

Introduction (Ask a Question)

The CoreCAN IP, developed by Microchip Technology, is a highly efficient and reliable Intellectual Property (IP) core designed for Controller Area Network (CAN) communication systems. It serves as a critical component in the implementation of CAN networks, providing robust communication capabilities for various applications.

Primary Function and Purpose

The primary function of the CoreCAN IP is to facilitate communication between different nodes in a CAN network. It ensures the accurate transmission and reception of data frames, enabling seamless data exchange in automotive, industrial, and other embedded systems. The CoreCAN IP is designed to meet the stringent requirements of CAN communication, offering high performance and reliability.

Role in CAN Communication Systems

In CAN communication systems, the CoreCAN IP plays a pivotal role by managing the data link layer of the CAN protocol. It handles tasks such as message framing, error detection, and arbitration, ensuring that data is transmitted efficiently and without collisions. By integrating the CoreCAN IP into their designs, developers can achieve robust and scalable CAN networks that meet industry standards.

Key Benefits

The CAN controller offers several key benefits, including reliability, cost-effectiveness, scalability, and customization.

Enhanced System Performance and Interoperability

By integrating the CAN controller, system performance is significantly enhanced. It ensures seamless interoperability between different components, leading to a more efficient and cohesive system.

Advantages Over Alternative CAN Controllers

Compared to alternative CAN controllers, this solution provides superior reliability and cost-effectiveness. Its scalability allows for easy expansion, and its customization options ensure it can be tailored to specific needs.

Overview (Ask a Question)

This section provides an overview of the CoreCAN IP.

Receive Path

This section provides information about receive path.

- The receive path in the system consists of a memory block that stores up to 32 messages.
- Each buffer is equipped with a configurable message filter, which can be set through registers to cover multiple fields, including ID, IDE (Identifier Extension), RTR (Remote Transmission Request), Data Byte 1, and Data Byte 2.
- Multiple message buffers can be assigned the same ID to store multiple messages with the same ID.
- The core includes an automatic Remote Transmission Request (RTR) response handler, which optionally generates an RTR interrupt.

Transmit Path

This section provides information about transmit path.

- The transmit path consists of a memory block that stores up to 32 transmit (Tx) messages. These messages can be transmitted with programmable priority arbitration. This feature allows precise control over the transmission order, ensuring that higher-priority messages are sent first.
- Additionally, a message abort command is available, enabling the immediate cancellation of pending transmissions if needed.
- The transmit path supports single-shot transmission, meaning that there is no automatic retransmission in the event of an error or arbitration loss.

Summary (Ask a Question)

Table 1. CoreCAN IP has the following characteristics:

Core Version	This document applies to the CoreCAN v2.0.
Supported Device Families	<ul style="list-style-type: none">• PolarFire® SoC• PolarFire• RT PolarFire• RTG4™• IGLOO® 2• SmartFusion® 2
Supported Tool Flow	Requires Libero® SoC v12.0 or later releases.
Supported Interfaces	<ul style="list-style-type: none">• AMBA 3 Advanced Peripheral Bus (APB) Interface• APB Target• System Bus Interface <p>For more information, see Functional Description.</p>
Licensing	Evaluation: The evaluation version is available with obfuscated and encrypted Verilog RTL with self-destruct logic. It supports user approximately four hours of functionality on silicon. It will not be license-locked and will be available free. Obfuscated: The obfuscated version is available with obfuscated and encrypted Verilog RTL. It supports unlimited functionality on silicon. Obfuscated version is license-locked. You must purchase this license separately.
Installation Instructions	The CoreCAN must be installed to the IP Catalog of Libero SoC software automatically through the IP Catalog update function in Libero SoC software. Alternatively, it can be manually downloaded from the catalog. Once the IP core is installed, it is configured, generated, and instantiated within SmartDesign for inclusion in the Libero project.
Device Utilization and Performance	A summary of utilization and performance information for is listed in the Device Utilization and Performance .

Features (Ask a Question)

This section lists the key features of the IP core, including performance, power efficiency, size, and integration ease.

- Message-based (not address-based) communication.
- Multi-master system where any node can transmit when the bus is free.
- Operating speeds up to 1 Mbit/s (high-speed CAN).
- Maximum bus length depends on speed (for example, 40m at 1 Mbit/s, 1000m at 50 kbit/s).
- Differential signaling (CAN-H and CAN-L) for noise immunity.
- Non-destructive arbitration using message IDs.
- Performance: High throughput and low latency.
- Power Efficiency: Optimized for low power consumption.

- Size: Compact design suitable for various applications.
- Integration Ease: Easy to integrate with existing systems.

Table of Contents

Introduction.....	1
Overview.....	1
Summary.....	2
Features.....	2
1. Functional Description of the CoreCAN IP.....	6
1.1. Tx Memory.....	6
1.2. Memory Arbitrer.....	6
1.3. Memory Controller.....	7
1.4. Tx Framer.....	7
1.5. Arbitration Controller.....	8
1.6. Rx Memory.....	8
1.7. Rx De-Framer.....	8
2. IP Core Parameters and Interface Signals.....	15
2.1. Configuration Settings.....	15
2.2. Inputs and Outputs Signals.....	15
2.3. Timing Diagrams.....	16
3. Implementation of CoreCAN IP in Libero Design Suite.....	18
3.1. Smart Design Integration.....	18
4. Register Summary	21
4.1. TxMessageY Word1 Data Register.....	50
4.2. TxMessageY Word0 Data Register.....	51
4.3. TxMessageY ID Register.....	52
4.4. TxMessageY Control Register.....	53
4.5. TxBuffer Status Register.....	55
4.6. RxMessageY Word1 Data Register.....	60
4.7. RxMessageY Word0 Data Register.....	61
4.8. RxMessageY ID Register.....	62
4.9. RxMessageY Control Register.....	63
4.10. RxMessageY AMR Register.....	65
4.11. RxMessageY AMR Data Register.....	66
4.12. RxMessageY ACR Register.....	67
4.13. RxMessageY ACR Data Register.....	68
4.14. RxBuffer Status Register.....	69
4.15. Interrupt Source Register.....	74
4.16. Interrupt Enable Register.....	77
4.17. Error Status Register.....	80
4.18. Configurator Register.....	81
4.19. Command Register.....	83
5. System Integration.....	84
6. Bit Time.....	85
7. Known Issues and Workarounds.....	86

8.	Discontinued Features and Devices.....	87
9.	Glossary.....	88
10.	Device Utilization and Performance.....	89
11.	Revision History.....	90
	Microchip FPGA Support.....	91
	Microchip Information.....	91
	Trademarks.....	91
	Legal Notice.....	91
	Microchip Devices Code Protection Feature.....	92

1. Functional Description of the CoreCAN IP [\(Ask a Question\)](#)

The CoreCAN IP is designed to provide robust and efficient communication capabilities for Controller Area Network (CAN) systems. It facilitates the transmission and reception of CAN messages, ensuring reliable data exchange in automotive and industrial applications.

Key Functional Blocks

The CoreCAN IP consists of several key functional blocks, each contributing to its overall functionality:

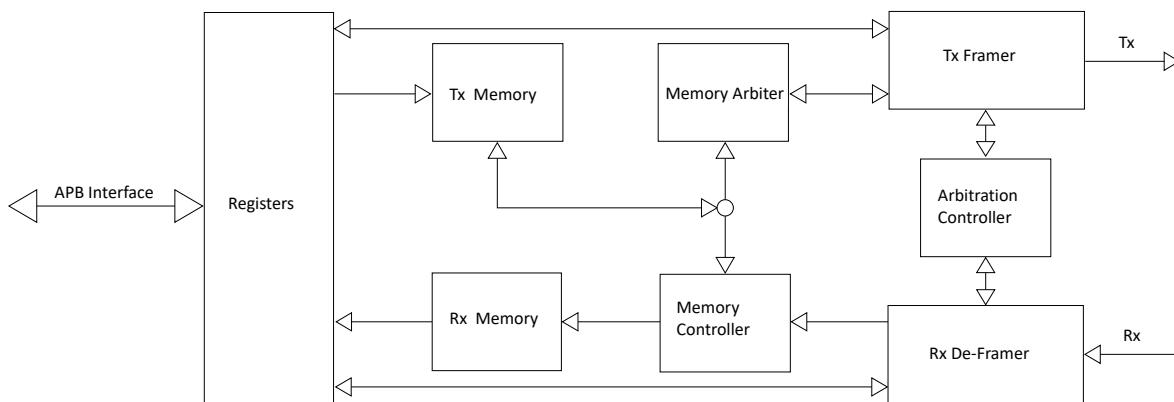
- Message Handler: Manages the transmission and reception of CAN messages, including message filtering and prioritization.
- Bit Timing Logic: Ensures accurate synchronization with the CAN bus by managing the timing of bit transmission and reception.
- Error Management Unit: Monitors and handles error conditions on the CAN bus, ensuring reliable communication by implementing error detection and correction mechanisms.
- Protocol Controller: Implements the CAN protocol, managing the framing, arbitration, acknowledgment, and error signaling processes.

Interactions with Other Components or Applications

The CoreCAN IP interacts with various components and applications within a CAN system:

- CAN Transceivers: The CoreCAN IP connects to CAN transceivers, which handle the physical layer signaling required for communication on the CAN bus.
- Diagnostic Tools: The CoreCAN IP can be accessed by diagnostic tools for monitoring and troubleshooting CAN communication issues.
- Other CAN Nodes: The CoreCAN IP communicates with other CAN nodes on the network, facilitating data exchange and coordination between different system components.

Figure 1-1. CoreCAN IP Block Diagram



1.1. Tx Memory [\(Ask a Question\)](#)

The core implements a memory block that can hold up to 32 transmit messages. The memory arbiter selects a message based on the chosen arbitration scheme. Upon transmission or arbitration loss, the priority arbiter re-evaluates the priority of the next message.

1.2. Memory Arbiter [\(Ask a Question\)](#)

The memory arbiter supports both round-robin and fixed-priority arbitration. The arbitration mode is selected using the configuration register:

- Round-Robin: The controller operates in a cyclical manner to read data from memory blocks, specifically from TxMessage 0 to TxMessage 31. It processes messages sequentially based on data availability, starting at TxMessage 0 and progressing to TxMessage 31. Once it reaches TxMessage 31, it loops back to TxMessage 0 and begins the cycle again. If data is available in any of these memory blocks, the corresponding TX_REQ flag is set, indicating data presence and allowing efficient message management.
- Fixed Priority: TxMessage 0 has the highest priority. The remaining messages are transmitted in sequential order.

1.3. Memory Controller [\(Ask a Question\)](#)

The memory controller block generates addresses for the memory block to store messages for transmission and reception. It checks memory availability and generates addresses accordingly. This is a common block for both Tx and Rx memory.

- Tx Memory: After receiving data from the register through the APB interface, the controller allocates a memory block based on availability and generates the corresponding address.
- Rx Memory: After receiving a message, it is validated against the message filter. If the message passes the filter, it is stored in the corresponding memory block as per the generated address. If a message with the same ID is already present in the corresponding memory block and no memory is available, the new message is discarded.

1.4. Tx Framer [\(Ask a Question\)](#)

This section describes the functionality of the Tx Framer.

Message Transmission Process

Writing the Message to an Empty Buffer

The core first checks for an empty transmit message holding buffer. An empty buffer is indicated when

```
TX_REQ = 0
```

, meaning the transmit request flag is not set. This ensures that the message is not overwritten.

Requesting Transmission

After the message is written to the buffer, the TX_REQ flag is set to 1 for that specific message buffer. Setting this flag triggers the transmission process.

Waiting for Transmission

Once TX_REQ is set to 1, the system enters a waiting state. The TX_REQ flag remains set (1) while the transmit request is pending. The buffer data must not be modified while the TX_REQ flag is set.

Message Priority Arbitration

If multiple messages are queued for transmission, an internal arbiter determines the next message to send. The selection follows a predefined arbitration scheme.

Serialization

The message is read from memory and converted into serialized data. A cyclic redundancy check (CRC) is generated and appended to the message.

Transmission Completion

After the message is successfully transmitted:

- The core automatically resets the TX_REQ flag to 0, indicating that the buffer is now free.
- The core sets the TxMsg interrupt status bit.

1.5. Arbitration Controller [\(Ask a Question\)](#)

Whenever the bus is free, any device may begin transmitting a message. If multiple nodes start transmitting messages simultaneously, the conflict is resolved through bitwise arbitration using the IDENTIFIER. The arbitration mechanism ensures that neither information nor time is lost.

- If a DATA FRAME and a REMOTE FRAME with the same IDENTIFIER are initiated simultaneously, the DATA FRAME prevails over the REMOTE FRAME.
- During arbitration, each transmitter compares the level of the bit it transmits with the level monitored on the bus.
- If these levels match, the unit continues to send data.
- Based on feedback from the controller, the Tx framer continuously sends data.
- If a recessive level is sent but a dominant level is monitored, the unit loses arbitration and must withdraw without sending any additional bits.

1.6. Rx Memory [\(Ask a Question\)](#)

The Core implements a memory block capable of storing up to 32 receive messages, each with its own message filter mask. This design provides precise control over which messages are accepted. The Core supports automatic replies to Remote Transmission Request (RTR) messages, enhancing system responsiveness.

- When a message is successfully received and accepted into a buffer, the corresponding MSGAV (Message Available) flag is set.
- The received message remains valid as long as the MSGAV flag remains set.
- To prepare the buffer for a new message, the host CPU must reset the MSGAV flag.

Once a new message is received, the Memory Controller searches through all receive buffers, starting from RxMessage0, until it finds an empty buffer to store the new message. This systematic approach ensures efficient message handling and storage within the Core.

A message is considered valid when:

- The receive message enable signal (rxmsg_en) is asserted.
- The acceptance filter matches the incoming message.

When the Memory Controller identifies a valid and empty memory block, it performs the following actions:

- Stores the incoming message in that block.
- Sets the corresponding MSGAV (Message Available) flag to 1, indicating successful reception.

If the rxint_en (Receive Interrupt Enable) flag is set for that buffer, the Core also asserts the RxMsg flag of the interrupt controller.

However, if the Memory Controller encounters a buffer that already contains a message (MSGAV = 1) and the Link Flag for that buffer is not set, it cannot store the new message in that buffer. In this case:

- The Core asserts the RxMsgLoss (Receive Message Loss) interrupt flag, signaling that a message has been lost due to buffer overflow.

Additionally, if an incoming message has its RTR flag set and the RTRreply flag of the matching buffer is set, the message is not stored, and instead, an RTR auto-reply request is issued.

1.7. Rx De-Framer [\(Ask a Question\)](#)

This section describes the functionality of the Rx De-Framer.

CAN Frame Reception Process

Message Filtering

When a CAN frame arrives at a node, it first passes through the message filtering stage. This stage determines whether the message is relevant to the receiving node based on its identifier.

Integrity Checks and Acknowledgment

If the message passes the filter, the controller performs the following checks to ensure data integrity:

- Cyclic Redundancy Check (CRC): Verifies data integrity.
- Bit Monitoring and Stuff Bit Checking: Detects transmission errors.

If the message passes all checks, the controller sends an acknowledgment (ACK) bit to confirm successful reception.

Data Storage and Processing

After acknowledgment, the controller stores the message in the Receive Memory Block. The stored data is then available for processing by the node's application layer.

Error Detection and Handling

The controller continuously monitors for transmission errors. If an error is detected, the controller initiates error-handling procedures, which may include:

- Transmitting Error Frames: Notifies other nodes about the issue.
- Discarding Corrupt Messages: Maintains system reliability.

1.7.1. Message Filter (Ask a Question)

This section describes the functionality of the message filter.

Message Filtering Mechanism

Each memory block has a dedicated filter that determines whether a received message should be accepted or discarded. The filtering process is managed by two key registers:

1. Acceptance Mask Register (AMR)
2. Acceptance Code Register (ACR)

These registers work together to match specific bits in an incoming message before storing it in the memory block.

Filtering Fields

The following message fields are evaluated during filtering:

- ID (Identifier)
- IDE (Identifier Extension)
- RTR (Remote Transmission Request)
- Data Byte 1
- Data Byte 2

Acceptance Mask Register (AMR) Functionality

The AMR determines which bits of the incoming message should be compared against the ACR:

- AMR = 0: The incoming bit must match the corresponding ACR bit for the message to be accepted.
- AMR = 1: The incoming bit is considered "don't care", meaning it will not be checked.

This approach allows for flexible filtering, ensuring that only relevant messages are processed while minimizing unnecessary storage and processing overhead.

The following examples show the acceptance register configuration.



Important: 'h' indicates hex value in the following examples.

1.7.1.1. Example 1 [\(Ask a Question\)](#)

IDE – 0

RTR – 0

ID – 11'h08E

To accept the message with the above configuration, RxMessage[n] AMR Register, RxMessage[n] AMR Data Register, RxMessage[n] ACR Register and RxMessage[n] ACR Data Register must be programmed as follows:

RxMessage[n] AMR Register: ID bits [28:18] must be 11'h000, ID bits [17:3] must be 18'h3FFF, IDE must be 0, and RTR must be 0.

RxMessage[n] AMR Data Register: 16'hFFFF

RxMessage[n] ACR Register: ID bits [28:18] must be 11'h08E, ID bits [17:3] must be 18'hxxxxx (don't care), IDE must be 0, and RTR must be 0.

RxMessage[n] ACR Data Register: 16'hxxxx (don't care)

Table 1-1. Example 1 - AMR Bits

RxMessage[n] AMR Register Name	RxMessage[n] AMR Bits	Value
ID[28:18]	31:21	11'h000
ID[17:0]	20:3	18'h3FFF
IDE	2	0
RTR	1	0
Reserved	0	0

Table 1-2. Example 1 - ACR Bits

RxMessage[n] ACR Register Name	RxMessage[n] ACR Bits	Value
ID[28:18]	31:21	11'h08E
ID[17:0]	20:3	18'hxxxxx
IDE	2	0
RTR	1	0
Reserved	0	0

1.7.1.2. Example 2 [\(Ask a Question\)](#)

IDE – 1

RTR – 0

ID – 29'h10A1238E

To accept the message with the above configuration, RxMessage[n] AMR Register, RxMessage[n] AMR Data Register, RxMessage[n] ACR Register and RxMessage[n] ACR Data Register must be programmed as follows:

RxMessage[n] AMR Register: ID bits [28:0] must be 29'h00000000, IDE must be 0 and RTR must be 0

RxMessage[n] AMR Data Register: 16'hFFFF

RxMessage[n] ACR Register: ID bits [28:0] must be 29'h10A1238E, IDE must be 1 and RTR must be 0

RxMessage[n] ACR Data Register: 16'hxxxx (don't care)

Table 1-3. Example 2 - AMR Bits

RxMessage[n] AMR Register Name	RxMessage[n] AMR Bits	Value
ID [28:0]	31:3	29'h00000000
IDE	2	0
RTR	1	0
Reserved	0	0

Table 1-4. Example 2 - ACR Bits

RxMessage[n] ACR Register Name	RxMessage[n] ACR Bits	Value
ID[28:0]	31:3	29'h10A1238E
IDE	2	1
RTR	1	0
Reserved	0	0

1.7.1.3. Example 3 [\(Ask a Question\)](#)

IDE – 0

RTR – 1

ID – 11'h02A

To accept the message with the above configuration, RxMessage[n] AMR Register, RxMessage[n] AMR Data Register, RxMessage[n] ACR Register and RxMessage[n] ACR Data Register must be programmed as follows:

RxMessage[n] AMR Register: ID bits [28:18] must be 11'h000, ID bits [17:3] must be 18'hFFFF, IDE must be 0, and RTR must be 0.

RxMessage[n] AMR Data Register: 16'hFFFF

RxMessage[n] ACR Register: ID bits [28:18] must be 11'h02A, ID bits [17:3] must be 18'hxxxxx (don't care) , IDE must be 0, and RTR must be 0.

RxMessage[n] ACR Data Register: 16'hxxxx (don't care)

Table 1-5. Example 3 - AMR Bits

RxMessage[n] AMR Register Name	RxMessage[n] AMR Bits	Value
ID [28:18]	31:21	11'h000
ID [17:0]	20:3	18'h3FFFF
IDE	2	0
RTR	1	0
Reserved	0	0

Table 1-6. Example 3 - ACR Bits

RxMessage[n] ACR Register Name	RxMessage[n] ACR Bits	Value
ID [28:18]	31:21	11'h02A
ID [17:0]	20:3	18'hxxxxx
IDE	2	0
RTR	1	1
Reserved	0	0

1.7.1.4. Example 4 [\(Ask a Question\)](#)

IDE – 0

RTR – 0

Word1 Data [31:16] - 16'h1234

To accept the message with the above configuration, RxMessage[n] AMR Register, RxMessage[n] AMR Data Register, RxMessage[n] ACR Register and RxMessage[n] ACR Data Register must be programmed as follows:

RxMessage[n] AMR Register: ID bits [28:0] must be 29'h1FFFFFFF, IDE must be 0, and RTR must be 0.

RxMessage[n] AMR Data Register: 16'h0000

RxMessage[n] ACR Register: ID bits [28:0] must be 29'hxxxxxxxx(don't care), IDE must be 0, and RTR must be 0.

RxMessage[n] ACR Data Register: 16'h1234

Table 1-7. Example 4 - AMR Bits

RxMessage[n] AMR Register Name	RxMessage[n] AMR Bits	Value
ID [28:0]	31:3	29'h1FFFFFFF
IDE	2	0
RTR	1	0
Reserved	0	0

Table 1-8. Example 4 - ACR Bits

RxMessage[n] ACR Register Name	RxMessage[n] ACR Bits	Value
ID [28:0]	31:3	29'hxxxxxxxx
IDE	2	0
RTR	1	0
Reserved	0	0

Table 1-9. Example 4 - AMR Data Bits

RxMessage[n] AMR Data Register Name	RxMessage[n] AMR Data Bits	Value
Data	15:0	16'h0000

Table 1-10. Example 4 - ACR Data Bits

RxMessage[n] ACR Data Register Name	RxMessage[n] ACR Data Bits	Value
Data	15:0	16'h1234

1.7.1.5. Example 5 [\(Ask a Question\)](#)

IDE – 0

RTR – 0

ID – 11'h05A

Word1 Data [31:24] - 8'hEB

To accept the message with the above configuration, RxMessage[n] AMR Register, RxMessage[n] AMR Data Register, RxMessage[n] ACR Register and RxMessage[n] ACR Data Register must be programmed as follows:

RxMessage[n] AMR Register: ID bits [28:18] must be 11'h000, ID bits [17:0] must be 18'h3FFF, IDE must be 0, and RTR must be 0.

RxMessage[n] AMR Data Register: Data byte 1 must be 8'h00 and data byte 2 must be 8'hFF

RxMessage[n] ACR Register: ID bits [28:18] must be 11'h05A, ID bits[17:0] must be 18'hxxxxx, IDE must be 0, and RTR must be 0.

RxMessage[n] ACR Data Register: Data byte 1 must be 8'hEB and data byte 2 must be 8'hxx (don't care)

Table 1-11. Example 5 - AMR Bits

RxMessage[n] AMR Register Name	RxMessage[n] AMR Bits	Value
ID [28:18]	31:21	11'h000
ID [17:0]	20:3	18'h3FFFF
IDE	2	0
RTR	1	0
Reserved	0	0

Table 1-12. Example 5 - ACR Bits

RxMessage[n] ACR Register Name	RxMessage[n] ACR Bits	Value
ID [28:18]	31:21	11'h05A
ID [17:0]	20:3	18'hxxxxx
IDE	2	0
RTR	1	0
Reserved	0	0

Table 1-13. Example 5 - AMR Data Bits

RxMessage[n] AMR Data Register Name	RxMessage[n] AMR Data Bits	Value
Data byte 1	15:8	8'h00
Data byte 2	7:0	8'hFF

Table 1-14. Example 5 - ACR Data Bits

RxMessage[n] ACR Data Register Name	RxMessage[n] ACR Data Bits	Value
Data byte 1	15:8	8'hEB
Data byte 2	7:0	8'hxx

1.7.2. RX Buffer Linking [\(Ask a Question\)](#)

Multiple buffers can be appended to form a buffer array format. When an RX buffer already contains a message (Msgavl flag is set) and another message is received from the RX-deframer, the new message can either be discarded or stored in a linked buffer, if buffer linking is enabled for that particular buffer. When the RX-deframer receives a message matching a specific buffer, the controller checks whether the link flag is enabled for that buffer. If the link flag is enabled, the message will be stored in the next buffer; otherwise, it will be discarded.

Requirements for RX Buffer Linking:

- All the buffers of the same array must have identical message filter settings (AMR and ACR must be identical).
- The last buffer of an array must not have its link flag set.

1.7.3. RTR Auto-Reply [\(Ask a Question\)](#)

This section describes the functionality of the RTR auto-reply.

Automatic RTR Message Handling in the Core:

- The Core supports automatic replies to Remote Transmission Request (RTR) messages across all 32 receive buffers.
- When an RTR message is received in a buffer with the RTReply flag set, the buffer automatically responds with its current content, eliminating the need for software intervention.

RTR Message Reply Process:

1. Upon receiving an RTR message request, the RTReply_pending flag is set, indicating that an automatic reply is being prepared.

2. This flag is cleared when:
 - The message is successfully sent.
 - The message buffer is disabled.
3. To abort a pending RTR reply, the RTRabort command can be issued. This command stops the reply process and clears the RTReply_pending flag.

RTR Auto-Reply Status and Interrupts:

- If the RTR auto-reply option is enabled, the RTRsent flag is asserted when the RTR auto-reply message is successfully sent.
- The RTRsent flag is cleared by writing a '1' to it.
- An RTR message interrupt is generated when both the RTRsent flag and rxint_en (Receive Interrupt Enable) are set, indicating that an RTR message has been successfully auto-replied.
- This interrupt is cleared by resetting the RTRsent flag, ensuring that the Core is ready to handle subsequent RTR messages without interference from previous operations.

Filtering and Matching Process

1. When a frame with the RTR bit set to '1' is received, the automatic reply mechanism is engaged if:
 - The filtering fields (ID, RTR, IDE) match the configured registers.
 - The received frame passes through the message filtering section.
2. Comparison Process:
 - The received frame's ID, RTR, and IDE fields are compared with the Acceptance Code Register (ACR) and Acceptance Mask Register (AMR) of the RX buffer.
 - The filtering logic follows the standard message filtering process as documented elsewhere.

Example: For example, let us say there are two nodes, CAN0 (CoreCAN) and CAN1 in the system. In order to configure CAN0 RX buffer 0 to transmit response on the detection of the RTR frame following configurations are required.

- Configures CAN0 RxMessage0 AMR Register and RxMessage0 ACR Register to accept RTR frames.
- Set BUFFER_ENABLE and RTR_REPLY bit to 1 of RxMessage0 Control Register along with other bits as per the requirement.
- Program RxMessage0 ID Register, RxMessage0 Word1 Data Register and RxMessage0 Word0 Data Register with the response which should be transmitted upon detection of the RTR frame.
- Send RTR frame from CAN1 node as per the filtering criteria configured in the CAN0 node.
- Upon detection of the RTR frame, CAN0 node reads the RxMessage0 ID Register, RxMessage0 Word1 Register and RxMessage0 Word0 Register and transmit the frame to node CAN1.

This mechanism enables efficient, hardware-automated responses to remote transmission requests, eliminating the need for CPU intervention.

2. IP Core Parameters and Interface Signals [\(Ask a Question\)](#)

2.1. Configuration Settings [\(Ask a Question\)](#)

The following table lists configuration parameters for the CoreCAN IP.

Table 2-1. CoreCAN IP Configuration Parameters

Parameter	Valid Values	Default Value	Description
RESET_TYPE	Asynchronous Synchronous	Asynchronous	This Parameter is used to select the Reset type.

The following table lists the clock ports for Core. The Clock Domain column in the following table indicates the input ports and the output ports expected to be synchronized in the respective clock domain.

Table 2-2. Core Clock Signals and Description

Port Name	Width	Direction	Description
PCLK	1	Input	System clock. Baud clock for both transmitter and receiver will be generated using this clock. It is recommended to operate this clock at 40MHz frequency. For the Evaluation license, four hour functionality is supported on silicon considering PCLK frequency is 40MHz. If PCLK frequency is other than 40MHz then number of hours supported for Evaluation license changes as per the PCLK frequency

2.2. Inputs and Outputs Signals [\(Ask a Question\)](#)

The following table lists the input and output signals, including the clock ports.

Table 2-3. Input and Output Signals

Port Name	Width	Direction	Clock Domain	Description
Reset Ports				
PRESETN	1	Input	PCLK	Active-low reset. It should be synchronized in PCLK clock domain externally.
 Important: The PRESETN is either Synchronous or asynchronous based on RESET_TYPE Parameter.				
APB Interface Ports				
PSEL	1	Input	PCLK	Module select signal
PENABLE	1	Input	PCLK	Bus transfer enable signal
PADDR	11	Input	PCLK	Address Bus
PWRITE	1	Input	PCLK	Read/write signal <ul style="list-style-type: none"> • '0': Read operation • '1': Write operation
PWDATA	32	Input	PCLK	Write data bus
PRDATA	32	Output	PCLK	Read data bus
PREADY	1	Output	PCLK	Ready indicator
PSLVERR	1	Output	PCLK	Optional signal. Indicates an error condition on an APB transfer.
INT_N	1	Output	PCLK	Interrupt request, active low. The signal is asserted when both particular Interrupt Source bit and the corresponding interrupt enable bit are set. This is level sensitive.
CAN Bus Interface Ports				

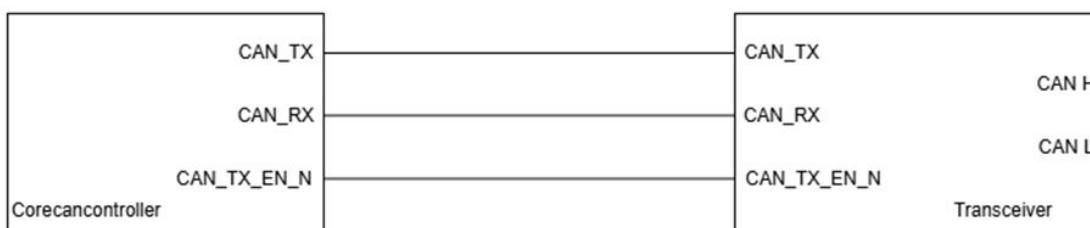
Table 2-3. Input and Output Signals (continued)

Port Name	Width	Direction	Clock Domain	Description
CAN_RX	1	Input	PCLK	Local receive signal (connect to can_rx_bus of external driver)
CAN_TX	1	Output	PCLK	CANbus transmit signal, connected to external driver
CAN_TX_EN_N	1	Output	PCLK	External driver control signal This is used to disable an external CAN transceiver <code>canbus_tx_ebl_n</code> is asserted when the CAN controller is stopped or if the CAN state is bus-off.

CANbus Transceiver Configuration

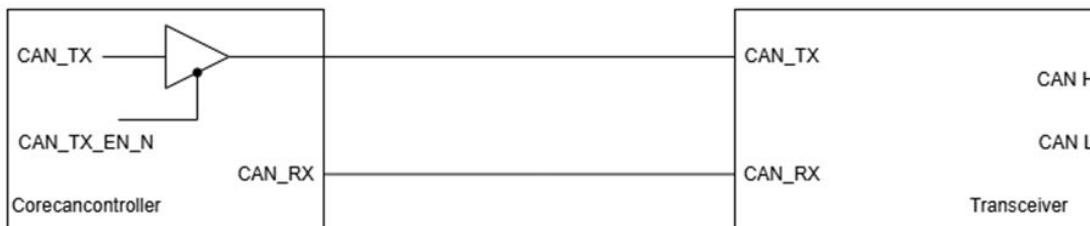
Standard CAN bus transceiver chips can be directly connected to the CAN interface pins. To reduce pin usage, a two-port configuration is also supported.

The following figure shows the 3 pin CAN bus interface.

Figure 2-1. 3 Pin CAN Bus Interface

3 Pin CAN Bus Interface

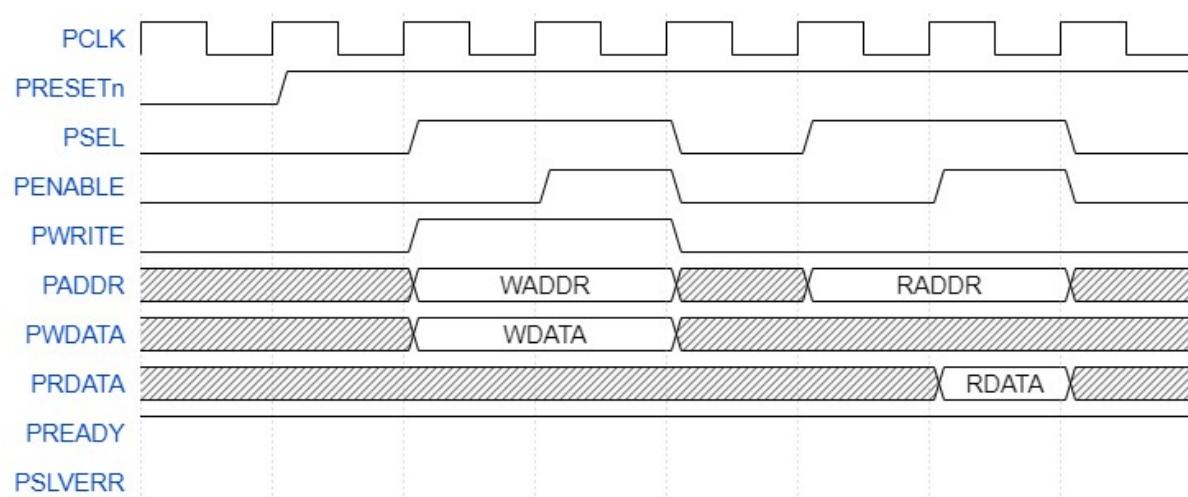
The following figure shows the 2 pin CAN bus interface.

Figure 2-2. 2 Pin CAN Bus Interface

2 Pin CAN Bus Interface

2.3. Timing Diagrams (Ask a Question)

The following figure shows the timing diagram for APB Interface.

Figure 2-3. APB Interface Timing Diagram

3. Implementation of CoreCAN IP in Libero Design Suite [\(Ask a Question\)](#)

The CoreCAN IP is implemented in the Libero Design Suite by following a series of steps that ensure proper integration and functionality within the design environment.

Steps for Integration

1. Open the Libero Design Suite and create a new project or open an existing project.
2. Import the CoreCAN IP into the project by selecting the appropriate IP catalog and adding it to the design.
3. Configure the CoreCAN IP parameters as required for the specific application.
4. Connect the CoreCAN IP to other design components using the design canvas and ensure proper signal routing.
5. Run design rule checks to verify the integration and resolve any issues that may arise.
6. Synthesize the design to generate the necessary netlist and other implementation files.
7. Perform place and route operations to map the design onto the target FPGA device.
8. Generate the bitstream file for programming the FPGA with the integrated CoreCAN IP.
9. Program the FPGA and verify the functionality of the CoreCAN IP within the overall design.



Important:

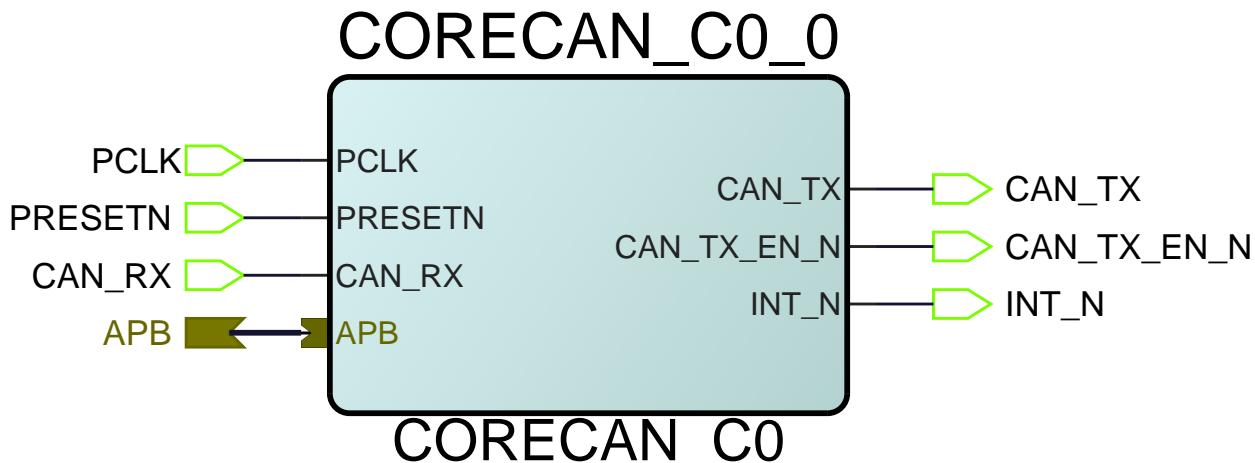
For information on using SmartDesign to instantiate, configure, connect, and generate cores, see the [Libero SOC online help](#).

3.1. Smart Design Integration [\(Ask a Question\)](#)

The CoreCAN IP is available in the SmartDesign IP Catalog. Users can easily locate and utilize this IP for their design needs.

The following figure shows an example of an instantiated view of the CoreCAN on the SmartDesign canvas.

Figure 3-1. Instantiation of CoreCAN on Smart Design Canvas



Steps for Instantiation, Configuration, and Core Generation in SmartDesign

1. Open the SmartDesign tool and navigate to the IP Catalog.
2. Search for the CoreCAN IP in the catalog.
3. Drag and drop the CoreCAN IP into your design canvas.
4. Double-click on the CoreCAN IP block to open the configuration window.
5. Configure the IP parameters as per your design requirements.
6. Click 'OK' to save the configuration settings.
7. Generate the core by clicking on the 'Generate' button in the SmartDesign toolbar.
8. Verify the generated core and integrate it into your overall design.

3.1.1. Configuration [\(Ask a Question\)](#)

This section describes how to configure the CoreCAN IP instance using the configuration GUI.

To configure the CoreCAN IP instance, perform the following steps:

1. Open the configuration GUI.
2. Select the CoreCAN IP instance from the list of available IPs.
3. Adjust the configuration settings as required.
4. Save the configuration.

Common Configuration Settings

The following are examples of common configuration settings for the CoreCAN IP instance:

- Baud Rate: Set the baud rate for CAN communication. Common values are 125 kbps, 250 kbps, 500 kbps, and 1 Mbps.
- Operating Mode: Select the operating mode for the controller, such as Normal, Listen-Only, or Loopback mode.
- Interrupts: Enable or disable interrupts for various events like message reception, transmission, and error conditions.
- Acceptance Filters: Configure acceptance filters to determine which CAN messages are processed by the controller.

3.1.2. Synthesis in Libero SoC [\(Ask a Question\)](#)

To run synthesis, perform the following steps:

1. With the configuration selected in the configuration GUI, set the design root appropriately.
2. Under **Implement Design**, on the **Design Flow** tab, right click **Synthesize** and click **Run**.

Step-by-step guide to synthesizing the CoreCAN IP

1. Open the Libero SoC software.
2. Create a new project or open an existing project.
3. Import the CoreCAN IP into your project.
4. Configure the CoreCAN IP according to your design requirements.
5. Run the synthesis tool to generate the synthesized netlist.
6. Review the synthesis report for any warnings or errors.
7. Make any necessary adjustments to the design and re-run the synthesis if needed.
8. Once the synthesis is successful, proceed to the implementation phase.

3.1.3. Place-and-Route in Libero SoC [\(Ask a Question\)](#)

To run the place and route, perform the following steps:

- With the configuration selected in the configuration GUI, set the design root appropriately.
- Under **Implement Design**, on the **Design Flow** tab, right click **Place and Route** and click **Run**.

Steps for Running Place-and-Route

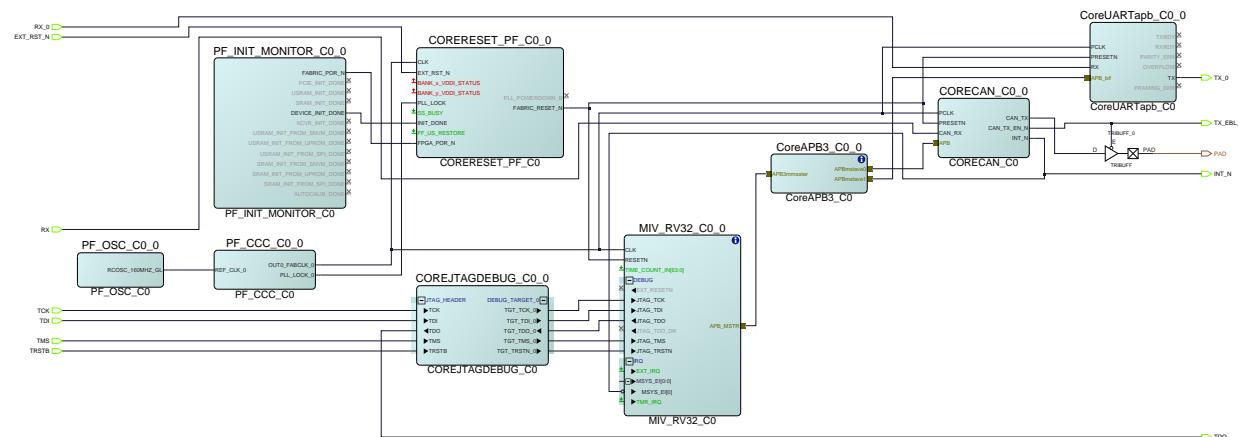
To run place-and-route in Libero SoC, perform the following steps:

- Open your project in Libero SoC.
- Ensure that the design is synthesized and optimized.
- Navigate to the Place-and-Route section in the design flow.
- Configure the place-and-route settings as required for your design.
- Run the place-and-route process.
- Review the place-and-route report for any issues or optimizations.
- Make any necessary adjustments and re-run the place-and-route if needed.

3.1.4. System Integration (Ask a Question)

This section provides example design along with detailed information to facilitate the integration of CoreCAN.

Figure 3-2. CoreCAN System Integration



This example design requires the following:

- CoreMIV_RV32 must be connected to the CoreCAN through CoreAPB3.
- CORERESET_PF must be used for all the resets.
- 40 MHz Clock must be used for all the logic in the design which is driven from PF_CCC.
- Top level port INT_N is connected to External System Interrupt port {MSYS_EI [0]} of CoreMIV_RV32 for the purpose of generating interrupt.
- CAN Controller is used to establish Communication with CAN Bus Network through CAN transceivers.
- CAN_TX_EN_N is an external driver control signal to drive the CoreCAN TX pin via TRIBUFF.
- CAN Controller TX and RX are connected to the CAN transceivers, output of the CAN Transceiver CAN_H and CAN_L are connected to the CAN Bus Network.
- Firmware application for CoreMIV_RV32 is used to configure registers of CoreCAN.

Execute the Libero flow with the timing-driven and high-effort place and route options enabled. This example design can be obtained from the Microchip technical support team.

4. Register Summary [\(Ask a Question\)](#)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	Interrupt Source Register	7:0	FORM_ERR	ACK_ERR	STUFF_ERR	BIT_ERR	OVR_LOAD	ARB_LOSS		
		15:8	OST_FAILURE	STUCK_AT_0	RTR_AUTO_M SG	RX_MSG	TX_MSG	RX_MSG_LOS S	BUS_OFF	CRC_ERR
		23:16								
		31:24								
0x04	Interrupt Enable Register	7:0	FORM_ERR_IE	ACK_ERR_IE	STUFF_ERR_IE	BIT_ERR_IE	OVR_LOAD_IE	ARB_LOSS_IE		INT_EBL
		15:8	OST_FAILURE _IE	STUCK_AT_0_I E	RTR_MSG_IE	RX_MSG_IE	TX_MSG_IE	RX_MSG_LOS S_IE	BUS_OFF_IE	CRC_ERR_IE
		23:16								
		31:24								
0x08	RxBuffer Status Register	7:0	RXBUFF7_STS	RXBUFF6_STS	RXBUFF5_STS	RXBUFF4_STS	RXBUFF3_STS	RXBUFF2_STS	RXBUFF1_STS	RXBUFF0_STS
		15:8	RXBUFF15_ST S	RXBUFF14_ST S	RXBUFF13_ST S	RXBUFF12_ST S	RXBUFF11_ST S	RXBUFF10_ST S	RXBUFF9_STS	RXBUFF8_STS
		23:16	RXBUFF23_ST S	RXBUFF22_ST S	RXBUFF21_ST S	RXBUFF20_ST S	RXBUFF19_ST S	RXBUFF18_ST S	RXBUFF17_ST S	RXBUFF16_ST S
		31:24	RXBUFF31_ST S	RXBUFF30_ST S	RXBUFF29_ST S	RXBUFF28_ST S	RXBUFF27_ST S	RXBUFF26_ST S	RXBUFF25_ST S	RXBUFF24_ST S
0x0C	TxBuffer Status Register	7:0	TXBUFF7_STS	TXBUFF6_STS	TXBUFF5_STS	TXBUFF4_STS	TXBUFF3_STS	TXBUFF2_STS	TXBUFF1_STS	TXBUFF0_STS
		15:8	TXBUFF15_ST S	TXBUFF14_ST S	TXBUFF13_ST S	TXBUFF12_ST S	TXBUFF11_ST S	TXBUFF10_ST S	TXBUFF9_STS	TXBUFF8_STS
		23:16	TXBUFF23_ST S	TXBUFF22_ST S	TXBUFF21_ST S	TXBUFF20_ST S	TXBUFF19_ST S	TXBUFF18_ST S	TXBUFF17_ST S	TXBUFF16_ST S
		31:24	TXBUFF31_ST S	TXBUFF30_ST S	TXBUFF29_ST S	TXBUFF28_ST S	TXBUFF27_ST S	TXBUFF26_ST S	TXBUFF25_ST S	TXBUFF24_ST S
0x10	Error Status Register	7:0								
		15:8								
		23:16						RXGTE96	TXGTE96	ERROR_STATE[1:0]
		31:24								
0x14	Command Register	7:0							LPBK_MODE	LISTEN_MODE
		15:8								RUN_STOP_MODE
		23:16								
		31:24								
0x18	Configurator Register	7:0								
		15:8								
		23:16								
		31:24								
0x1C ... 0x1F	Reserved									
0x20	TxMessage0 Control Register	7:0						TXINTR_WPN	TX_INTEBL	TX_ABORT
		15:8								
		23:16	WPN		RTR	IDE			DLC[3:0]	
		31:24								
0x24	TxMessage0 ID Register	7:0				TX_ID[4:0]				
		15:8								
		23:16								
		31:24								
0x28	TxMessage0 Word1 Data Register	7:0						WORD1_TXDATA[7:0]		
		15:8						WORD1_TXDATA[15:8]		
		23:16						WORD1_TXDATA[23:16]		
		31:24						WORD1_TXDATA[31:24]		
0x2C	TxMessage0 Word0 Data Register	7:0						WORD0_TXDATA[7:0]		
		15:8						WORD0_TXDATA[15:8]		
		23:16						WORD0_TXDATA[23:16]		
		31:24						WORD0_TXDATA[31:24]		

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x30	TxMessage1 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x34	TxMessage1 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x38	TxMessage1 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				
0x3C	TxMessage1 Word0 Data Register	7:0				WORD0_TXDATA[7:0]				
		15:8				WORD0_TXDATA[15:8]				
		23:16				WORD0_TXDATA[23:16]				
		31:24				WORD0_TXDATA[31:24]				
0x40	TxMessage2 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x44	TxMessage2 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x48	TxMessage2 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				
0x4C	TxMessage2 Word0 Data Register	7:0				WORD0_TXDATA[7:0]				
		15:8				WORD0_TXDATA[15:8]				
		23:16				WORD0_TXDATA[23:16]				
		31:24				WORD0_TXDATA[31:24]				
0x50	TxMessage3 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x54	TxMessage3 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x58	TxMessage3 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				
0x5C	TxMessage3 Word0 Data Register	7:0				WORD0_TXDATA[7:0]				
		15:8				WORD0_TXDATA[15:8]				
		23:16				WORD0_TXDATA[23:16]				
		31:24				WORD0_TXDATA[31:24]				
0x60	TxMessage4 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x64	TxMessage4 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x68	TxMessage4 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x6C	TxMessage4 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x70	TxMessage5 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x74	TxMessage5 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x78	TxMessage5 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x7C	TxMessage5 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x80	TxMessage6 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x84	TxMessage6 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x88	TxMessage6 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x8C	TxMessage6 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x90	TxMessage7 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x94	TxMessage7 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x98	TxMessage7 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x9C	TxMessage7 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0xA0	TxMessage8 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0xA4	TxMessage8 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xA8	TxMessage8 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0xAC	TxMessage8 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0xB0	TxMessage9 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0xB4	TxMessage9 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0xB8	TxMessage9 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0xBC	TxMessage9 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0xC0	TxMessage10 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0xC4	TxMessage10 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0xC8	TxMessage10 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0xCC	TxMessage10 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0xD0	TxMessage11 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0xD4	TxMessage11 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0xD8	TxMessage11 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0xDC	TxMessage11 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0xE0	TxMessage12 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE4	TxMessage12 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0xE8	TxMessage12 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0xEC	TxMessage12 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					
0xF0	TxMessage13 Control Register	7:0				TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ	
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0xF4	TxMessage13 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0xF8	TxMessage13 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0xFC	TxMessage13 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					
0x0100	TxMessage14 Control Register	7:0				TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ	
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0104	TxMessage14 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x0108	TxMessage14 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0x010C	TxMessage14 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					
0x0110	TxMessage15 Control Register	7:0				TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ	
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0114	TxMessage15 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x0118	TxMessage15 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0x011C	TxMessage15 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0120	TxMessage16 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0124	TxMessage16 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x0128	TxMessage16 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				
0x012C	TxMessage16 Word0 Data Register	7:0				WORD0_TXDATA[7:0]				
		15:8				WORD0_TXDATA[15:8]				
		23:16				WORD0_TXDATA[23:16]				
		31:24				WORD0_TXDATA[31:24]				
0x0130	TxMessage17 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0134	TxMessage17 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x0138	TxMessage17 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				
0x013C	TxMessage17 Word0 Data Register	7:0				WORD0_TXDATA[7:0]				
		15:8				WORD0_TXDATA[15:8]				
		23:16				WORD0_TXDATA[23:16]				
		31:24				WORD0_TXDATA[31:24]				
0x0140	TxMessage18 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0144	TxMessage18 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x0148	TxMessage18 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				
0x014C	TxMessage18 Word0 Data Register	7:0				WORD0_TXDATA[7:0]				
		15:8				WORD0_TXDATA[15:8]				
		23:16				WORD0_TXDATA[23:16]				
		31:24				WORD0_TXDATA[31:24]				
0x0150	TxMessage19 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0154	TxMessage19 ID Register	7:0		TX_ID[4:0]						
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x0158	TxMessage19 Word1 Data Register	7:0				WORD1_TXDATA[7:0]				
		15:8				WORD1_TXDATA[15:8]				
		23:16				WORD1_TXDATA[23:16]				
		31:24				WORD1_TXDATA[31:24]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x015C	TxMessage19 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x0160	TxMessage20 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0164	TxMessage20 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x0168	TxMessage20 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x016C	TxMessage20 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x0170	TxMessage21 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0174	TxMessage21 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x0178	TxMessage21 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x017C	TxMessage21 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x0180	TxMessage22 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0184	TxMessage22 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x0188	TxMessage22 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x018C	TxMessage22 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x0190	TxMessage23 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0194	TxMessage23 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0198	TxMessage23 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x019C	TxMessage23 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x01A0	TxMessage24 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x01A4	TxMessage24 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x01A8	TxMessage24 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x01AC	TxMessage24 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x01B0	TxMessage25 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x01B4	TxMessage25 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x01B8	TxMessage25 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x01BC	TxMessage25 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x01C0	TxMessage26 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x01C4	TxMessage26 ID Register	7:0			TX_ID[4:0]					
		15:8					TX_ID[12:5]			
		23:16					TX_ID[20:13]			
		31:24					TX_ID[28:21]			
0x01C8	TxMessage26 Word1 Data Register	7:0					WORD1_TXDATA[7:0]			
		15:8					WORD1_TXDATA[15:8]			
		23:16					WORD1_TXDATA[23:16]			
		31:24					WORD1_TXDATA[31:24]			
0x01CC	TxMessage26 Word0 Data Register	7:0					WORD0_TXDATA[7:0]			
		15:8					WORD0_TXDATA[15:8]			
		23:16					WORD0_TXDATA[23:16]			
		31:24					WORD0_TXDATA[31:24]			
0x01D0	TxMessage27 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01D4	TxMessage27 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x01D8	TxMessage27 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0x01DC	TxMessage27 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					
0x01E0	TxMessage28 Control Register	7:0				TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ	
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x01E4	TxMessage28 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x01E8	TxMessage28 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0x01EC	TxMessage28 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					
0x01F0	TxMessage29 Control Register	7:0				TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ	
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x01F4	TxMessage29 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x01F8	TxMessage29 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0x01FC	TxMessage29 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					
0x0200	TxMessage30 Control Register	7:0				TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ	
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0204	TxMessage30 ID Register	7:0			TX_ID[4:0]					
		15:8				TX_ID[12:5]				
		23:16				TX_ID[20:13]				
		31:24				TX_ID[28:21]				
0x0208	TxMessage30 Word1 Data Register	7:0			WORD1_TXDATA[7:0]					
		15:8			WORD1_TXDATA[15:8]					
		23:16			WORD1_TXDATA[23:16]					
		31:24			WORD1_TXDATA[31:24]					
0x020C	TxMessage30 Word0 Data Register	7:0			WORD0_TXDATA[7:0]					
		15:8			WORD0_TXDATA[15:8]					
		23:16			WORD0_TXDATA[23:16]					
		31:24			WORD0_TXDATA[31:24]					

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0210	TxMessage31 Control Register	7:0					TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
		15:8								
		23:16	WPN		RTR	IDE		DLC[3:0]		
		31:24								
0x0214	TxMessage31 ID Register	7:0		TX_ID[4:0]						
		15:8		TX_ID[12:5]						
		23:16		TX_ID[20:13]						
		31:24		TX_ID[28:21]						
0x0218	TxMessage31 Word1 Data Register	7:0		WORD1_TXDATA[7:0]						
		15:8		WORD1_TXDATA[15:8]						
		23:16		WORD1_TXDATA[23:16]						
		31:24		WORD1_TXDATA[31:24]						
0x021C	TxMessage31 Word0 Data Register	7:0		WORD0_TXDATA[7:0]						
		15:8		WORD0_TXDATA[15:8]						
		23:16		WORD0_TXDATA[23:16]						
		31:24		WORD0_TXDATA[31:24]						
0x0220	RxMessage0 Control Register	7:0	LINK_FLAG	RX_INTEBL	RTR_REPL0	BUFFER_ENA BLE	RTRABORT	RTREPL0_PEN DING	MSGAV	
		15:8								
		23:16		RTR	IDE		DLC[3:0]			
		31:24								
0x0224	RxMessage0 ID Register	7:0	RX_ID[4:0]							
		15:8		RX_ID[12:5]						
		23:16		RX_ID[20:13]						
		31:24		RX_ID[28:21]						
0x0228	RxMessage0 Word1 Data Register	7:0		WORD1_RXDATA[7:0]						
		15:8		WORD1_RXDATA[15:8]						
		23:16		WORD1_RXDATA[23:16]						
		31:24		WORD1_RXDATA[31:24]						
0x022C	RxMessage0 Word0 Data Register	7:0		WORD0_RXDATA[7:0]						
		15:8		WORD0_RXDATA[15:8]						
		23:16		WORD0_RXDATA[23:16]						
		31:24		WORD0_RXDATA[31:24]						
0x0230	RxMessage0 AMR Register	7:0	ACPT_MASK_ID[4:0]				ACPT_MASK_I DE	ACPT_MASK_RTR		
		15:8		ACPT_MASK_ID[12:5]						
		23:16		ACPT_MASK_ID[20:13]						
		31:24		ACPT_MASK_ID[28:21]						
0x0234	RxMessage0 ACR Register	7:0	ACPT_CODE_ID[4:0]				ACPT_CODE_I DE	ACPT_CODE_RTR		
		15:8		ACPT_CODE_ID[12:5]						
		23:16		ACPT_CODE_ID[20:13]						
		31:24		ACPT_CODE_ID[28:21]						
0x0238	RxMessage0 AMR Data Register	7:0		ACPT_MASK_BOTE2[7:0]						
		15:8		ACPT_MASK_BOTE1[7:0]						
		23:16								
		31:24								
0x023C	RxMessage0 ACR Data Register	7:0	ACPT_CODE_BOTE2[7:0]							
		15:8		ACPT_CODE_BOTE1[7:0]						
		23:16								
		31:24								
0x0240	RxMessage1 Control Register	7:0	LINK_FLAG	RX_INTEBL	RTR_REPL1	BUFFER_ENA BLE	RTRABORT	RTREPL1_PEN DING	MSGAV	
		15:8								
		23:16		RTR	IDE		DLC[3:0]			
		31:24								
0x0244	RxMessage1 ID Register	7:0	RX_ID[4:0]							
		15:8		RX_ID[12:5]						
		23:16		RX_ID[20:13]						
		31:24		RX_ID[28:21]						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0248	RxMessage1 Word1 Data Register	7:0					WORD1_RXDATA[7:0]			
		15:8					WORD1_RXDATA[15:8]			
		23:16					WORD1_RXDATA[23:16]			
		31:24					WORD1_RXDATA[31:24]			
0x024C	RxMessage1 Word0 Data Register	7:0					WORD0_RXDATA[7:0]			
		15:8					WORD0_RXDATA[15:8]			
		23:16					WORD0_RXDATA[23:16]			
		31:24					WORD0_RXDATA[31:24]			
0x0250	RxMessage1 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR
		15:8					ACPT_MASK_ID[12:5]			
		23:16					ACPT_MASK_ID[20:13]			
		31:24					ACPT_MASK_ID[28:21]			
0x0254	RxMessage1 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR
		15:8					ACPT_CODE_ID[12:5]			
		23:16					ACPT_CODE_ID[20:13]			
		31:24					ACPT_CODE_ID[28:21]			
0x0258	RxMessage1 AMR Data Register	7:0					ACPT_MASK_B1TE2[7:0]			
		15:8					ACPT_MASK_B1TE1[7:0]			
		23:16								
		31:24								
0x025C	RxMessage1 ACR Data Register	7:0					ACPT_CODE_B1TE2[7:0]			
		15:8					ACPT_CODE_B1TE1[7:0]			
		23:16								
		31:24								
0x0260	RxMessage2 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL2	BUFFER_ENABLE	RTRABORT	RTREPL2_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0264	RxMessage2 ID Register	7:0			RX_ID[4:0]					
		15:8					RX_ID[12:5]			
		23:16					RX_ID[20:13]			
		31:24					RX_ID[28:21]			
0x0268	RxMessage2 Word1 Data Register	7:0					WORD1_RXDATA[7:0]			
		15:8					WORD1_RXDATA[15:8]			
		23:16					WORD1_RXDATA[23:16]			
		31:24					WORD1_RXDATA[31:24]			
0x026C	RxMessage2 Word0 Data Register	7:0					WORD0_RXDATA[7:0]			
		15:8					WORD0_RXDATA[15:8]			
		23:16					WORD0_RXDATA[23:16]			
		31:24					WORD0_RXDATA[31:24]			
0x0270	RxMessage2 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR
		15:8					ACPT_MASK_ID[12:5]			
		23:16					ACPT_MASK_ID[20:13]			
		31:24					ACPT_MASK_ID[28:21]			
0x0274	RxMessage2 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR
		15:8					ACPT_CODE_ID[12:5]			
		23:16					ACPT_CODE_ID[20:13]			
		31:24					ACPT_CODE_ID[28:21]			
0x0278	RxMessage2 AMR Data Register	7:0					ACPT_MASK_B2TE2[7:0]			
		15:8					ACPT_MASK_B2TE1[7:0]			
		23:16								
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x027C	RxMessage2 ACR Data Register	7:0					ACPT_CODE_B2TE2[7:0]			
		15:8					ACPT_CODE_B2TE1[7:0]			
		23:16								
		31:24								
0x0280	RxMessage3 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL3	BUFFER_ENABLE	RTRABORT	RTREPL3_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0284	RxMessage3 ID Register	7:0	RX_ID[4:0]							
		15:8	RX_ID[12:5]							
		23:16	RX_ID[20:13]							
		31:24	RX_ID[28:21]							
0x0288	RxMessage3 Word1 Data Register	7:0	WORD1_RXDATA[7:0]							
		15:8	WORD1_RXDATA[15:8]							
		23:16	WORD1_RXDATA[23:16]							
		31:24	WORD1_RXDATA[31:24]							
0x028C	RxMessage3 Word0 Data Register	7:0	WORD0_RXDATA[7:0]							
		15:8	WORD0_RXDATA[15:8]							
		23:16	WORD0_RXDATA[23:16]							
		31:24	WORD0_RXDATA[31:24]							
0x0290	RxMessage3 AMR Register	7:0	ACPT_MASK_ID[4:0]					ACPT_MASK_I	ACPT_MASK_DE	
		15:8	ACPT_MASK_ID[12:5]							
		23:16	ACPT_MASK_ID[20:13]							
		31:24	ACPT_MASK_ID[28:21]							
0x0294	RxMessage3 ACR Register	7:0	ACPT_CODE_ID[4:0]					ACPT_CODE_I	ACPT_CODE_DE	
		15:8	ACPT_CODE_ID[12:5]							
		23:16	ACPT_CODE_ID[20:13]							
		31:24	ACPT_CODE_ID[28:21]							
0x0298	RxMessage3 AMR Data Register	7:0	ACPT_MASK_B3TE2[7:0]							
		15:8	ACPT_MASK_B3TE1[7:0]							
		23:16								
		31:24								
0x029C	RxMessage3 ACR Data Register	7:0	ACPT_CODE_B3TE2[7:0]							
		15:8	ACPT_CODE_B3TE1[7:0]							
		23:16								
		31:24								
0x02A0	RxMessage4 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL4	BUFFER_ENABLE	RTRABORT	RTREPL4_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x02A4	RxMessage4 ID Register	7:0	RX_ID[4:0]							
		15:8	RX_ID[12:5]							
		23:16	RX_ID[20:13]							
		31:24	RX_ID[28:21]							
0x02A8	RxMessage4 Word1 Data Register	7:0	WORD1_RXDATA[7:0]							
		15:8	WORD1_RXDATA[15:8]							
		23:16	WORD1_RXDATA[23:16]							
		31:24	WORD1_RXDATA[31:24]							
0x02AC	RxMessage4 Word0 Data Register	7:0	WORD0_RXDATA[7:0]							
		15:8	WORD0_RXDATA[15:8]							
		23:16	WORD0_RXDATA[23:16]							
		31:24	WORD0_RXDATA[31:24]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02B0	RxMessage4 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x02B4	RxMessage4 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x02B8	RxMessage4 AMR Data Register	7:0				ACPT_MASK_B4TE2[7:0]				
		15:8				ACPT_MASK_B4TE1[7:0]				
		23:16								
		31:24								
0x02BC	RxMessage4 ACR Data Register	7:0				ACPT_CODE_B4TE2[7:0]				
		15:8				ACPT_CODE_B4TE1[7:0]				
		23:16								
		31:24								
0x02C0	RxMessage5 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL5	BUFFER_ENABLE	RTRABORT	RTREPL5_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x02C4	RxMessage5 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x02C8	RxMessage5 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x02CC	RxMessage5 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x02D0	RxMessage5 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x02D4	RxMessage5 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x02D8	RxMessage5 AMR Data Register	7:0				ACPT_MASK_B5TE2[7:0]				
		15:8				ACPT_MASK_B5TE1[7:0]				
		23:16								
		31:24								
0x02DC	RxMessage5 ACR Data Register	7:0				ACPT_CODE_B5TE2[7:0]				
		15:8				ACPT_CODE_B5TE1[7:0]				
		23:16								
		31:24								
0x02E0	RxMessage6 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL6	BUFFER_ENABLE	RTRABORT	RTREPL6_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02E4	RxMessage6 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x02E8	RxMessage6 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x02EC	RxMessage6 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x02F0	RxMessage6 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR
		15:8						ACPT_MASK_ID[12:5]		
		23:16						ACPT_MASK_ID[20:13]		
		31:24						ACPT_MASK_ID[28:21]		
0x02F4	RxMessage6 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR
		15:8						ACPT_CODE_ID[12:5]		
		23:16						ACPT_CODE_ID[20:13]		
		31:24						ACPT_CODE_ID[28:21]		
0x02F8	RxMessage6 AMR Data Register	7:0			ACPT_MASK_B6TE2[7:0]					
		15:8			ACPT_MASK_B6TE1[7:0]					
		23:16								
		31:24								
0x02FC	RxMessage6 ACR Data Register	7:0			ACPT_CODE_B6TE2[7:0]					
		15:8			ACPT_CODE_B6TE1[7:0]					
		23:16								
		31:24								
0x0300	RxMessage7 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL7	BUFFER_ENABLE	RTRABORT	RTREPL7_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0304	RxMessage7 ID Register	7:0			RX_ID[4:0]					
		15:8						RX_ID[12:5]		
		23:16						RX_ID[20:13]		
		31:24						RX_ID[28:21]		
0x0308	RxMessage7 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x030C	RxMessage7 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x0310	RxMessage7 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR
		15:8						ACPT_MASK_ID[12:5]		
		23:16						ACPT_MASK_ID[20:13]		
		31:24						ACPT_MASK_ID[28:21]		
0x0314	RxMessage7 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR
		15:8						ACPT_CODE_ID[12:5]		
		23:16						ACPT_CODE_ID[20:13]		
		31:24						ACPT_CODE_ID[28:21]		

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0318	RxMessage7 AMR Data Register	7:0					ACPT_MASK_B7TE2[7:0]			
		15:8					ACPT_MASK_B7TE1[7:0]			
		23:16								
		31:24								
0x031C	RxMessage7 ACR Data Register	7:0					ACPT_CODE_B7TE2[7:0]			
		15:8					ACPT_CODE_B7TE1[7:0]			
		23:16								
		31:24								
0x0320	RxMessage8 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL8	BUFFER_ENA BLE	RTRABORT	RTREPL8_PEN DING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0324	RxMessage8 ID Register	7:0			RX_ID[4:0]					
		15:8					RX_ID[12:5]			
		23:16					RX_ID[20:13]			
		31:24					RX_ID[28:21]			
0x0328	RxMessage8 Word1 Data Register	7:0					WORD1_RXDATA[7:0]			
		15:8					WORD1_RXDATA[15:8]			
		23:16					WORD1_RXDATA[23:16]			
		31:24					WORD1_RXDATA[31:24]			
0x032C	RxMessage8 Word0 Data Register	7:0					WORD0_RXDATA[7:0]			
		15:8					WORD0_RXDATA[15:8]			
		23:16					WORD0_RXDATA[23:16]			
		31:24					WORD0_RXDATA[31:24]			
0x0330	RxMessage8 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_ DE	ACPT_MASK_ RTR	
		15:8					ACPT_MASK_ID[12:5]			
		23:16					ACPT_MASK_ID[20:13]			
		31:24					ACPT_MASK_ID[28:21]			
0x0334	RxMessage8 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_ DE	ACPT_CODE_ RTR	
		15:8					ACPT_CODE_ID[12:5]			
		23:16					ACPT_CODE_ID[20:13]			
		31:24					ACPT_CODE_ID[28:21]			
0x0338	RxMessage8 AMR Data Register	7:0					ACPT_MASK_B8TE2[7:0]			
		15:8					ACPT_MASK_B8TE1[7:0]			
		23:16								
		31:24								
0x033C	RxMessage8 ACR Data Register	7:0					ACPT_CODE_B8TE2[7:0]			
		15:8					ACPT_CODE_B8TE1[7:0]			
		23:16								
		31:24								
0x0340	RxMessage9 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL9	BUFFER_ENA BLE	RTRABORT	RTREPL9_PEN DING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0344	RxMessage9 ID Register	7:0			RX_ID[4:0]					
		15:8					RX_ID[12:5]			
		23:16					RX_ID[20:13]			
		31:24					RX_ID[28:21]			
0x0348	RxMessage9 Word1 Data Register	7:0					WORD1_RXDATA[7:0]			
		15:8					WORD1_RXDATA[15:8]			
		23:16					WORD1_RXDATA[23:16]			
		31:24					WORD1_RXDATA[31:24]			
0x034C	RxMessage9 Word0 Data Register	7:0					WORD0_RXDATA[7:0]			
		15:8					WORD0_RXDATA[15:8]			
		23:16					WORD0_RXDATA[23:16]			
		31:24					WORD0_RXDATA[31:24]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0350	RxMessage9 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_ RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0354	RxMessage9 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_ RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0358	RxMessage9 AMR Data Register	7:0				ACPT_MASK_B9TE2[7:0]				
		15:8				ACPT_MASK_B9TE1[7:0]				
		23:16								
		31:24								
0x035C	RxMessage9 ACR Data Register	7:0				ACPT_CODE_B9TE2[7:0]				
		15:8				ACPT_CODE_B9TE1[7:0]				
		23:16								
		31:24								
0x0360	RxMessage10 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL10	BUFFER_ENA BLE	RTRABORT	RTREPL10_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0364	RxMessage10 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0368	RxMessage10 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x036C	RxMessage10 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x0370	RxMessage10 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_ RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0374	RxMessage10 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_ RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0378	RxMessage10 AMR Data Register	7:0				ACPT_MASK_B10TE2[7:0]				
		15:8				ACPT_MASK_B10TE1[7:0]				
		23:16								
		31:24								
0x037C	RxMessage10 ACR Data Register	7:0				ACPT_CODE_B10TE2[7:0]				
		15:8				ACPT_CODE_B10TE1[7:0]				
		23:16								
		31:24								
0x0380	RxMessage11 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL11	BUFFER_ENA BLE	RTRABORT	RTREPL11_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0384	RxMessage11 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0388	RxMessage11 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x038C	RxMessage11 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x0390	RxMessage11 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0394	RxMessage11 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0398	RxMessage11 AMR Data Register	7:0			ACPT_MASK_B11TE2[7:0]					
		15:8			ACPT_MASK_B11TE1[7:0]					
		23:16								
		31:24								
0x039C	RxMessage11 ACR Data Register	7:0			ACPT_CODE_B11TE2[7:0]					
		15:8			ACPT_CODE_B11TE1[7:0]					
		23:16								
		31:24								
0x03A0	RxMessage12 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL12	BUFFER_ENABLE	RTRABORT	RTREPL12_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x03A4	RxMessage12 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x03A8	RxMessage12 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x03AC	RxMessage12 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x03B0	RxMessage12 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x03B4	RxMessage12 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03B8	RxMessage12 AMR Data Register	7:0				ACPT_MASK_B12TE2[7:0]				
		15:8				ACPT_MASK_B12TE1[7:0]				
		23:16								
		31:24								
0x03BC	RxMessage12 ACR Data Register	7:0				ACPT_CODE_B12TE2[7:0]				
		15:8				ACPT_CODE_B12TE1[7:0]				
		23:16								
		31:24								
0x03C0	RxMessage13 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL13	BUFFER_ENA BLE	RTRABORT	RTREPL13_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x03C4	RxMessage13 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x03C8	RxMessage13 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x03CC	RxMessage13 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x03D0	RxMessage13 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x03D4	RxMessage13 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x03D8	RxMessage13 AMR Data Register	7:0				ACPT_MASK_B13TE2[7:0]				
		15:8				ACPT_MASK_B13TE1[7:0]				
		23:16								
		31:24								
0x03DC	RxMessage13 ACR Data Register	7:0				ACPT_CODE_B13TE2[7:0]				
		15:8				ACPT_CODE_B13TE1[7:0]				
		23:16								
		31:24								
0x03E0	RxMessage14 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL14	BUFFER_ENA BLE	RTRABORT	RTREPL14_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x03E4	RxMessage14 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x03E8	RxMessage14 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x03EC	RxMessage14 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03F0	RxMessage14 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x03F4	RxMessage14 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x03F8	RxMessage14 AMR Data Register	7:0				ACPT_MASK_B14TE2[7:0]				
		15:8				ACPT_MASK_B14TE1[7:0]				
		23:16								
		31:24								
0x03FC	RxMessage14 ACR Data Register	7:0				ACPT_CODE_B14TE2[7:0]				
		15:8				ACPT_CODE_B14TE1[7:0]				
		23:16								
		31:24								
0x0400	RxMessage15 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL15	BUFFER_ENA BLE	RTRABORT	RTREPL15_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0404	RxMessage15 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0408	RxMessage15 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x040C	RxMessage15 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x0410	RxMessage15 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0414	RxMessage15 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0418	RxMessage15 AMR Data Register	7:0				ACPT_MASK_B15TE2[7:0]				
		15:8				ACPT_MASK_B15TE1[7:0]				
		23:16								
		31:24								
0x041C	RxMessage15 ACR Data Register	7:0				ACPT_CODE_B15TE2[7:0]				
		15:8				ACPT_CODE_B15TE1[7:0]				
		23:16								
		31:24								
0x0420	RxMessage16 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL16	BUFFER_ENA BLE	RTRABORT	RTREPL16_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0424	RxMessage16 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0428	RxMessage16 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x042C	RxMessage16 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x0430	RxMessage16 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0434	RxMessage16 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0438	RxMessage16 AMR Data Register	7:0			ACPT_MASK_B16TE2[7:0]					
		15:8			ACPT_MASK_B16TE1[7:0]					
		23:16								
		31:24								
0x043C	RxMessage16 ACR Data Register	7:0			ACPT_CODE_B16TE2[7:0]					
		15:8			ACPT_CODE_B16TE1[7:0]					
		23:16								
		31:24								
0x0440	RxMessage17 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL17	BUFFER_ENABLE	RTRABORT	RTREPL17_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0444	RxMessage17 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0448	RxMessage17 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x044C	RxMessage17 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x0450	RxMessage17 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0454	RxMessage17 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0458	RxMessage17 AMR Data Register	7:0				ACPT_MASK_B17TE2[7:0]				
		15:8				ACPT_MASK_B17TE1[7:0]				
		23:16								
		31:24								
0x045C	RxMessage17 ACR Data Register	7:0				ACPT_CODE_B17TE2[7:0]				
		15:8				ACPT_CODE_B17TE1[7:0]				
		23:16								
		31:24								
0x0460	RxMessage18 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL18	BUFFER_ENABLE	RTRABORT	RTREPL18_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0464	RxMessage18 ID Register	7:0			RX_ID[4:0]					
		15:8					RX_ID[12:5]			
		23:16					RX_ID[20:13]			
		31:24					RX_ID[28:21]			
0x0468	RxMessage18 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x046C	RxMessage18 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x0470	RxMessage18 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_ID	ACPT_MASK_RTR		
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0474	RxMessage18 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_ID	ACPT_CODE_RTR		
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0478	RxMessage18 AMR Data Register	7:0			ACPT_MASK_B18TE2[7:0]					
		15:8			ACPT_MASK_B18TE1[7:0]					
		23:16								
		31:24								
0x047C	RxMessage18 ACR Data Register	7:0			ACPT_CODE_B18TE2[7:0]					
		15:8			ACPT_CODE_B18TE1[7:0]					
		23:16								
		31:24								
0x0480	RxMessage19 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL19	BUFFER_ENABLE	RTRABORT	RTREPL19_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0484	RxMessage19 ID Register	7:0			RX_ID[4:0]					
		15:8					RX_ID[12:5]			
		23:16					RX_ID[20:13]			
		31:24					RX_ID[28:21]			
0x0488	RxMessage19 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x048C	RxMessage19 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0490	RxMessage19 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0494	RxMessage19 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0498	RxMessage19 AMR Data Register	7:0				ACPT_MASK_B19TE2[7:0]				
		15:8				ACPT_MASK_B19TE1[7:0]				
		23:16								
		31:24								
0x049C	RxMessage19 ACR Data Register	7:0				ACPT_CODE_B19TE2[7:0]				
		15:8				ACPT_CODE_B19TE1[7:0]				
		23:16								
		31:24								
0x04A0	RxMessage20 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL20	BUFFER_ENA BLE	RTRABORT	RTREPL20_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x04A4	RxMessage20 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x04A8	RxMessage20 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x04AC	RxMessage20 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x04B0	RxMessage20 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x04B4	RxMessage20 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x04B8	RxMessage20 AMR Data Register	7:0				ACPT_MASK_B20TE2[7:0]				
		15:8				ACPT_MASK_B20TE1[7:0]				
		23:16								
		31:24								
0x04BC	RxMessage20 ACR Data Register	7:0				ACPT_CODE_B20TE2[7:0]				
		15:8				ACPT_CODE_B20TE1[7:0]				
		23:16								
		31:24								
0x04C0	RxMessage21 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL21	BUFFER_ENA BLE	RTRABORT	RTREPL21_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04C4	RxMessage21 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x04C8	RxMessage21 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x04CC	RxMessage21 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x04D0	RxMessage21 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR
		15:8						ACPT_MASK_ID[12:5]		
		23:16						ACPT_MASK_ID[20:13]		
		31:24						ACPT_MASK_ID[28:21]		
0x04D4	RxMessage21 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR
		15:8						ACPT_CODE_ID[12:5]		
		23:16						ACPT_CODE_ID[20:13]		
		31:24						ACPT_CODE_ID[28:21]		
0x04D8	RxMessage21 AMR Data Register	7:0			ACPT_MASK_B21TE2[7:0]					
		15:8			ACPT_MASK_B21TE1[7:0]					
		23:16								
		31:24								
0x04DC	RxMessage21 ACR Data Register	7:0			ACPT_CODE_B21TE2[7:0]					
		15:8			ACPT_CODE_B21TE1[7:0]					
		23:16								
		31:24								
0x04E0	RxMessage22 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL22	BUFFER_ENABLE	RTRABORT	RTREPL22_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x04E4	RxMessage22 ID Register	7:0			RX_ID[4:0]					
		15:8						RX_ID[12:5]		
		23:16						RX_ID[20:13]		
		31:24						RX_ID[28:21]		
0x04E8	RxMessage22 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x04EC	RxMessage22 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x04F0	RxMessage22 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR
		15:8						ACPT_MASK_ID[12:5]		
		23:16						ACPT_MASK_ID[20:13]		
		31:24						ACPT_MASK_ID[28:21]		
0x04F4	RxMessage22 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR
		15:8						ACPT_CODE_ID[12:5]		
		23:16						ACPT_CODE_ID[20:13]		
		31:24						ACPT_CODE_ID[28:21]		

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04F8	RxMessage22 AMR Data Register	7:0				ACPT_MASK_B22TE2[7:0]				
		15:8				ACPT_MASK_B22TE1[7:0]				
		23:16								
		31:24								
0x04FC	RxMessage22 ACR Data Register	7:0				ACPT_CODE_B22TE2[7:0]				
		15:8				ACPT_CODE_B22TE1[7:0]				
		23:16								
		31:24								
0x0500	RxMessage23 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL23	BUFFER_ENA BLE	RTRABORT	RTREPL23_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0504	RxMessage23 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0508	RxMessage23 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x050C	RxMessage23 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x0510	RxMessage23 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE		
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0514	RxMessage23 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE		
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0518	RxMessage23 AMR Data Register	7:0			ACPT_MASK_B23TE2[7:0]					
		15:8			ACPT_MASK_B23TE1[7:0]					
		23:16								
		31:24								
0x051C	RxMessage23 ACR Data Register	7:0			ACPT_CODE_B23TE2[7:0]					
		15:8			ACPT_CODE_B23TE1[7:0]					
		23:16								
		31:24								
0x0520	RxMessage24 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL24	BUFFER_ENA BLE	RTRABORT	RTREPL24_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0524	RxMessage24 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0528	RxMessage24 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x052C	RxMessage24 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0530	RxMessage24 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0534	RxMessage24 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0538	RxMessage24 AMR Data Register	7:0				ACPT_MASK_B24TE2[7:0]				
		15:8				ACPT_MASK_B24TE1[7:0]				
		23:16								
		31:24								
0x053C	RxMessage24 ACR Data Register	7:0				ACPT_CODE_B24TE2[7:0]				
		15:8				ACPT_CODE_B24TE1[7:0]				
		23:16								
		31:24								
0x0540	RxMessage25 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL25	BUFFER_ENA BLE	RTRABORT	RTREPL25_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0544	RxMessage25 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0548	RxMessage25 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x054C	RxMessage25 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x0550	RxMessage25 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0554	RxMessage25 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0558	RxMessage25 AMR Data Register	7:0				ACPT_MASK_B25TE2[7:0]				
		15:8				ACPT_MASK_B25TE1[7:0]				
		23:16								
		31:24								
0x055C	RxMessage25 ACR Data Register	7:0				ACPT_CODE_B25TE2[7:0]				
		15:8				ACPT_CODE_B25TE1[7:0]				
		23:16								
		31:24								
0x0560	RxMessage26 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL26	BUFFER_ENA BLE	RTRABORT	RTREPL26_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0564	RxMessage26 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0568	RxMessage26 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x056C	RxMessage26 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x0570	RxMessage26 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0574	RxMessage26 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x0578	RxMessage26 AMR Data Register	7:0			ACPT_MASK_B26TE2[7:0]					
		15:8			ACPT_MASK_B26TE1[7:0]					
		23:16								
		31:24								
0x057C	RxMessage26 ACR Data Register	7:0			ACPT_CODE_B26TE2[7:0]					
		15:8			ACPT_CODE_B26TE1[7:0]					
		23:16								
		31:24								
0x0580	RxMessage27 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL27	BUFFER_ENABLE	RTRABORT	RTREPL27_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x0584	RxMessage27 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0588	RxMessage27 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x058C	RxMessage27 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x0590	RxMessage27 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x0594	RxMessage27 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0598	RxMessage27 AMR Data Register	7:0					ACPT_MASK_B27TE2[7:0]			
		15:8					ACPT_MASK_B27TE1[7:0]			
		23:16								
		31:24								
0x059C	RxMessage27 ACR Data Register	7:0					ACPT_CODE_B27TE2[7:0]			
		15:8					ACPT_CODE_B27TE1[7:0]			
		23:16								
		31:24								
0x05A0	RxMessage28 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL28	BUFFER_ENABLE	RTRABORT	RTREPL28_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x05A4	RxMessage28 ID Register	7:0			RX_ID[4:0]					
		15:8					RX_ID[12:5]			
		23:16					RX_ID[20:13]			
		31:24					RX_ID[28:21]			
0x05A8	RxMessage28 Word1 Data Register	7:0					WORD1_RXDATA[7:0]			
		15:8					WORD1_RXDATA[15:8]			
		23:16					WORD1_RXDATA[23:16]			
		31:24					WORD1_RXDATA[31:24]			
0x05AC	RxMessage28 Word0 Data Register	7:0					WORD0_RXDATA[7:0]			
		15:8					WORD0_RXDATA[15:8]			
		23:16					WORD0_RXDATA[23:16]			
		31:24					WORD0_RXDATA[31:24]			
0x05B0	RxMessage28 AMR Register	7:0				ACPT_MASK_ID[4:0]		ACPT_MASK_ID	ACPT_MASK_RTR	
		15:8					ACPT_MASK_ID[12:5]			
		23:16					ACPT_MASK_ID[20:13]			
		31:24					ACPT_MASK_ID[28:21]			
0x05B4	RxMessage28 ACR Register	7:0				ACPT_CODE_ID[4:0]		ACPT_CODE_ID	ACPT_CODE_RTR	
		15:8					ACPT_CODE_ID[12:5]			
		23:16					ACPT_CODE_ID[20:13]			
		31:24					ACPT_CODE_ID[28:21]			
0x05B8	RxMessage28 AMR Data Register	7:0					ACPT_MASK_B28TE2[7:0]			
		15:8					ACPT_MASK_B28TE1[7:0]			
		23:16								
		31:24								
0x05BC	RxMessage28 ACR Data Register	7:0					ACPT_CODE_B28TE2[7:0]			
		15:8					ACPT_CODE_B28TE1[7:0]			
		23:16								
		31:24								
0x05C0	RxMessage29 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL29	BUFFER_ENABLE	RTRABORT	RTREPL29_PENDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x05C4	RxMessage29 ID Register	7:0			RX_ID[4:0]					
		15:8					RX_ID[12:5]			
		23:16					RX_ID[20:13]			
		31:24					RX_ID[28:21]			
0x05C8	RxMessage29 Word1 Data Register	7:0					WORD1_RXDATA[7:0]			
		15:8					WORD1_RXDATA[15:8]			
		23:16					WORD1_RXDATA[23:16]			
		31:24					WORD1_RXDATA[31:24]			
0x05CC	RxMessage29 Word0 Data Register	7:0					WORD0_RXDATA[7:0]			
		15:8					WORD0_RXDATA[15:8]			
		23:16					WORD0_RXDATA[23:16]			
		31:24					WORD0_RXDATA[31:24]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x05D0	RxMessage29 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x05D4	RxMessage29 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x05D8	RxMessage29 AMR Data Register	7:0				ACPT_MASK_B29TE2[7:0]				
		15:8				ACPT_MASK_B29TE1[7:0]				
		23:16								
		31:24								
0x05DC	RxMessage29 ACR Data Register	7:0				ACPT_CODE_B29TE2[7:0]				
		15:8				ACPT_CODE_B29TE1[7:0]				
		23:16								
		31:24								
0x05E0	RxMessage30 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL30	BUFFER_ENA BLE	RTRABORT	RTREPL30_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								
0x05E4	RxMessage30 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x05E8	RxMessage30 Word1 Data Register	7:0				WORD1_RXDATA[7:0]				
		15:8				WORD1_RXDATA[15:8]				
		23:16				WORD1_RXDATA[23:16]				
		31:24				WORD1_RXDATA[31:24]				
0x05EC	RxMessage30 Word0 Data Register	7:0				WORD0_RXDATA[7:0]				
		15:8				WORD0_RXDATA[15:8]				
		23:16				WORD0_RXDATA[23:16]				
		31:24				WORD0_RXDATA[31:24]				
0x05F0	RxMessage30 AMR Register	7:0			ACPT_MASK_ID[4:0]			ACPT_MASK_I DE	ACPT_MASK_RTR	
		15:8				ACPT_MASK_ID[12:5]				
		23:16				ACPT_MASK_ID[20:13]				
		31:24				ACPT_MASK_ID[28:21]				
0x05F4	RxMessage30 ACR Register	7:0			ACPT_CODE_ID[4:0]			ACPT_CODE_I DE	ACPT_CODE_RTR	
		15:8				ACPT_CODE_ID[12:5]				
		23:16				ACPT_CODE_ID[20:13]				
		31:24				ACPT_CODE_ID[28:21]				
0x05F8	RxMessage30 AMR Data Register	7:0				ACPT_MASK_B30TE2[7:0]				
		15:8				ACPT_MASK_B30TE1[7:0]				
		23:16								
		31:24								
0x05FC	RxMessage30 ACR Data Register	7:0				ACPT_CODE_B30TE2[7:0]				
		15:8				ACPT_CODE_B30TE1[7:0]				
		23:16								
		31:24								
0x0600	RxMessage31 Control Register	7:0		LINK_FLAG	RX_INTEBL	RTR_REPL31	BUFFER_ENA BLE	RTRABORT	RTREPL31_PE NDING	MSGAV
		15:8								
		23:16			RTR	IDE		DLC[3:0]		
		31:24								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0604	RxMessage31 ID Register	7:0			RX_ID[4:0]					
		15:8				RX_ID[12:5]				
		23:16				RX_ID[20:13]				
		31:24				RX_ID[28:21]				
0x0608	RxMessage31 Word1 Data Register	7:0			WORD1_RXDATA[7:0]					
		15:8			WORD1_RXDATA[15:8]					
		23:16			WORD1_RXDATA[23:16]					
		31:24			WORD1_RXDATA[31:24]					
0x060C	RxMessage31 Word0 Data Register	7:0			WORD0_RXDATA[7:0]					
		15:8			WORD0_RXDATA[15:8]					
		23:16			WORD0_RXDATA[23:16]					
		31:24			WORD0_RXDATA[31:24]					
0x0610	RxMessage31 AMR Register	7:0			ACPT_MASK_ID[4:0]		ACPT_MASK_I	ACPT_MASK_DE	ACPT_MASK_RTR	
		15:8			ACPT_MASK_ID[12:5]					
		23:16			ACPT_MASK_ID[20:13]					
		31:24			ACPT_MASK_ID[28:21]					
0x0614	RxMessage31 ACR Register	7:0			ACPT_CODE_ID[4:0]		ACPT_CODE_I	ACPT_CODE_DE	ACPT_CODE_RTR	
		15:8			ACPT_CODE_ID[12:5]					
		23:16			ACPT_CODE_ID[20:13]					
		31:24			ACPT_CODE_ID[28:21]					
0x0618	RxMessage31 AMR Data Register	7:0			ACPT_MASK_B31TE2[7:0]					
		15:8			ACPT_MASK_B31TE1[7:0]					
		23:16								
		31:24								
0x061C	RxMessage31 ACR Data Register	7:0			ACPT_CODE_B31TE2[7:0]					
		15:8			ACPT_CODE_B31TE1[7:0]					
		23:16								
		31:24								

4.1. TxMessageY Word1 Data Register [\(Ask a Question\)](#)

Name: TxMessageY Word1 Data Register

Offset: 0x28 + n*0x10 [n=0..31]

Reset: 0x0

Property: Write-only

TxMessageY Buffer: TxMessageY Word1 Data Register

Bit	31	30	29	28	27	26	25	24
WORD1_TXDATA[31:24]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
WORD1_TXDATA[23:16]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
WORD1_TXDATA[15:8]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
WORD1_TXDATA[7:0]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WORD1_TXDATA[31:0] The byte mapping can be set using the BYTE_ORDER configuration bit
When BYTE_ORDER= 0

Value	Description
31:24	1
23:16	2
15:8	3
7:0	4

When BYTE_ORDER= 1

Value	Description
31:24	4
23:16	3
15:8	2
7:0	1

The following table shows data transmission from MSB to LSB based on the DLC configuration

Value	Description
1	31:24
2	31:16
3	31:8
4	31:0

4.2. TxMessageY Word0 Data Register [\(Ask a Question\)](#)

Name: TxMessageY Word0 Data Register

Offset: 0x2C + n*0x10 [n=0..31]

Reset: 0x0

Property: Write-only

TxMessageY Buffer: TxMessageY Word0 Data Register

Bit	31	30	29	28	27	26	25	24
WORD0_TXDATA[31:24]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
WORD0_TXDATA[23:16]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
WORD0_TXDATA[15:8]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
WORD0_TXDATA[7:0]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WORD0_TXDATA[31:0] The byte mapping can be set using the BYTE_ORDER configuration bit
When BYTE_ORDER= 0

Value	Description
31:24	5
23:16	6
15:8	7
7:0	8

When BYTE_ORDER= 1

Value	Description
31:24	8
23:16	7
15:8	6
7:0	5

The following table shows data transmission from MSB to LSB based on the DLC configuration

Value	Description
5	31:24
6	31:16
7	31:8
Greater than or equal to 8	31:0

4.3. TxMessageY ID Register [\(Ask a Question\)](#)

Name: TxMessageY ID Register

Offset: 0x24 + n*0x10 [n=0..31]

Reset: 0x0

Property: Write-only

TxMessageY Buffer: Identifier Register

Bit	31	30	29	28	27	26	25	24
TX_ID[28:21]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
TX_ID[20:13]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
TX_ID[12:5]								
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
TX_ID[4:0]								
Access	W	W	W	W	W			
Reset	0	0	0	0	0			

Bits 31:3 – TX_ID[28:0] Identifier MSB 11 bits [28:18] is used from MSB to LSB for the standard message format and bits [28:0] is used from MSB to LSB for the extended message format.

4.4. TxMessageY Control Register [\(Ask a Question\)](#)

Name: TxMessageY Control Register

Offset: 0x20 + n*0x10 [n=0..31]

Reset: 0x0

Property: RW/W

TxMessageY Buffer: Control Flags Register

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	W		W	W	W	W	W	W
Reset	0		0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	TXINTR_WPN	TX_INTEBL	TX_ABORT	TX_REQ
Access					W	W	R/W	R/W
Reset					0	0	0	0

Bit 23 – WPN WPN, Write Protect Not. The readback value of this bit is undefined. Info: Using the WPN flag enables simple retransmission of the same message by only having to set the TX_REQ flag without taking care of the special flags

Value	Description
0b0	Bits [21:16] of this register remain unchanged
0b1	Bits [21:16] of this register are modified, default.

Bit 21 – RTR Remote Bit

Value	Description
0b0	Standard message
0b1	RTR message

Bit 20 – IDE Extended Identifier Bit

Value	Description
0b0	Standard format message
0b1	Extended format message

Bits 19:16 – DLC[3:0] Data Length Code. Invalid values are transmitted as they are, but the number of data bytes is limited to eight.

Value	Description
0	Message has 0 data byte
1	Message has 1 data byte
2	Message has 2 data bytes

DLC (continued)

Value	Description
3	Message has 3 data bytes
4	Message has 4 data bytes
5	Message has 5 data bytes
6	Message has 6 data bytes
7	Message has 7 data bytes
Greater than or equal to 8	Message has 8 data bytes

Bit 3 - TXINTR_WPN Transmit Interrupt Write Protect Not.

Value	Description
0b0	Bit [2] of this register remains unchanged
0b1	Bit [2] of this register is modified, default.

Bit 2 - TX_INTEBL Transmit Interrupt Enable

Value	Description
0b0	Disable interrupt
0b1	Enable interrupt. Successful message transmission sets the TX_MSG flag in the interrupt source register

Bit 1 - TX_ABORT Transmit Abort Request

Value	Description
0b0	When this bit is set to 0, the CoreCAN retransmits the message when arbitration loss occurs.
0b1	Requests removal of a pending message. The message is removed the next time an arbitration loss occurs. The flag is cleared when the message is removed or when the message wins arbitration. The TX_REQ flag is released at the same time.

Bit 0 - TX_REQ Transmit Request

Value	Description
0b0	When this bit is programmed to 0, the CoreCAN does not initiate message transmission. The CoreCAN set this bit 0 when the transmit request is served.
0b1	When this bit is programmed to 1, the CoreCAN initiates message transmission. The CoreCAN set this bit 1 when the transmit request is pending.

4.5. TxBuffer Status Register [\(Ask a Question\)](#)

Name: TxBuffer Status Register

Offset: 0x00C

Reset: 0x0

Property: Read-only

Tx Message Memory Status Indicator Register

Bit	31	30	29	28	27	26	25	24
	TXBUFF31_ST	TXBUFF30_ST	TXBUFF29_ST	TXBUFF28_ST	TXBUFF27_ST	TXBUFF26_ST	TXBUFF25_ST	TXBUFF24_ST
Access	S	S	S	S	S	S	S	S
Reset	R	R	R	R	R	R	R	R
	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TXBUFF23_ST	TXBUFF22_ST	TXBUFF21_ST	TXBUFF20_ST	TXBUFF19_ST	TXBUFF18_ST	TXBUFF17_ST	TXBUFF16_ST
Access	S	S	S	S	S	S	S	S
Reset	R	R	R	R	R	R	R	R
	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TXBUFF15_ST	TXBUFF14_ST	TXBUFF13_ST	TXBUFF12_ST	TXBUFF11_ST	TXBUFF10_ST	TXBUFF9_STS	TXBUFF8_STS
Access	S	S	S	S	S	S	R	R
Reset	R	R	R	R	R	R	0	0
	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TXBUFF7_STS	TXBUFF6_STS	TXBUFF5_STS	TXBUFF4_STS	TXBUFF3_STS	TXBUFF2_STS	TXBUFF1_STS	TXBUFF0_STS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – TXBUFF31_STS

Value	Description
0b1	Indicates that the transmit message buffer 31 request is pending
0b0	Indicates that the transmit message buffer 31 is ready to accept new request

Bit 30 – TXBUFF30_STS

Value	Description
0b1	Indicates that the transmit message buffer 30 request is pending
0b0	Indicates that the transmit message buffer 30 is ready to accept new request

Bit 29 – TXBUFF29_STS

Value	Description
0b1	Indicates that the transmit message buffer 29 request is pending
0b0	Indicates that the transmit message buffer 29 is ready to accept new request

Bit 28 – TXBUFF28_STS

Value	Description
0b1	Indicates that the transmit message buffer 28 request is pending
0b0	Indicates that the transmit message buffer 28 is ready to accept new request

Bit 27 – TXBUFF27_STS

Value	Description
0b1	Indicates that the transmit message buffer 27 request is pending
0b0	Indicates that the transmit message buffer 27 is ready to accept new request

Bit 26 – TXBUFF26_STS

Value	Description
0b1	Indicates that the transmit message buffer 26 request is pending
0b0	Indicates that the transmit message buffer 26 is ready to accept new request

Bit 25 – TXBUFF25_STS

Value	Description
0b1	Indicates that the transmit message buffer 25 request is pending
0b0	Indicates that the transmit message buffer 25 is ready to accept new request

Bit 24 – TXBUFF24_STS

Value	Description
0b1	Indicates that the transmit message buffer 24 request is pending
0b0	Indicates that the transmit message buffer 24 is ready to accept new request

Bit 23 – TXBUFF23_STS

Value	Description
0b1	Indicates that the transmit message buffer 23 request is pending
0b0	Indicates that the transmit message buffer 23 is ready to accept new request

Bit 22 – TXBUFF22_STS

Value	Description
0b1	Indicates that the transmit message buffer 22 request is pending
0b0	Indicates that the transmit message buffer 22 is ready to accept new request

Bit 21 – TXBUFF21_STS

Value	Description
0b1	Indicates that the transmit message buffer 21 request is pending

TXBUFF21_STS (continued)

Value	Description
0b0	Indicates that the transmit message buffer 21 is ready to accept new request

Bit 20 – TXBUFF20_STS

Value	Description
0b1	Indicates that the transmit message buffer 20 request is pending
0b0	Indicates that the transmit message buffer 20 is ready to accept new request

Bit 19 – TXBUFF19_STS

Value	Description
0b1	Indicates that the transmit message buffer 19 request is pending
0b0	Indicates that the transmit message buffer 19 is ready to accept new request

Bit 18 – TXBUFF18_STS

Value	Description
0b1	Indicates that the transmit message buffer 18 request is pending
0b0	Indicates that the transmit message buffer 18 is ready to accept new request

Bit 17 – TXBUFF17_STS

Value	Description
0b1	Indicates that the transmit message buffer 17 request is pending
0b0	Indicates that the transmit message buffer 17 is ready to accept new request

Bit 16 – TXBUFF16_STS

Value	Description
0b1	Indicates that the transmit message buffer 16 request is pending
0b0	Indicates that the transmit message buffer 16 is ready to accept new request

Bit 15 – TXBUFF15_STS

Value	Description
0b1	Indicates that the transmit message buffer 15 request is pending
0b0	Indicates that the transmit message buffer 15 is ready to accept new request

Bit 14 – TXBUFF14_STS

Value	Description
0b1	Indicates that the transmit message buffer 14 request is pending
0b0	Indicates that the transmit message buffer 14 is ready to accept new request

Bit 13 – TXBUFF13_STS

Value	Description
0b1	Indicates that the transmit message buffer 13 request is pending
0b0	Indicates that the transmit message buffer 13 is ready to accept new request

Bit 12 – TXBUFF12_STS

Value	Description
0b1	Indicates that the transmit message buffer 12 request is pending
0b0	Indicates that the transmit message buffer 12 is ready to accept new request

Bit 11 – TXBUFF11_STS

Value	Description
0b1	Indicates that the transmit message buffer 11 request is pending
0b0	Indicates that the transmit message buffer 11 is ready to accept new request

Bit 10 – TXBUFF10_STS

Value	Description
0b1	Indicates that the transmit message buffer 10 request is pending
0b0	Indicates that the transmit message buffer 10 is ready to accept new request

Bit 9 – TXBUFF9_STS

Value	Description
0b1	Indicates that the transmit message buffer 9 request is pending
0b0	Indicates that the transmit message buffer 9 is ready to accept new request

Bit 8 – TXBUFF8_STS

Value	Description
0b1	Indicates that the transmit message buffer 8 request is pending
0b0	Indicates that the transmit message buffer 8 is ready to accept new request

Bit 7 – TXBUFF7_STS

Value	Description
0b1	Indicates that the transmit message buffer 7 request is pending
0b0	Indicates that the transmit message buffer 7 is ready to accept new request

Bit 6 – TXBUFF6_STS

Value	Description
0b1	Indicates that the transmit message buffer 6 request is pending

TXBUFF6_STS (continued)

Value	Description
0b0	Indicates that the transmit message buffer 6 is ready to accept new request

Bit 5 - TXBUFF5_STS

Value	Description
0b1	Indicates that the transmit message buffer 5 request is pending
0b0	Indicates that the transmit message buffer 5 is ready to accept new request

Bit 4 - TXBUFF4_STS

Value	Description
0b1	Indicates that the transmit message buffer 4 request is pending
0b0	Indicates that the transmit message buffer 4 is ready to accept new request

Bit 3 - TXBUFF3_STS

Value	Description
0b1	Indicates that the transmit message buffer 3 request is pending
0b0	Indicates that the transmit message buffer 3 is ready to accept new request

Bit 2 - TXBUFF2_STS

Value	Description
0b1	Indicates that the transmit message buffer 2 request is pending
0b0	Indicates that the transmit message buffer 2 is ready to accept new request

Bit 1 - TXBUFF1_STS

Value	Description
0b1	Indicates that the transmit message buffer 1 request is pending
0b0	Indicates that the transmit message buffer 1 is ready to accept new request

Bit 0 - TXBUFF0_STS

Value	Description
0b1	Indicates that the transmit message buffer 0 request is pending
0b0	Indicates that the transmit message buffer 0 is ready to accept new request

4.6. RxMessageY Word1 Data Register [\(Ask a Question\)](#)

Name: RxMessageY Word1 Data Register

Offset: 0x0228 + n*0x20 [n=0..31]

Reset: 0x0

Property: Read/Write

RxMessageY Buffer:: RxMessageY Word1 Data Register

Bit	31	30	29	28	27	26	25	24
WORD1_RXDATA[31:24]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
WORD1_RXDATA[23:16]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
WORD1_RXDATA[15:8]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
WORD1_RXDATA[7:0]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WORD1_RXDATA[31:0] The byte mapping can be set using the BYTE_ORDER configuration bitWhen BYTE_ORDER= 0

Value	Description
31:24	1
23:16	2
15:8	3
7:0	4

When BYTE_ORDER= 1

Value	Description
31:24	4
23:16	3
15:8	2
7:0	1

The following table shows data stored into this register from MSB to LSB based on the received DLC

Value	Description
1	31:24
2	31:16
3	31:8
4	31:0

4.7. RxMessageY Word0 Data Register [\(Ask a Question\)](#)

Name: RxMessageY Word0 Data Register

Offset: 0x022C + n*0x20 [n=0..31]

Reset: 0x0

Property: Read/Write

RxMessageY Buffer: RxMessageY Word0 Data Register

Bit	31	30	29	28	27	26	25	24
WORD0_RXDATA[31:24]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
WORD0_RXDATA[23:16]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
WORD0_RXDATA[15:8]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
WORD0_RXDATA[7:0]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WORD0_RXDATA[31:0] The byte mapping can be set using the BYTE_ORDER configuration bitWhen BYTE_ORDER= 0

Value	Description
31:24	5
23:16	6
15:8	7
7:0	8

When BYTE_ORDER= 1

Value	Description
31:24	8
23:16	7
15:8	6
7:0	5

The following table shows data stored into this register from MSB to LSB based on the received DLC

Value	Description
5	31:24
6	31:16
7	31:8
Greater than or equal to 8	31:0

4.8. RxMessageY ID Register [\(Ask a Question\)](#)

Name: RxMessageY ID Register
Offset: 0x0224 + n*0x20 [n=0..31]
Reset: 0x0
Property: Read/Write

RxMessageY Buffer:: RxMessageY Identifier Register

Bit	31	30	29	28	27	26	25	24
RX_ID[28:21]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
RX_ID[20:13]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
RX_ID[12:5]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
RX_ID[4:0]								
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bits 31:3 – RX_ID[28:0] Identifier ID [28:0] For the standard message format, only MSB 11 bits [28:18] of identifier are valid. For the extended message format, all 29 bits of the identifier are valid

4.9. RxMessageY Control Register [\(Ask a Question\)](#)

Name: RxMessageY Control Register

Offset: 0x0220 + n*0x20 [n=0..31]

Reset: 0x0

Property: RW/W/RW1C

RxMessageY Buffer: Control Flags Register

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access			RTR	IDE		DLC[3:0]		
Reset			R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access		LINK_FLAG	RX_INTEBL	RTR_REPLY	BUFFER_ENA BLE	RTRABORT	RTREPLY_PEN DING	MSGAV
Reset		R/W	R/W	R/W	R/W	R/W	RW1C	RW1C

Bit 21 – RTR Remote Bit

Value	Description
0b0	Indicates that the regular message is received
0b1	Indicates that the RTR message is received

Bit 20 – IDE Extended Identifier Bit

Value	Description
0b0	Indicates that the standard format message is received
0b1	Indicates that the extended format message is received

Bit 17 – WPN Write Protect Not Low. The readback value of this bit is undefined.

Value	Description
0b0	Bit [6:3] of this register remain unchanged.
0b1	Bit [6:3] of this register are modified, default.

Bits 19:16 – DLC[3:0] Data Length Code

Value	Description
0	The message received contains 0 valid data byte
1	The message received contains 1 data byte
2	The message received contains 2 data bytes
3	The message received contains 3 data bytes
4	The message received contains 4 data bytes

DLC (continued)

Value	Description
5	The message received contains 5 data bytes
6	The message received contains 6 data bytes
7	The message received contains 7 data bytes
Greater than or equal to 8	The message received contains 8 data bytes

Bit 6 – LINK_FLAG Link Flag

Value	Description
0b0	This buffer is not linked to the next buffer.
0b1	This buffer is linked with next buffer.

Bit 5 – RX_INTEBL Receive Interrupt Enable

Value	Description
0b0	Disable interrupt.
0b1	Enable interrupt. Successful message reception sets the RX_MSG flag in the interrupt source register.

Bit 4 – RTR_REPLY Automatic message reply upon receipt of an RTR message.

Value	Description
0b0	Disable automatic RTR message handling.
0b1	Enable automatic RTR message handling.

Bit 3 – BUFFER_ENABLE Receive Buffer0 Enable

Value	Description
0b0	Receive buffer0 is disabled.
0b1	Receive buffer0 is enabled.

Bit 2 – RTRABORT RTR Abort Request

Value	Description
0b0	When this bit is set to 0, the CoreCAN retransmits the RTR message when arbitration loss occurs.
0b1	Requests the removal of a pending RTR message reply. The message is removed the next time an arbitration loss occurs. The flag is cleared when the message is removed or when the message wins arbitration. The TX_REQ flag is released at the same time.

Bit 1 – RTREPLY_PENDING RTReply_pending

Value	Description
0b0	When RTR_REPLY is set to 1, this bit indicates that after receiving the RTR message, RTR reply request has been sent.
0b1	When RTR_REPLY is set to 1, this bit indicates that after receiving the RTR message, RTR reply request is pending.

Bit 0 – MSGAV Message Available When RTR_REPLY is set to 0, this bit is set to 1 by the CoreCAN when buffer contains a valid message. When RTR_REPLY is set to 1, this bit is set to 1 by the CoreCAN when RTR reply request is sent. Writing 1, clears this bit.

4.10. RxMessageY AMR Register [\(Ask a Question\)](#)

Name: RxMessageY AMR Register

Offset: 0x0230 + n*0x20 [n=0..31]

Reset: 0x0

Property: Read/Write

RxMessageY Buffer: RxMessageY Acceptance Mask Register

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

Bits 31:3 – ACPT_MASK_ID[28:0] Acceptance mask for identifier

Bit 2 – ACPT_MASK_IDE Acceptance mask for IDE

Bit 1 – ACPT_MASK_RTR Acceptance mask for RTR

4.11. RxMessageY AMR Data Register [\(Ask a Question\)](#)

Name: RxMessageY AMR Data Register

Offset: 0x0238 + n*0x20 [n=0..31]

Reset: 0x0

Property: Read/Write

RxMessageY Buffer:: RxMessageY Acceptance Mask Data Register

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	ACPT_MASK_BYTE1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ACPT_MASK_BYTE2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – ACPT_MASK_BYTE1[7:0] Acceptance mask for data byte 1

Bits 7:0 – ACPT_MASK_BYTE2[7:0] Acceptance mask for data byte 2

4.12. RxMessageY ACR Register [\(Ask a Question\)](#)

Name: RxMessageY ACR Register

Offset: 0x0234 + n*0x20 [n=0..31]

Reset: 0x0

Property: Read/Write

RxMessageY Buffer: RxMessageY Acceptance Code Register

Bit	31	30	29	28	27	26	25	24
ACPT_CODE_ID[28:21]								
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
ACPT_CODE_ID[20:13]								
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
ACPT_CODE_ID[12:5]								
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
ACPT_CODE_ID[4:0]								
					ACPT_CODE_ID DE	ACPT_CODE_ID RTR		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:3 – ACPT_CODE_ID[28:0] Acceptance code for identifier

Bit 2 – ACPT_CODE_IDE Acceptance code for IDE

Bit 1 – ACPT_CODE_RTR Acceptance code for RTR

4.13. RxMessageY ACR Data Register [\(Ask a Question\)](#)

Name: RxMessageY ACR Data Register
Offset: 0x023C + n*0x20 [n=0..31]
Reset: 0x0
Property: Read/Write

RxMessageY Buffer:: RxMessageY Acceptance Code Data Register

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
				ACPT_CODE_BYTE1[7:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				ACPT_CODE_BYTE2[7:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – ACPT_CODE_BYTE1[7:0] Acceptance mask for data byte 1

Bits 7:0 – ACPT_CODE_BYTE2[7:0] Acceptance mask for data byte 2

4.14. RxBuffer Status Register [\(Ask a Question\)](#)

Name: RxBuffer Status Register

Offset: 0x008

Reset: 0x0

Property: Read-only

Rx Message Memory Status Indicator Register

Bit	31	30	29	28	27	26	25	24
	RXBUFF31_ST	RXBUFF30_ST	RXBUFF29_ST	RXBUFF28_ST	RXBUFF27_ST	RXBUFF26_ST	RXBUFF25_ST	RXBUFF24_ST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RXBUFF23_ST	RXBUFF22_ST	RXBUFF21_ST	RXBUFF20_ST	RXBUFF19_ST	RXBUFF18_ST	RXBUFF17_ST	RXBUFF16_ST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RXBUFF15_STS	RXBUFF14_STS	RXBUFF13_STS	RXBUFF12_STS	RXBUFF11_STS	RXBUFF10_STS	RXBUFF9_STS	RXBUFF8_STS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RXBUFF7_STS	RXBUFF6_STS	RXBUFF5_STS	RXBUFF4_STS	RXBUFF3_STS	RXBUFF2_STS	RXBUFF1_STS	RXBUFF0_STS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – RXBUFF31_STS

Value	Description
0b1	Indicates that the receive message buffer 31 has received new message
0b0	Indicates that the receive message buffer 31 has not received new message

Bit 30 – RXBUFF30_STS

Value	Description
0b1	Indicates that the receive message buffer 30 has received new message
0b0	Indicates that the receive message buffer 30 has not received new message

Bit 29 – RXBUFF29_STS

Value	Description
0b1	Indicates that the receive message buffer 29 has received new message
0b0	Indicates that the receive message buffer 29 has not received new message

Bit 28 – RXBUFF28_STS

Value	Description
0b1	Indicates that the receive message buffer 28 has received new message
0b0	Indicates that the receive message buffer 28 has not received new message

Bit 27 – RXBUFF27_STS

Value	Description
0b1	Indicates that the receive message buffer 27 has received new message
0b0	Indicates that the receive message buffer 27 has not received new message

Bit 26 – RXBUFF26_STS

Value	Description
0b1	Indicates that the receive message buffer 26 has received new message
0b0	Indicates that the receive message buffer 26 has not received new message

Bit 25 – RXBUFF25_STS

Value	Description
0b1	Indicates that the receive message buffer 25 has received new message
0b0	Indicates that the receive message buffer 25 has not received new message

Bit 24 – RXBUFF24_STS

Value	Description
0b1	Indicates that the receive message buffer 24 has received new message
0b0	Indicates that the receive message buffer 24 has not received new message

Bit 23 – RXBUFF23_STS

Value	Description
0b1	Indicates that the receive message buffer 23 has received new message
0b0	Indicates that the receive message buffer 23 has not received new message

Bit 22 – RXBUFF22_STS

Value	Description
0b1	Indicates that the receive message buffer 22 has received new message
0b0	Indicates that the receive message buffer 22 has not received new message

Bit 21 – RXBUFF21_STS

Value	Description
0b1	Indicates that the receive message buffer 21 has received new message

RXBUFF21_STS (continued)

Value	Description
0b0	Indicates that the receive message buffer 21 has not received new message

Bit 20 – RXBUFF20_STS

Value	Description
0b1	Indicates that the receive message buffer 20 has received new message
0b0	Indicates that the receive message buffer 20 has not received new message

Bit 19 – RXBUFF19_STS

Value	Description
0b1	Indicates that the receive message buffer 19 has received new message
0b0	Indicates that the receive message buffer 19 has not received new message

Bit 18 – RXBUFF18_STS

Value	Description
0b1	Indicates that the receive message buffer 18 has received new message
0b0	Indicates that the receive message buffer 18 has not received new message

Bit 17 – RXBUFF17_STS

Value	Description
0b1	Indicates that the receive message buffer 17 has received new message
0b0	Indicates that the receive message buffer 17 has not received new message

Bit 16 – RXBUFF16_STS

Value	Description
0b1	Indicates that the receive message buffer 16 has received new message
0b0	Indicates that the receive message buffer 16 has not received new message

Bit 15 – RXBUFF15_STS

Value	Description
0b1	Indicates that the receive message buffer 15 has received new message
0b0	Indicates that the receive message buffer 15 has not received new message

Bit 14 – RXBUFF14_STS

Value	Description
0b1	Indicates that the receive message buffer 14 has received new message
0b0	Indicates that the receive message buffer 14 has not received new message

Bit 13 – RXBUFF13_STS

Value	Description
0b1	Indicates that the receive message buffer 13 has received new message
0b0	Indicates that the receive message buffer 13 has not received new message

Bit 12 – RXBUFF12_STS

Value	Description
0b1	Indicates that the receive message buffer 12 has received new message
0b0	Indicates that the receive message buffer 12 has not received new message

Bit 11 – RXBUFF11_STS

Value	Description
0b1	Indicates that the receive message buffer 11 has received new message
0b0	Indicates that the receive message buffer 11 has not received new message

Bit 10 – RXBUFF10_STS

Value	Description
0b1	Indicates that the receive message buffer 10 has received new message
0b0	Indicates that the receive message buffer 10 has not received new message

Bit 9 – RXBUFF9_STS

Value	Description
0b1	Indicates that the receive message buffer 9 has received new message
0b0	Indicates that the receive message buffer 9 has not received new message

Bit 8 – RXBUFF8_STS

Value	Description
0b1	Indicates that the receive message buffer 8 has received new message
0b0	Indicates that the receive message buffer 8 has not received new message

Bit 7 – RXBUFF7_STS

Value	Description
0b1	Indicates that the receive message buffer 7 has received new message
0b0	Indicates that the receive message buffer 7 has not received new message

Bit 6 – RXBUFF6_STS

Value	Description
0b1	Indicates that the receive message buffer 6 has received new message

RXBUFF6_STS (continued)

Value	Description
0b0	Indicates that the receive message buffer 6 has not received new message

Bit 5 – RXBUFF5_STS

Value	Description
0b1	Indicates that the receive message buffer 5 has received new message
0b0	Indicates that the receive message buffer 5 has not received new message

Bit 4 – RXBUFF4_STS

Value	Description
0b1	Indicates that the receive message buffer 4 has received new message
0b0	Indicates that the receive message buffer 4 has not received new message

Bit 3 – RXBUFF3_STS

Value	Description
0b1	Indicates that the receive message buffer 3 has received new message
0b0	Indicates that the receive message buffer 3 has not received new message

Bit 2 – RXBUFF2_STS

Value	Description
0b1	Indicates that the receive message buffer 2 has received new message
0b0	Indicates that the receive message buffer 2 has not received new message

Bit 1 – RXBUFF1_STS

Value	Description
0b1	Indicates that the receive message buffer 1 has received new message
0b0	Indicates that the receive message buffer 1 has not received new message

Bit 0 – RXBUFF0_STS

Value	Description
0b1	Indicates that the receive message buffer 0 has received new message
0b0	Indicates that the receive message buffer 0 has not received new message

4.15. Interrupt Source Register (Ask a Question)

Name: Interrupt Source Register

Offset: 0x000

Reset: 0x0

Property: RW1C

Interrupt Status Register

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	OST_FAILURE	STUCK_AT_0	RTR_AUTO_MSG	RX_MSG	TX_MSG	RX_MSG_LOSS	BUS_OFF	CRC_ERR
Access	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FORM_ERR	ACK_ERR	STUFF_ERR	BIT_ERR	OVR_LOAD	ARB_LOSS		
Access	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C		
Reset	0	0	0	0	0	0		

Bit 15 – OST_FAILURE

Value	Description
0b1	Indicates that the buffer set for single shot transmission experienced an arbitration loss or a bus error during transmission.
0b0	Normal operation

Bit 14 – STUCK_AT_0

Value	Description
0b1	Indicates that the RX input remains stuck at 0 (dominant level) for more than 11 consecutive bit times.
0b0	Normal operation

Bit 13 – RTR_AUTO_MSG

Value	Description
0b1	Indicates that the RTR auto-reply message is sent.
0b0	Indicates that the RTR auto-reply message is not sent.

Bit 12 – RX_MSG

Value	Description
0b1	When RX_INTEBL flag of receive message buffer is set to 1, this bit indicates the new message is successfully received and available in a receive buffer.

RX_MSG (continued)

Value	Description
0b0	Indicates that the new message is not received

Bit 11 - TX_MSG

Value	Description
0b1	When TX_INTEBL flag of transmit message buffer is set to 1, this bit indicates the message is successfully sent from the transmit buffer
0b0	Indicates that the message is not sent from a transmit buffer

Bit 10 - RX_MSG_LOSS

Value	Description
0b1	Newly received message cannot be stored because the target message buffer is full i.e. target message buffers MSGAV flag is set
0b0	Newly received message is not lost

Bit 9 - BUS_OFF

Value	Description
0b1	Indicates that the CANController entered the bus-off error state.
0b0	Indicates the normal operation

Bit 8 - CRC_ERR

Value	Description
0b1	Indicates that the crc error is detected
0b0	Indicates that the crc error is not detected

Bit 7 - FORM_ERR

Value	Description
0b1	Indicates that the form error is detected
0b0	Indicates that the form error is not detected

Bit 6 - ACK_ERR

Value	Description
0b1	Indicates that the acknowledgement error is detected
0b0	Indicates that the acknowledgement is received

Bit 5 - STUFF_ERR

Value	Description
0b1	Indicates that the bit stuffing error is detected
0b0	Indicates that the bit stuffing error is not detected

Bit 4 - BIT_ERR

Value	Description
0b1	Indicates that the bit error is detected
0b0	Indicates that the bit error is not detected

Bit 3 – OVR_LOAD

Value	Description
0b1	Indicates that the overload message is detected
0b0	Indicates that the overload message is not detected

Bit 2 – ARB_LOSS

Value	Description
0b1	Indicates the message arbitration is lost while sending a message. The message transmission will be retried once the CAN bus is idle again.
0b0	Indicates the arbitration is not lost and message transmitted successful.

4.16. Interrupt Enable Register (Ask a Question)

Name: Interrupt Enable Register

Offset: 0x004

Reset: 0x0

Property: Read/Write

Interrupt Enable Register

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	OST_FAILURE_	STUCK_AT_0_	RTR_MSG_IE	RX_MSG_IE	TX_MSG_IE	RX_MSG_LOS_S_IE	BUS_OFF_IE	CRC_ERR_IE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FORM_ERR_IE	ACK_ERR_IE	STUFF_ERR_IE	BIT_ERR_IE	OVR_LOAD_IE	ARB_LOSS_IE		INT_EBL
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 15 – OST_FAILURE_IE

Value	Description
0b1	Enable interrupt generation when one-shot transmission failure is detected
0b0	Disable interrupt generation when one-shot transmission failure is detected

Bit 14 – STUCK_AT_0_IE

Value	Description
0b1	Enable interrupt generation when stuck at dominant error is detected
0b0	Enable interrupt generation when stuck at dominant error is detected

Bit 13 – RTR_MSG_IE

Value	Description
0b1	Enable interrupt generation when RTR auto reply message is sent
0b0	Disable interrupt generation when RTR auto reply message is sent

Bit 12 – RX_MSG_IE

Value	Description
0b1	Enable interrupt generation when new message is received in receive buffers
0b0	Disable interrupt generation when one new message is received in receive buffers.

Bit 11 – TX_MSG_IE

Value	Description
0b1	Enable interrupt generation when message is transmitted from the transmit buffers
0b0	Disable interrupt generation when message is transmitted from the transmit buffers

Bit 10 – RX_MSG_LOSS_IE

Value	Description
0b1	Enable interrupt generation when newly received message is lost
0b0	Disable interrupt generation when newly received message is lost

Bit 9 – BUS_OFF_IE

Value	Description
0b1	Enable interrupt generation when CANController goes into bus off state
0b0	Disable interrupt generation when CANController goes into bus off state

Bit 8 – CRC_ERR_IE

Value	Description
0b1	Enable interrupt generation when crc error is detected
0b0	Disable interrupt generation when crc error is detected

Bit 7 – FORM_ERR_IE

Value	Description
0b1	Enable interrupt generation when form error is detected
0b0	Disable interrupt generation when form error is detected

Bit 6 – ACK_ERR_IE

Value	Description
0b1	Enable interrupt generation when acknowledgement is not received
0b0	Disable interrupt generation when acknowledgement is not received

Bit 5 – STUFF_ERR_IE

Value	Description
0b1	Enable interrupt generation when stuff error is detected
0b0	Disable interrupt generation when stuff error is detected

Bit 4 - BIT_ERR_IE

Value	Description
0b1	Enable interrupt generation when bit error is detected
0b0	Disable interrupt generation when bit error is detected

Bit 3 - OVR_LOAD_IE

Value	Description
0b1	Enable interrupt generation when overload message is detected
0b0	Disable interrupt generation when overload message is detected

Bit 2 - ARB_LOSS_IE

Value	Description
0b1	Enable interrupt generation when arbitration loss is detected
0b0	Disable interrupt generation when arbitration loss is detected

Bit 0 - INT_EBL

Value	Description
0b1	Enable global interrupt. When this bit is set to 1, interrupt source register is valid.
0b0	Disable all interrupts

4.17. Error Status Register [\(Ask a Question\)](#)

Name: Error Status Register

Offset: 0x010

Reset: 0x0

Property: Read-only

CAN Error Status Register

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access					RXGTE96	TXGTE96	ERROR_STATE[1:0]	
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 19 – RXGTE96

Value	Description
0b1	Indicates the receive error counter is greater or equal 96
0b0	Indicates the receive error counter is less than 96

Bit 18 – TXGTE96

Value	Description
0b1	Indicates the transmit error counter is greater or equal 96
0b0	Indicates the transmit error counter is less than 96

Bits 17:16 – ERROR_STATE[1:0]

The error state of the CoreCAN node

Value	Description
0b00	Indicates active error state (normal operation)
0b01	Indicates passive error state
0b1x	Indicates bus off state

Bits 15:8 – RX_ERR_CNT[7:0] The receive error counter according to the CAN 2.0 specification. When in bus-off state, this counter is used to count 128 groups of 11 recessive bits.

Bits 7:0 – TX_ERR_CNT[7:0] The transmitter error counter according to the CAN 2.0 standard. When it is greater than 255, it is fixed at 255.

4.18. Configurator Register [\(Ask a Question\)](#)

Name: Configurator Register

Offset: 0x018

Reset: 0x0

Property: Read/Write

CAN Configurator Register

Bit	31	30	29	28	27	26	25	24
CFG_BITRATE[14:8]								
Access		R/W						
Reset		0	0	0	0	0	0	0
CFG_BITRATE[7:0]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0
BYTE_ORDER CFG_ARBITER CFG_TSEG1[3:0]								
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
CFG_TSEG2[2:0] AUTO_RESTA RT CFG_SJW[1:0]								
Access	R/W							
Reset	0	0	0	0	0	0	0	0

Bits 30:16 – CFG_BITRATE[14:0] Prescaler for generating the time quantum which defines the TQ.

Value	Description
0	One time quantum equals 1 clock cycle.
1	One time quantum equals 2 clock cycles.
...	...
32767	One time quantum equals 32768 clock cycles.

Bit 13 – BYTE_ORDER The byte position of the CAN receives and transmit data fields can be modified to match the endian setting of the processor or the used CAN protocol.

Value	Description
0b0	Data byte position is not swapped (big endian).
0b1	Data byte position is swapped (little endian).

Bit 12 – CFG_ARBITER Transmit buffer arbiter

Value	Description
0b0	Round robin arbitration.
0b1	Fixed priority arbitration.

Bits 11:8 – CFG_TSEG1[3:0] Time segment 1. Length of the first-time segment TSEG1=CFG_TSEG1+1 Time segment 1 includes the propagation time. CFG_TSEG1=0 and CFG_TSEG1=1 are not allowed.

Bits 7:5 – CFG_TSEG2[2:0] Time segment 2. Length of the second time segment TSEG2 = CFG_TSEG2 + 1. CFG_TSEG2=0 is not allowed.

Bit 4 – AUTO_RESTART

Value	Description
0b0	After bus-off, the CoreCAN must be restarted. This is the recommended setting.
0b1	After the bus-off, the CoreCAN is restarting automatically after 128 groups of 11 recessive bits.

Bits 3:2 – CFG_SJW[1:0]

4.19. Command Register [\(Ask a Question\)](#)

Name: Command Register
Offset: 0x014
Reset: 0x0
Property: RW/R

CAN Command Register

Bit	31	30	29	28	27	26	25	24
	MAJOR_VER[3:0]					MINOR_VER[3:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access						LPBK_MODE	LISTEN_MODE	RUN_STOP_MODE
Reset						R/W	R/W	R/W
						0	0	0

Bits 31:28 – MAJOR_VER[3:0] Major version

Bits 27:24 – MINOR_VER[3:0] Minor version

Bit 2 – LPBK_MODE

Value	Description
0b0	Indicates that the normal operation
0b1	Indicates that the loopback mode is enabled

Bit 1 – LISTEN_MODE

Value	Description
0b0	Indicates that the CoreCAN is in active mode
0b1	Indicates that the CoreCAN is in listen only mode. The output CAN_TX port is driven at R level

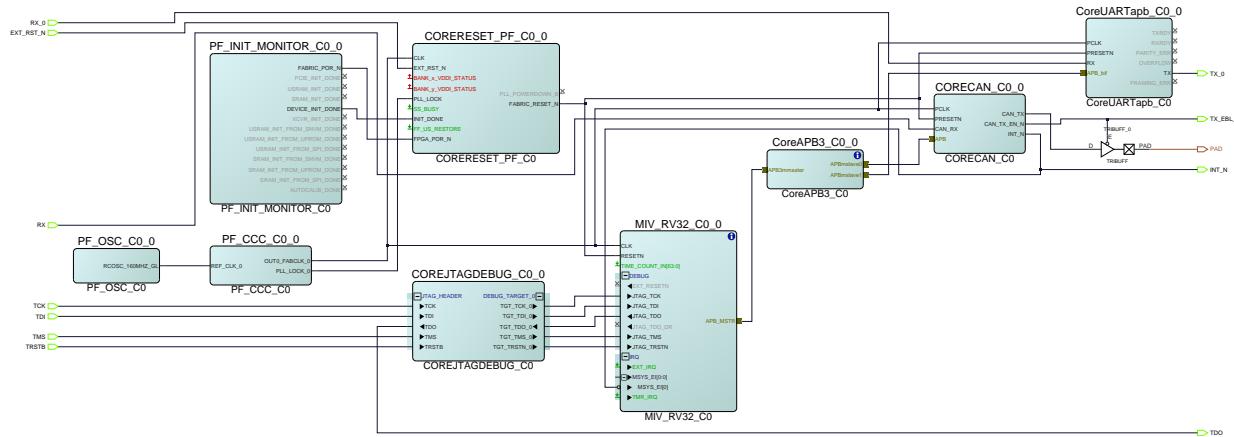
Bit 0 – RUN_STOP_MODE

Value	Description
0b0	Write 0 to set the CoreCAN into stop mode. Returns 0 when stopped.
0b1	Write 1 to set the CoreCAN into run mode. Returns 1 when running.

5. System Integration (Ask a Question)

This section provides example design along with detailed information to facilitate the integration of CoreCAN.

Figure 5-1. CoreCAN System Integration



This example design requires the following:

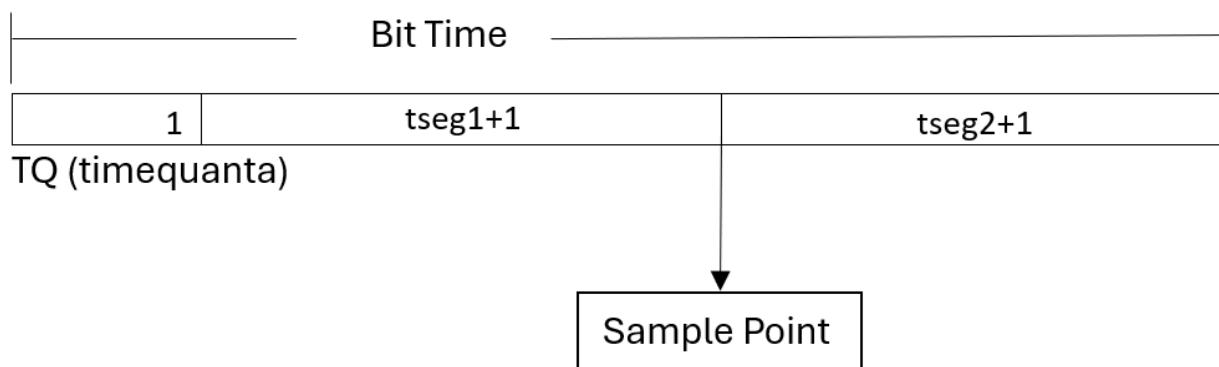
- CoreMIV_RV32 must be connected to the CoreCAN through CoreAPB3.
- CORERESET_PF must be used for all the resets.
- 40 MHz Clock must be used for all the logic in the design which is driven from PF_CCC.
- Top level port INT_N is connected to External System Interrupt port {MSYS_EI [0]} of CoreMIV_RV32 for the purpose of generating interrupt.
- CAN Controller is used to establish Communication with CAN Bus Network through CAN transceivers.
- CAN_TX_EN_N is an external driver control signal to drive the CoreCAN TX pin via TRIBUFF.
- CAN Controller TX and RX are connected to the CAN transceivers, output of the CAN Transceiver CAN_H and CAN_L are connected to the CAN Bus Network.
- Firmware application for CoreMIV_RV32 is used to configure registers of CoreCAN.

Execute the Libero flow with the timing-driven and high-effort place and route options enabled. This example design can be obtained from the Microchip technical support team.

6. Bit Time [\(Ask a Question\)](#)

The following table recommends values for tseg1, tseg2 and prescaler values based on bit rate and clock frequency. These values should be configured properly in configurator register. If the values are not as per calculation, then the node loses synchronization with the CAN bus.

Bit Rate	Clock Frequency	CFG_TSEG1	CFG_TSEG2	CFG_Bitrate
125 Kbps	40 MHZ	9	4	19
250 Kbps	40 MHZ	9	4	9
500 Kbps	40 MHZ	9	4	4
1 Mbps	40 MHZ	4	3	3



$$\text{bittime} = (1 + (\text{tseg1} + 1) + (\text{tseg2} + 1)) * \text{timequanta}$$

$$\text{timequanta} = \text{bitrate} + 1/\text{fclk}$$

7. Known Issues and Workarounds [\(Ask a Question\)](#)

There are no known limitations and workarounds. As this is the first release.

8. Discontinued Features and Devices [\(Ask a Question\)](#)

There are no discontinued features and devices. As this is the first release.

9. Glossary [\(Ask a Question\)](#)

The following are the list of terms and definitions used in the document.

Table 9-1. Terms and Definitions

Term	Definition
ACR	Acceptance Code Register
AMR	Acceptance Mask Register
APB	Advanced Peripheral Bus
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
RTR	Remote Transmission Request

10. Device Utilization and Performance (Ask a Question)

This section provides details on the CoreCAN's device utilization and performance.

Table 10-1. CoreCAN Utilization

Device Details		Resources			Performance (MHz)	RAMs	
Family	Device	LUTs	DFF	Logic Elements		LSRAM	μSRAM
PolarFire®	MPF300T	4050	1700	4300	124	4	0



Important: The data in this table is achieved using typical synthesis and layout settings. The CDR reference clock source is set to **Dedicated** with other configurator values unchanged.

11. Revision History [\(Ask a Question\)](#)

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Table 11-1. Revision History

Revision	Date	Description
A	05/2025	Initial release

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

Trademarks

The "Microchip" name and logo, the "M" logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries ("Microchip Trademarks"). Information regarding Microchip Trademarks can be found at <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>.

ISBN: 979-8-3371-1186-5

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

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

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.