, f_C = 1.0 MHz, V_{DD} = 0V **(Note 1)**
I_{DD}	Operating Current	—	10	mA	V_{DD} = 5.5V, F_{OSC} = 25 MHz, F_{CLK} = 1 MHz, SO = Open
I_{DDS}	Standby Current (Sleep mode)	—	8	µA	$\overline{CS}$, $\overline{TxnRTS}$ = V_{DD}, Inputs tied to V_{DD} or V_{SS}, -40°C to +125°C

Note 1: Characterized, not 100% tested.

TABLE 7-2: OSCILLATOR TIMING CHARACTERISTICS

Oscillator Timing Characteristics[1]		Extended (E): T_{AMB} = -40°C to +125°C; V_{DD} = 2.7V to 5.5V			
Sym.	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
F_{OSC}	Clock In Frequency	1	25	MHz	
T_{OSC}	Clock In Period	40	1000	ns	
t_{DUTY}	Duty Cycle (External Clock input)	0.45	0.55	—	$T_{OSH}/(T_{OSH} + T_{OSL})$

Note 1: Characterized, not 100% tested.

TABLE 7-3: CAN INTERFACE AC CHARACTERISTICS

CAN Interface AC Characteristics		Extended (E): T_{AMB} = -40°C to +125°C; V_{DD} = 2.7V to 5.5V			
Sym.	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
t_{WF}	Wake-up Noise Filter	100	—	ns	

TABLE 7-4: $\overline{\text{RESET}}$ AC CHARACTERISTICS

$\overline{\text{RESET}}$ AC Characteristics		Extended (E): T_{AMB} = -40°C to +125°C; V_{DD} = 2.7V to 5.5V			
Sym.	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
t_{RL}	$\overline{\text{RESET}}$ Pin Low Time	2	—	µs	

TABLE 7-5: CLKOUT PIN AC CHARACTERISTICS

CLKOUT Pin AC/DC Characteristics			Extended (E): T_{AMB} = -40°C to +125°C; V_{DD} = 2.7V to 5.5V			
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
	$t_{HCLKOUT}$	CLKOUT Pin High Time	10	—	ns	T_{OSC} = 40 ns **(Note 1)**
	$t_{LCLKOUT}$	CLKOUT Pin Low Time	10	—	ns	T_{OSC} = 40 ns **(Note 1)**
	$t_{RCLKOUT}$	CLKOUT Pin Rise Time	—	10	ns	Measured from 0.3 V_{DD} to 0.7 V_{DD} **(Note 1)**
	$t_{FCLKOUT}$	CLKOUT Pin Fall Time	—	10	ns	Measured from 0.7 V_{DD} to 0.3 V_{DD} **(Note 1)**
	$t_{DCLKOUT}$	CLKOUT Propagation Delay	—	100	ns	**(Note 1)**
15	t_{HSOF}	Start-of-Frame High Time	—	2 T_{OSC}	ns	**(Note 1)**
16	t_{DSOF}	Start-of-Frame Propagation Delay	—	2 T_{OSC} + 0.5 T_Q	ns	Measured from CAN bit sample point, device is a receiver, BRP[5:0] = 0 in the CNF1 register **(Note 2)**

Note 1: All CLKOUT mode functionality and output frequency are tested at device frequency limits; however, the CLKOUT prescaler is set to divide-by-one. Characterized, not 100% tested.

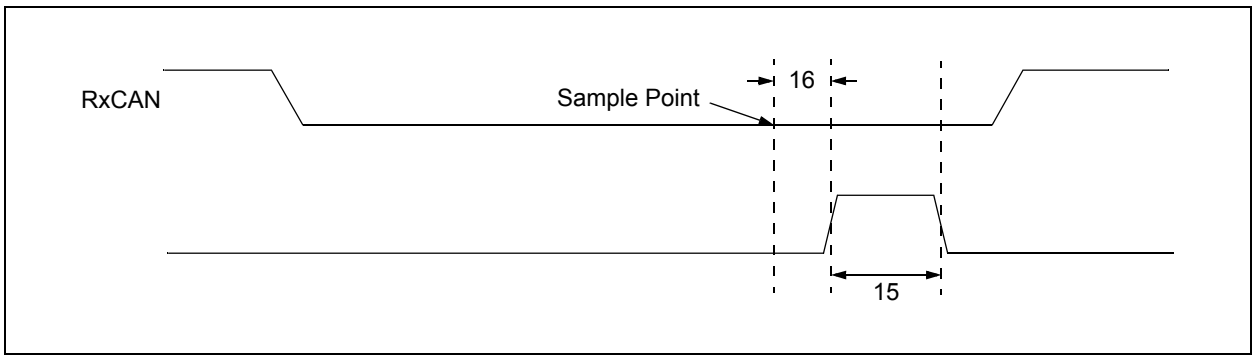
2: Characterized, not 100% tested.

TABLE 7-6: SPI INTERFACE AC CHARACTERISTICS

SPI Interface AC Characteristics			Extended (E): T_{AMB} = -40°C to +125°C; V_{DD} = 2.7V to 5.5V			
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
	F_{CLK}	Clock Frequency	—	10	MHz	
1	t_{CSS}	$\overline{CS}$ Setup Time	50	—	ns	
2	t_{CSH}	$\overline{CS}$ Hold Time	50	—	ns	
3	t_{CSD}	$\overline{CS}$ Disable Time	50	—	ns	
4	t_{SU}	Data Setup Time	10	—	ns	
5	t_{HD}	Data Hold Time	10	—	ns	
6	t_R	Clock Rise Time	—	2	µs	**(Note 1)**
7	t_F	Clock Fall Time	—	2	µs	**(Note 1)**
8	t_{HI}	Clock High Time	45	—	ns	
9	t_{LO}	Clock Low Time	45	—	ns	
10	t_{CLD}	Clock Delay Time	50	—	ns	
11	t_{CLE}	Clock Enable Time	50	—	ns	
12	t_V	Output Valid from Clock Low	—	45	ns	
13	t_{HO}	Output Hold Time	0	—	ns	
14	t_{DIS}	Output Disable Time	—	100	ns	

Note 1: Characterized, not 100% tested.

FIGURE 7-1: START-OF-FRAME PIN AC CHARACTERISTICS



7.3 CAN Transceiver Characteristics

TABLE 7-7: DC CHARACTERISTICS

Electrical Characteristics: Extended (E): T_{AMB} = -40°C to +125°C; V_{DDA} = 4.5V to 5.5V, V_{IO} = 2.7V to 5.5V, R_L = 60Ω; unless otherwise specified.						
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SUPPLY						
V_{DDA} Pin						
Voltage Range	V_{DDA}	4.5	—	5.5		
Supply Current	I_{DD}	—	5	10	mA	Recessive; $V_{TXD} = V_{DDA}$
		—	45	70		Dominant; V_{TXD} = 0V
Standby Current	I_{DDS}	—	5	15	µA	Includes I_{IO}
High Level of the POR Comparator	V_{PORH}	3.8	—	4.3	V	
Low Level of the POR Comparator	V_{PORL}	3.4	—	4.0	V	
Hysteresis of POR Comparator	V_{PORD}	0.3	—	0.8	V	
V_{IO} Pin						
Digital Supply Voltage Range	V_{IO}	2.7	—	5.5	V	
Supply Current on V_{IO}	I_{IO}	—	4	30	µA	Recessive; $V_{TXD} = V_{IO}$
		—	85	500		Dominant; V_{TXD} = 0V
Standby Current	I_{DDS}	—	0.3	1	µA	**(Note 1)**
Undervoltage Detection on V_{IO}	$V_{UVD(IO)}$	—	1.2	—	V	**(Note 1)**
BUS LINE (CANH, CANL) TRANSMITTER						
CANH, CANL: Recessive Bus Output Voltage	$V_{O(R)}$	2.0	0.5 V_{DDA}	3.0	V	$V_{TXD} = V_{DDA}$; No load
CANH, CANL: Bus Output Voltage in Standby	$V_{O(S)}$	-0.1	0.0	+0.1	V	STBY = $V_{TXD} = V_{DDA}$; No load
Recessive Output Current	$I_{O(R)}$	-5	—	+5	mA	-24V < V_{CAN} < +24V
CANH: Dominant Output Voltage	$V_{O(D)}$	2.75	3.50	4.50	V	T_{XD} = 0; R_L = 50 to 65Ω
CANL: Dominant Output Voltage		0.50	1.50	2.25		R_L = 50 to 65Ω
Symmetry of Dominant Output Voltage ($V_{DD} - V_{CANH} - V_{CANL}$)	$V_{O(D)(M)}$	-400	0	+400	mV	$V_{TXD} = V_{SS}$ **(Note 1)**
Dominant: Differential Output Voltage	$V_{O(DIFF)}$	1.5	2.0	3.0	V	$V_{TXD} = V_{SS}$; R_L = 50 to 65Ω, Figure 7-2, Figure 7-4
Recessive: Differential Output Voltage		-120	0	12	mV	$V_{TXD} = V_{DDA}$, Figure 7-2, Figure 7-4
		-500	0	50	mV	$V_{TXD} = V_{DDA}$; No load, Figure 7-2, Figure 7-4
CANH: Short-Circuit Output Current	$I_{O(SC)}$	-120	85	—	mA	$V_{TXD} = V_{SS}$; V_{CANH} = 0V; CANL: floating
		-100	—	—	mA	Same as above, but V_{DDA} = 5V, T_{AMB} = +25°C **(Note 1)**
CANL: Short-Circuit Output Current		—	75	+120	mA	$V_{TXD} = V_{SS}$; V_{CANL} = 18V; CANH: floating
		—	—	+100	mA	Same as above, but V_{DD} = 5V, T_{AMB} = +25°C **(Note 1)**

Note 1: Characterized; not 100% tested.

2: -12V to 12V is ensured by characterization, tested from -2V to 7V.

TABLE 7-7: DC CHARACTERISTICS (CONTINUED)

Electrical Characteristics: Extended (E): T_{AMB} = -40°C to +125°C; V_{DDA} = 4.5V to 5.5V, V_{IO} = 2.7V to 5.5V, R_L = 60Ω; unless otherwise specified.						
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
BUS LINE (CANH; CANL) RECEIVER						
Recessive Differential Input Voltage	$V_{DIFF(R)(I)}$	-1.0	—	+0.5	V	Normal mode; -12V < $V_{(CANH, CANL)}$ < +12V; see Figure 7-6 **(Note 2)**
		-1.0	—	+0.4		Standby mode; -12V < $V_{(CANH, CANL)}$ < +12V; see Figure 7-6 **(Note 2)**
Dominant Differential Input Voltage	$V_{DIFF(D)(I)}$	0.9	—	V_{DDA}	V	Normal mode; -12V < $V_{(CANH, CANL)}$ < +12V; see Figure 7-6 **(Note 2)**
		1.0	—	V_{DDA}		Standby mode; -12V < $V_{(CANH, CANL)}$ < +12V; see Figure 7-6 **(Note 2)**
Differential Receiver Threshold	$V_{TH(DIFF)}$	0.5	0.7	0.9	V	Normal mode; -12V < $V_{(CANH, CANL)}$ < +12V; see Figure 7-6 **(Note 2)**
		0.4	—	1.15		Standby mode; -12V < $V_{(CANH, CANL)}$ < +12V; see Figure 7-6 **(Note 2)**
Differential Input Hysteresis	$V_{HYS(DIFF)}$	50	—	200	mV	Normal mode; see Figure 7-6 **(Note 1)**
Common-Mode Input Resistance	R_{IN}	10	—	30	kΩ	**(Note 1)**
Common-Mode Resistance Matching	$R_{IN(M)}$	-1	0	+1	%	V_{CANH} = V_{CANL} **(Note 1)**
Differential Input Resistance	$R_{IN(DIFF)}$	10	—	100	kΩ	**(Note 1)**
Common-Mode Input Capacitance	$C_{IN(CM)}$	—	—	20	pF	V_{TXD} = V_{DDA} **(Note 1)**
Differential Input Capacitance	$C_{IN(DIFF)}$	—	—	10		V_{TXD} = V_{DDA} **(Note 1)**
CANH, CANL: Input Leakage	I_{LI}	-5	—	+5	µA	V_{DDA} = V_{TXD} = V_{STBY} = 0V, V_{IO} = 0V, V_{CANH} = V_{CANL} = 5V
DIGITAL INPUT PINS (T_{XD}, STBY)						
High-Level Input Voltage	V_{IH}	0.7 V_{IO}	—	V_{IO} + 0.3	V	
Low-Level Input Voltage	V_{IL}	-0.3	—	0.3 V_{IO}	V	
High-Level Input Current	I_{IH}	-1	—	+1	µA	
T_{XD}: Low-Level Input Current	$I_{IL(TXD)}$	-270	-150	-30	µA	
STBY: Low-Level Input Current	$I_{IL(STBY)}$	-30	—	-1	µA	
RECEIVE DATA (R_{XD}) OUTPUT						
High-Level Output Voltage	V_{OH}	V_{IO} – 0.4	—	—	V	I_{OH} = -1 mA; typical -2 mA
Low-Level Output Voltage	V_{OL}	—	—	0.4	V	I_{OL} = 4 mA; typical 8 mA
THERMAL SHUTDOWN						
Shutdown Junction Temperature	$T_{J(SD)}$	165	175	185	°C	-12V < $V_{(CANH, CANL)}$ < +12V **(Note 1)**
Shutdown Temperature Hysteresis	$T_{J(HYST)}$	20	—	30	°C	-12V < $V_{(CANH, CANL)}$ < +12V **(Note 1)**

Note 1: Characterized; not 100% tested.

2: -12V to 12V is ensured by characterization, tested from -2V to 7V.

TABLE 7-8: AC CHARACTERISTICS

Electrical Characteristics: Extended (E): T_{AMB} = -40°C to +125°C; V_{DDA} = 4.5V to 5.5V, V_{IO} = 2.7V to 5.5V, R_L = 60Ω; unless otherwise specified.							
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
1	t_{BIT}	Bit Time	1	—	69.44	µs	
2	f_{BIT}	Bit Frequency	14.4	—	1000	kHz	
3	$t_{TXD-BUSON}$	Delay T_{XD} Low to Bus Dominant	—	—	70	ns	
4	$t_{TXD-BUSOFF}$	Delay T_{XD} High to Bus Recessive	—	—	125	ns	
5	$t_{BUSON-RXD}$	Delay Bus Dominant to R_{XD}	—	—	70	ns	
6	$t_{BUSOFF-RXD}$	Delay Bus Recessive to R_{XD}	—	—	110	ns	
7	$t_{TXD-RXD}$	Propagation Delay T_{XD} to R_{XD}	—	—	125	ns	Negative edge on T_{XD}
8			—	—	235		Positive edge on T_{XD}
9	$t_{FLTR(WAKE)}$	Delay Bus Dominant to R_{XD} (Standby mode)	0.5	1	4	µs	Standby mode
10	t_{WAKE}	Delay Standby to Normal mode	5	25	40	µs	Negative edge on STBY
11	t_{PDT}	Permanent Dominant Detect Time	—	1.25	—	ms	T_{XD} = 0V
12	t_{PDTR}	Permanent Dominant Timer Reset	—	100	—	ns	The shortest Recessive pulse on T_{XD} or CAN bus to reset permanent Dominant timer

FIGURE 7-2: PHYSICAL BIT REPRESENTATION AND SIMPLIFIED BIAS IMPLEMENTATION

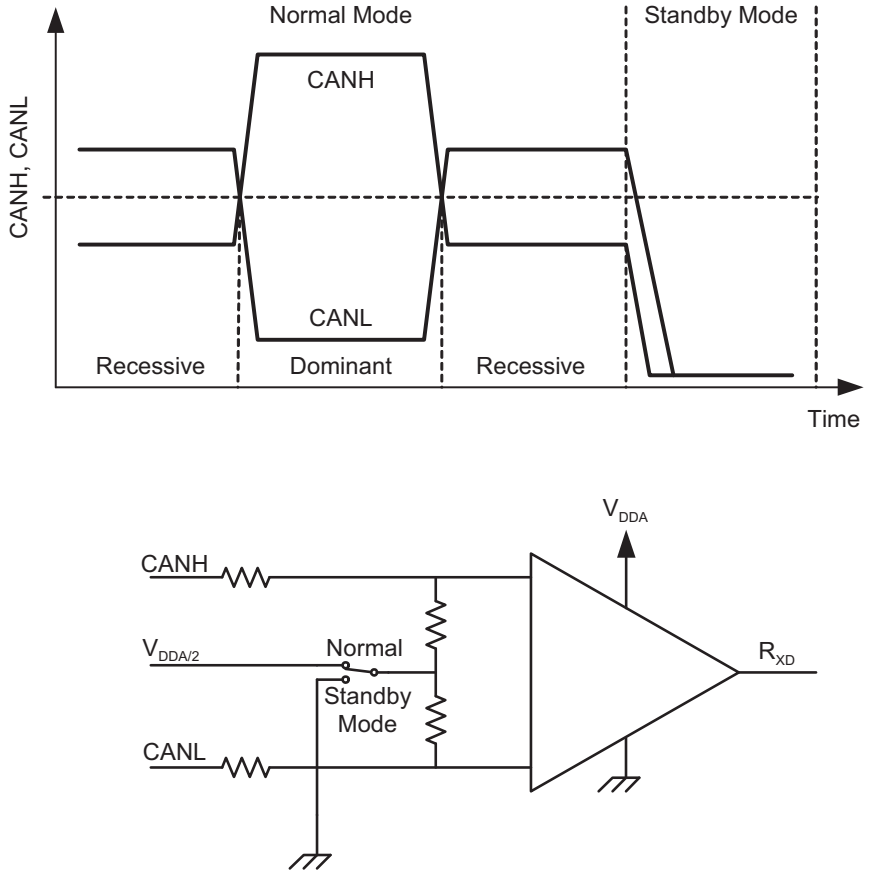
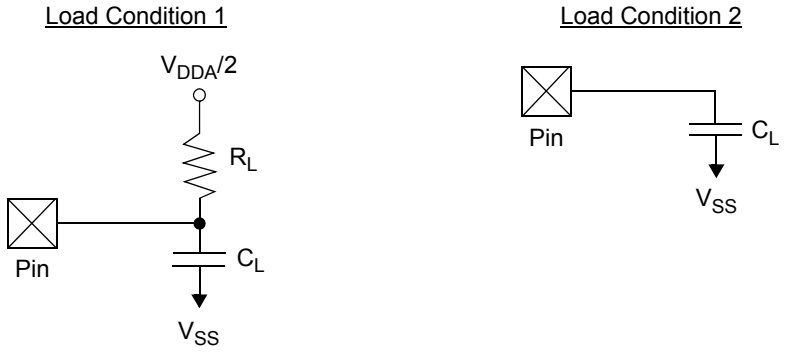


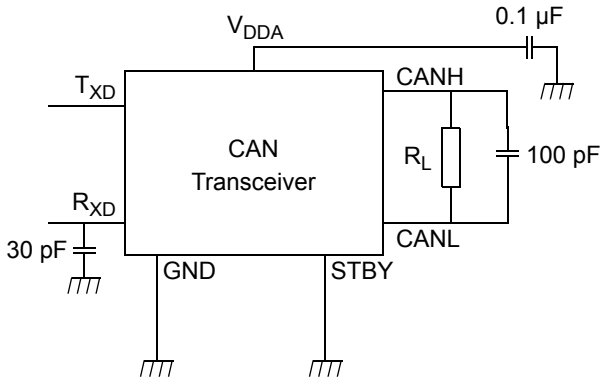
FIGURE 7-3: TEST LOAD CONDITIONS



R_L = 464 Ω

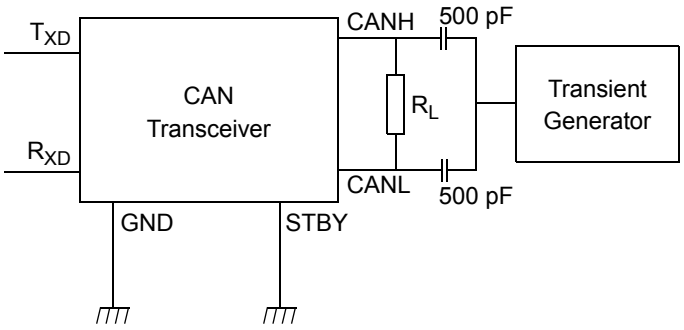
C_L = 50 pF for all digital pins

FIGURE 7-4: TEST CIRCUIT FOR ELECTRICAL CHARACTERISTICS



Note: V_{IO} is connected to V_{DDA}.

FIGURE 7-5: TEST CIRCUIT FOR AUTOMOTIVE TRANSIENTS



Note: The waveforms of the applied transients shall be in accordance with ISO-7637, Part 1, Test Pulses 1, 2, 3a and 3b.

V_{IO} is connected to V_{DDA}.

FIGURE 7-6: HYSTERESIS OF THE RECEIVER

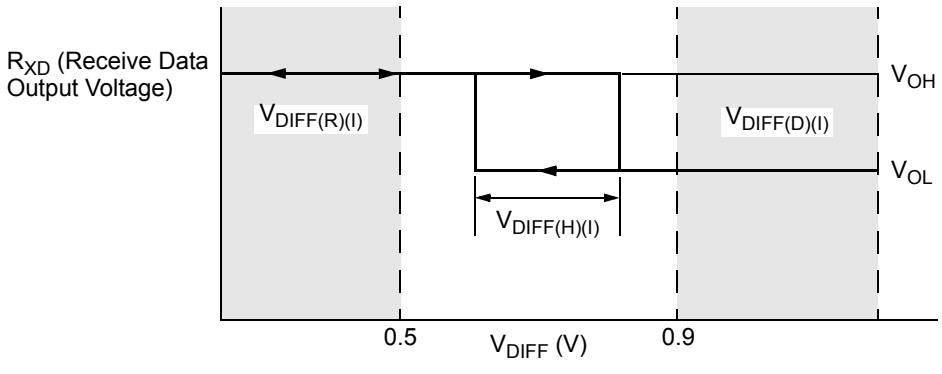


FIGURE 7-7: TIMING DIAGRAM FOR AC CHARACTERISTICS

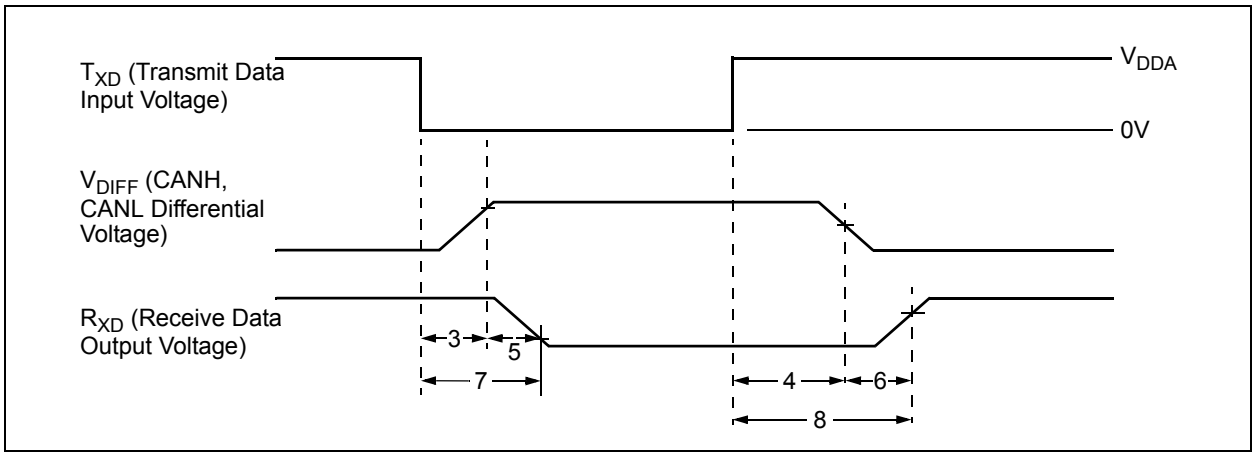


FIGURE 7-8: TIMING DIAGRAM FOR WAKE-UP FROM STANDBY

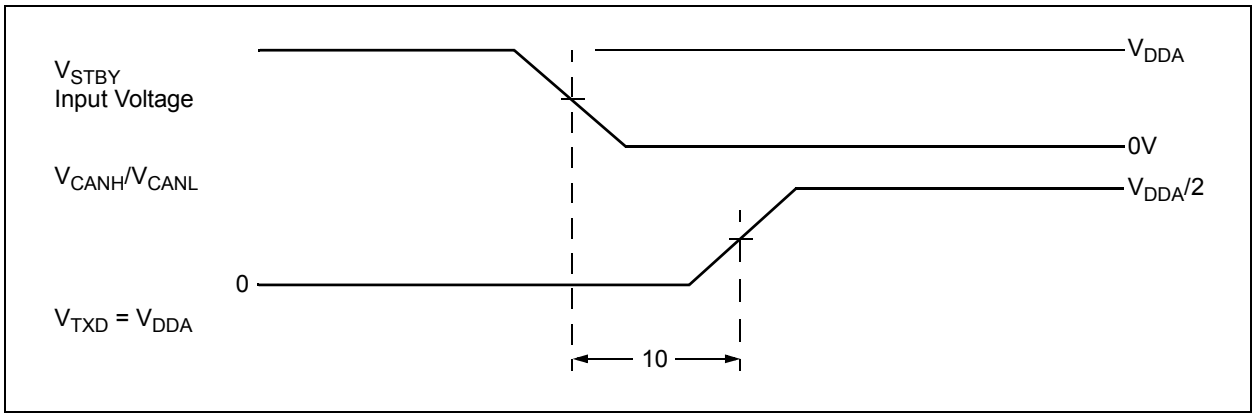
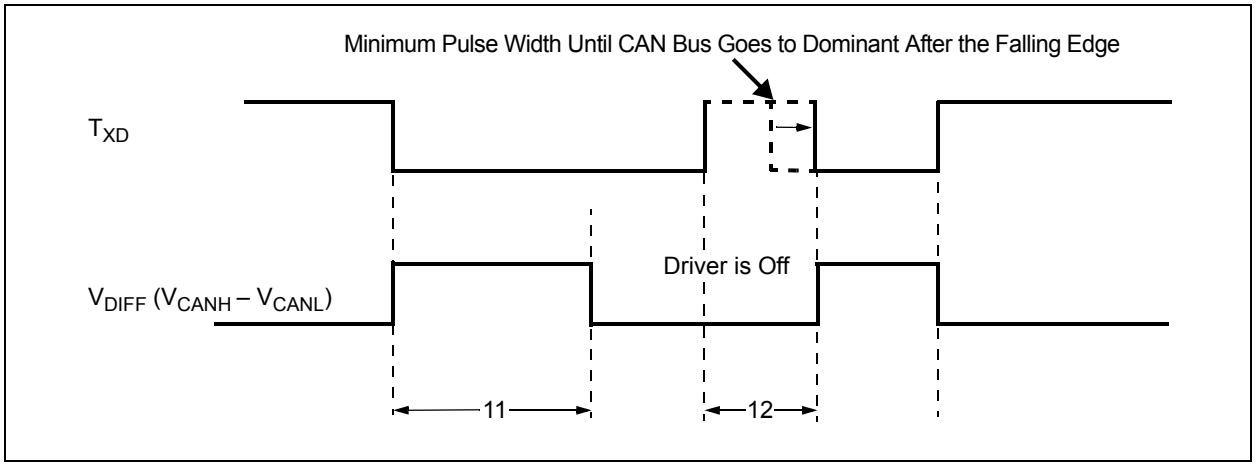


FIGURE 7-9: PERMANENT DOMINANT TIMER RESET DETECT



7.4 Thermal Specifications

TABLE 7-9: THERMAL SPECIFICATIONS

Parameter	Symbol	Min.	Typ.	Max.	Units	Test Conditions
Temperature Ranges						
Specified Temperature Range	T_A	-40	—	+125	°C	
Operating Temperature Range	T_A	-40	—	+125	°C	
Storage Temperature Range	T_A	-65	—	+150	°C	
Thermal Package Resistances						
Thermal Resistance, 28L-QFN 6x6 mm	θ_{JA}	—	32.8	—	°C/W	
Thermal Resistance, 28L-SSOP	θ_{JA}	—	80	—	°C/W	

7.5 Terms and Definitions

A number of terms are defined in ISO-11898 that are used to describe the electrical characteristics of a CAN transceiver device. These terms and definitions are summarized in this section.

7.5.1 BUS VOLTAGE

V_{CANL} and V_{CANH} denote the voltages of the bus line wires, CANL and CANH, relative to the ground of each individual CAN node.

7.5.2 COMMON-MODE BUS VOLTAGE RANGE

Boundary voltage levels of V_{CANL} and V_{CANH}, with respect to ground for which proper operation will occur, if up to the maximum number of CAN nodes are connected to the bus.

7.5.3 DIFFERENTIAL INTERNAL CAPACITANCE, C_{DIFF} (OF A CAN NODE)

Capacitance seen between CANL and CANH during the Recessive state, when the CAN node is disconnected from the bus (see Figure 7-10).

7.5.4 DIFFERENTIAL INTERNAL RESISTANCE, R_{DIFF} (OF A CAN NODE)

Resistance seen between CANL and CANH during the Recessive state when the CAN node is disconnected from the bus (see Figure 7-10).

7.5.5 DIFFERENTIAL VOLTAGE, V_{DIFF} (OF CAN BUS)

Differential voltage of the two-wire CAN bus, value: $V_{DIFF} = V_{CANH} - V_{CANL}$.

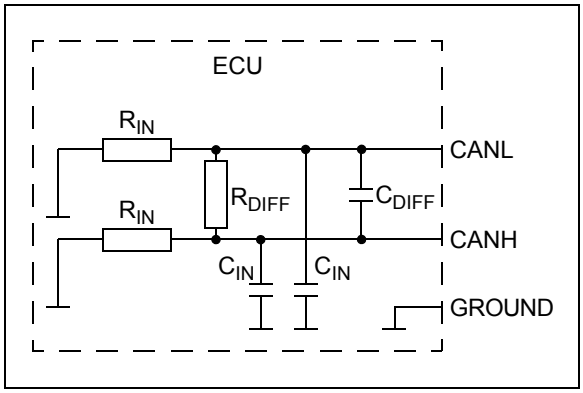
7.5.6 INTERNAL CAPACITANCE, C_{IN} (OF A CAN NODE)

Capacitance seen between CANL (or CANH) and ground, during the Recessive state, when the CAN node is disconnected from the bus (see Figure 7-10).

7.5.7 INTERNAL RESISTANCE, R_{IN} (OF A CAN NODE)

Resistance seen between CANL (or CANH) and ground, during the Recessive state, when the CAN node is disconnected from the bus (see Figure 7-10).

FIGURE 7-10: PHYSICAL LAYER DEFINITIONS

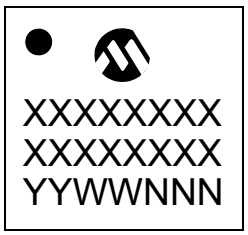


NOTES:

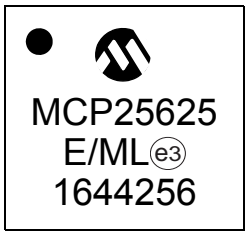
8.0 PACKAGING INFORMATION

8.1 Package Marking Information

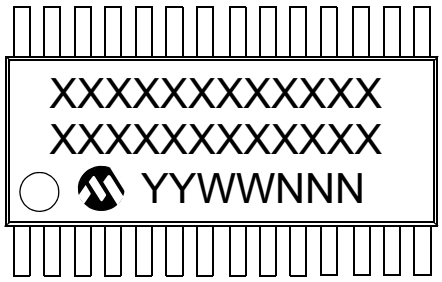
28-Lead QFN (6x6 mm)



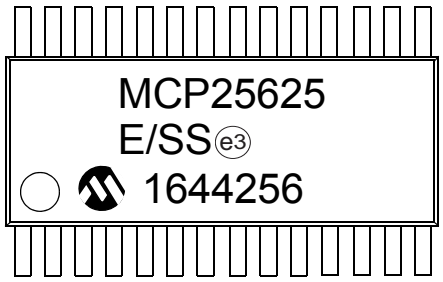
Example



28-Lead SSOP (5.30 mm)



Example

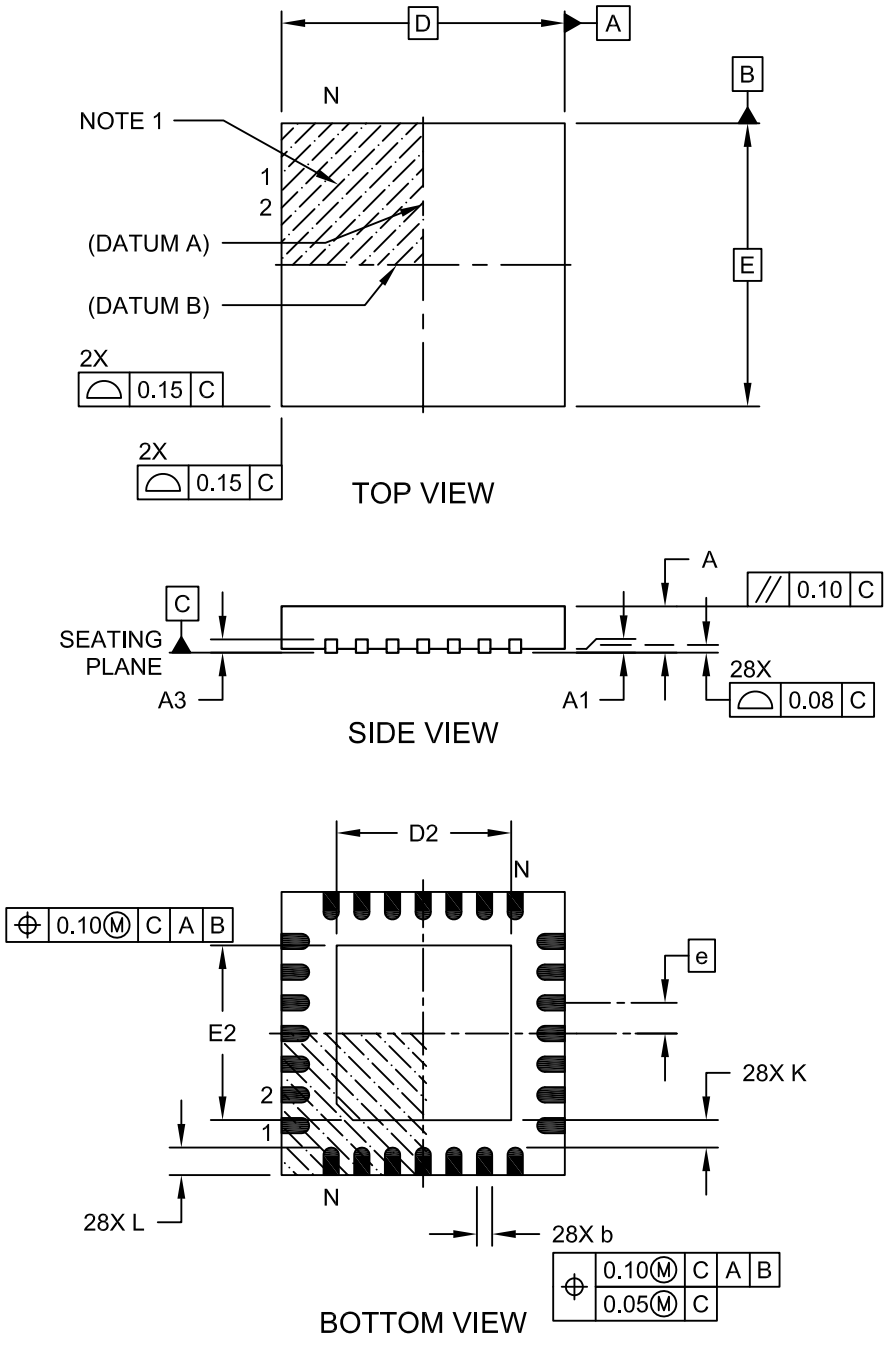


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN]
With 0.55 mm Terminal Length

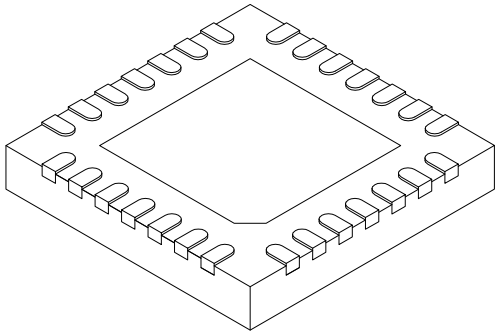
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-105C Sheet 1 of 2

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

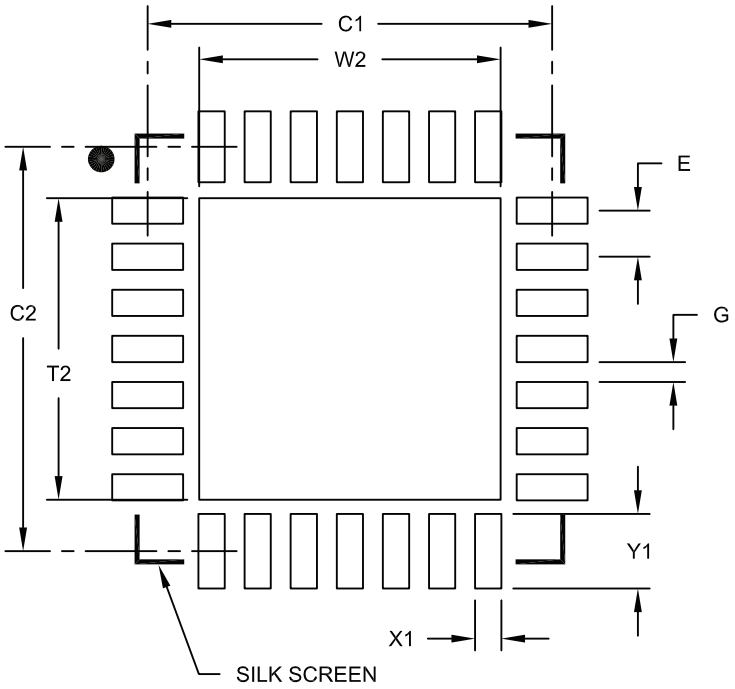
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105C Sheet 2 of 2

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

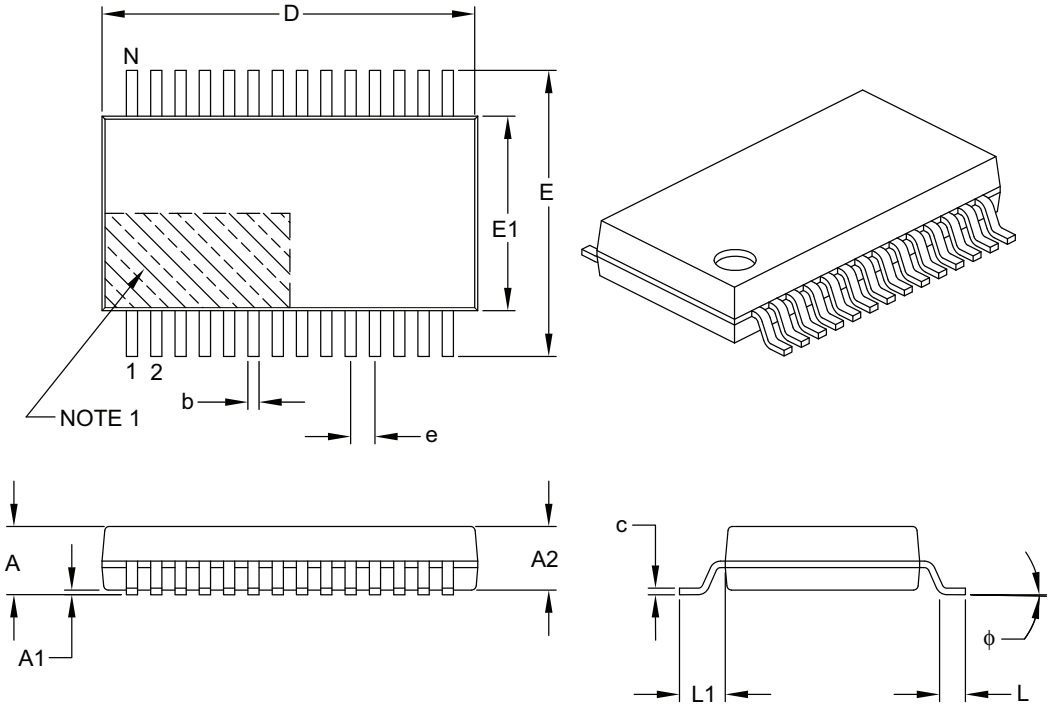
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

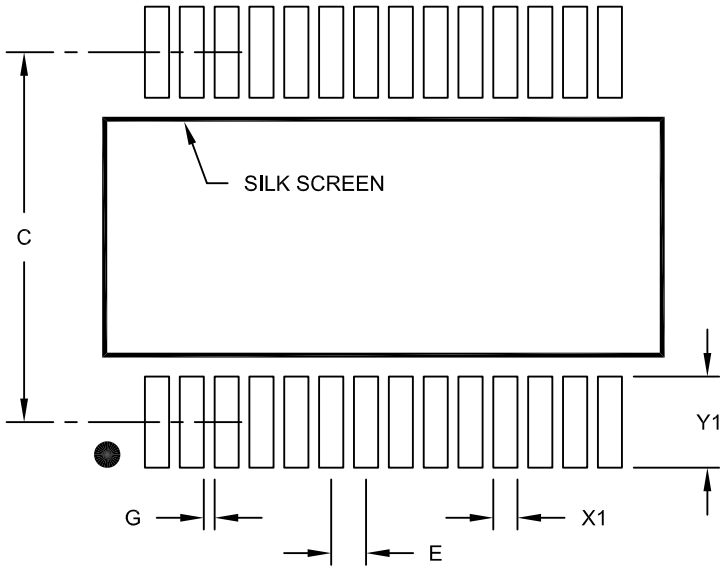
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

APPENDIX A: REVISION HISTORY

Revision C (January 2019)

The following is the list of modifications:

1. Updated Figure 1-2.
2. Updated **Section 3.13.1, Oscillator Start-up Timer**.
3. Updated Table 4-2.
4. Updated Register 4-34.

Revision B (January 2017)

The following is the list of modifications:

1. The usage of the RXMx bits setting, '01' and '10', in the RXBxCTRL registers (Register 4-9 and Register 4-10) is not recommended.
2. Updated Figure 3-11.
3. Updated Table 3-3.

Revision A (March 2014)

- Original Release of this Document.

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X] (1)	-X	/XX
Device	Tape and Reel Option	Temperature Range	Package

Device:	MCP25625 CAN Controller with Integrated Transceiver		
Temperature Range:	E	=	-40°C to +125°C (Extended)
Tape and Reel Option:	Blank	=	Standard packaging (tube)
	T	=	Tape and Reel
Package:	ML	=	Plastic Quad Flat, No Lead Package – 6x6 mm Body with 0.55 mm Terminal Length, 28-Lead
	SS	=	Plastic Shrink Small Outline – 5.30 mm Body 28-Lead

Examples:

a) MCP25625-E/ML: Extended Temperature, 28-Lead 6x6 QFN Package

b) MCP25625-E/SS: Extended Temperature, 28-Lead SSOP Package

c) MCP25625T-E/ML: Tape and Reel, Extended Temperature, 28-Lead 6x6 QFN Package

d) MCP25625T-E/SS: Tape and Reel, Extended Temperature, 28-Lead SSOP Package

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-4057-4

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820