



# Gowin Controller Area Network (CAN) Controller IP User Guide

IPUG527-1.2.2E, 10/25/2024

**Copyright © 2024 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.**

**GOWIN** is a trademark of Guangdong Gowin Semiconductor Corporation and is registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

#### **Disclaimer**

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

**Revision History**

| Date       | Version | Description                                 |
|------------|---------|---------------------------------------------|
| 01/08/2019 | 1.0E    | Initial version published.                  |
| 06/28/2019 | 1.1E    | APB, Wishbone, and REG bus interface added. |
| 06/01/2020 | 1.2E    | Function description of CAN IP updated.     |
| 12/29/2023 | 1.2.1E  | Table 5-1 Pin Description updated.          |
| 10/25/2024 | 1.2.2E  | "Support CAN FD" option added.              |

# Contents

|                                                      |            |
|------------------------------------------------------|------------|
| <b>Contents .....</b>                                | <b>i</b>   |
| <b>List of Figures .....</b>                         | <b>iii</b> |
| <b>List of Tables .....</b>                          | <b>iv</b>  |
| <b>1 About This Guide .....</b>                      | <b>1</b>   |
| 1.1 Purpose .....                                    | 1          |
| 1.2 Related Documents .....                          | 1          |
| 1.3 Abbreviations and Terminology .....              | 2          |
| 1.4 Support and Feedback .....                       | 2          |
| <b>2 Overview .....</b>                              | <b>3</b>   |
| <b>3 Features and Performance .....</b>              | <b>4</b>   |
| 3.1 Key Features .....                               | 4          |
| 3.2 Working Frequency .....                          | 5          |
| 3.3 Resource Utilization .....                       | 5          |
| <b>4 Structure and Functional Description .....</b>  | <b>6</b>   |
| 4.1 Module Structure .....                           | 6          |
| 4.2 Functional Description .....                     | 7          |
| 4.3 Function Block Diagram .....                     | 7          |
| 4.4 Function Block Description .....                 | 7          |
| 4.4.1 Bit Timing Logic (BTL) .....                   | 7          |
| 4.4.2 Bit Stream Processing (BSP) .....              | 8          |
| 4.4.3 Baud Rate Prescaler (BRP) .....                | 8          |
| 4.4.4 Rx Acceptance Filters .....                    | 8          |
| 4.4.5 Shared Buffer .....                            | 8          |
| 4.4.6 Transmit Scheduler .....                       | 14         |
| 4.4.7 Configuration and Status Registers (CSR) ..... | 14         |
| 4.4.8 CPU Interface .....                            | 14         |
| <b>5 Interface List .....</b>                        | <b>17</b>  |
| 5.1 PIN Diagram .....                                | 17         |
| 5.2 Pin Description .....                            | 17         |
| <b>6 Registers .....</b>                             | <b>21</b>  |

|                                                 |           |
|-------------------------------------------------|-----------|
| <b>7 Interrupt.....</b>                         | <b>34</b> |
| <b>8 Clocking .....</b>                         | <b>35</b> |
| <b>9 Reset.....</b>                             | <b>37</b> |
| <b>10 Example Initialization Sequence .....</b> | <b>38</b> |
| 10.1 Power-up .....                             | 38        |
| 10.2 Initialization .....                       | 38        |
| <b>11 Interface Configuration .....</b>         | <b>39</b> |
| <b>12 Reference Design .....</b>                | <b>41</b> |

# List of Figures

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| Figure 4-1 Module Structure .....                                         | 6  |
| Figure 4-2 CAN IP Functional Block Diagram .....                          | 7  |
| Figure 4-3 Shared Buffer Illustration .....                               | 9  |
| Figure 4-4 Receive Buffer/FIFO Structure and Format.....                  | 10 |
| Figure 4-5 Transmit High0-Priority Buffer/FIFO Structure AND Format ..... | 12 |
| Figure 4-6 GRE Write Operation Immediate Response .....                   | 14 |
| Figure 4-7 GRE Write Operation Delayed Response .....                     | 15 |
| Figure 4-8 GRE Read Operation Immediate Response .....                    | 15 |
| Figure 4-9 GRE Read Operation Delayed Response .....                      | 16 |
| Figure 5-1 CAN Pinout Diagram .....                                       | 17 |
| Figure 8-1 Clock Structure .....                                          | 36 |
| Figure 12-1 CAN Configuration Interface .....                             | 39 |

# List of Tables

|                                                         |    |
|---------------------------------------------------------|----|
| Table 1-1 Abbreviations and Terminologies .....         | 2  |
| Table 3-1 Resource Utilization .....                    | 5  |
| Table 4-1 rxbhdr1 Format.....                           | 11 |
| Table 4-2 rxhdr2 Format.....                            | 11 |
| Table 4-3 txhdr1 Format.....                            | 13 |
| Table 4-4 txhdr2 Format.....                            | 13 |
| Table 5-1 Pin Description .....                         | 17 |
| Table 6-1 Address Map and Description of Registers..... | 21 |

# 1 About This Guide

## 1.1 Purpose

Gowin CAN Controller IP user guide is designed to help users to quickly learn the features of Gowin CAN Controller IP by providing descriptions of functions, GUI, and reference design, etc. The software screenshots and the supported products listed in this manual are based on Gowin Software 1.9.10.03. As the software is subject to change without notice, some information may not remain relevant and may need to be adjusted according to the software that is in use.

## 1.2 Related Documents

The latest user guides are available on GOWINSEMI Website. You can find the related documents at [www.gowinsemi.com](http://www.gowinsemi.com):

- [DS100, GW1N series of FPGA Products Data Sheet](#)
- [DS117, GW1NR series of FPGA Products Data Sheet](#)
- [DS102, GW2A series of FPGA Products Data Sheet](#)
- [DS226, GW2AR series of FPGA Products Data Sheet](#)
- [DS971, GW2AN-18X &9X Data Sheet](#)
- [DS976, GW2AN-55 Data Sheet](#)
- [DS961, GW2ANR series of FPGA Products Data Sheet](#)
- [SUG100, Gowin Software User Guide](#)
- [DS1228, Arora V FPGA Products Overview](#)

## 1.3 Abbreviations and Terminology

The abbreviations and terminologies used in this manual are set out in Table 1-1 below.

**Table 1-1 Abbreviations and Terminologies**

| Abbreviations and Terminology | Name                          |
|-------------------------------|-------------------------------|
| CAN                           | Controller Area Network       |
| FPGA                          | Field Programmable Gate Array |
| IP                            | Intellectual Property         |
| LUT                           | Look-up Table                 |

## 1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly using the information provided below.

Website: [www.gowinsemi.com](http://www.gowinsemi.com)

E-mail: [support@gowinsemi.com](mailto:support@gowinsemi.com)

# 2 Overview

This document describes the Controller Area Network (CAN) controller IP that implements the CAN2.0A, CAN2.0B as well as newer high performance Non ISO CAN-FD protocols. It can be integrated into devices that require CAN connectivity commonly used in automotive and industrial applications.

**Table 2-1 Gowin CAN Controller IP Overview**

| Gowin CAN Controller IP |                     |
|-------------------------|---------------------|
| Logic Resource          | See Table 3-1       |
| Delivered Doc.          |                     |
| Design Files            | Verilog (encrypted) |
| Reference Designs       | Verilog             |
| TestBench               | Verilog             |
| Test and Design Flow    |                     |
| Integrated Software     | GowinSynthesis      |
| Application Software    | Gowin Software      |

**Note!**

For the devices supported, you can click [here](#) to get the information.

# 3 Features and Performance

## 3.1 Key Features

- Compliant with CAN 2.0A and CAN 2.0B protocols, as well as ISO 11898-1 standard
- Support CAN-FD
  - Non-ISO CAN FD
  - Compliant to Bosch Protocol
  - Compliant to ISO/DIS 11898-1
- Independent System Clock(SYSCLK) and CAN Bus Clock(CANCLK)
  - SYSCLK is CPU interface Bus Clock and IP System Clock, which needs to be Continuous Clock and uninterruptible
  - CANCLK can either be independent or connected to SYSCLK
- Flexible shared buffering scheme that implements optimal buffer size for storing Transmit and Receive messages in a given application
  - The total buffer depth can be configured through parameters during IP generation
  - Transmit buffer, receiver buffer and high-priority transmit buffer
  - CPU can individually configure the depth of each buffer for transmission, reception, and high-priority transmission
- The receive filter can be configured for 1 to 16 filters
- Supports four CPU connection modes
  - AHB Slave Interface
  - APB Slave Interface
  - Wishbone Slave Interface
  - Register (REG) Interface
- Programmable Baud Rate Prescaler (BRP)
  - Generate Time Quantum Clock from CANCLK
  - 8-bit BRP register to have div-by-4 up to div-by-255

## 3.2 Working Frequency

The system supports two clocks: sysclk and canclk. Sysclk is typically a high-frequency clock, for example, 50MHz. Canclk is the CAN bus clock, and the TQ clock is derived by dividing canclk, depending on the CAN data bit width. In the IP design, sysclk and canclk are completely treated as asynchronous clocks, without requiring a defined phase relationship. In typical applications,  $\text{sysclk} \geq \text{canclk}$  is sufficient.

## 3.3 Resource Utilization

Gowin CAN Controller IP employs the Verilog language, which is used in the GW1N-9, GW1NR-9, GW2A, and GW2AR series of FPGA devices. Different modes have different resource consumption.

Table 3-1 presents an overview of the resource utilization. To validate alternative applications, please see the later release information.

**Table 3-1 Resource Utilization**

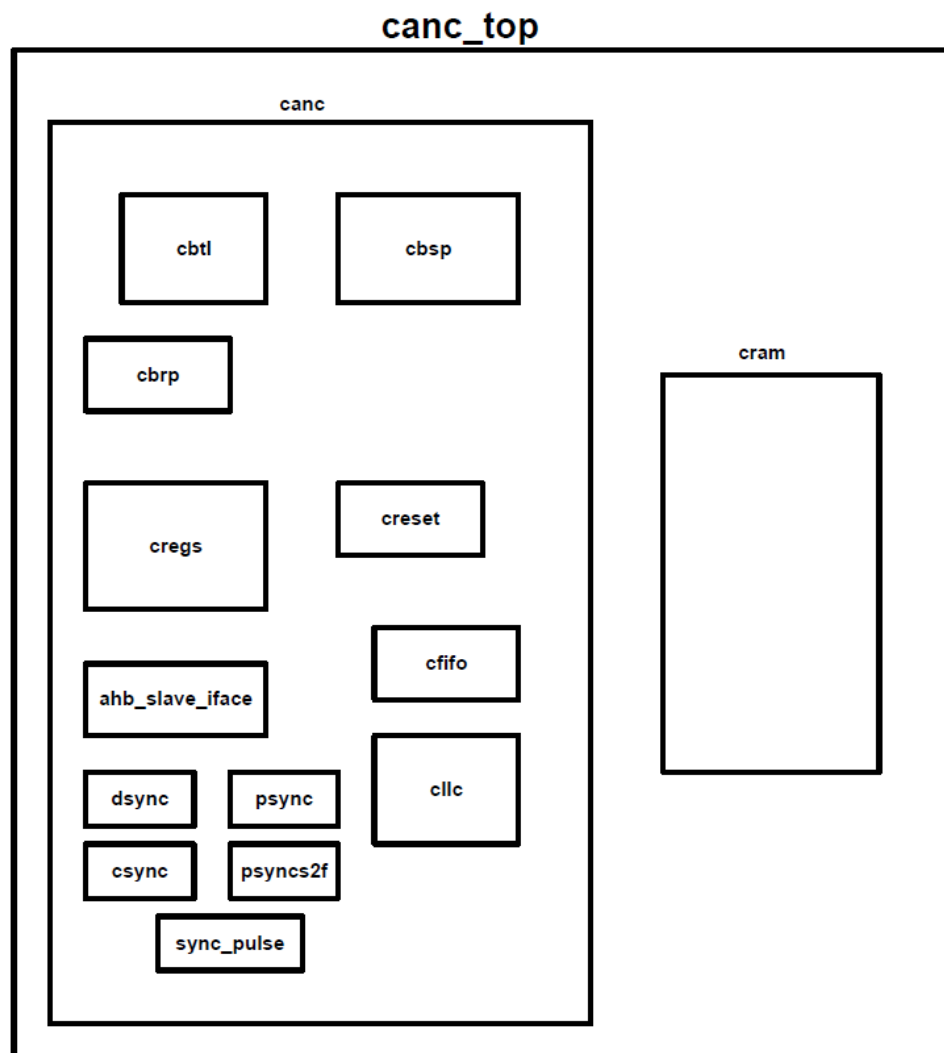
| LUTs | REGs | Block Rams | Device Series | Speed Level |
|------|------|------------|---------------|-------------|
| 4109 | 2217 | 1          | GW2A18        | -8          |

# 4 Structure and Functional Description

## 4.1 Module Structure

The Figure 4-1 shows internal module structure of the IP.

Figure 4-1 Module Structure

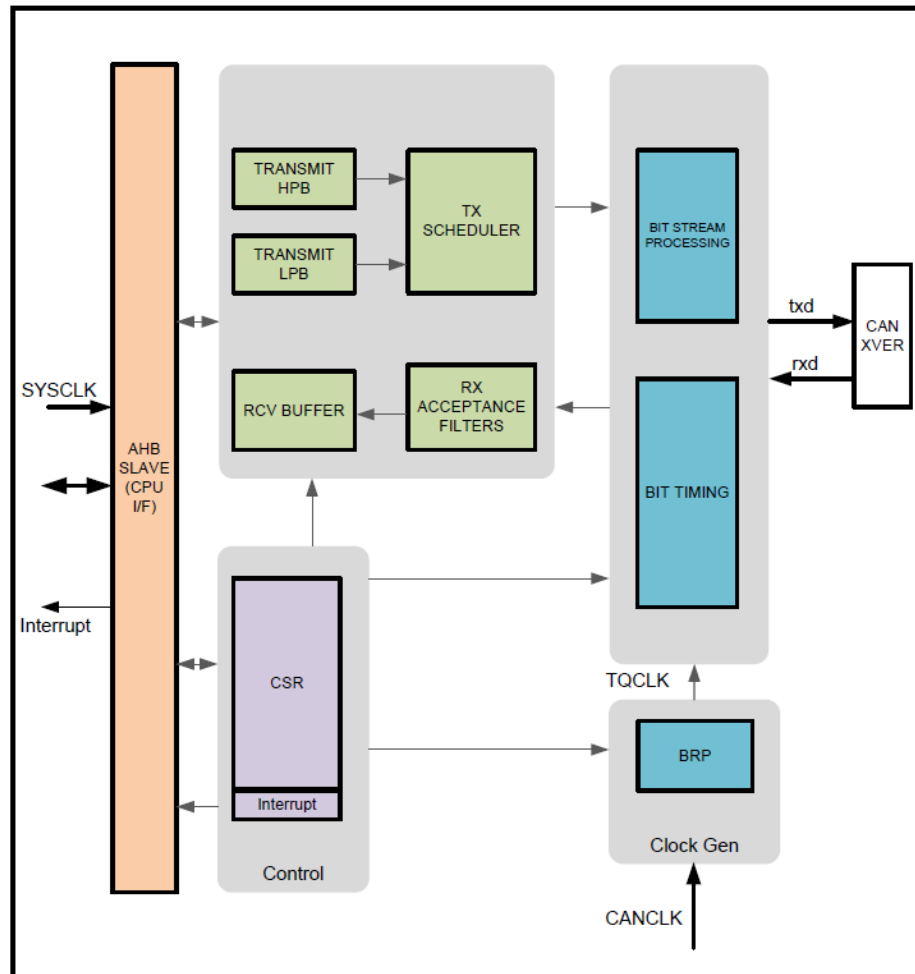


## 4.2 Functional Description

The CAN Bus Controller IP has been architected and designed to simplify the job of implementing CAN protocol in devices used automobile and industrial applications.

## 4.3 Function Block Diagram

Figure 4-2 CAN IP Functional Block Diagram



## 4.4 Function Block Description

The following sub-sections describe the behavior of major functions.

### 4.4.1 Bit Timing Logic (BTL)

The primary function of Bit Timing logic is to align the controller to incoming CAN traffic using Sync, Propagation Segment and Phase Segments. The segments are measured in units of quanta counted using a Time Quantum (TQ) Clock, generated by Baud Rate Prescaler (BRP) by dividing the CANCLK by a programmable value. Once synchronization is achieved, it generates sample reference to Bit Stream Processing (BSP) for higher level frame decoding function.

### 4.4.2 Bit Stream Processing (BSP)

The BSP block performs various higher level protocol functions typically classified as MAC functions. These include, checking and removal of CRC bits and stuffed bits in the receive direction. Similarly, the BSP block serializes the stream in the transmit direction, adds stuffing bits as needed and computes and adds CRC bits. It transfers complete frames on to the CAN Bus after arbitrating for the bus. It automatically retransmits the bus if access is lost in the arbitration process.

### 4.4.3 Baud Rate Prescaler (BRP)

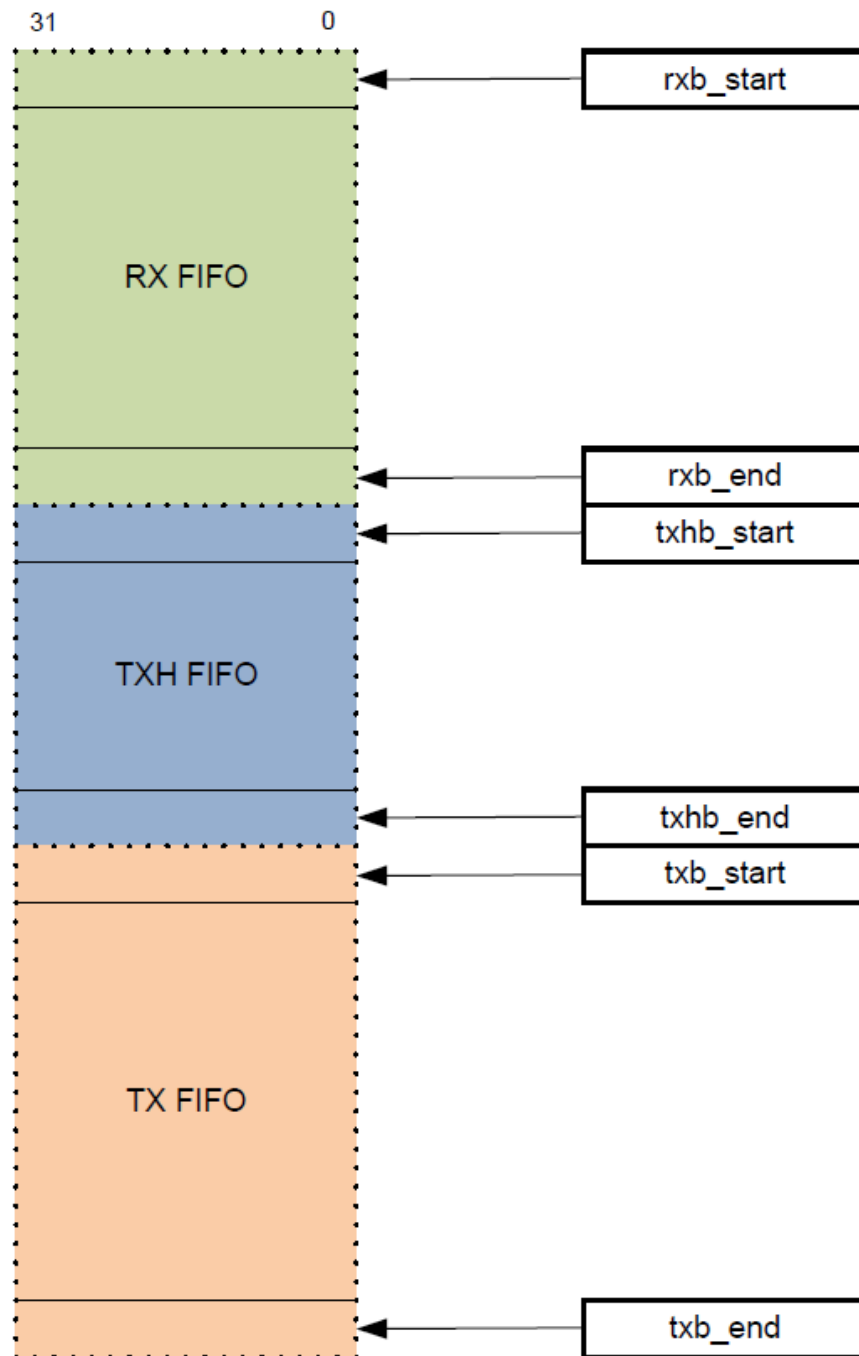
The Baud Rate Prescaler (BRP) block generates the Time Quantum Clock (TQ) by dividing input CANCLK by a programmed prescaler value, BRP. The register is 8-bits wide and supports up to divide-by-255. The TQ clock is used by BTL to align and sample the bit-stream.

### 4.4.4 Rx Acceptance Filters

This block filters incoming frames by comparing 11-bit or 29-bit IDENTIFIER in each message with number of Acceptance Filter Registers (AFR) along with corresponding Acceptance Filter Masks (AFM). The messages that pass any of the filters are accepted and stored in Rx Buffer. The AFR and AFM are individually programmable by the CPU. The number of filters is 1-16.

### 4.4.5 Shared Buffer

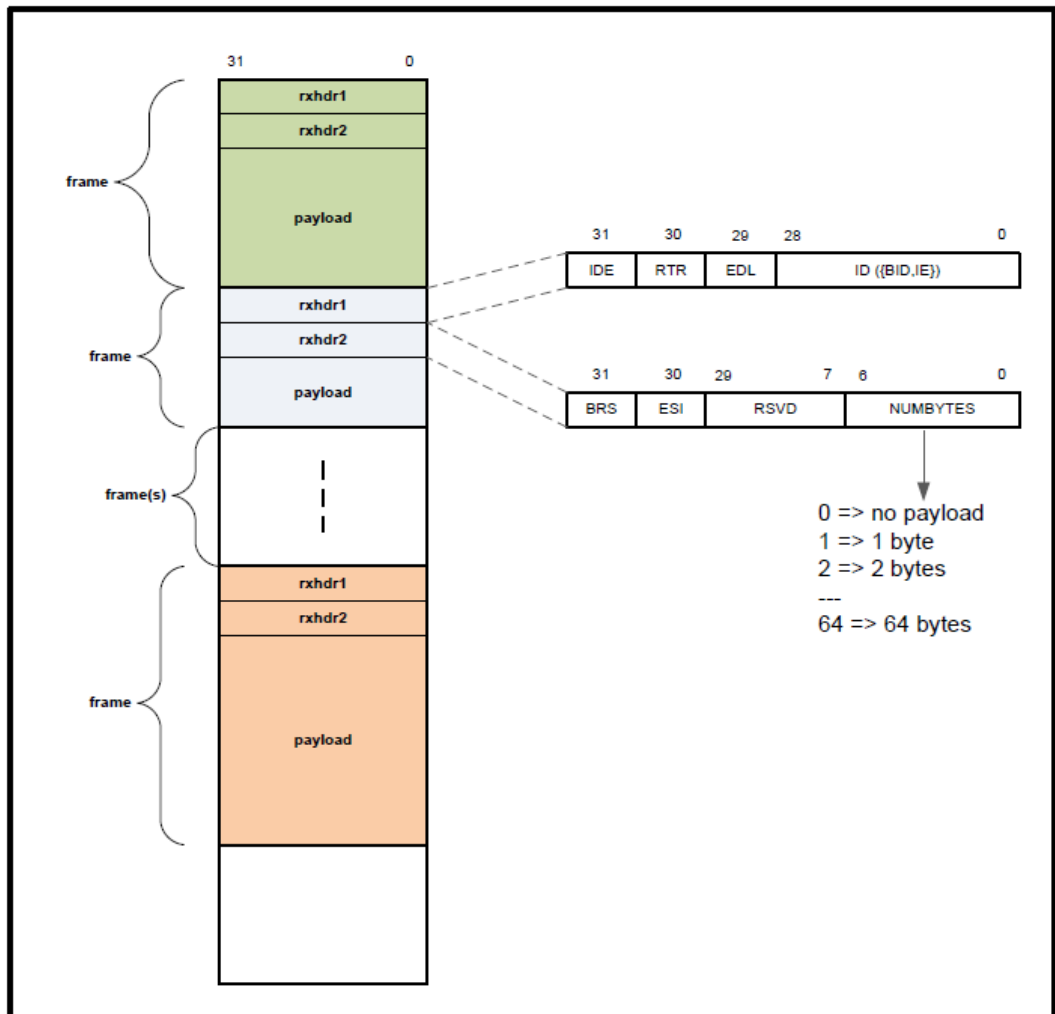
The CAN Controller IP implements shared buffering scheme to optimize the area for various applications. A single storage RAM is soft-partitioned into three individual circular FIFO. These three partitions serve as Receive Buffer (RB), Transmit High-Priority Buffer (THPB) and the Transmit Buffer (TB). The size of each partition is configured by the CPU when the controller is disabled (CMD.enable = 0). This gives massive flexibility in buffer space optimization depending on the applications the SoC is used in. The bytes are stored in big-endian format where byte 0 stored at bits 31:24, byte 1 at 23:16, byte 2 at 15:8 and byte 3 at 7:0. The Figure 4-2 illustrates it. The following sub-sections describe the format and function of individual FIFOs.

**Figure 4-3 Shared Buffer Illustration****Receive Buffer**

The Receive Buffer (RXB) stores received frames that are error free and passed acceptance filters. From here, they are read by the CPU over AHB Slave interface. Each message contains a pair of 32-bit headers indicating message identifier, length and other flags so CPU knows how many bytes to read. They are serviced first-in-first-out to CPU via a 32-bit window register. The receive buffer depth is run-time configurable. The number of messages stored varies depending on the depth and length of

individual messages. The Figure 4-3 shows the structure and format of the Rx FIFO.

**Figure 4-4 Receive Buffer/FIFO Structure and Format**



As can be seen, the controller writes two 32-bit headers; rxhdr1, rxhdr2 at the head of each received frame. These headers contain frame length and other control bits received. The CPU uses NUMBYTES field of rxhdr2 to know how many more words to read/pop beyond rxhdr2 to get next received frame. After the entire frame is read, the CPU checks the Rx Buffer status (RXBSTS.rxbdepth) to see if there are more frames received. Alternatively, the CPU could wait for RCVFVLD interrupt. The Table 4-1 and Table 4-2 describe the fields of rxhdr1, rxhdr2 respectively.

**Table 4-1 rxbhdr1 Format**

| 31                                                                                                                                 | 30                                                                                                                                        | 29                                                                                                                     | 28:0                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| IDE                                                                                                                                | RTR                                                                                                                                       | EDL                                                                                                                    | ID                                                                                                                          |
| Identifier Extension<br>1=>Extended<br>Format Frame with<br>29-bit identifier<br>0=>Base Format<br>Frame with 11-bit<br>identifier | Remote Transmit<br>Request<br>Reflects received<br>RTR bit<br>1 => Remote<br>Transmission<br>Request<br>0 => Normal frame<br>with payload | Extended Data<br>Length.<br>It is EDL bit<br>received in the<br>frame.<br>1 => FD format<br>frame<br>0 => Normal frame | Identifier<br><br>29-bit Identifier<br>28:18 => BID<br>17:0 => IE<br>Only bits 28:18<br>are valid for base<br>format frames |

**Table 4-2 rxhdr2 Format**

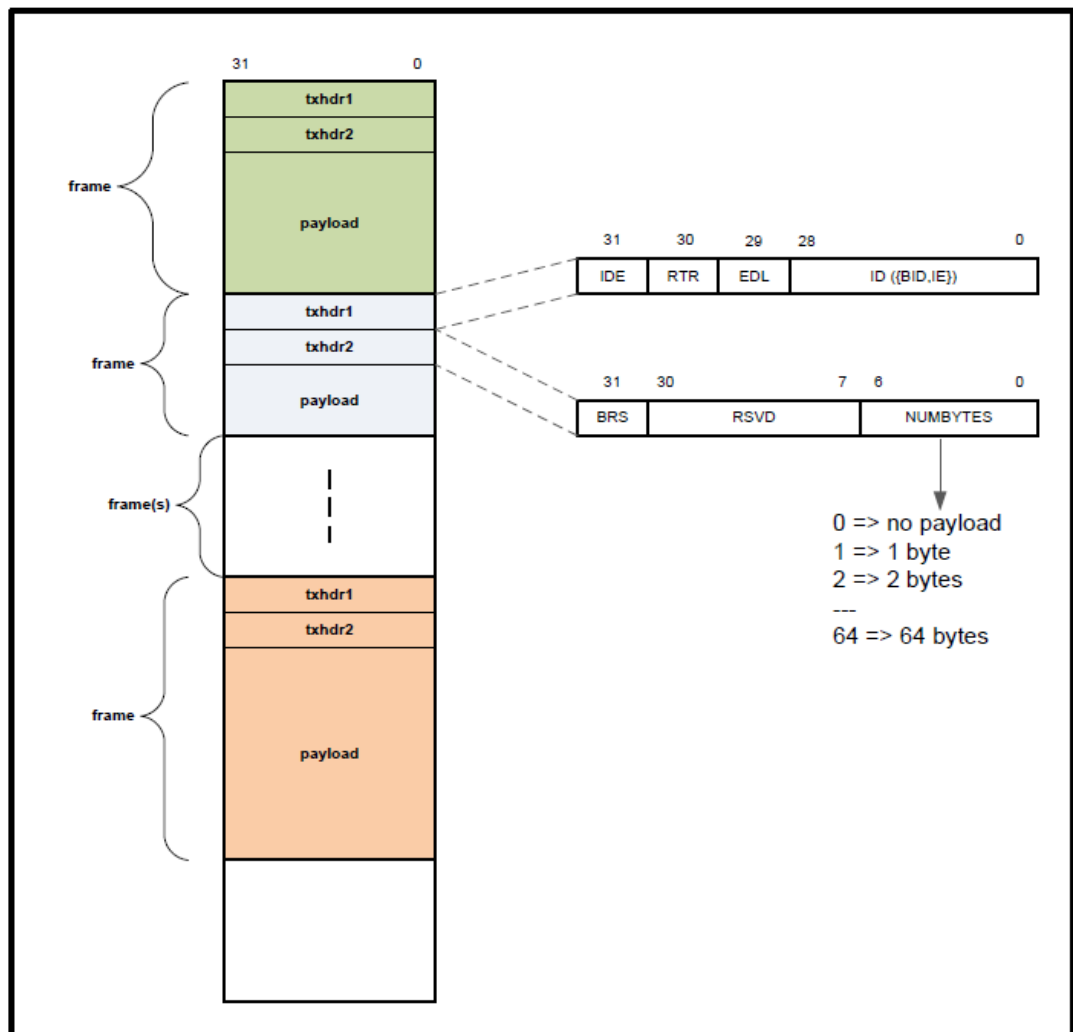
| 31                                                                                                                                                                                                                                                              | 30                                                                                                                                                                                                                                          | 29:7                             | 6:0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BRS                                                                                                                                                                                                                                                             | ESI                                                                                                                                                                                                                                         | RSVD                             | NUMBYTES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Bit Rate Switch<br>Only exists in FD<br>format frames.<br>Reflects the received<br>BRS bit.<br>1 => Bit rate is<br>switched to a<br>pre-configured<br>alternate rate during<br>Data Phase<br>0 => Bit rate is<br>nominal (not<br>switched) during<br>Data Phase | Error State<br>Indicator<br>Only exists in FD<br>format frames.<br>Reflects the<br>received ESI bit.<br>1 => Transmitter of<br>the message is in<br>error-passive state.<br>0 => Transmitter of<br>the message is in<br>error-active state. | Reserved. Not<br>currently used. | Frame Payload<br>Length in bytes.<br>Indicates number<br>of payload bytes of<br>the received frame,<br>also indicates<br>number of bytes for<br>CPU to read<br>beyond rxhdr2.<br><b>Note!</b> Messages<br>are stored in 32-bit<br>words and CPU<br>reads complete<br>words and discards<br>trailing bytes if any<br>in the last word<br>depending on the<br>length.<br>0 => No payload<br>RTR frames have<br>no payload<br>1 => 1 byte<br>2 => 2 bytes<br>----<br>64 => 64 bytes<br>OTHERS => Not<br>used |

### Transmit High-Priority Buffer

The Transmit High Priority Buffer (TXHB) is used for storing high priority messages that the CPU wants be transmitted at the higher priority. It operates as a circular FIFO. The Transmit Scheduler attends to these messages ahead of messages stored in low (normal) priority buffer. The

depth of TXHB is configurable by CPU. The number of messages that can be stored in TXHB depends on the configured depth and size of individual messages. Similar to Rx Buffer, each message in TXHB Buffer/FIFO has two header words containing frame length and other control information needed by the controller. The CPU writes the two headers; txhdr1, txhdr2 ahead of the payload for each packet. The Figure 4-4 shows the structure and format of the messages stored in THPB.

**Figure 4-5 Transmit High0-Priority Buffer/FIFO Structure AND Format**



As shown, the CPU writes two 32-bit headers; txhdr1, txhdr2 at the head of each received frame. These headers contain frame length and other control bits received. The controller uses NUMBYTES field of txhdr2 to know how many more words to read/pop beyond txhdr2 to get to end of the next transmit frame. The CPU checks to see if there is room for the next frame before writing/pushing next frame. The status is available in TXHBSTS.txhbspace. The space read should be greater than equal to the size of the payload of the next frame to be transmitted rounded to words + 2 (to account for the two headers). The TXBSTS.txbwerr may be checked at the end of writing each frame to see if the frame caused FIFO to overflow. The frame is discarded if overflow happens midway through the CPU pushing the frame into the FIFO. The Table 4-3 and Table 4-4 describe the

fields of txhdr1, txhdr2 respectively.

**Table 4-3 txhdr1 Format**

| 31                                                                                                                                                                                                             | 30                                                                                                                                                                  | 29                                                                                                                                                                                       | 28:0                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| IDE                                                                                                                                                                                                            | RTR                                                                                                                                                                 | EDL                                                                                                                                                                                      | ID                                                                                                   |
| Identifier Extension<br>Indicates if the controller transmits the frame in extended format or normal format.<br>1=>Extended Format Frame with 29-bit identifier<br>0=>Base Format Frame with 11-bit identifier | Remote Transmit Request<br>Indicates if it is Remote Transmission Request and has no payload.<br>1 => Remote Transmission Request<br>0 => Normal frame with payload | Extended Data Length. It is EDL bit received in the frame.<br>Indicates if the controller transmits the frame in FD format or normal format<br>1 => FD format frame<br>0 => Normal frame | 29-bit Identifier.<br>28:18 => BID<br>17:0 => IE<br>Only bits 28:18 are valid for base format frames |

**Table 4-4 txhdr2 Format**

| 31                                                                                                                                                                                                                                                        | 20:7                          | 6:0                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BRS                                                                                                                                                                                                                                                       | RSVD                          | NUMBYTES                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Bit Rate Switch<br>Only exists in FD format frames<br>Indicates if Bit rate is switched during Data Phase.<br>1 => Bit rate is switched to a pre-configured alternate rate during Data Phase<br>0 => Bit rate is nominal (not switched) during Data Phase | Reserved. Not currently used. | Frame Payload Length in bytes.<br>Indicates number of payload bytes of the received frame, also indicates number of bytes for CPU to write beyond rxhdr2. <b>Note!</b> Messages are stored in 32-bit words and CPU reads complete words and discards trailing bytes if any in the last word depending on the length.<br>0 => No payload<br>RTR frames have no payload<br>1 => 1 byte<br>2 => 2 bytes<br>----<br>64 => 64 bytes<br>OTHERS => Not used |

### Transmit Buffer (TXB)

The Transmit Buffer (TXB) stores frames that are marked by CPU as normal priority. The messages out of this buffer are serviced FIFO style and

scheduled at a lower priority, i.e., transferred if THPB doesn't have any complete messages. The depth of the TXB is also configurable by CPU. The structure and format of the Transmit Buffer/FIFO is similar to Transmit High-Priority Buffer described in the section above.

#### 4.4.6 Transmit Scheduler

The Tx Scheduler block implements a Strict-Priority (SP) scheduling between messages available in TB and THPB where messages in THPB have higher priority. The messages from both the buffers are serviced first-in-first out.

#### 4.4.7 Configuration and Status Registers (CSR)

The Configuration and Status Register (CSR) block implements various command, control and status registers needed for the operation of the controller and to pass the status to CPU. These include an Interrupt Status Register (ISR) as well.

#### 4.4.8 CPU Interface

Users can configure CPU interface as one of AHB slave interface, APB slave interface, Wishbone slave interface, and REG interface. Among them, AHB slave interface, APB slave interface and Wishbone slave interface comply with relevant protocol.

REG interfaces are divided into immediate response and delayed response. Immediate response: IP responds as soon as `cpu_cs` and `cpu_write/cpu_read` are valid. Delayed response: IP does not respond until `n` cycles of `cpu_cs` and `cpu_write/cpu_read` are valid. The immediate response and delayed responses cannot be configured by the user, depending on the address of the register read by the user.

The REG interface operation timing is shown below.

**Figure 4-6 GRE Write Operation Immediate Response**

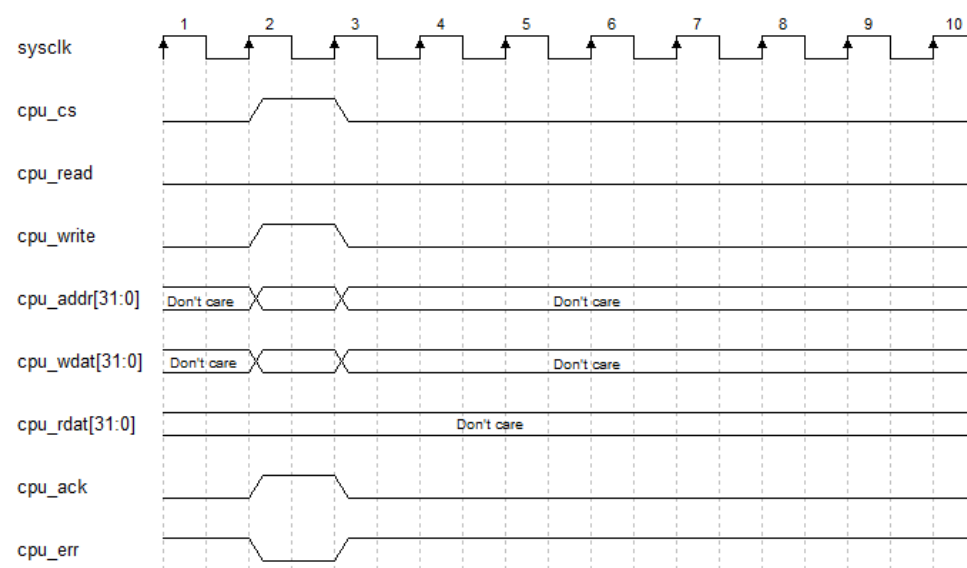


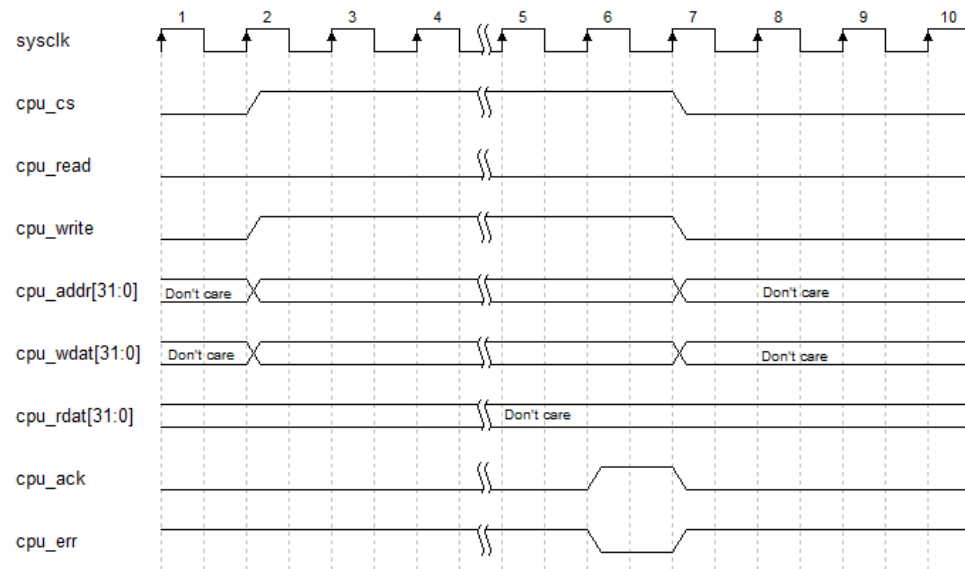
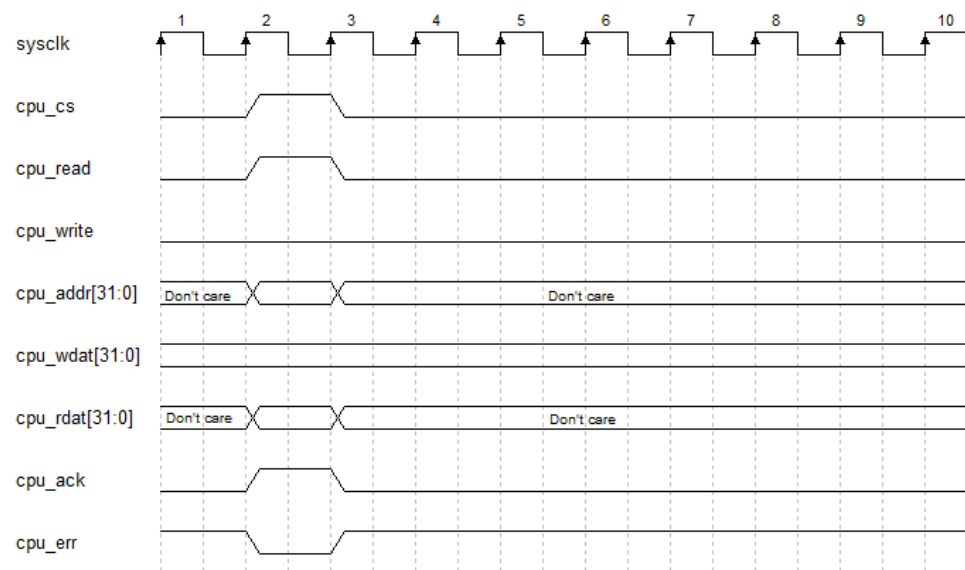
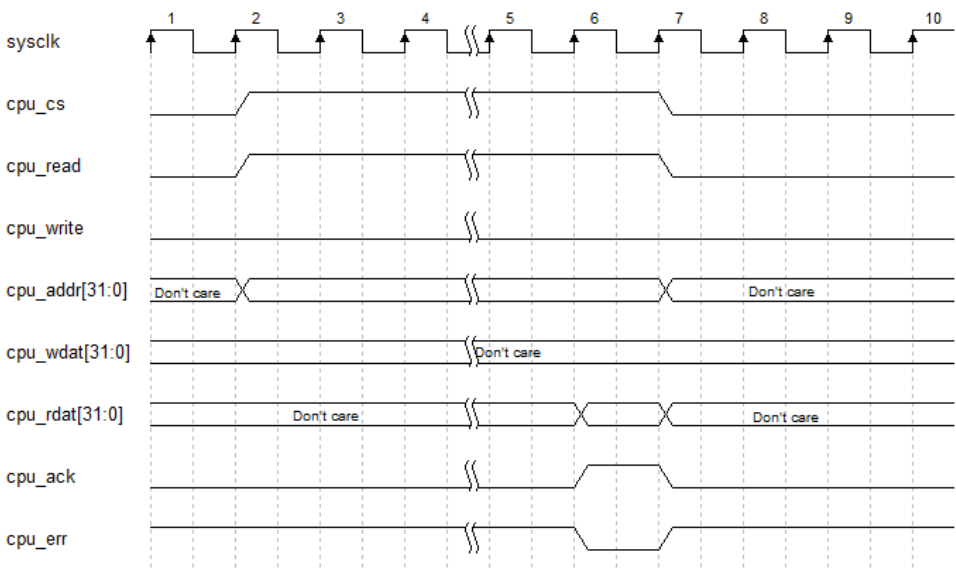
**Figure 4-7 GRE Write Operation Delayed Response****Figure 4-8 GRE Read Operation Immediate Response**

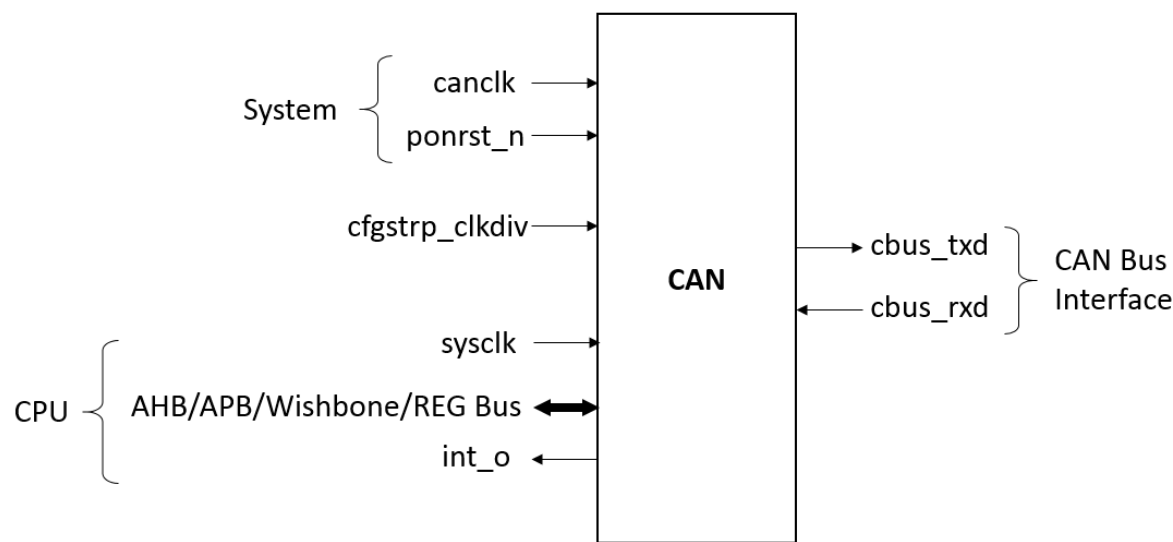
Figure 4-9 GRE Read Operation Delayed Response



# 5 Interface List

## 5.1 PIN Diagram

Figure 5-1 CAN Pinout Diagram



## 5.2 Pin Description

Table 5-1 Pin Description

| Pin              | Direction | Description                                                                                                                                                                                                                                                                                                                                         |
|------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Interface |           |                                                                                                                                                                                                                                                                                                                                                     |
| sysclk           | In        | System Clock Input. Serves as CPU interface clock. Typically AHB Clock if CPU is ARM. All the CSRs and message FIFOs are implemented in sysclk domain for fast CPU access. The sysclk is independent of canclk used for CAN Bus interface, and no phase relationship is assumed between the two, provided that $\text{sysclk} \geq \text{canclk}$ . |
| canclk           | In        | CAN Bus Interface Clock.                                                                                                                                                                                                                                                                                                                            |

| Pin                                | Direction      | Description                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                    |                | Serves as input clock for the internal prescaler that generates Time Quantum(TQ) clock used for sampling as well as transmitting the bitstream over the CAN Bus. Should be at least 2X of the desired time-quantum clock.                                                                                                                            |
| ponrst_n                           | In             | Active low Power On Reset. Should stay asserted for at least 4 sysclk clock periods. May be tied to hreset_n if AHB is the host CPU interface.                                                                                                                                                                                                       |
| cfgstrp_clkdiv[7:0]                | In (Pin Strap) | Initial Pin-Strap Clock Divider value. Interpreted as follows:<br>0..3 => Reserved<br>4 => div-by-4<br>5 => div-by-5<br>---<br>---<br>255 => div-by_255<br><b>Note!</b> The cfgstrp_clkdiv is read during power on reset and serves as initial divider value. The CPU may alter the frequency later by writing to BRP(baud rate prescaler) register. |
| CAN Bus Interface                  |                |                                                                                                                                                                                                                                                                                                                                                      |
| cbus_rxd                           | In             | CAN Bus serial data single-ended input from the transceiver.<br>Connected to RXD output pin of the transceiver.                                                                                                                                                                                                                                      |
| cbus_txd                           | Out            | CAN Bus serial data single-ended output. Connected to TXD input pin of the transceiver.                                                                                                                                                                                                                                                              |
| CPU Bus Interface - AHB-Lite Slave |                |                                                                                                                                                                                                                                                                                                                                                      |
| sysclk                             | In             | AHB system clock.<br>All the AHB clock domain signals are sampled and driven in this AHB system clock. In addition, internal CSR and message buffer also use system clock to accelerate CPU access speed.                                                                                                                                            |
| t_hsel                             | In             | Slave select                                                                                                                                                                                                                                                                                                                                         |
| t_haddr[13:0]                      | In             | Byte address                                                                                                                                                                                                                                                                                                                                         |
| t_hwdata[31:0]                     | In             | Write data                                                                                                                                                                                                                                                                                                                                           |
| t_hrdata[31:0]                     | Out            | Read data                                                                                                                                                                                                                                                                                                                                            |
| t_hwrite                           | In             | Write / Read direction indication                                                                                                                                                                                                                                                                                                                    |
| t_hsize[1:0]                       | In             | Size ( Byte – 8bit , Half word – 16bit, Word -32 bit)                                                                                                                                                                                                                                                                                                |
| t_htrans[1:0]                      | In             | Transfer type<br>2'b00 - Idle<br>2'b10 - Non Sequential                                                                                                                                                                                                                                                                                              |

| Pin                              | Direction | Description                                                                                                                                                                                                         |
|----------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| t_hresp[1:0]                     | Out       | Transfer Response<br>2'b00 – Okay<br>We are not sending Error Responsne.                                                                                                                                            |
| t_hready                         | Out       | Transfer complete                                                                                                                                                                                                   |
| t_hready_in                      | In        | Slave Ready input                                                                                                                                                                                                   |
| CPU Bus Interface-APB Slave      |           |                                                                                                                                                                                                                     |
| sysclk                           | In        | APB system clock.<br>All APB clock domain signals are sampled and driven in the APB system clock. In addition, internal CSR and message buffer also use system clock to accelerate CPU access speed.                |
| PADDR[13:0]                      | In        | Byte address                                                                                                                                                                                                        |
| PWDATA[31:0]                     | In        | Write data                                                                                                                                                                                                          |
| PRDATA[31:0]                     | Out       | Read data                                                                                                                                                                                                           |
| PSEL                             | In        | Slave select                                                                                                                                                                                                        |
| PWRITE                           | In        | Write/Read direction indication                                                                                                                                                                                     |
| PENABLE                          | In        | Enable signal                                                                                                                                                                                                       |
| PREADY                           | Out       | Ready                                                                                                                                                                                                               |
| PSLVERR                          | Out       | Transfer fail                                                                                                                                                                                                       |
| CPU Bus Interface-Wishbone Slave |           |                                                                                                                                                                                                                     |
| sysclk                           | In        | Wishbone system clock.<br>All wishbone clock domain signals are sampled and driven in the wishbone system clock. In addition, internal CSR and message buffer also use system clock to accelerate CPU access speed. |
| CYC_I                            | In        | Bus cycle signal                                                                                                                                                                                                    |
| STB_I                            | In        | Strobe signal                                                                                                                                                                                                       |
| WE_I                             | In        | Write/Read direction indication                                                                                                                                                                                     |
| ADR_I[13:0]                      | In        | Address signal                                                                                                                                                                                                      |
| DAT_I[31:0]                      | In        | Write data                                                                                                                                                                                                          |
| ACK_O                            | Out       | Acknowledge signal                                                                                                                                                                                                  |
| DAT_O[31:0]                      | Out       | Read data                                                                                                                                                                                                           |
| CPU Bus Interface-REG            |           |                                                                                                                                                                                                                     |
| sysclk                           | In        | REG system clock.<br>All REG clock domain signals are sampled and driven in the REG system clock. In addition, internal CSR and message buffer also use system clock to accelerate CPU access speed.                |
| cpu_cs                           | In        | Chip selection signal                                                                                                                                                                                               |
| cpu_read                         | In        | Read enable                                                                                                                                                                                                         |
| cpu_write                        | In        | Write enbale                                                                                                                                                                                                        |
| cpu_addr[13:0]                   | In        | Address signal                                                                                                                                                                                                      |

| Pin                     | Direction | Description                      |
|-------------------------|-----------|----------------------------------|
| cpu_wdat[31:0]          | In        | Write data                       |
| cpu_rdat[31:0]          | Out       | Read data                        |
| cpu_ack                 | Out       | Transfer acknowledge response    |
| cpu_err                 | Out       | Transfer error indication        |
| CPU Interruption Signal |           |                                  |
| int_o                   | In        | Interruption signal, active high |

# 6Registers

The CAN Controller IP uses 64KB address space, so decodes lower 16-bits of CPU/System Bus address as offset value for register bank. It maps all the control and status registers as well as window registers for accessing Tx and Rx buffers into this address space. All registers DWORD (32-bit) accessed. The address map and description of registers are as shown in Table 6-1.

**Table 6-1 Address Map and Description of Registers**

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|----------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0000       | SRST     | None                | <p>Software Reset Register</p> <p>CPU writing '1' to bit 0 initiates hard reset of the controller which resets the entire controller just like power-on-reset.</p> <p>Writing '0' de-asserts the reset.</p> <p>0 =&gt; Soft Reset</p> <p>1 =&gt; Completely resets the controller including configuration and status registers. The BRP is not reset and continue to generate tqclk</p> <p>0 =&gt; Reset is de-asserted</p> <p>Others =&gt; RSVD</p> <p><b>Note!</b> Internally soft reset is synchronized to tqclk domain which typically runs at much lower frequency, could be in the order of KHz. Therefore, CPU has to keep srst asserted for extended period of time to completely reset the controller. The minimum duration is at least 4 tqclk clock periods. It has to wait an additional 4 tqclk clock periods after de-assertion before proceeding to re-initialization. Since BRP register that controls the tqclk clock period itself is reset during soft reset, it defaults</p> |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------|----------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |          |                     | to (canclk frequency / BRP.brp[7:0]).<br>The BRP.brp[7:0] default value of the register which is set to 0x4 (div-by-4 of canclk frequency)                                                                                                                                                                                                                                                                    |
| 0x0004       | CMD      | None                | Command Register<br>0 => enable (default: 0x0)<br>1'b1: Operational mode<br>1'b0: Command mode<br>Others => RSVD<br>Writing a '1' to enable bit puts the controller in the operational state. It is written to '1' after programming rest of the operational parameters such as BRP, BTN and BTD.                                                                                                             |
| 0x0008       | BRP      | None                | Baud Rate Prescaler Register.<br>7:0 => brp (default: 0x0)<br>Others => RSVD<br>Prescaler value used to generate TQ clock from canclk by integer division.<br>8'd0, 8'd1, 8'd2, 8'd3 => reserved.<br>(div_by_4 chosen by default)<br>8'd4 => div-by-4<br>8'd5 => div-by-5<br>---<br>8'd255 => div-by-255<br><b>Note!</b> It is programmed before enabling the controller by writing a '1' to CMD. enable bit. |
| 0x000C       | BTN      | None                | Bit Timing (Nominal) Register.<br>This register defines bit timing parameters for nominal or normal rate. Units are in TQ clock periods.<br>5:0 => phseg1_nom (default: 0x0)<br>Defines the width of PROP_SEG + PHASE_SEG1 window.<br>13:8 => phseg2_nom (default: 0x0)<br>Defines the width of PHASE_SEG2 window.<br>28:24 => sjw_nom (default: 0x0)<br>Resynchronization Jump Width                         |
| 0x0010       | BTD      | None                | Bit Timing (Data) Register.<br>This register defines bit timing parameters for optional higher                                                                                                                                                                                                                                                                                                                |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |          |                     | <p>data rate allowed in FDS and used during DATA PHASE. The higher rate DATA PHASE is indicated by recessive BRS bit. Units are in TQ clock periods.</p> <p>3:0 =&gt; phseg1_d (default: 0x0)<br/>Defines the width of PROP_SEG + PHASE_SEG1 window.</p> <p>11:8 =&gt; phseg2_d (default: 0x0)<br/>Defines the width of PHASE_SEG2 window.</p> <p>26:24 =&gt; sjw_d (default: 0x0)<br/>Resynchronization Jump Width</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 0x0020       | IS       | Read-Only           | <p>Interrupt Status Register<br/>Indicates status of various events.</p> <p>CPU writes '1' to Interrupt Clear (IC) register to clear the individual bits.</p> <p>0 =&gt; RCVFVLD (default: 0x0)<br/>Rx FIFO Valid. When read as '1', indicates Receive FIFO has at least one valid frame received and available for CPU to read via Rx FIFO window register RXB. A '0' indicates Rx FIFO buffer is empty or doesn't contain any valid Rx messages.</p> <p>1 =&gt; RXBFULL (default: 0x0)<br/>Rx FIFO Overflow. When read as "1", it indicates that an RX buffer overflow occurs while a valid message was being written, and the message is dropped.</p> <p>4 =&gt; TXFOVF (default: 0x0)<br/>Tx FIFO Overflow indicator. The partial frame is discarded and not transmitted if Tx buffer overflows midway through the transmitting frame. This is the error scenario as normal access procedure doesn't result in this bit getting set. The CPU is expected to check the space in the Tx Buffer/FIFO before beginning to write the frame into Tx FIFO</p> <p>5 =&gt; TXHFOVF (default: 0x0)<br/>Tx High-Priority FIFO Overflow indicator. The partial frame is discarded and not transmitted if</p> |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|----------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |          |                     | <p>Tx buffer overflows midway through the transmitting frame. This is the error scenario as normal access procedure doesn't result in this bit getting set. The CPU is expected to check the space in the Tx Buffer/FIFO before beginning to write the frame into Tx HP FIFO</p> <p>8 =&gt; ERRSTS (default: 0x0)</p> <p>When read '1', indicates that one or more bits in ERRSTS register (0x60) have been set. Upon detection, CPU reads ERRSTS register contents for further details, i.e., know type of errors that have been set. CPU first clears all the bits that have been set in ERRSTS register, followed by this bit itself.</p> <p>27 =&gt; TXMSGOK</p> <p>When read '1', indicates last Tx message transmitted successfully. The message ID is recorded in TXMSGSTS register</p> <p>26 =&gt; TXMSGRETRY</p> <p>When read '1', indicates that last Tx message was retried due to earlier attempt not successful. The message ID is recorded in TXMSGSTS register</p> <p>25 =&gt; TXMSGFAIL</p> <p>When read '1', indicates that the last Tx message transmission including retry attempts failed and message was discarded. The message ID is recorded in TXMSGSTS register.</p> <p>23 =&gt; TXHMSGOK</p> <p>When read '1', indicates last HP Tx message transmitted successfully. The message ID is recorded in TXHMSGSTS register</p> <p>22 =&gt; TXHMSGRETRY</p> <p>When read '1', indicates that last HP Tx message was retried due to earlier attempt not successful. The message ID is recorded in TXHMSGSTS</p> |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------|----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |          |                     | <p>register</p> <p>21 =&gt; TXHMSGFAIL<br/>When read '1', indicates that the last HP Tx message transmission including retry attempts failed and message was discarded. The message ID is recorded in TXHMSGSTS register.</p> <p>31 =&gt; BUSOFF (default: 0x0)<br/>When read '1', indicates that controller is in BUSOFF state, due to excessive transmit errors, ie., transmit error count (TEC) &gt;= 256.</p> <p>OTHERS =&gt; RSVD</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 0x0024       | IE       | None                | <p>Interrupt Enable Register</p> <p>Enable for each interrupt source listed in IS register.</p> <p>0 =&gt; RCVFVLDE (default: 0x0)<br/>Enable for RCVFVLD</p> <p>1 =&gt; RXBFULLE (default: 0x0)<br/>Enable for RXBFULL</p> <p>4 =&gt; TXFOVFE (default: 0x0)<br/>Enable for TXFOVF</p> <p>5 =&gt; TXHFOVFE (default: 0x0)<br/>Enable for TXHFOVF</p> <p>8 =&gt; ERRSTSE (default: 0x0)<br/>Enable for ERRSTS</p> <p>21 =&gt; TXHMSGFAILE (default: 0x0)<br/>Enable for TXHMSGFAIL</p> <p>22 =&gt; TXHMSGRETRYE (default: 0x0)<br/>Enable for TXHMSGRETRY</p> <p>23 =&gt; TXHMSGOKE (default: 0x0)<br/>Enable for TXHMSGOK</p> <p>25 =&gt; TXMSGFAILE (default: 0x0)<br/>Enable for TXMSGFAILE</p> <p>26 =&gt; TXMSGRETRYE (default: 0x0)<br/>Enable for TXMSGRETRY</p> <p>27 =&gt; TXMSGOKE (default: 0x0)<br/>Enable for TXMSGOK</p> <p>31 =&gt; BUSOFFE (default: 0x0)<br/>Enable for BUSOFF</p> <p>OTHERS =&gt; RSVD</p> |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0028       | IC       | Write-Only          | <p>Interrupt Clear Register</p> <p>Clear for each interrupt source listed in IS register.</p> <p>0 =&gt; RCVFVLD (default: 0x0)</p> <p>Clear for RCVFVLD</p> <p>1 =&gt; RXBFULL</p> <p>Clear for RXBFULL</p> <p>4 =&gt; TXFOVF (default: 0x0)</p> <p>Clear for TXFOVF</p> <p>5 =&gt; TXHFOVF (default: 0x0)</p> <p>Clear for TXHFOVF</p> <p>8 =&gt; ERRSTS (default: 0x0)</p> <p>Clear for ERRSTS</p> <p>21 =&gt; TXHMSGFAIL (default: 0x0)</p> <p>Clear for TXHMSGFAIL</p> <p>22 =&gt; TXHMSGRETRY (default: 0x0)</p> <p>Clear for TXHMSGRETRY</p> <p>23 =&gt; TXHMSGOK (default: 0x0)</p> <p>Clear for TXHMSGOK</p> <p>25 =&gt; TXMSGFAIL (default: 0x0)</p> <p>Clear for TXMSGFAIL</p> <p>26 =&gt; TXMSGRETRY (default: 0x0)</p> <p>Clear for TXMSGRETRY</p> <p>27 =&gt; TXMSGOK (default: 0x0)</p> <p>Clear for TXMSGOK</p> <p>31 =&gt; BUSOFF (default: 0x0)</p> <p>Clear for BUSOFF</p> <p>OTHERS =&gt; RSVD</p> |
| 0x0030       | CFG      | None                | <p>Configuration Register</p> <p>0 =&gt; iso fd (default 0x0)</p> <p>0: Non-ISO FD mode (Bosch)</p> <p>1: ISO FD mode</p> <p>If ISO FD mode enables stuff counter at the beginning of the CRC field as described in ISO_DIS_11898-1 specification</p> <p>4 =&gt; disprotexceponres (default 0x0)</p> <p>0: recessive 'res' bit is treated as protocol exception event</p> <p>1: recessive 'res' bit is treated as FORM-ERROR</p> <p>is applicable and interpreted only if the node is operating in ISO FD mode</p>                                                                                                                                                                                                                                                                                                                                                                                                     |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|----------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0040       | RXBCFG   | None                | <p>Rx Buffer/FIFO Configuration Register</p> <p>Defines the Rx Buffer size (in 32-bit words) of Receive Buffer. It is a soft partition in a shared buffer shared by TXBUF and TXHBUF.</p> <p>15:0 =&gt; rxb_start (default: 0x0)<br/>Start offset of Rx Buffer</p> <p>31:16 =&gt; rxb_end (default: 0x0)<br/>End offset of Rx Buffer</p> <p><b>Note!</b> Number of bits used/programmed should match shared buffer address width, ie., for example, if shared buffer depth is 256 words indicating a 1K Bytes total buffer space, only lower 8-bits are used, upper bits if any programmed alias to lower 256 locations and may overlap with other two FIFO.</p>  |
| 0x0044       | TXBCFG   | None                | <p>Tx Buffer/FIFO Configuration Register</p> <p>Defines the Tx Buffer size (in 32-bit words) of Receive Buffer. It is a soft partition in a shared buffer shared by RXBUF and TXHBUF.</p> <p>15:0 =&gt; txb_start (default: 0x0)<br/>Start offset of Rx Buffer</p> <p>31:16 =&gt; txb_end (default: 0x0)<br/>End offset of Rx Buffer</p> <p><b>Note!</b> Number of bits used/programmed should match shared buffer address width, ie., for example, if shared buffer depth is 256 words indicating a 1K Bytes total buffer space, only lower 8-bits are used, upper bits if any programmed alias to lower 256 locations and may overlap with other two FIFOs.</p> |
| 0x0048       | TXHBCFG  | None                | <p>Tx High-Priority Buffer/FIFO Configuration Register</p> <p>Defines the Tx HP Buffer size (in 32-bit words) of Receive Buffer. It is a soft partition in a shared buffer shared by RXBUF and TXBUF.</p> <p>15:0 =&gt; txhb_start (default: 0x0)<br/>Start offset of Rx Buffer</p> <p>31:16 =&gt; txhb_end (default: 0x0)</p>                                                                                                                                                                                                                                                                                                                                    |

| Offset/Range | Register      | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------|---------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |               |                     | <p>End offset of Rx Buffer</p> <p><b>Note!</b> Number of bits used/programmed should match shared buffer address width, ie., for example, if shared buffer depth is 256 words indicating a 1KB total buffer space, only lower 8-bits are used, upper bits if any programmed alias to lower 256 locations and may overlap with other two FIFOs.</p>                                                                                                                                                                                    |
| 0x0050       | TXBRET<br>RY  | None                | <p>Tx Buffer Retry Count (default: 0x0)</p> <p>Defines number of retries of a tx message in case of transmission errors. The default value (0x0) implies unlimited number of retries until the message is transmitted successfully.</p> <p><b>Note!</b> The message is discarded if the message is failed after number of retries in limited retry mode.</p> <p>The number of retries can be set to:</p> <p>0 =&gt; infinite<br/> 1 =&gt; 1 time<br/> 2 =&gt; 2 times<br/> ...<br/> 15 =&gt; 15 times</p>                             |
| 0x0054       | TXHBRE<br>TRY | None                | <p>Tx High-Priority Buffer Retry Count (default: 0x0)</p> <p>Defines number of retries of a high-priority tx message in case of transmission errors. The default value (0x0) implies unlimited number of retries until the message is transmitted successfully.</p> <p><b>Note!</b> The message is discarded if the message is failed after number of retries in limited retry mode.</p> <p>The number of retries can be set to.</p> <p>0 =&gt; infinite<br/> 1 =&gt; 1 time<br/> 2 =&gt; 2 times<br/> ...<br/> 15 =&gt; 15 times</p> |

| Offset/Range | Register  | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|-----------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0058       | TXMSGSTS  | Read-Only           | <p>Transmit Message Status Register</p> <p>Records the status of last transmit message</p> <p>31:30 =&gt; txmsgsts</p> <p>Message Status</p> <p>2'b00: OK - message transmitted successfully</p> <p>2'b10: RETRIED - previous attempt failed and is being retried</p> <p>2'b11: FAILED and message was discarded</p> <p>28:0 =&gt; txmsgid</p> <p>Message ID.</p> <p>Only bits 28:18 valid for frames without extended header</p>                    |
| 0x005C       | TXHMSGSTS | Read-Only           | <p>Transmit High-Priority Message Status Register</p> <p>Records the status of last HP transmit message</p> <p>31:30 =&gt; txhmsgsts</p> <p>Message Status</p> <p>2'b00: OK - message transmitted successfully</p> <p>2'b10: RETRIED - previous attempt failed and is being retried</p> <p>2'b11: FAILED and message was discarded</p> <p>28:0 =&gt; txhmsgid</p> <p>Message ID.</p> <p>Only bits 28:18 valid for frames without extended header</p> |
| 0x0060       | ERRSTS    | None                | <p>Error Status Register.</p> <p>Records error events as defined in the Spec. until CPU clears by writing a '1' to the corresponding bits.</p> <p>0 =&gt; STUFF-ERROR</p> <p>1 =&gt; BIT-ERROR</p> <p>2 =&gt; FORM-ERROR</p> <p>3 =&gt; ACK-ERROR</p> <p>4 =&gt; CRC-ERROR</p>                                                                                                                                                                       |

| Offset/Range           | Register    | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------|-------------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0064                 | ERRCNT<br>R | Read-Only           | <p>Error Counter Register.</p> <p>Reflects the values of transmit and receive error counters, TEC and REC. The TEC and REC are incremented and decremented as defined in the Spec. This register shows a snapshot of these counters.</p> <p>8:0 =&gt; REC (default: 0x00)</p> <p>24:16 =&gt; TEC (default : 0x00)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 0x0100-0x100+<br>(4*N) | AF          | None                | <p>Receive Acceptance Filter Register</p> <p>Each filter has corresponding bit-mask register that CPU can program to selectively match a subset of identifier bits or other fields.</p> <p>Each AF is 32-bit register, the first filter starts at offset 0x100 and subsequent filters are located 4-bytes apart.</p> <p>Number of Acceptance Filters, N = 1-16.</p> <p>Each AF has the following bitfields.</p> <p>31 =&gt; Enable</p> <p>'1' =&gt; Filter is valid and used for matching incoming frames</p> <p>'0' =&gt; Filter is not valid, not used to match incoming frames</p> <p>28:18 =&gt; Base Identifier (BID)</p> <p>17:0 =&gt; Identifier Extension (IE)</p> <p>IE is matched/used with incoming extension frames if filter is configured to be extended identifier by setting IDE bit</p> <p>30 =&gt; IDE</p> <p>Selects if the filter is to be used with extension frames or normal frames</p> <p>'1' =&gt; Use with extension frames</p> <p>'0' =&gt; Use with normal frames</p> <p>29 =&gt; EDL (Extended Data Length)</p> <p>Used to match frames that are in FD format, indicated by recessive EDL bit</p> <p>'1' =&gt; Matches FD frames, normal frames fail this filter</p> <p>'0' =&gt; Matches normal frames,</p> |

| Offset/Range           | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------|----------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        |          |                     | FD format frames fail this filter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 0x0140-0x140+<br>(4*N) | AFM      | None                | <p>Receive Acceptance Filter Mask Register</p> <p>Mask vector for each acceptance filter. Only bits that are masked ('1') are compared/matched in each filter, the bits are not masked ('0') are treated as wild-cards. Each AFM is 32-bit register, the first filter starts at offset 0x100 and subsequent filters are located 4-bytes apart.</p> <p>Number of Acceptance Filters, N = 1-16.</p> <p>Each AF has the following bitfields.</p> <p>28:18 =&gt; Base Identifier (BID) Mask</p> <p>Bitmask for BID field</p> <p>17:0 =&gt; Identifier Extension (IE) Mask Bitmask for IE field.</p> <p>OTHERS =&gt; RSVD</p> |
| 0x0200                 | RXB      | Read-Only           | <p>Receive Buffer/FIFO Window Register.</p> <p>Provides indirect access to Rx FIFO buffer.</p> <p>Each CPU read is treated as FIFO pop where next word from the Rx FIFO is loaded. The CPU starts reading when it learns that Rx FIFO is valid (contains at least one valid frame) until all the words of the frame have been popped or read. The first 2 words contain special header inserted by the controller and contain control information such as length(decoded value of DLC) and other control flags such as IDE, EDL, BRS etc.</p>                                                                            |
| 0x0204                 | TXB      | Write-Only          | <p>Transmit Buffer/FIFO Window Register.</p> <p>Provides indirect access to Tx FIFO.</p> <p>Each CPU write to this registers pushes a word into Tx FIFO and internal FIFO write-pointer is incremented. Once it learns of space availability, the CPU executes repeated writes until all bytes of the Tx frame have</p>                                                                                                                                                                                                                                                                                                  |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------|----------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |          |                     | been written (pushed) into the Tx FIFO. The first two writes contain header words that help the controller in formatting and transmission of the payload. The headers contain the length (to be encoded into DLC field by the controller) and other control information such as BID, IE, IDE, RTR, EDL and BRS.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 0x0208       | TXHB     | Write-Only          | <p>Transmit High-Priority Buffer/FIFO Window Register. Provides indirect access to THPB FIFO.</p> <p>Each CPU write to this registers pushes a word into THPB FIFO and internal FIFO write-pointer is incremented. Once it learns of space availability, the CPU executes repeated writes until all bytes of the Tx frame have been written (pushed) into the THPB FIFO. The first two writes contain header words that help the controller in formatting and transmission of the payload. The headers contain the length (to be encoded into DLC field by the controller) and other control information such as BID, IE, IDE, RTR, EDL and BRS.</p>                                                                                  |
| 0x020C       | TXBSTS   | Read-Only           | <p>Transmit Buffer/FIFO Status. Indicates the current status of Tx FIFO, read by CPU before writing/pushing a new transmit frame into the FIFO.</p> <p>15:0 =&gt; txbspace<br/>Space available in 32-bit words. The CPU is expected to check the space before writing each new transmit frame. The space should be &gt;= number of words in payload + 2 words (to account for transmit headers to be filled in/written by the CPU)</p> <p>31 =&gt; txbwerr<br/>If the (CPU) writes (pushes) the data causing a TX FIFO overflow, then the bit is set. It is recommended that the CPU reads the bit after the last frame of each send has been written to TX fifo to ensure that the bit is zero. <b>Note!</b> If overflow happens</p> |

| Offset/Range | Register | Access Restrictions | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|----------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |          |                     | the transmit packet is discarded and not transmitted onto CAN bus.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 0x0210       | TXHBSTS  | Read-Only           | <p>Transmit High-Priority Buffer/FIFO Status.</p> <p>Indicates the current status of Tx HP FIFO, read by CPU before writing/pushing a new transmit frame into the FIFO.</p> <p>15:0 =&gt; txhbospace</p> <p>Space available in 32-bit words. The CPU is expected to check the space before writing each new transmit frame. The space should be &gt;= number of words in payload + 2 words (to account for transmit headers to be filled in/written by the CPU)</p> <p>31 =&gt; txhbwerr</p> <p>If the CPU writes (pushes) the data causing a TXH FIFO overflow, then the bit is set. It is recommended that the CPU read the bit after the last frame of each sending/transmission is written to TX FIFO to ensure that the bit is zero. <b>Note!</b> If overflow happens the transmit packet is discarded and not transmitted onto CAN bus.</p> |
| 0x0214       | RXBSTS   | Read-Only           | <p>Receive Buffer/FIFO Status.</p> <p>Indicates the current status of the Rx FIFO. The CPU can read it to know if the Rx FIFO has any frames waiting to be read. This is an alternative way to waiting on RCVFVLD interrupt.</p> <p><b>Note!</b> Any value greater than zero indicates at least one complete frame as the status is updated by the controller on frame boundaries.</p> <p>Units in 32-bit words.</p> <p>15:0 =&gt; rxbdepth</p> <p>Occupancy level of Rx FIFO</p>                                                                                                                                                                                                                                                                                                                                                                 |

# 7 Interrupt

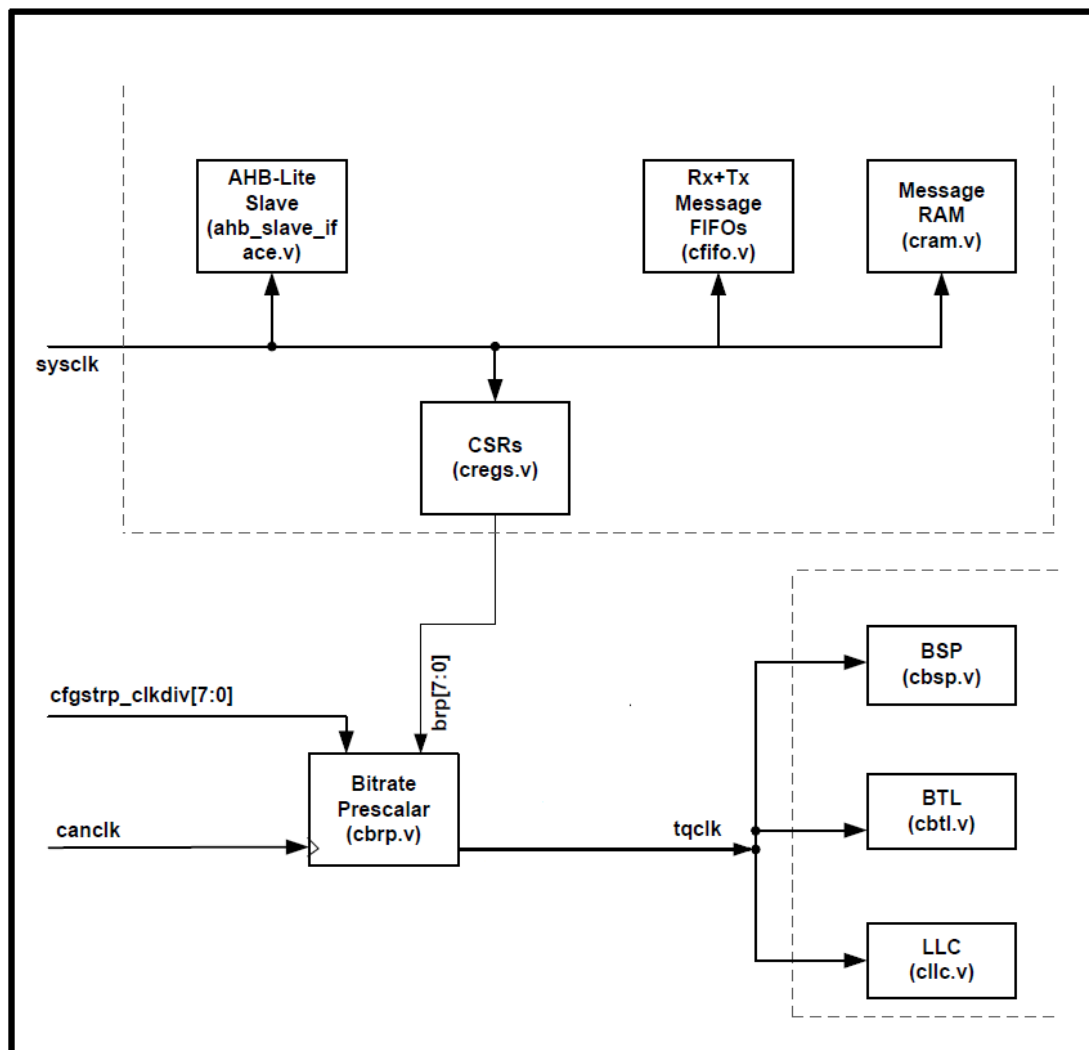
The CAN Controller IP implements a level interrupt that is active-high. It is a standard mechanism where all the enabled individual interrupt sources are ORed together to produce final interrupt output. The interrupt sources are defined in the Interrupt Status (IS) registers and the Interrupt Enable has corresponding masks.

# 8 Clocking

The Figure 8-1 shows Clocking scheme used in the CAN Controller IP. The IP has two clock input sources; canclk and sysclk. The canclk serves as reference time base for bitstream transfer from and to CAN Bus. The canclk is divided by the Bit-Rate Prescaler(BRP) to generate Time Quantum Clock (tqclk) that is the basis for serial bit timing. The BRP supports integer division and can go up to div-by-255. The frequency of the canclk must be at least four times that of the desired tqclk.

The sysclk is the second input source clock for the IP. It serves as CPU interface clock and IP System Clock and therefore tied to AHB Interface clock if ARM CPU is used, typically runs at much higher speed than the canclk. It is assumed that sysclk is either same or faster than canclk, however, no phase relationship is expected and are treated as asynchronous to each other.

Figure 8-1 Clock Structure



# 9 Reset

The CAN Controller IP uses asynchronous reset mechanism wherein all registers are reset asynchronously. The reset input going to flops is pre-synchronized to respective clock domain so the de-assertion of reset event is synchronous to the edge of the clock. The IP has two sources of reset; active-low power-on-reset and CPU initiated software reset. The power-on-reset may be tied to reset in AHB based systems. Both resets serve as hard resets where the controller is completely reset with the exception that software reset doesn't reset CPU interface block and one configuration register that implements software reset itself.

# 10 Example Initialization Sequence

The following is an example high-level CPU sequence of the Controller IP initialization.

## 10.1 Power-up

1. Set the Prescaler pin-strap `appstrp_clkdiv[7:0]` value to desired value
2. Start `sysclk` and `cancclk`
3. Assert power on reset `ponrst_n` as low
4. Wait for at least 8 `cancclk` clock periods.
5. De-assert `ponrst_n`
6. Wait for at least 32 `tqclk` clock periods and proceed to Initialization sequence outlined in Section 9.2.

**Note!** `Tqclk` clock period depends on `cancclk` frequency and the divider indicated on `appstrp_clkdiv [7:0]` input pin-strap. For example, for a divider value of 0x4, wait until 128 `cancclk` clock periods.

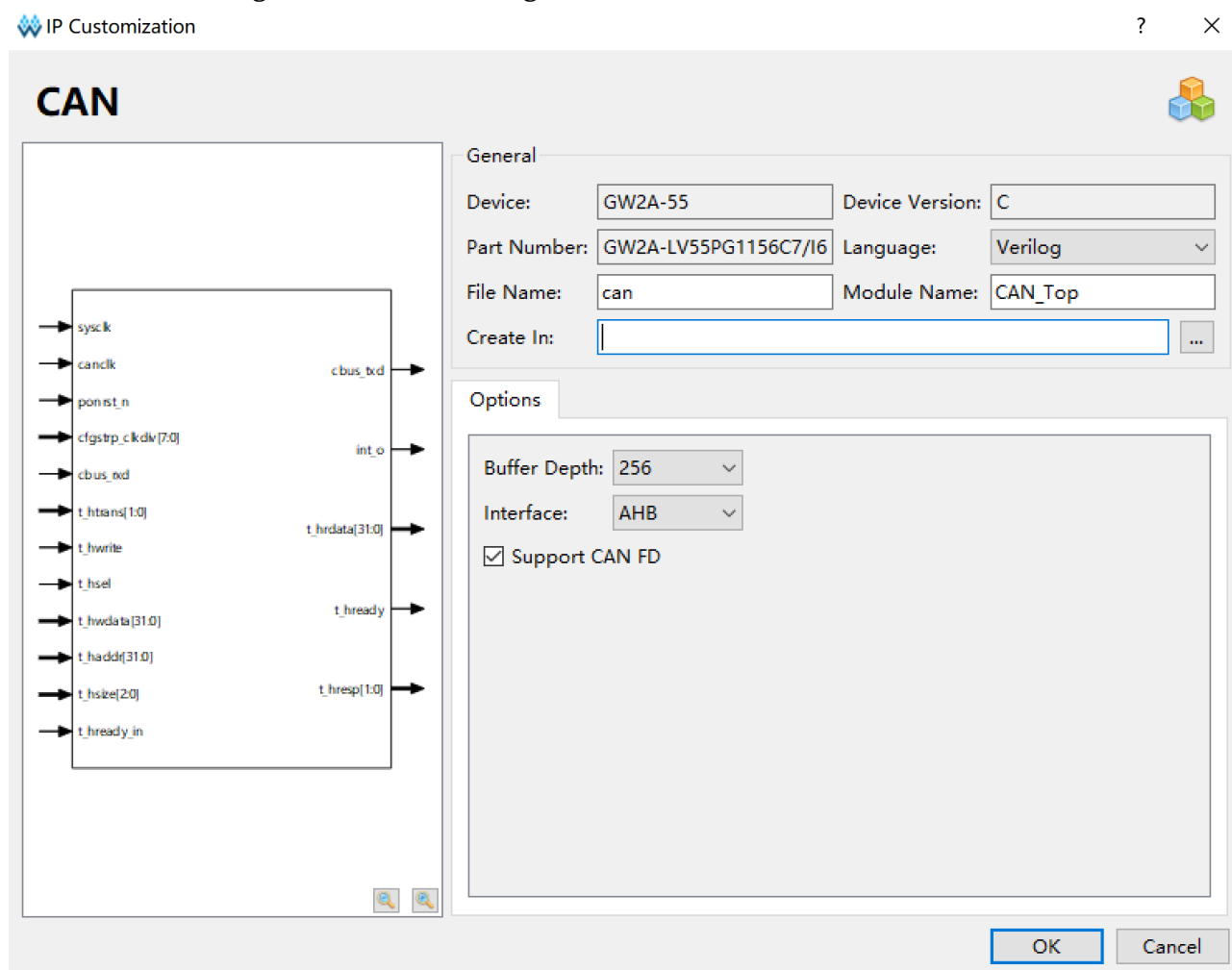
## 10.2 Initialization

1. Program BRP register with desired value of the divider in the Prescaler
2. Program framing related registers `BTN`, `BTD`
3. Configure buffer boundaries on registers `RXBCFG`, `TXBCFG`, and `TXHBCFG` to initialize transmit and receive buffer zones.
4. Setup enables for Interrupt sources in `IE` register.
5. Set up Acceptance Filters in the configuration registers `AF` and the corresponding masks in `AFM`.
6. Put the controller in the operational mode by writing a '1' to `CMD.enable` bit. Now the controller becomes operational and Bit Stream Processor (BSP) enters INTEGRATING state, after finding 11 consecutive recessive bits it enters IDLE state and ready to transmit and receive messages.

# 11 Interface Configuration

You can invoke and configure the Gowin CAN Controller IP using the IP Core Generator tool in the IDE. The CAN configuration interface is shown in Figure 12-1.

**Figure 11-1 CAN Configuration Interface**



- You can configure the depth of the buffer by modifying the value in "Buffer Depth" option.

- You can configure the type of CPU interface by modifying the value in "Interface" option.
- You can configure whether the IP supports CAN FD by checking the "Support CAN FD" option.

# 12 Reference Design

See the [reference design](#) of CAN for details.

