# CANFD INTERFACE (RS-CANFD)

This section contains a generic description of the CANFD interface (RS-CANFD).

The first part of this section describes all specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RS-CANFD.

### Features

#### Number of Units and Channels

#### Functional Overview

**Table 102.1 RS-CANFD Module Specifications**

|  |  |
| --- | --- |
| **Item** | **Specification** |
| Communication | CAN functionality conform to CAN-FD ISO 11898-1 (2015) |
| Protocol engine Version | RS-CANFD\_PE V3.0 |
| Gateway Function | CAN 2.0 <> CAN 2.0 CAN 2.0 <> CAN-FD Gateway (Only 8 Byte Payload) CAN-FD <> CAN-FD |
| Data transfer rate | up to 1Mbps for arbitration phase and up to 8Mbps for data phase , individually for each CAN channel |
| Proposed min. operation frequency peripheral clock/APB clock | 133.33MHz |
| Data Link Layer clock (DLLC) | 40MHz(Max) |
| Input/Output pins | TX/RX |
| CAN channels | 8 |
| Selectable ID type | 11-bit Standard ID |
|  | 11-bit Standard ID + 18-bit Extended ID |
| Selectable Frame Type | Data Frame (RTR = 0) (CAN and CAN-FD frames) |
|  | Remote Frame (RTR = 1) (only CAN frames) |
| Variable Data Byte Count for Data Frames | DLC range: 0 to F |
| Message Buffer | 64 transmit message buffers per channel \*\*1 4 transmission queue per channel Automatic message transfer into transmission queues supported |
|  | 256 shared buffers for RXMB and FIFO buffers per channel \*\*1 - RXMB: - Up to 16\*(n+1) reception message buffers, shared among all the CAN channels - FIFO buffers: - 8 Reception FIFO Buffers - Up to 3\*(n+1) FIFOs individually configurable as Reception FIFO / Transmission FIFO / CAN to CAN Gateway FIFO |
| Automatic delay interval timer for transmission | The delay timer can be applied to: -Transmission FIFO -CAN to CAN Gateway FIFO |
| Enhanced reception filtering | support of 11bit and 29bit CAN identifier |
|  | programmable 29 bit CAN identifier acceptance filter mask for each entry |
|  | programmable GW routing capability for each channel (up to 8 routing destinations) |
|  | RTR and IDE masking |
|  | DLC filter |
|  | Message buffer payload overload protection |
|  | Payload filter |
|  | Updating AFL entry during communication |
| General SW Support | Automatic label information added to receive message (for upper SW layer support) |
| Timer | TX and RX Time Stamp function |
| Power down function | Module start stop function for each CAN node (Channel & Global Sleep Mode) |
| RAM | RAM ECC protected |
| Bus traffic measurement | CAN bus traffic measurement of each Channel is possible |

#### Features of RS-CANFD

RS-CANFD has two interface modes (classical CAN mode and CAN FD mode), and uses different registers in each  
mode. The different register names RSCFD{unit}XXX and RSCFD{unit}CFDXXX are used in the different interface  
modes (XXX can be any character). In this document, the registers common to the two modes are indicated as  
RSCFD{unit}CFDXXX.

##### Number of Units and Channels

This microcontroller has the following number of RS-CANFD units.

**Number of Units and Channels**

|  |  |
| --- | --- |
| **Product Name** | **R-Car V4H** |
| Number of Units | 1 |
| Number of Channels | 8 |
| Number of Channel per Unit | 8 |
| Transmission Message Buffers per Channel | 64 |
| Shared Buffers for per Channel | 256 |
| Total Message Buffer | 2560 |

**Table 102.1 Index**

|  |  |
| --- | --- |
| **Index** | **Description** |
| unit | Throughout this section, the individual RS-CANFD units are generically indicated by the index “unit”. For example, RSCFD[unit]CFDGCTR is the global control register of the RSCFDunit unit. |
| n | Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index “n”. For example, RSCFD[unit]CFDCnSTS is the channel n status register. |
| m | The individual channels of RSCFD unit is generically indicated by the index “m”. For example, CANFDm\_RX is receive data input. |
| d | The individual common FIFO buffers are generically indicated by the index “d”. For example, RSCFD[unit]CFDCFCCd is the common FIFO buffer configuration/control register. |
| a | The individual receive FIFO buffers are generically identified by the index “a”. For example, RSCFD[unit]CFDRFSTSa is the receive FIFO buffer status register. |
| p | Data field registers of common FIFO buffers and receive FIFO buffers are identified by “p”. For example, RSCFD[unit]CFDCFDFp is the common FIFO access data field register. |
| i | The individual transmit control buffers are generically indicated by the index “i”. For example, RSCFD[unit]CFDTMCi is transmit message buffer control register. |
| j | The individual transmit status buffers are generically indicated by the index “j”. For example, RSCFD[unit]CFDTMSTSj is transmit message buffer status register. |
| k | The individual RAM tests for CAN are generically indicated by the index “k”. For example, RSCFD[unit]CFDRPGACCk is the RAM test page access register. |
| f | The transmit message buffer transmission are generically indicated by index “f”. For example, RSCFD[unit]CFDTMTRSTSf is the transmit message buffer transmission request status register. |
| w | A global register which setting for channels indicated by the index "w". For example, RSCFD[unit]CFDGAFLCFGw is global Acceptance Filter List configuration register. |
| v | A global register which setting for 04 channels indicated by the index “v”. For example, RSCFD[unit]CFDGTINTSTSv is the global transmit interrupt status register. |
| t | The receive message buffer new data are generically indicated by index “t”. For example, RSCFD[unit]CFDRMNDt is a receive message buffer new data register. |

Note 1. The functions and descriptions of registers in this section are for the RS-CANFDs that has 8 channels (m = 0 to 7).  
When referring to information with indices, regard the index values as the ones corresponding to the target product.  
Note 2. In some figure or table, this section use acronyms: CH, Ch, Ch instead of "channel".

**Table 102.2 Indices for Individual Products**

|  |  |
| --- | --- |
| **Index Correspondence of Each Product** | **R-CarV4H** |
| unit | 0 |
| n | 0 to 7 |
| m | 0 to 7 |
| d | 0 to 23 |
| a | 0 to 7 |
| p | 0 to 15 |
| i | 0 to 511 |
| j | 0 to 511 |
| k | 0 to 63 |
| f | 0 to 15 |
| w | 0 to 3 for RSCFD[unit]CFDGAFLCFGw 0 to 1 for RSCFD[unit]CFDGPFLCFGw |
| v | 0 to 1 |
| t | 0 to 3 |

#### **Interface Mode**

RS-CANFD has the following two interface modes.  
• Classical CAN mode: Only classical CAN frames are handled.  
• CAN FD mode: both the classical CAN frames and CAN FD frames are handled.  
Two modes use different register maps with the same base address, and the register maps are switched by switching the modes.  
Interface modes can be switched using the CLOE bit in the RSCFD{unit}CFDCnFDCFG register.

#### Block Diagram

The block diagram which shows connection with a peripheral module is shown

A diagram of a computer

Description automatically generated

**Figure 102.1 RS-CANFD Module Block Diagram (in classical CAN mode)**

CANFDɸ must be set as follows.  
R-Car V4H: 80(MHz)

Peripheral bus clock (pclk): SASYNCPERD2ɸ  
clkc: CANFDɸ  
clk\_xincan: CAN\_CLK  
clk\_ram: SASYNCPERD1ɸ

#### External Pins

Table 102.3 shows the CAN module pin.  
Pin switching is required for pins which are multiplexed with other Function. For details, see section 6, Pin Function  
Controller (PFC).  
**Table 102.3 Pin Configuration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **I/O** | **Function** |
| CANFDm\_RX | — | Input | Pins for receiving data |
| CANFDm\_TX | — | Output | Pins for transmitting data |
| CAN\_CLK | — | Input | Input pin used for external clock input. |

#### Connected module

**Table 102.4 Connected module**

|  |  |  |
| --- | --- | --- |
| Module name | Connected module name | function of the related module |
| CANFD | AXI | Register access of CPU |
| CPG | Clock output |  |
| PFC | Selection of external pins |  |
| Module Standby | Clock stop control |  |
| Software Reset | Soft reset execution |  |
| INTC | Interruption |  |
| SYS-DMAC | DMA transmission |  |
| CRC | CRC module |  |
| MFIS | MFIS modul |  |

##### Register Base Address

RSCFD{unit} base addresses are listed in the following table.  
RSCFD{unit} register addresses are given as offsets from the base addresses in general.  
**Table 102.5 Register Base Address**

|  |  |
| --- | --- |
| **Base Address Name** | **Base Address** |
| <RSCFD0\_base> | H'E666 0000 |

#### Register Configuration

This section describes all registers of RS-CANFD

##### List of Registers

RSCFD registers are listed in the table below.

For details about Register Base Addresses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Register Name** | **Symbol RSCFD0+{registers of RS-CANFD}** | **R/W (Refer to Figure 4.2)** | **Address** | **Access Size** |
| Channel n Nominal Bitrate Configuration Register | RSCFD0CFDCnNCFG | - | <RSCFD0\_base> + 0000h + n\*0010h | 8, 16, 32 |
| Channel n Control Registers | RSCFD0CFDCnCTR | - | <RSCFD0\_base> + 0004h + n\*0010h | 8, 16, 32 |
| Channel n Status Registers | RSCFD0CFDCnSTS | - | <RSCFD0\_base> + 0008h + n\*0010h | 8, 16, 32 |
| Channel n Error Flag Registers | RSCFD0CFDCnERFL | - | <RSCFD0\_base> + 000Ch + n\*0010h | 8, 16, 32 |
| Global IP Version Register | RSCFD0CFDGIPV | - | <RSCFD0\_base> + 0080h | 8, 16, 32 |
| Global Configuration Register | RSCFD0CFDGCFG | - | <RSCFD0\_base> + 0084h | 8, 16, 32 |
| Global Control Register | RSCFD0CFDGCTR | - | <RSCFD0\_base> + 0088h | 8, 16, 32 |
| Global Status Register | RSCFD0CFDGSTS | - | <RSCFD0\_base> + 008Ch | 8, 16, 32 |
| Global Error Flag Register | RSCFD0CFDGERFL | - | <RSCFD0\_base> + 0090h | 8, 16, 32 |
| Global Timestamp Counter Register | RSCFD0CFDGTSC | - | <RSCFD0\_base> + 0094h | 16, 32 |
| Global Acceptance Filter List Entry Control Register | RSCFD0CFDGAFLECTR | - | <RSCFD0\_base> + 0098h | 8, 16, 32 |
| Global Acceptance Filter List Configuration Register w | RSCFD0CFDGAFLCFGw | - | <RSCFD0\_base> + 009Ch + w\*0004h | 8, 16, 32 |
| RX Message Buffer Number Register | RSCFD0CFDRMNB | - | <RSCFD0\_base> + 00Ach | 8, 16, 32 |
| RX Message Buffer New Data Register t | RSCFD0CFDRMNDt | - | <RSCFD0\_base> + 00B0h + t\*0004h | 8, 16, 32 |
| RX FIFO Configuration / Control Registers a | RSCFD0CFDRFCCa | - | <RSCFD0\_base> + 00C0h + a\*0004h | 8, 16, 32 |
| RX FIFO Status Registers a | RSCFD0CFDRFSTSa | - | <RSCFD0\_base> + 00E0h + a\*0004h | 8, 16, 32 |
| RX FIFO Pointer Control Registers a | RSCFD0CFDRFPCTRa | - | <RSCFD0\_base> + 0100h + a\*0004h | 8, 16, 32 |
| Common FIFO Configuration / Control Registers d | RSCFD0CFDCFCCd | - | <RSCFD0\_base> + 0120h + d\*0004h | 8, 16, 32 |
| Common FIFO Configuration / Control Enhancement Registers d | RSCFD0CFDCFCCEd | - | <RSCFD0\_base> + 0180h + d\*0004h | 8, 16, 32 |
| Common FIFO Status Registers d | RSCFD0CFDCFSTSd | - | <RSCFD0\_base> + 01E0h + d\*0004h | 8, 16, 32 |
| Common FIFO Pointer Control Registers d | RSCFD0CFDCFPCTRd | - | <RSCFD0\_base> + 0240h + d\*0004h | 8, 16, 32 |
| FIFO Empty Status Register | RSCFD0CFDFESTS | - | <RSCFD0\_base> + 02A0h | 8, 16, 32 |
| FIFO Full Status Register | RSCFD0CFDFFSTS | - | <RSCFD0\_base> + 02A4h | 8, 16, 32 |
| FIFO Message Lost Status Register | RSCFD0CFDFMSTS | - | <RSCFD0\_base> + 02A8h | 8, 16, 32 |
|  |  |  |  |  |
|  |  |  |  |  |
| RX FIFO Interrupt Flag Status Register | RSCFD0CFDRFISTS | - | <RSCFD0\_base> + 02Ach | 8, 16, 32 |
| Common FIFO RX Interrupt Flag Status Register | RSCFD0CFDCFRISTS | - | <RSCFD0\_base> + 02B0h | 8, 16, 32 |
| Common FIFO TX Interrupt Flag Status Register | RSCFD0CFDCFTISTS | - | <RSCFD0\_base> + 02B4h | 8, 16, 32 |
| Common FIFO One Frame RX Interrupt Flag Status Register | RSCFD0CFDCFOFRISTS | - | <RSCFD0\_base> + 02B8h | 8, 16, 32 |
| Common FIFO One Frame TX Interrupt Flag Status Register | RSCFD0CFDCFOFTISTS | - | <RSCFD0\_base> + 02BCh | 8, 16, 32 |
| Common FIFO Message Over Write Status Register | RSCFD0CFDCFMOWSTS | - | <RSCFD0\_base> + 02C0h | 8, 16, 32 |
| FIFO FDC Full Status Register | RSCFD0CFDFFFSTS | - | <RSCFD0\_base> + 02C4h | 8, 16, 32 |
| TX Message Buffer Control Registers i | RSCFD0CFDTMCi | - | <RSCFD0\_base> + 02D0h + i\*0001h | 8 |
| TX Message Buffer Status Registers j | RSCFD0CFDTMSTSj | - | <RSCFD0\_base> + 07D0h + j\*0001h | 8 |
| TX Message Buffer Transmission Request Status Register f | RSCFD0CFDTMTRSTSf | - | <RSCFD0\_base> + 0CD0h + f\*0004h | 8, 16, 32 |
| TX Message Buffer Transmission Abort Request Status Register f | RSCFD0CFDTMTARSTSf | - | <RSCFD0\_base> + 0D70h + f\*0004h | 8, 16, 32 |
| TX Message Buffer Transmission Completion Status Register f | RSCFD0CFDTMTCSTSf | - | <RSCFD0\_base> + 0E10h + f\*0004h | 8, 16, 32 |
| TX Message Buffer Transmission Abort Status Register f | RSCFD0CFDTMTASTSf | - | <RSCFD0\_base> + 0EB0h + f\*0004h | 8, 16, 32 |
| TX Message Buffer Interrupt Enable Configuration Register f | RSCFD0CFDTMIECf | - | <RSCFD0\_base> + 0F50h + f\*0004h | 8, 16, 32 |
| TX Queue Configuration / Control Registers 0 [n] | RSCFD0CFDTXQCC0[n] | - | <RSCFD0\_base> + 1000h + n\*0004h | 8, 16, 32 |
| TX Queue Status Registers 0 [n] | RSCFD0CFDTXQSTS0[n] | - | <RSCFD0\_base> + 1020h + n\*0004h | 8, 16, 32 |
| TX Queue Pointer Control Registers 0 [n] | RSCFD0CFDTXQPCTR0[n] | - | <RSCFD0\_base> + 1040h + n\*0004h | 8, 16, 32 |
| TX Queue Configuration / Control Registers 1 [n] | RSCFD0CFDTXQCC1[n] | - | <RSCFD0\_base> + 1060h + n\*0004h | 8, 16, 32 |
| TX Queue Status Registers 1 [n] | RSCFD0CFDTXQSTS1[n] | - | <RSCFD0\_base> + 1080h + n\*0004h | 8, 16, 32 |
| TX Queue Pointer Control Registers 1 [n] | RSCFD0CFDTXQPCTR1[n] | - | <RSCFD0\_base> + 10A0h + n\*0004h | 8, 16, 32 |
| TX Queue Configuration / Control Registers 2 [n] | RSCFD0CFDTXQCC2[n] | - | <RSCFD0\_base> + 10C0h + n\*0004h | 8, 16, 32 |
| TX Queue Status Registers 2 [n] | RSCFD0CFDTXQSTS2[n] | - | <RSCFD0\_base> + 10E0h + n\*0004h | 8, 16, 32 |
| TX Queue Pointer Control Registers 2 [n] | RSCFD0CFDTXQPCTR2[n] | - | <RSCFD0\_base> + 1100h + n\*0004h | 8, 16, 32 |
| TX Queue Configuration / Control Registers 3 [n] | RSCFD0CFDTXQCC3[n] | - | <RSCFD0\_base> + 1120h + n\*0004h | 8, 16, 32 |
| TX Queue Status Registers 3 [n] | RSCFD0CFDTXQSTS3[n] | - | <RSCFD0\_base> + 1140h + n\*0004h | 8, 16, 32 |
| TX Queue Pointer Control Registers 3 [n] | RSCFD0CFDTXQPCTR3[n] | - | <RSCFD0\_base> + 1160h + n\*0004h | 8, 16, 32 |
| TX Queue Empty Status Register | RSCFD0CFDTXQESTS | - | <RSCFD0\_base> + 1180h | 8, 16, 32 |
| TX Queue Full Interrupt Status Register | RSCFD0CFDTXQFISTS | - | <RSCFD0\_base> + 1184h | 8, 16, 32 |
| TX Queue Message Lost Status Register | RSCFD0CFDTXQMSTS | - | <RSCFD0\_base> + 1188h | 8, 16, 32 |
| TX Queue Message Overwrite Status Register | RSCFD0CFDTXQOWSTS | - | <RSCFD0\_base> + 118Ch | 8, 16, 32 |
| TX Queue Interrupt Status Register | RSCFD0CFDTXQISTS | - | <RSCFD0\_base> + 1190h | 8, 16, 32 |
| TX Queue One Frame TX Interrupt Status Register | RSCFD0CFDTXQOFTISTS | - | <RSCFD0\_base> + 1194h | 8, 16, 32 |
| TX Queue One Frame RX Interrupt Status Register | RSCFD0CFDTXQOFRISTS | - | <RSCFD0\_base> + 1198h | 8, 16, 32 |
| TX Queue Full Status Register | RSCFD0CFDTXQFSTS | - | <RSCFD0\_base> + 119Ch | 8, 16, 32 |
| TX History List Configuration / Control Register n | RSCFD0CFDTHLCCn | - | <RSCFD0\_base> + 1200h + n\*0004h | 8, 16, 32 |
| TX History List Status Register n | RSCFD0CFDTHLSTSn | - | <RSCFD0\_base> + 1220h + n\*0004h | 8, 16, 32 |
| TX History List Pointer Control Registers n | RSCFD0CFDTHLPCTRn | - | <RSCFD0\_base> + 1240h + n\*0004h | 8, 16, 32 |
| Global TX Interrupt Status Register v | RSCFD0CFDGTINTSTSv | - | <RSCFD0\_base> + 1300h + v\*4 | 8, 16, 32 |
| Global Test Configuration Register | RSCFD0CFDGTSTCFG | - | <RSCFD0\_base> + 1308h | 8, 16, 32 |
| Global Test Control Register | RSCFD0CFDGTSTCTR | - | <RSCFD0\_base> + 130Ch | 8, 16, 32 |
| Global FD Configuration register | RSCFD0CFDGFDCFG | - | <RSCFD0\_base> + 1314h | 8, 16, 32 |
| Global FD CRC Configuration register | RSCFD0CFDGCRCCFG | - | <RSCFD0\_base> + 1318h | 8, 16, 32 |
| Global Lock Key Register | RSCFD0CFDGLOCKK | - | <RSCFD0\_base> + 131Ch | 16, 32 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Global AFL Ignore Entry Register | RSCFD0CFDGAFLIGNENT | - | <RSCFD0\_base> + 1324h | 8, 16, 32 |
| Global AFL Ignore Control Register | RSCFD0CFDGAFLIGNCTR | - | <RSCFD0\_base> + 1328h | 16, 32 |
| DMA Transfer Control Register | RSCFD0CFDCDTCT | - | <RSCFD0\_base> + 1330h | 8, 16, 32 |
| DMA Transfer Status Register | RSCFD0CFDCDTSTS | - | <RSCFD0\_base> + 1334h | 8, 16, 32 |
| DMA TX Transfer Control Register | RSCFD0CFDCDTTCT | - | <RSCFD0\_base> + 1340h | 8, 16, 32 |
| DMA TX Transfer Status Register | RSCFD0CFDCDTTSTS | - | <RSCFD0\_base> + 1344h | 8, 16, 32 |
| Global RX Interrupt Status Register n | RSCFD0CFDGRINTSTSn | - | <RSCFD0\_base> + 1350h + n\*0004h | 8, 16, 32 |
| Pretended Network Filter List Entry control Register | RSCFD0CFDGPFLECTR | - | <RSCFD0\_base> + 1370h | 8, 16, 32 |
| Pretended Network Filter List Entry Configuration Register u | RSCFD0CFDGPFLCFGu | - | <RSCFD0\_base> + 1374h + u \* 0004h | 8, 16, 32 |
| Global SW reset Register | RSCFD0CFDGRSTC | - | <RSCFD0\_base> + 1380h | 16, 32 |
| Global Flexible CAN mode Configuration Register | RSCFD0CFDGFCMC | - | <RSCFD0\_base> + 1384h | 8, 16, 32 |
| Global Flexible transmission buffer assignment Configuration Register | RSCFD0CFDGFTBAC | - | <RSCFD0\_base> + 138Ch | 8, 16, 32 |
| Global Virtual Machine Mode configuration Register | RSCFD0CFDGFFIMC | - | <RSCFD0\_base> + 1390h | 16, 32 |
| Global Virtual Machine Error Interrupt Select Register | RSCFD0CFDGVMEIS | - | <RSCFD0\_base> + 1394h | 8, 16, 32 |
| Global Virtual Machine Common FIFO TXQ configuration Register n | RSCFD0CFDVMCFGn | - | <RSCFD0\_base> + 13A0h + n\*0004h | 8, 16, 32 |
| Global Virtual Machine RX FIFO configuration Register | RSCFD0CFDVMRFCFG | - | <RSCFD0\_base> + 13C0h | 8, 16, 32 |
| Virtual Machine Interrupt Status Register n | RSCFD0CFDVMISTSn | - | <RSCFD0\_base> + 13E0h + n\*0004h | 8, 16, 32 |
| Channel n Data Bitrate Configuration Register | RSCFD0CFDCnDCFG | - | <RSCFD0\_base> + 1400h + n\*0020h | 8, 16, 32 |
| Channel n CAN-FD Configuration Register | RSCFD0CFDCnFDCFG | - | <RSCFD0\_base> + 1404h + n\*0020h | 8, 16, 32 |
| Channel n CAN-FD Control Register | RSCFD0CFDCnFDCTR | - | <RSCFD0\_base> + 1408h + n\*0020h | 8, 16, 32 |
| Channel n CAN-FD Status Register | RSCFD0CFDCnFDSTS | - | <RSCFD0\_base> + 140Ch + n\*0020h | 8, 16, 32 |
| Channel n CAN-FD CRC Register | RSCFD0CFDCnFDCRC | - | <RSCFD0\_base> + 1410h + n\*0020h | 8, 16, 32 |
| Channel n Bus load Control Register | RSCFD0CFDCnBLCT | - | <RSCFD0\_base> + 1418h + n\*0020h | 8, 16, 32 |
| Channel n Bus load Status Register | RSCFD0CFDCnBLSTS | - | <RSCFD0\_base> + 141Ch + n\*0020h | 8, 16, 32 |
| Global Acceptance Filter List ID Registers r = [1…10]h | RSCFD0CFDGAFLIDr | - | <RSCFD0\_base> + 1800h + (r-1)\*0010h | 8, 16, 32 |
| Global Acceptance Filter List Mask Registers r = [1…10]h | RSCFD0CFDGAFLMr | - | <RSCFD0\_base> + 1804h + (r-1)\*0010h | 8, 16, 32 |
| Global Acceptance Filter List Pointer 0 Registers r = [1…10]h | RSCFD0CFDGAFLP0r | - | <RSCFD0\_base> + 1808h + (r-1)\*0010h | 8, 16, 32 |
| Global Acceptance Filter List Pointer 1 Registers r = [1…10]h | RSCFD0CFDGAFLP1r | - | <RSCFD0\_base> + 180Ch + (r-1)\*0010h | 8, 16, 32 |
| Global Pretended Network Filter List ID Registers s = [1…4] | RSCFD0CFDGPFLIDs | - | <RSCFD0\_base> + 1A00h + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Mask Registers s = [1...4] | RSCFD0CFDGPFLMs | - | <RSCFD0\_base> + 1A04h + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Pointer 0 Registers s = [1...4] | RSCFD0CFDGPFLP0s | - | <RSCFD0\_base> + 1A08h + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Pointer 1 Registers s = [1...4] | RSCFD0CFDGPFLP1s | - | <RSCFD0\_base> + 1A0Ch + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Filter Payload Type Registers s = [1...4] | RSCFD0CFDGPFLPTs | - | <RSCFD0\_base> + 1A10h + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Payload Data 0 Registers s = [1...4] | RSCFD0CFDGPFLPD0s | - | <RSCFD0\_base> + 1A14h + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Payload Mask 0 Registers s = [1...4] | RSCFD0CFDGPFLPM0s | - | <RSCFD0\_base> + 1A18h + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Payload Data 1 Registers s = [1...4] | RSCFD0CFDGPFLPD1s | - | <RSCFD0\_base> + 1A1Ch + (s-1) \* 0040h | 8, 16, 32 |
| Global Pretended Network Filter List Payload Mask 1 Registers s = [1...4] | RSCFD0CFDGPFLPM1s | - | <RSCFD0\_base> + 1A20h + (s-1) \* 0040h | 8, 16, 32 |
| Channel n TX History List Access Registers 0 | RSCFD0CFDTHLACC0[n] | - | <RSCFD0\_base> + 8000h + n\*0008h | 8, 16, 32 |
| Channel n TX History List Access Registers 1 | RSCFD0CFDTHLACC1[n] | - | <RSCFD0\_base> + 8004h + n\*0008h | 8, 16, 32 |
| RAM Test Page Access Registers k | RSCFD0CFDRPGACCk | - | <RSCFD0\_base> + 8400h + k\*0004h | 8, 16, 32 |
| RX Message Buffer ID Registers | RSCFD0CFDRMID | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| RX Message Buffer Pointer Registers | RSCFD0CFDRMPTR | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| RX Message Buffer CAN-FD Status Register | RSCFD0CFDRMFDSTS | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
|  |  |  |  |  |
|  |  |  |  |  |
| RX Message Buffer Data Field p Registers | RSCFD0CFDRMDFp | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| RX FIFO Access ID Registers | RSCFD0CFDRFID | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| RX FIFO Access Pointer Register | RSCFD0CFDRFPTR | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| RX FIFO Access CAN-FD Status Register | RSCFD0CFDRFFDSTS | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| RX FIFO Access Data Field p Registers | RSCFD0CFDRFDFp | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| Common FIFO Access ID Registers | RSCFD0CFDCFID | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| Common FIFO Access Pointer Registers | RSCFD0CFDCFPTR | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| Common FIFO Access CAN-FD Control/Status Register | RSCFD0CFDCFFDCSTS | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| Common FIFO Access Data Field p Registers | RSCFD0CFDCFDFp | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| TX Message Buffer ID Registers | RSCFD0CFDTMID | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| TX Message Buffer Pointer Registers | RSCFD0CFDTMPTR | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| TX Message Buffer CAN-FD Control Register | RSCFD0CFDTMFDCTR | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |
| TX Message Buffer Data Field p Registers | RSCFD0CFDTMDFp | - | <RSCFD0\_base> + Refer to Figure 4.2 | 8, 16, 32 |

### Register Description

#### Details of Interface Mode Related Registers

Refer to the section 4 of “102\_uciaprcn0140\_IPSpec\_v1p06.pdf”

### Operation

Refer to the section 5 to 18 of “102\_uciaprcn0140\_IPSpec\_v1p06.pdf”

### Usage Notes

No description on this section.

### Safety Mechanisms

#### ECC for CAN-FD (IP uciapecc7w10)

##### Overview

ECC IP – Function block diagram:

A diagram of a program code

Description automatically generated

**Figure 102.2 ECC Block diagram**EDC protection is applied to internal buffer (Acceptance filter list RAM, Message Buffer RAM) of CAN-FD.  
**(1) Error Detection and Correction**-Seven-bit ECC data is appended to the 32-bit RAM data.  
-This ECC module provides 2-bit ECC error detection and 1-bit ECC error detection and correction.  
\*Note: This module is not capable of reliably detecting errors in three or more bits. If errors occur in three or more bits,  
the module may detect the errors as 1- or 2-bit ECC errors or not detect any errors. Depending on the settings, this may  
lead to the correction of a bit that was not actually inverted.  
**(2) Enabling or Disabling ECC Error Detection and Correction**-ECC error detection can be either enabled or disabled.  
-One-bit ECC error correction can be either enabled or disabled.  
-If all the bits of RAM output data are stuck to 0 or 1, it is detected as a 2-bit ECC error.  
**(3) Error Notification**-The ECM is notified when 2-bit ECC errors are detected (this can be enabled or disabled).  
-The ECM is notified when 1-bit ECC errors are detected (this can be enabled or disabled).  
-Once the ECM has been notified of an error, even if another ECC error is detected, the ECM is not notified until the error  
status bit corresponding to the initial error is cleared.  
**(4) Error Status**-Detection of 2-bit and 1-bit ECC errors can be monitored.  
-Special registers are provided to clear error status.  
**(5) Address Capture**-Only one address at which an ECC error has occurred can be captured.  
-When a 2-bit or 1-bit ECC error is detected, the error-causing address is captured. It is used to capture only when the  
first error is detected after the flag is cleared.  
**(6) Testing Function (Error Insertion)**-By setting the test mode, register values can be used as the data to be output to the RAM. When a peripheral module writes  
to the RAM, the ECEDB[31:0] register value can be written to the RAM data section, and the ECERDB[6:0] register  
value can be written to the ECC redundant bit section.  
-By setting the test mode, the ECC redundant bit section can be latched when RAM data is read, and the value can be  
confirmed.  
-By setting the test mode, the ECC redundant bit (encoding circuit) and syndrome code (decoding circuit), which are  
generated from the input data, can be confirmed.

##### Hardware Description

A diagram of a computer system

Description automatically generated

**Figure 102.3 EDC protection for CAN-FD**

**Table 102.6 Register Configuration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Register Name |  | Abbreviation | R/W | Address | Initial Value | Size |
| ECC control register |  | EC710CTL | R/W | H’E66C\_8000 | H’0000 0000 | 32 |
| ECC test mode control register |  | EC710TMC | R/W | H’E66C\_8004 | H’0000 0000 | 32 |
| ECC redundant bit data control test register |  | EC710TRC | R | H’E66C\_8008 | H’0000 0000 | 32 |
| ECC encoder and decoder data test register |  | EC710TED | R/W | H’E66C\_800C | H’0000 0000 | 32 |
| ECC error address register |  | EC710EAD | R | H’E66C\_8010 | H’0000 0000 | 32 |