RH850/U2B Group

R01AN7038EJ0100
Rev.1.00

Data CRC Function K (KCRC)

Introduction

This application note describes how to use the Data CRC Function K (KCRC) on the RH850/U2Bx.

Aim of this document and software is to provide supplemental information for the function on RH850/U2C. It is not intended to implement in the design for mass production.

There is no guarantee to update in this document and software to reflect the latest manual, errata, technical update and development environment. You are fully responsible for the incorporation or any other use of the information of this document in the design of your product or system, and please refer to latest manual, errata, technical update, and development environment.

Target Device

RH850/U2Bx

Contents

1. Introduction 3
1.1 Functions Used 3
2. Functional Overview of Data CRC Function K 4
2.1 CRC Generation Procedure 5
3. Operation Overview 6
3.1 Example 1 Verification of Transmitted Data by CRC Embedded in SPI Frame 6
3.1.1 Overview 6
3.1.2 Operating Conditions for Functions Used 8
3.1.2.1 Operating Conditions on Transmitter Side 8
3.1.2.2 Operating Conditions on Receiver Side 9
3.1.3 Software Description 10
3.1.3.1 Transmitter Side Settings 10
3.1.3.2 Receiver Side Settings 15
3.1.4 Operation Sequence 20
3.2 Example 2 Verification of Transmitted Data by CRC Embedded in CAN FD Frame 22
3.2.1 Overview 22
3.2.2 Operating Conditions for Functions Used 23
3.2.2.1 Operating Conditions on Transmitter Side 23
3.2.2.2 Operating Conditions on Receiver Side 24
3.2.3 Software Description 25
3.2.3.1 Transmitter Side Settings 25
3.2.3.2 Receiver Side Settings 30
3.2.4 Operation Sequence 35
Revision History 37

1. Introduction

This application note describes how to use the Data CRC Function K (KCRC) of the RH850/U2Bx and software examples.

1.1 Functions Used

The RH850/U2Bx hardware features used in this application note are listed below.

- Data CRC Function K (KCRC)
- DMA (sDMAC)
- Multichannel Serial Peripheral Interface (MSPI)
- CAN FD Interface (RS-CANFD)

2. Functional Overview of Data CRC Function K

The Data CRC Function K can be used to verify or generate data streams of any length and various bit widths protected by a CRC (Cyclic Redundancy Check).

- Supported CRC polynomials:
  - 64-bit CRC64ECMA
    42F0E1EB A9EA3693H: $X^{64} + X^{62} + X^{57} + X^{55} + X^{54} + X^{53} + X^{52} + X^{47} + X^{46} + X^{45} + X^{40} + X^{39} + X^{38} + X^{37} + X^{35} + X^{33} + X^{32} + X^{31} + X^{29} + X^{27} + X^{24} + X^{23} + X^{22} + X^{21} + X^{19} + X^{17} + X^{13} + X^{12} + X^{10} + X^{9} + X^{7} + X^{4} + X + 1$
  - 32-bit CRC32P4
    F4ACFB13H: $X^{32} + X^{31} + X^{30} + X^{29} + X^{28} + X^{26} + X^{23} + X^{21} + X^{19} + X^{18} + X^{15} + X^{14} + X^{13} + X^{12} + X^{11} + X^{9} + X^{8} + X^{4} + X + 1$
  - 32-bit Ethernet CRC
    04C1 1DB7H: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$
  - 32-bit CRC32 Reversed Polynomial
    EDB88320H: $X^{32} + X^{31} + X^{30} + X^{29} + X^{27} + X^{26} + X^{24} + X^{23} + X^{21} + X^{20} + X^{19} + X^{15} + X^{9} + X^{8} + X^{5} + 1$
  - 32-bit CRC32C (Castagnoli)
    1EDC 6F41H: $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^{9} + X^{8} + X^{6} + 1$
  - 16-bit CCITT CRC
    1021H: $X^{16} + X^{12} + X^{5} + 1$
  - 16-bit Baicheva00
    90D9H: $X^{16} + X^{15} + X^{12} + X^{7} + X^{6} + X^{4} + X^{3} + 1$
  - 15-bit CRC15CAN
    4599H: $X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$
  - 16-bit ARC
    8005H: $X^{16} + X^{15} + X^{2} + 1$
  - 8-bit SAE J1850 CRC
    1DH: $X^{8} + X^{4} + X^{3} + X^{2} + 1$
  - 8-bit 0x2F CRC
    2FH: $X^{8} + X^{5} + X^{3} + X^{2} + X + 1$
- CRC generation for any data block length
- After initialization of the KCRC data register, every write access to the KCRC input register (KCRCnDIN) generates a new CRC according to the chosen polynomial and the result is stored in the KCRC data register (KCRCnDOUT).

2.1 CRC Generation Procedure

The Data CRC Function K generates a CRC of any data block length. The data is transferred to the data CRC function in 8-bit, 16-bit, or 32-bit increments. Before the first write access to the KCRCnDIN register, an initial start value must be set in the KCRCnDOUT register.

One clock after the last write access to the KCRCnDIN register, the result can be read back from the KCRCnDOUT register.

Figure 2-1 shows a flowchart of CRC generation in the Data CRC Function K.

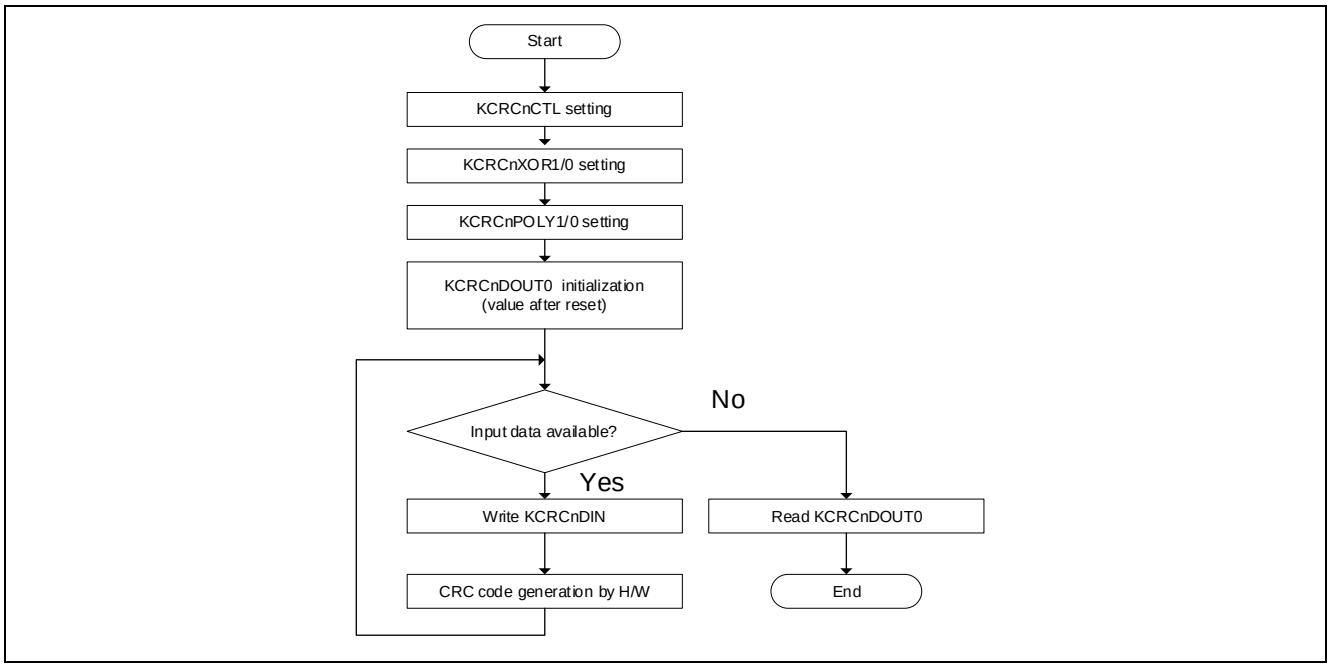

Figure 2-1 Flowchart of CRC generation in the Data CRC Function K

All registers of the KCRC can be read/write accessed via DMA independently from the CPU. This application note introduces a method of CPU-independent CRC generation using sDMAC in conjunction.

3. Operation Overview

3.1 Example 1 Verification of Transmitted Data by CRC Embedded in SPI Frame

3.1.1 Overview

By storing the CRC in the SPI frame and sending it, the validity of the transmitted data can be verified. In this operation example, data read/write to RAM and registers other than the initial settings are performed using sDMAC and do not involve the CPU.

A 16-bit CCITT CRC is used as the CRC generation polynomial; SPI communication is performed using MSPI, sending and receiving 16-bit data eight times and sending the CRC as the eighth piece of data.

Figure 3-1 shows a diagram of this operation example. Processes (1) through (6) in the diagram are executed by the sDMAC, and process (7) is executed by the CPU.

Transmitter

(1) The transmit data is written from RAM to the KCRC input register to generate a CRC.
(2) The generated CRC is stored at the end of the transmitted data stored in RAM.
(3) The transmit data is written to the MSPI transmit data register to start transmission.

Receiver

(4) The received data is read from the receive data register in the MSPI and stored in RAM.
(5) The received data stored in RAM is written to the KCRC input register, and CRC' is generated.
(6) The generated CRC' is stored to the RAM.
(7) The CPU compares the generated CRC' with the received CRC to verify the validity of the transmitted data.

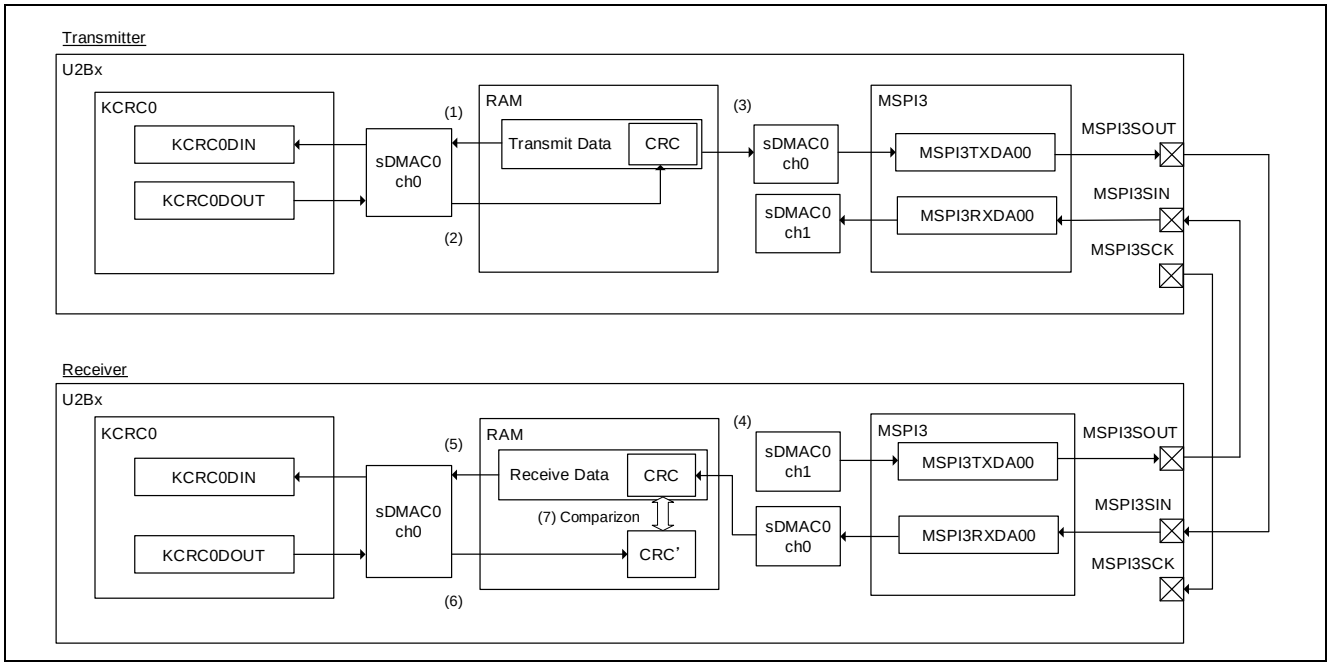

Figure 3-1 Operation Diagram

The operations of the sDMAC in this operation example are defined in the descriptor memory and are executed sequentially.

A descriptor contains all the register bits needed to describe a single transfer task on a channel. A channel can process multiple descriptors in a chain by reloading register settings from descriptor memory. When the DMA transfer defined by the current register setting is complete, the next descriptor is loaded from descriptor memory. The address of the descriptor to be loaded is defined by the DMAjDPPTR_n.PTR bit, and the continuation of the descriptor operation is defined by the DMAjDPPTR_n.CF bit.

Figure 3-2 shows an example of descriptor memory settings in this operation example.

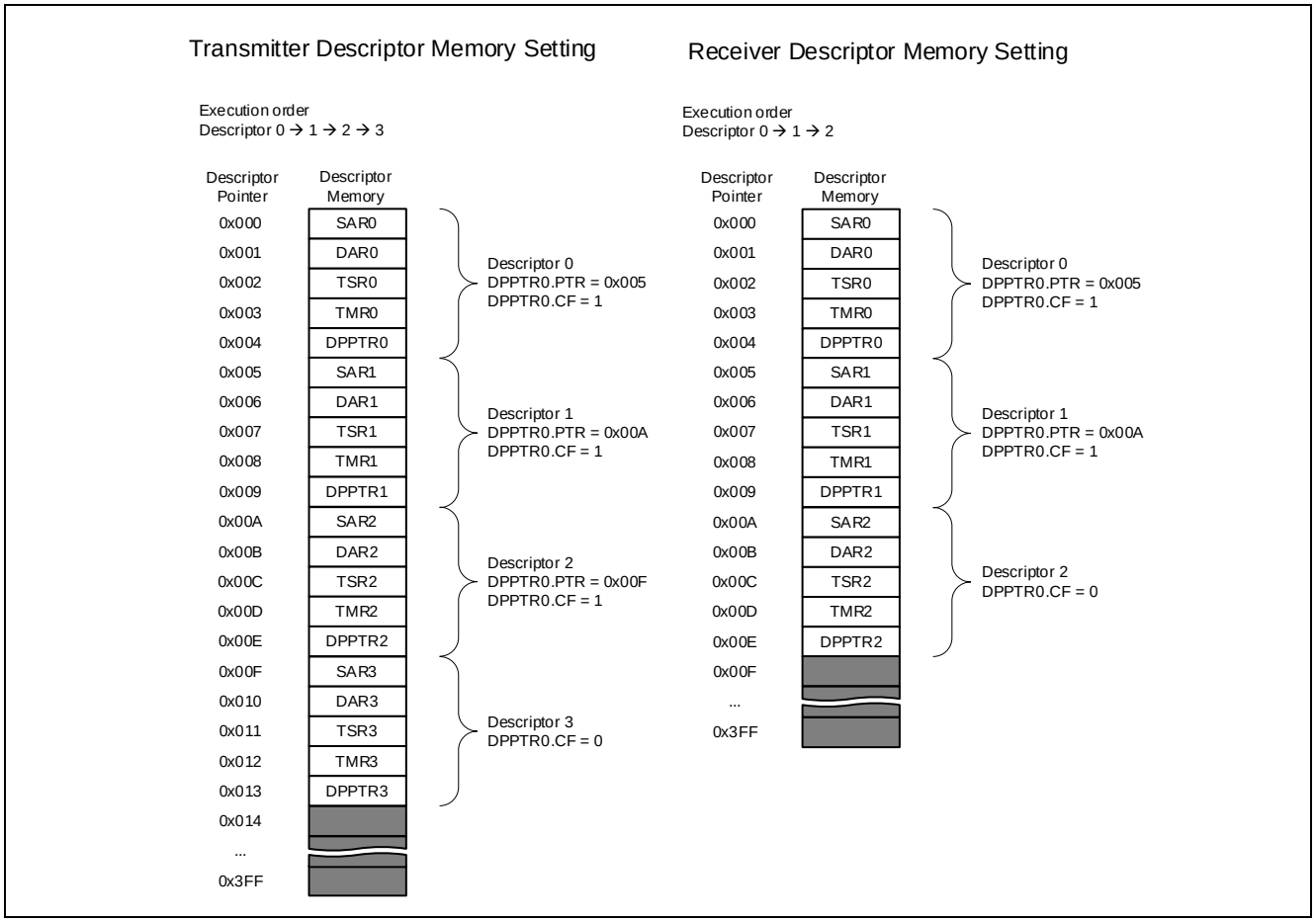

Figure 3-2 Descriptor Memory Setting Example

3.1.2 Operating Conditions for Functions Used

The operating conditions for the functions used in this operation example are shown below.

3.1.2.1 Operating Conditions on Transmitter Side

The operating conditions on the transmitter side are shown below.

Table 3-1 KCRC0 Settings

Item	Setting
Input Data Width	16 bits
CRC Polynomial	16-bit CCITT CRC

Table 3-2 MSPI3 Settings

Item	Setting
MSPI3 Operation Clock	80MHz (CLK_HSB_MSPI)
Channel used	ch3
Operation Mode	Master Mode
Transfer Clock Frequency	10 MHz
Transmit Data Length	16 bits
Transmit Data	0x1111, 0x2222, 0x3333, 0x4444, 0x5555, 0x6666, 0x7777, CRC (16 bits)
Interrupt (DMA Trigger)	Transmit Status Interrupt

Table 3-3 sDMAC0 Settings

Item	Setting
sDMAC0 Operation Clock	100 MHz
Channel used	ch0, ch1
Interrupt	None

Table 3-4 Port Settings

Item	Setting
Port used	P01_5 MSPI3SI
	P01_6 MSPI3SO
	P01_7 MSPI3SC

3.1.2.2 Operating Conditions on Receiver Side

The operating conditions on the receiver side are shown below.

Table 3-5 KCRC0 Settings

Item	Setting
Input Data Width	16 bits
CRC Polynomial	16-bit CCITT CRC

Table 3-6 MSPI3 Settings

Item	Setting
MSPI3 Operation Clock	80 MHz (CLK_HSB_MSPI)
Channel used	ch3
Operation Mode	Slave Mode
Transfer Clock Frequency	10 MHz
Interrupt (DMA Trigger)	Receive Status Interrupt

Table 3-7 sDMAC0 Settings

Item	Setting
sDMAC0 Operation Clock	100MHz
Channel used	ch0, ch1
Interrupt	Transfer End Interrupt

Table 3-8 Port Settings

Item	Setting
Port used	P01_5 MSPI3SI
	P01_6 MSPI3SO
	P01_7 MSPI3SC

3.1.3 Software Description

An example of each register setting used in this operation example is shown below.

3.1.3.1 Transmitter Side Settings

An example of each register setting used on the transmitter side is shown below.

Table 3-9 KCRC0 Settings

Register	Set value	Function
KCRC0CTL	0x000F0001	CRC input data size: 16 bits
		CRC polynomial size: 16 bits
KCRC0POLY1	0x00000000	CRC polynomial: CCITT16
KCRC0POLY0	0x00001021	
KCRC0DOUT0	0x0000FFFF	Start value for CRC generation

Table 3-10 MSPI Settings

Register	Set value	Function
MSPI3CTL0	0x01	MSPI function enabled
MSPI3CTL1	0x18000000	Master Mode
		Sampling at the next edge upon master reception.
		The default level of MSPInSCK is high.
		Set MSPInSOUT to low level.
		The MSPInCS signal is active low.
MSPI3CFG00	0x30700003	Transmission enabled
		Reception enabled
		Direct memory mode
		Channel priority level : 7
		The channel lock operation disabled
		When a last frame ends, MSPInCHENm is cleared and the channel operation ends.
		Transmit Interrupt enabled
		Reception Interrupt enabled
MSPI3CFG01	0x03010000	Clock phase: MSPInSCK is high during idle time
		Data phase: Shifting bits out for transmission takes place on odd-numbered edges
		MSB first
		The idle time is not inserted each end of a frame.
		MSPInCS returns to the inactive level after the last frame end.
		No parity check
MSPI3CFG02	0x10	Frame length: 16 bits
MSPI3CFG03	0x0004	Communication clock frequency: MSPI3CLK/8 (10MHz)
MSPI3SEUP0	0x0002	SCK delay time: 2
MSPI3HOLD0	0x0001	CS negation delay time: 1
MSPI3IDLE0	0x0001	Idle time: 1
MSPI3INDA0	0x0008	Inter-data time: 8
MSPI3CFSET0	0x0008	Number of frame count: 8
MSPI3SSEL0	0x0001	Activate CS0
MSPI3CSTS0	0x0003	Channel activated and enabled

Table 3-11 PBG Register Settings

Register	Set value	Function
PBGERRSLV30 PBGKCPROT	0xA5A5A501	Enable write access to the PBG30 registers.
PBG30 PBGPROT1_6	0x10000001	SPID for KCRC0: set SPID of sDMAC and CPU0 to 1
PBG30 PBGPROT1_11	0x10000001	SPID for MSPI3: set SPID of sDMAC and CPU0 to 1

Table 3-12 DMA/DTS Trigger Register Settings

Register	Set value	Function
MSPITG TGCTL3	0x0000000E	Trigger 1 = INTMSPInTX0, Trigger 2 = INTMSPInRX0

Table 3-13 sDMAC0 ch0 Register Settings for Transmission

Register	Set value	Function
DMA0CM_0	0x00001C00	Channel Master SPID: 0x1C
		Supervisor mode
DMA0CHCR_0	0x0302	Descriptor enabled
		Start DMA transfer after the channel configuration is copied from the descriptor memory.
		Channel Address Error Notification disabled
		Channel Address Error Interrupt disabled
		Descriptor Step End Interrupt disabled
		Transfer End Interrupt enabled
		DMA transfer disabled
DMA0DPCR_0	0x0000000F	Update flag of Descriptor DMAjSAR_n register update enabled DMAjDAR_n register update enabled DMAjTSR_n register update enabled DMAjTMR_n register update enabled DMAjGIAI_n register update disabled DMAjGOAI_n register update disabled DMAjSIAI_n register update disabled DMAjSOAI_n register update disabled DMAjSGCR_n register update disabled DMAjRS_n register update disabled DMAjBUFCR_n register update disabled
DMA0RS_0	0x00010084	Transfer count per hardware request: 1
		Transfer limit per hardware request: Transaction size indicated by DMAjTMR_n.STS * DMAjRS_n.TC
		Pre-load function disabled
		DRQ initialize disabled
		DMA request source: DMAMSPI24 (group0-132)
DMATRGSEL DMACSEL0_8	0xFFFFFCFF	MSPI3 (group0-132)
DMA0DPPTR_0	0x00000001	Address pointer of Descriptor: 0x000
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled
DMA0OR	0x0001	Priority Mode: CH0 > CH1 > ... > CH14 > CH15
		Enable DMA transfer on all channels

Table 3-14 sDMAC0 ch1 Register Settings for Dummy Data Reception

Register	Set value	Function
DMA0CM_1	0x00001C00	Channel Master SPID: 0x1C
		Supervisor mode
DMA0CHCR_1	0x0302	Descriptor enabled
		Start DMA transfer after the channel configuration is copied from the descriptor memory.
		Channel Address Error Notification disabled
		Channel Address Error Interrupt disabled
		Descriptor Step End Interrupt disabled
		Transfer End Interrupt enabled
		DMA transfer disabled
DMA0SAR_1	-	Source Address: (unsigned long) &MSPI3.RXDA00.UINT32 (MSPI3 Receive Data Register)
DMA0DAR_1	-	Destination Address: (unsigned long) &r_dat
DMA0TSR_1	0x00000020	Transfer Size: 32 bytes
DMA0TMR_1	0x00001022	Transfer Request Source: Hardware Request
		Fixed destination address
		Fixed source address
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0RS_1	0x00010085	Transfer count per hardware request: 1
		Transfer limit per hardware request: Transaction size indicated by DMAjTMR_n.STS * DMAjRS_n.TC
		Pre-load function disabled
		DRQ initialize disabled
		DMA request source: DMAMSPI25 (group0-133)
DMATRGSEL DMACSEL0_8	0xFFFFF3FF	MSPI3 (group0-133)

Table 3-15 Descriptor 0 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long)s_dat (First address of the transmit data)
DMA0DAR_0	-	Destination Address: (unsigned long)&KCRC0.DIN.UINT32 (KCRC0 Input Register)
DMA0TSR_0	0x0000000E	Transfer Size: 14 bytes
DMA0TMR_0	0x00000111	Transfer Request Source: Auto Request
		Fixed destination address
		Source address is incremented based on source transaction size
		DMA destination transaction size: 2-byte unit transfer
		DMA source transaction size: 2-byte unit transfer
DMA0DPPTR_0	0x00000015	Address pointer of Descriptor: 0x014
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-16 Descriptor 1 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long)&KCRC0.DOUT.UINT32 (KCRC0 Output Register)
DMA0DAR_0	-	Destination Address: (unsigned long) &s_dat[7] (Last address of the transmit data)
DMA0TSR_0	0x00000002	Transfer Size: 2 bytes
DMA0TMR_0	0x00000012	Transfer Request Source: Auto Request
		Fixed destination address
		Fixed source address
		DMA destination transaction size: 2-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000029	Address pointer of Descriptor: 0x028
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-17 Descriptor 2 Settings

Register	Set value	Function
DMA0SAR_0	-	Source address: (unsigned long)s_dat (First address of the transmit data)
DMA0DAR_0	-	Destination address: (unsigned long)&MSPI3.TXDA00.UINT32 (MSPI3 transmit data register)
DMA0TSR_0	0x00000002	Transfer size: 2 bytes
DMA0TMR_0	0x00000111	Transfer request source: Auto Request
		Fixed destination address
		Source address is incremented based on source transaction size
		DMA destination transaction size: 2-byte unit transfer
		DMA source transaction size: 2-byte unit transfer
DMA0DPPTR_0	0x0000003D	Address pointer of Descriptor: 0x03C
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-18 Descriptor 3 Settings

Register	Set value	Function
DMA0SAR_0	-	Source address: (unsigned long) &s_dat[1] (address of the second transmit data)
DMA0DAR_0	-	Destination address: (unsigned long)&MSPI3.TXDA00.UINT32 (MSPI3 transmit data register)
DMA0TSR_0	0x0000000E	Transfer size: 14 bytes
DMA0TMR_0	0x00001111	Transfer request source: Hardware request
		Fixed destination address
		Source address is incremented based on source transaction size
		DMA destination transaction size: 2-byte unit transfer
		DMA source transaction size: 2-byte unit transfer
DMA0DPPTR_0	0x00000000	Descriptor Interrupt disabled
		Continuation flag of Descriptor disabled

Table 3-19 shows a list of functions used in this operation example.

Table 3-19 List of Functions

Function	Overview
main_pe0	Call each function.
mspi_init	Initialize the MSPI.
sdmac_snd_init	Initialize the sDMAC ch0 for transmission.
sdmac_rcv_init	Initialize the sDMAC ch1 for dummy data reception.
descriptor_init	Initialize the descriptor RAM.
kcrc_init	Initialize the KCRC.
port_init	Initialize I/O ports.
intc_init	Initialize the interrupt controller.

3.1.3.2 Receiver Side Settings

An example of each register setting used on the receiver side is shown below.

Table 3-20 KCRC0 Settings

Register	Set value	Function
KCRC0CTL	0x000F0001	CRC input data size: 16 bits
		CRC polynomial size: 16 bits
KCRC0XOR1	0x00000000	CRC XOR mask value (upper part): 0x00000000
KCRC0XOR0	0x00000000	CRC XOR mask value (lower part): 0x00000000
KCRC0POLY1	0x00000000	CRC polynomial: CCITT16
KCRC0POLY0	0x00001021	
KCRC0DOUT0	0x0000FFFF	Start value for CRC generation

Table 3-21 MSPI3 Settings

Register	Set value	Function
MSPI3CTL1	0x18000000	Slave Mode
		The default level of MSPInSCK is high.
		Set MSPInSOUT to low after macro enable and holds the level after each transfer.
		The MSPInCS signal is active low.
MSPI3CTL0	0x01	MSPI function enabled
MSPI3CFG00	0x30700003	Transmission enabled
		Reception enabled
		Direct memory mode
		Channel priority level: 7
		The channel lock operation disabled
		When a last frame ends, MSPInCHENm is cleared and the channel operation ends.
		Transmit Interrupt enabled
		Reception Interrupt enabled
MSPI3CFG01	0x03010000	Clock phase and data phase: default setting
		MSB first
		The idle time is not inserted each end of a frame.
		MSPInCS returns to the inactive level after the last frame end.
		No parity check
MSPI3CFG02	0x0010	Frame length: 16 bits
MSPI3CFG03	0x0001	Communication clock frequency: MSPI3CLK/2 (default setting)
MSPI3SEUP0	0x0001	SCK delay time: 1
MSPI3HOLD0	0x0001	CS negation delay time: 1
MSPI3IDLE0	0x0001	Idle time: 1
MSPI3INDA0	0x0008	Inter-data time: 8
MSPI3CFSET0	0x0008	Number of frame count: 8
MSPI3CSTS0	0x0003	Channel activated and enabled

Table 3-22 PBG Register Settings

Register	Set value	Function
PBGERRSLV30 PBGKCPROT	0xA5A5A501	Enable write access to the PBG30 registers.
PBG30 PBGPROT1_6	0x10000001	SPID for KCRC0: set SPID of sDMAC and CPU0 to 1
PBG30 PBGPROT1_11	0x10000001	SPID for MSPI3: set SPID of sDMAC and CPU0 to 1

Table 3-23 DMA/DTS Trigger Register Settings

Register	Set value	Function
MSPITG TGCTL3	0x0000000E	Trigger 1 = INTMSPInTX0, Trigger 2 = INTMSPInRX0

Table 3-24 sDMAC0 ch0 Register Settings for Reception

Register	Set value	Function
DMA0CM_0	0x00001C00	Channel Master SPID: 0x1C
		Supervisor mode
DMA0CHCR_0	0x0302	Descriptor enabled
		Start DMA transfer after the channel configuration is copied from the descriptor memory.
		Channel Address Error Notification disabled
		Channel Address Error Interrupt disabled
		Descriptor Step End Interrupt disabled
		Transfer End Interrupt enabled
		DMA transfer disabled
DMA0DPCR_0	0x0000000F	Update flag of Descriptor DMAjSAR_n register update enabled DMAjDAR_n register update enabled DMAjTSR_n register update enabled DMAjTMR_n register update enabled DMAjGIAI_n register update disabled DMAjGOAI_n register update disabled DMAjSIAI_n register update disabled DMAjSOAI_n register update disabled DMAjSGCR_n register update disabled DMAjRS_n register update disabled DMAjBUFCR_n register update disabled
DMA0RS_0	0x00010085	Transfer count per hardware request: 1
		Transfer limit per hardware request: Transaction size indicated by DMAjTMR_n.STS * DMAjRS_n.TC
		Pre-load function disabled
		DRQ initialize disabled
		DMA request source: DMAMSPI25 (group0-133)
DMA0DPPTR_0	0x00000001	Address pointer of Descriptor: 0x000
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled
DMATRGSEL DMACSEL0_8	0xFFFFF3FF	MSPI3 (group0-133)
DMA0OR	0x0001	Priority Mode: CH0 > CH1 > ... > CH14 > CH15
		Enable DMA transfer on all channels

Table 3-25 sDMAC0 ch1 Register Settings for Dummy Data Transmission

Register	Set value	Function
DMA0CM_1	0x00001C00	Channel Master SPID: 0x1C
		Supervisor mode
DMA0CHCR_1	0x0302	Descriptor enabled
		Start DMA transfer after the channel configuration is copied from the descriptor memory.
		Channel Address Error Notification disabled
		Channel Address Error Interrupt disabled
		Descriptor Step End Interrupt disabled
		Transfer End Interrupt enabled
		DMA transfer disabled
DMA0SAR_1	-	Source address: (unsigned long)&dummy32 (Dummy data)
DMA0DAR_1	-	Destination address: (unsigned long)&MSPI3.TXDA00.UINT32 (MSPI transmit data register)
DMA0TSR_1	0x0000001C	Transfer Size: 28 bytes
DMA0TMR_1	0x00001022	Transfer Request Source: Hardware Request
		Fixed destination address
		Fixed source address
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0RS_1	0x00010084	Transfer count per hardware request: 1
		Transfer limit per hardware request: Transaction size indicated by DMAjTMR_n.STS * DMAjRS_n.TC
		Pre-load function disabled
		DRQ initialize disabled
		DMA request source: DMAMSPI24 (group0-132)
DMATRGSEL DMACSEL0_8	0xFFFFFCFF	MSPI3 (group0-132)

Table 3-26 Descriptor 0 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long)&MSPI3.RXDA00.UINT32 (MSPI3 Receive Data Register)
DMA0DAR_0	-	Destination Address: (unsigned long) r_dat (First address of the receive data)
DMA0TSR_0	0x00000010	Transfer Size: 16 bytes
DMA0TMR_0	0x00001411	Transfer Request Source: Hardware Request
		Destination address is incremented based on destination transaction size
		Fixed source address
		DMA destination transaction size: 2-byte unit transfer
		DMA source transaction size: 2-byte unit transfer
DMA0DPPTR_0	0x00000015	Address pointer of Descriptor: 0x014
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-27 Descriptor 1 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long) r_dat (First address of the receive data)
DMA0DAR_0	-	Destination Address: (unsigned long) &KCRC0.DIN.UINT32 (KCRC0 Input Register)
DMA0TSR_0	0x0000000E	Transfer Size: 14 bytes
DMA0TMR_0	0x00000111	Transfer Request Source: Auto Request
		Fixed destination address
		Source address is incremented based on source transaction size
		DMA destination transaction size: 2-byte unit transfer
		DMA source transaction size: 2-byte unit transfer
DMA0DPPTR_0	0x00000029	Address pointer of Descriptor: 0x028
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-28 Descriptor 2 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long)&KCRC0.DOUT.UINT32 (KCRC0 output register)
DMA0DAR_0	-	Destination Address: (unsigned long) &kcrc_out (Address of CRC result variable)
DMA0TSR_0	0x00000002	Transfer Size: 2 bytes
DMA0TMR_0	0x00000021	Transfer Request Source: Auto Request
		Fixed destination address
		Fixed source address
		DMA destination transaction size: 2-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000000	Descriptor Interrupt disabled
		Continuation flag of Descriptor disabled

Table 3-29 Interrupt Register Settings

Register	Set value	Function
EIBD70	0x00000000	Bind interrupt to PE0 (CPU0)
EIC70	0x0040	Table reference method, priority level 0

Table 3-30 shows a list of functions used in this operation example.

Table 3-30 List of Functions

Function	Overview
main_pe0	Call each function.
mspi_init	Initialize the MSPI.
sdmac_snd_init	Initialize the sDMAC ch0 for reception.
sdmac_rcv_init	Initialize the sDMAC ch1 for dummy data transmission.
descriptor_init	Initialize the descriptor RAM.
kcrc_init	Initialize the KCRC.
port_init	Initialize I/O ports.
intc_init	Initialize the interrupt controller.

3.1.4 Operation Sequence

Figure 3-3 shows the sequence of operation for the transmit operation. The number in parentheses () is the number of the corresponding descriptor. DMA transfer with each descriptor setting is performed as follows.

(0) The transmit data is written from RAM to the KCRC input register to generate a CRC.
(1) The CRC generated in step (0) is read and written to the end of the transmitted data.
(2) The first data is written to the transmit data register of the MSPI to start transmission.
(3) The remaining data is written to the transmit data register, triggered by an MSPI communication status interrupt.

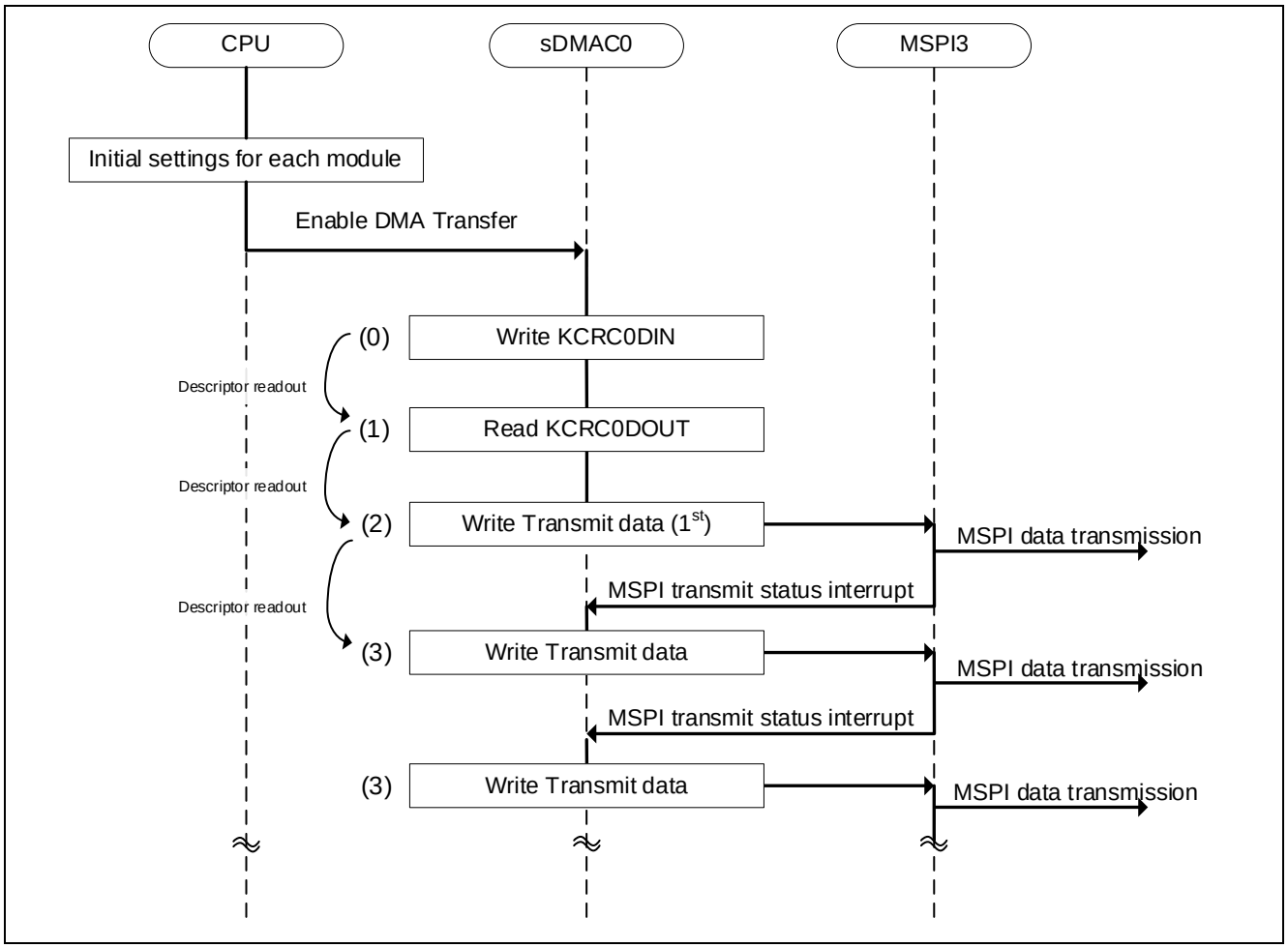

Figure 3-3 Operation Sequence of Transmission

Figure 3-4 shows the sequence of operations for the receive operation. The number in parentheses () is the number of the corresponding descriptor. DMA transfers with each descriptor setting are performed as follows.

(0) The receive data register of the MSPI is read by triggering the receive status interrupt of the MSPI.
(1) The received data read in step (0) is written to the KCRC input register to generate a CRC.
(2) The CRC generated in step (1) is read back and stored in RAM.

After all DMA transfers are completed, an sDMAC transfer end interrupt is generated. The CPU compares the CRC received in the interrupt process with the generated CRC to verify the validity of the transmitted data.

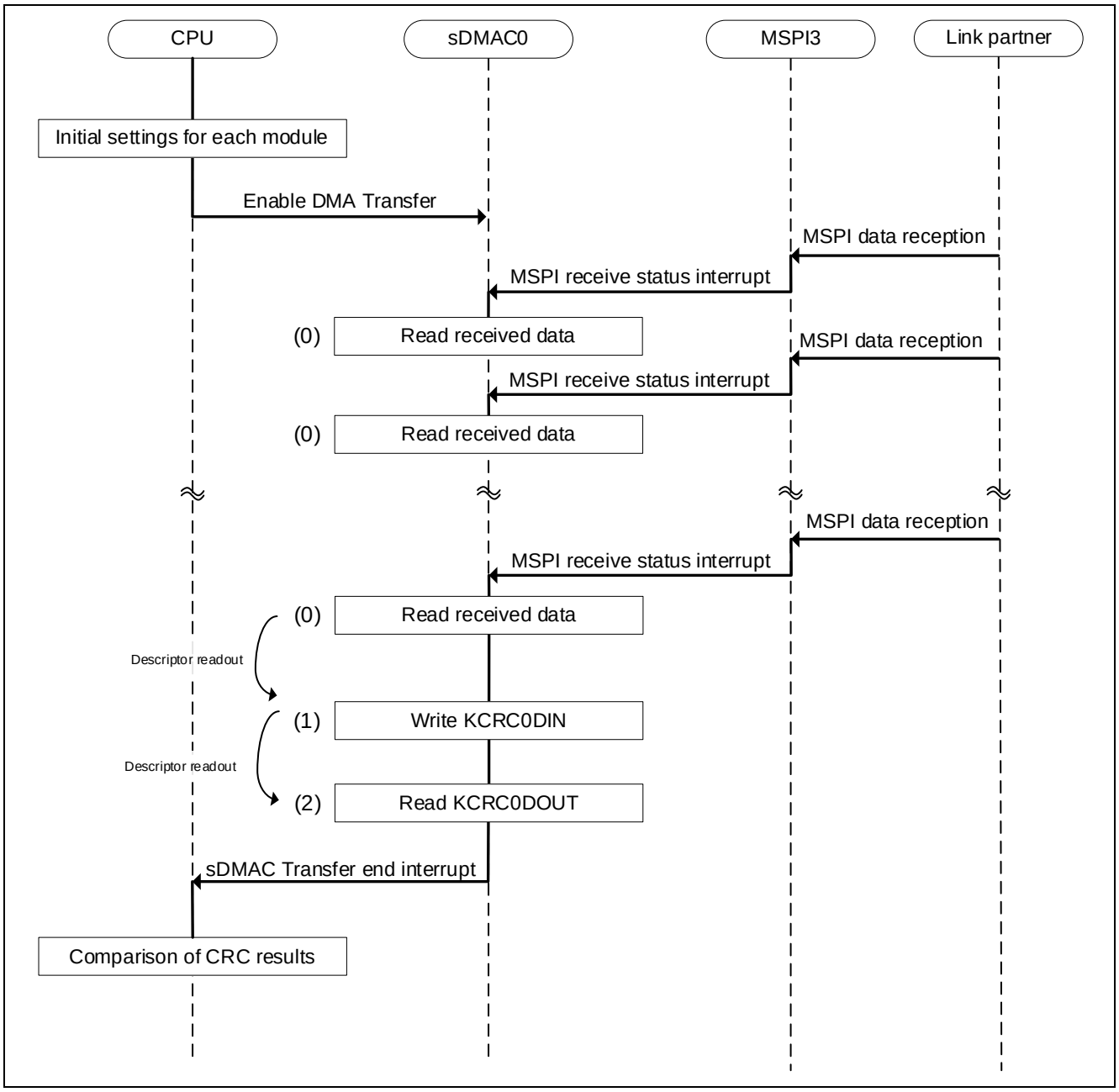

Figure 3-4 Operation Sequence of Reception

3.2 Example 2 Verification of Transmitted Data by CRC Embedded in CAN FD Frame

3.2.1 Overview

By storing the CRC in the CAN FD frame and sending it, the validity of the transmitted data can be verified. In this operation example, data read/write to RAM and registers other than the initial settings are performed using sDMAC and do not involve the CPU.

A 32-bit Ethernet CRC is used as the CRC polynomial, and CAN FD communication is performed using RS-CANFD. 64 bytes of data are transmitted and received once, and the CRC is stored in the last 4 bytes and transmitted. Figure 3-5 shows a diagram of this operation example.

Transmitter

(1) The transmit data is written from RAM to the KCRC input register to generate a CRC.
(2) The generated CRC is stored at the end of the transmitted data stored in RAM.
(3) The transmit data is written to the transmit/receive FIFO of RS-CANFD and transmission is started.

Receiver

(4) The received data is read from the receive FIFO of RS-CANFD and written to RAM.
(5) The receive data placed in RAM is written to the KCRC input register to generate a CRC'.
(6) The generated CRC' is stored to the RAM.
(7) The CPU comperes the generated CRC' with the received CRC to verify the validity of the transmitted data.

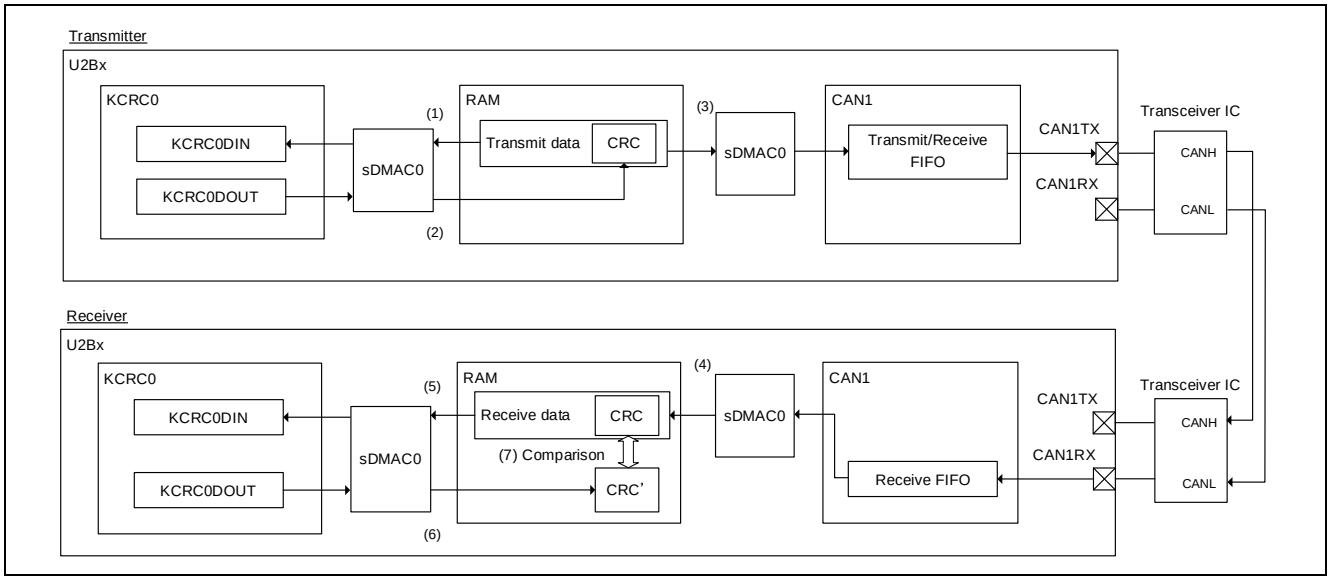

Figure 3-5 Operation Diagram

The operations of the sDMAC in this operation example are defined in the descriptor memory and are executed sequentially. For an overview of descriptors, see 3.1.1 Overview.

3.2.2 Operating Conditions for Functions Used

The operating conditions for the functions used in this operation example are shown below.

3.2.2.1 Operating Conditions on Transmitter Side

The operating conditions on the transmitter side are shown below.

Table 3-31 KCRC0 Settings

Item	Setting
Input Data Width	32 bits
CRC polynomial	32-bit Ethernet CRC

Table 3-32 RS-CANFD0 Settings

Item	Setting
Operation Clock	40 MHz
Channel used	ch1
Operation Mode	CAN FD mode
Nominal bit rate	1 Mbps
Data bit rate	2 Mbps
Transmit data length	64 bytes
Transmit data	0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0, 0x23,0x45,0x67,0x89,0xab,0xcd,0xef,0x01, 0x34,0x56,0x78,0x9a,0x12,0x34,0x56,0x78, 0x9a,0xbc,0xde,0xf0,0x23,0x45,0x67,0x89, 0xab,0xcd,0xef,0x01,0x34,0x56,0x78,0x9a, 0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0, 0x23,0x45,0x67,0x89,0xab,0xcd,0xef,0x01, 0x34,0x56,0x78,0x9a, CRC (32 bits)
Transmission buffer	Transmit/Receive FIFO

Table 3-33 sDMAC0 Settings

Item	Setting
sDMAC0 Operation Clock	100 MHz
Channel used	ch0

Table 3-34 Port Settings

Item	Setting
Port used	P02_10 ALT_IN7: CAN1RX
	P02_7 ALT_OUT7: CAN1TX

3.2.2.2 Operating Conditions on Receiver Side

The operating conditions on the receiver side are shown below.

Table 3-35 KCRC0 Settings

Item	Setting
Input Data Width	32 bits
CRC Polynomial	32-bit Ethernet CRC

Table 3-36 RS-CANFD0 Settings

Item	Setting
Operation Clock	40 MHz
Channel used	ch1
Operation Mode	CAN FD mode
Nominal bit rate	1 Mbps
Data bit rate	2 Mbps
Receive buffer	Receive FIFO
DMA Trigger	Receive FIFO DMA request 0

Table 3-37 sDMAC0 Settings

Item	Setting
sDMAC0 Operation Clock	100 MHz
Channel used	ch0
Interrupt	Transfer End Interrupt

Table 3-38 Port Settings

Item	Setting
Port used	P02_10 ALT_IN7: CAN1RX
	P02_7 ALT_OUT7: CAN1TX

3.2.3 Software Description

An example of each register setting used in this operation example is shown below.

3.2.3.1 Transmitter Side Settings

An example of each register setting used on the transmitter side is shown below.

Table 3-39 KCRC0 Register Settings

Register	Set value	Function
KCRC0CTL	0x001F0120	CRC input data size: 32 bits
		CRC polynomial size: 32 bits
KCRC0XOR1	0x00000000	CRC XOR mask value (upper part): 0x00000000
KCRC0XOR0	0xFFFFFFFF	CRC XOR mask value (lower part): 0xFFFFFFFF
KCRC0POLY1	0x00000000	CRC polynomial: CRC32
KCRC0POLY0	0x04C11DB7	
KCRC0DOUT0	0xFFFFFFFF	Start value for CRC generation

Table 3-40 RS-CANFD0 Register Settings

Register	Set value	Function
RSCFD0CFDC1NCFG	0x0D181000	Nominal Bit Rate: 1 Mbps
RSCFD0CFDC1DCFG	0x026B0000	Data Bit Rate: 2 Mbps
RSCFD0CFDCFCC3	0x00411175	Transmit Buffer Link: Transmit buffer 3
		Transmit/Receive FIFO Mode Select: Transmit mode
		Transmit/Receive FIFO Interrupt Source Select: A FIFO transmit interrupt request is generated each time a message has been transmitted
		Transmit/Receive FIFO Buffer Depth: 8 messages
		Transmit/Receive FIFO Buffer Payload Storage Size: 64 bytes
		Transmit/Receive FIFO Transmit Interrupt enabled
		Transmit/Receive FIFO Buffer enabled
RSCFD0CFDCFID3	0x000007F0	Standard ID
		Data frame
		Transmit history data is not stored in the buffer
		Transmit/Receive FIFO Buffer ID Data: 0x7F0
RSCFD0CFDCFPTR3	0xF0120000	Transmit/Receive FIFO Buffer DLC Data: 64 data bytes
		Transmit/Receive FIFO Buffer Label Data: 0x12
RSCFD0CFDCFFDCSTS3	0x00000006	CANFD frame received or to transmit
		The bit rate in the data area changes.

Table 3-41 PBG Register Settings

Register	Set value	Function
PBGERRSLV30 PBGKCPROT	0xA5A5A501	Enable write access to the PBG30 registers.
PBGERRSLV100 PBGKCPROT	0xA5A5A501	Enable write access to the PBG100 registers.
PBG30 PBGPROT1_6	0x10000001	SPID for KCRC0: set SPID of sDMAC and CPU0 to 1
PBG100 PBGPROT1_4	0x10000001	SPID for RSCAN0-CAN0: set SPID of sDMAC and CPU0 to 1

Table 3-42 sDMAC Register Settings

Register	Set value	Function
DMA0CM_0	0x00001C00	Channel Master SPID: 0x1C
		Supervisor mode
DMA0CHCR_0	0x0302	Descriptor enabled
		Start DMA transfer after the channel configuration is copied from the descriptor memory.
		Channel Address Error Notification disabled
		Channel Address Error Interrupt disabled
		Descriptor Step End Interrupt disabled
		Transfer End Interrupt enabled
		DMA transfer disabled
DMA0DPCR_0	0x0000000F	Update flag of Descriptor DMAjSAR_n register update enabled DMAjDAR_n register update enabled DMAjTSR_n register update enabled DMAjTMR_n register update enabled DMAjGIAI_n register update disabled DMAjGOAI_n register update disabled DMAjSIAI_n register update disabled DMAjSOAI_n register update disabled DMAjSGCR_n register update disabled DMAjRS_n register update disabled DMAjBUFCR_n register update disabled
DMA0DPPTR_0	0x00000001	Address pointer of Descriptor: 0x000
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled
DMA0OR	0x0001	Priority Mode: CH0 > CH1 > ... > CH14 > CH15
		Enable DMA transfer on all channels

Table 3-43 Descriptor 0 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long) tx_buf_table[0].DB (First address of the transmit data)
DMA0DAR_0	-	Destination Address: (unsigned long) &KCRC0.DIN.UINT32 (KCRC0 input register)
DMA0TSR_0	0x0000003C	Transfer Size: 60 bytes
DMA0TMR_0	0x00000122	Transfer Request Source: Auto Request
		Fixed destination address
		Source address is incremented based on source transaction size
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000015	Address pointer of Descriptor: 0x014
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-44 Descriptor 1 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long)&KCRC0.DOUT.UINT32 (KCRC0 output register)
DMA0DAR_0	-	Destination Address: (unsigned long)&tx_buf_table[0].DB[60] (Last address of the transmit data)
DMA0TSR_0	0x00000004	Transfer Size: 4 bytes
DMA0TMR_0	0x00000022	Transfer Request Source: Auto Request
		Fixed destination address
		Fixed source address
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000029	Address pointer of Descriptor: 0x028
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-45 Descriptor 2 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long) tx_buf_table (First address of the transmit data)
DMA0DAR_0	-	Destination Address: (unsigned long) &RSCFD0.CFDCFID3.UINT32 (Transmit/Receive FIFO buffer access ID register)
DMA0TSR_0	0x0000004C	Transfer Size: 76 bytes
DMA0TMR_0	0x00000522	Transfer Request Source: Auto Request
		Destination address is incremented based on destination transaction size
		Source address is incremented based on source transaction size
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x0000003D	Address pointer of Descriptor: 0x03C
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-46 Descriptor 3 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long) &can_fifo_ptr_inc (Address of data to be written to the Transmit/Receive FIFO buffer pointer control register)
DMA0DAR_0	-	Destination Address: (unsigned long) &RSCFD0.CFDCFPCTR3.UINT32 (Transmit/Receive FIFO buffer pointer control register)
DMA0TSR_0	0x00000004	Transfer Size: 4 bytes
DMA0TMR_0	0x00000022	Transfer Request Source: Auto Request
		Fixed destination address
		Fixed source address
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000000	Descriptor Interrupt disabled
		Continuation flag of Descriptor disabled

Table 3-47 shows a list of functions used in this operation example.

Table 3-47 List of Functions

Function	Overview
main_pe0	Call each function.
PORT_Init	Initialize I/O ports.
R_CAN_Init	Initialize RS-CANFD.
R_CAN_GlobalStart	Start the global operation of the RS-CANFD.
R_CAN_ChStart	Start the channel operation of the RS-CANFD.
sdmac_snd_init	Initialize the sDMAC.
descriptor_init	Initialize the descriptor RAM.
kcrc_init	Initialize the KCRC.
intc_init	Initialize the interrupt controller.

3.2.3.2 Receiver Side Settings

An example of each register setting used on the receiver side is shown below.

Table 3-48 KCRC0 Register Settings

Register	Set value	Function
KCRC0CTL	0x001F0120	CRC input data size: 32 bits
		CRC polynomial size: 32 bits
KCRC0XOR1	0x00000000	CRC XOR mask value (upper part): 0x00000000
KCRC0XOR0	0xFFFFFFFF	CRC XOR mask value (lower part): 0xFFFFFFFF
KCRC0POLY1	0x00000000	CRC polynomial: CRC32
KCRC0POLY0	0x04C11DB7	
KCRC0DOUT0	0xFFFFFFFF	Start value for CRC generation

Table 3-49 RS-CANFD0 Register Settings

Register	Set value	Function
RSCFD0CFDC1NCFG	0x0D181000	Nominal Bit Rate: 1 Mbps
RSCFD0CFDC1DCFG	0x026B0000	Data Bit Rate: 2 Mbps
RSCFD0CFDRFCC1	0x00000271	Receive FIFO Interrupt Source Select : An interrupt occurs for each message received.
		Receive FIFO Buffer Depth: 8 messages
		Receive FIFO Buffer Payload Storage Size: 64 bytes
		Receive FIFO Interrupt disabled
		Receive FIFO Buffer enabled
RSCFD0CFDGAFLECTR	0x00000100	Receive Rule Table Write enabled
RSCFD0CFDGAFLCFG0	0x00000001	Number of receive rules for channel 1: 1
RSCFD0CFDGAFLID0	0x000007F0	Standard ID
		Data Frame
		Receive Rule Target Message: When a message transmitted from another CAN node is received
		Receive rule ID: 0x7F0
RSCFD0CFDGAFLM0	0xC00007FF	The IDE bit is compared.
		The RTR bit is compared.
		The corresponding ID bit is compared.
RSCFD0CFDGAFLP0_0	0xF1230000	Receive Rule DLC: 64 data bytes
		No receive buffer is used.
RSCFD0CFDGAFLP1_0	0x00000002	Receive FIFO buffer 1 selected
RSCFD0CFDCDTCT	0x00000002	DMA Transfer Request for Receive FIFO buffer 1 enabled

Table 3-50 PBG Register Settings

Register	Set value	Function
PBGERRSLV30 PBGKCPROT	0xA5A5A501	Enable write access to the PBG30 registers.
PBG30 PBGPROT1_6	0x10000001	SPID for KCRC0: set SPID of sDMAC and CPU0 to 1

Table 3-51 sDMAC Register Settings

Register	Set value	Function
DMA0CM_0	0x00001C00	Channel Master SPID: 0x1C
		Supervisor mode
DMA0CHCR_0	0x0302	Descriptor enabled
		Start DMA transfer after the channel configuration is copied from the descriptor memory.
		Channel Address Error Notification disabled
		Channel Address Error Interrupt disabled
		Descriptor Step End Interrupt disabled
		Transfer End Interrupt enabled
		DMA transfer disabled
DMA0DPCR_0	0x0000000F	Update flag of Descriptor DMAjSAR_n register update enabled DMAjDAR_n register update enabled DMAjTSR_n register update enabled DMAjTMR_n register update enabled DMAjGIAI_n register update disabled DMAjGOAI_n register update disabled DMAjSIAI_n register update disabled DMAjSOAI_n register update disabled DMAjSGCR_n register update disabled DMAjRS_n register update disabled DMAjBUFCR_n register update disabled
DMA0RS_0	0x001000C3	Transfer count per hardware request: 16
		Transfer limit per hardware request: Transaction size indicated by DMAjTMR_n.STS * DMAjRS_n.TC
		Pre-load function disabled
		DRQ initialize disabled
		DMA request source: INTRCANRFDREQ1(group0-195)
DMA0DPPTR_0	0x00000001	Address pointer of Descriptor: 0x000
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled
DMATRGSEL DMACSEL0_12	0xFFFFFF3F	CAN0(group0-195)

Table 3-52 Descriptor 0 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long) &RSCFD0.CFDRFDF0_0.UINT32 (Receive FIFO buffer access data field)
DMA0DAR_0	-	Destination Address: (unsigned long) r_dat (First address of the receive data)
DMA0TSR_0	0x00000004	Transfer Size: 4 bytes
DMA0TMR_0	0x00001522	Transfer Request Source: Hardware Request
		Destination address is incremented based on destination transaction size
		Source address is incremented based on source transaction size
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000015	Address pointer of Descriptor: 0x014
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-53 Descriptor 1 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long) &RSCFD0.CFDRFDF1_0.UINT32 (Receive FIFO buffer access data field)
DMA0DAR_0	-	Destination Address: (unsigned long) r_dat [1] (Address of the element with index No. 1 in the received data)
DMA0TSR_0	0x0000003C	Transfer Size: 60 bytes
DMA0TMR_0	0x00000522	Transfer Request Source: Auto Request
		Destination address is incremented based on destination transaction size
		Source address is incremented based on source transaction size
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000029	Address pointer of Descriptor: 0x028
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-54 Descriptor 2 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long) r_dat (First address of the receive data)
DMA0DAR_0	-	Destination Address: (unsigned long)&KCRC0.DIN.UINT32 (KCRC input register)
DMA0TSR_0	0x0000003C	Transfer Size: 60 bytes
DMA0TMR_0	0x00000122	Transfer Request Source: Auto Request
		Fixed destination address
		Source address is incremented based on source transaction size
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x0000003D	Address pointer of Descriptor: 0x03C
		Descriptor Interrupt disabled
		Continuation flag of Descriptor enabled

Table 3-55 Descriptor 3 Settings

Register	Set value	Function
DMA0SAR_0	-	Source Address: (unsigned long)&KCRC0.DOUT.UINT32 (KCRC0 output register)
DMA0DAR_0	-	Destination Address: (unsigned long) &kcrc_out (Address of CRC result variable)
DMA0TSR_0	0x00000004	Transfer Size: 4 bytes
DMA0TMR_0	0x00000022	Transfer Request Source: Auto Request
		Fixed destination address
		Fixed source address
		DMA destination transaction size: 4-byte unit transfer
		DMA source transaction size: 4-byte unit transfer
DMA0DPPTR_0	0x00000000	Descriptor Interrupt disabled
		Continuation flag of Descriptor disabled

Table 3-56 Interrupt Register Settings

Register	Set value	Function
EIBD70	0x00000000	Bind interrupt to PE0 (CPU0)
EIC70	0x0040	Table reference method, priority level 0

Table 3-57 shows a list of functions used in this operation example.

Table 3-57 List of Functions

Function	Overview
main_pe0	Call each function.
PORT_Init	Initialize I/O ports.
R_CAN_Init	Initialize RS-CANFD.
R_CAN_GlobalStart	Start the global operation of the RS-CANFD.
R_CAN_ChStart	Start the channel operation of the RS-CANFD.
sdmac_rcv_init	Initialize the sDMAC.
descriptor_init	Initialize the descriptor RAM.
kcrc_init	Initialize the KCRC.
intc_init	Initialize the interrupt controller.

3.2.4 Operation Sequence

Figure 3-6 shows the sequence of operation for the transmit operation. The number in parentheses () is the number of the corresponding descriptor. DMA transfer with each descriptor setting is performed as follows.

(0) The transmit data is written from RAM to the KCRC input register to generate a CRC.
(1) The CRC generated in step (0) is read and written to the end of the transmitted data.
(2) The transmit data is written to the Transmit/Receive FIFO of the RS-CANFD.
(3) Transmission is initiated by writing FFh to the transmit/receive FIFO buffer pointer control register.

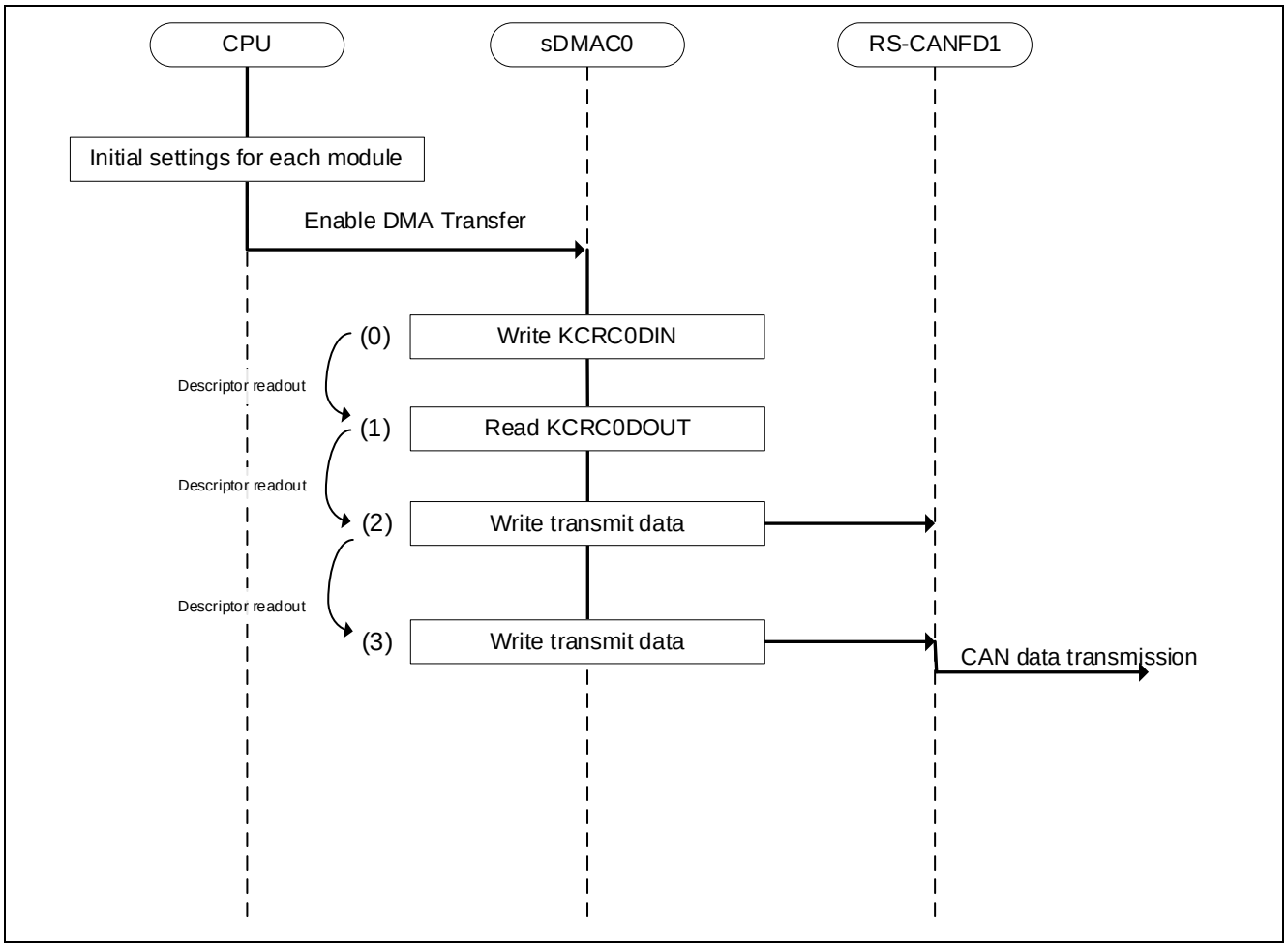

Figure 3-6 Operation Sequence of Transmission

Figure 3-7 shows the sequence of operations for the receive operation. The number in parentheses () is the number of the corresponding descriptor. DMA transfers with each descriptor setting are performed as follows.

(0) The receive data is read from the Receive FIFO triggered by a Receive FIFO DMA request from RS-CANFD.
(1) The received data read in step (0) is written to the KCRC input register to generate a CRC.
(2) The CRC generated in step (1) is read back and stored in RAM.

After all DMA transfers are completed, an sDMAC transfer end interrupt is generated. The CPU compares the CRC received in the interrupt process with the generated CRC to verify the validity of the transmitted data.

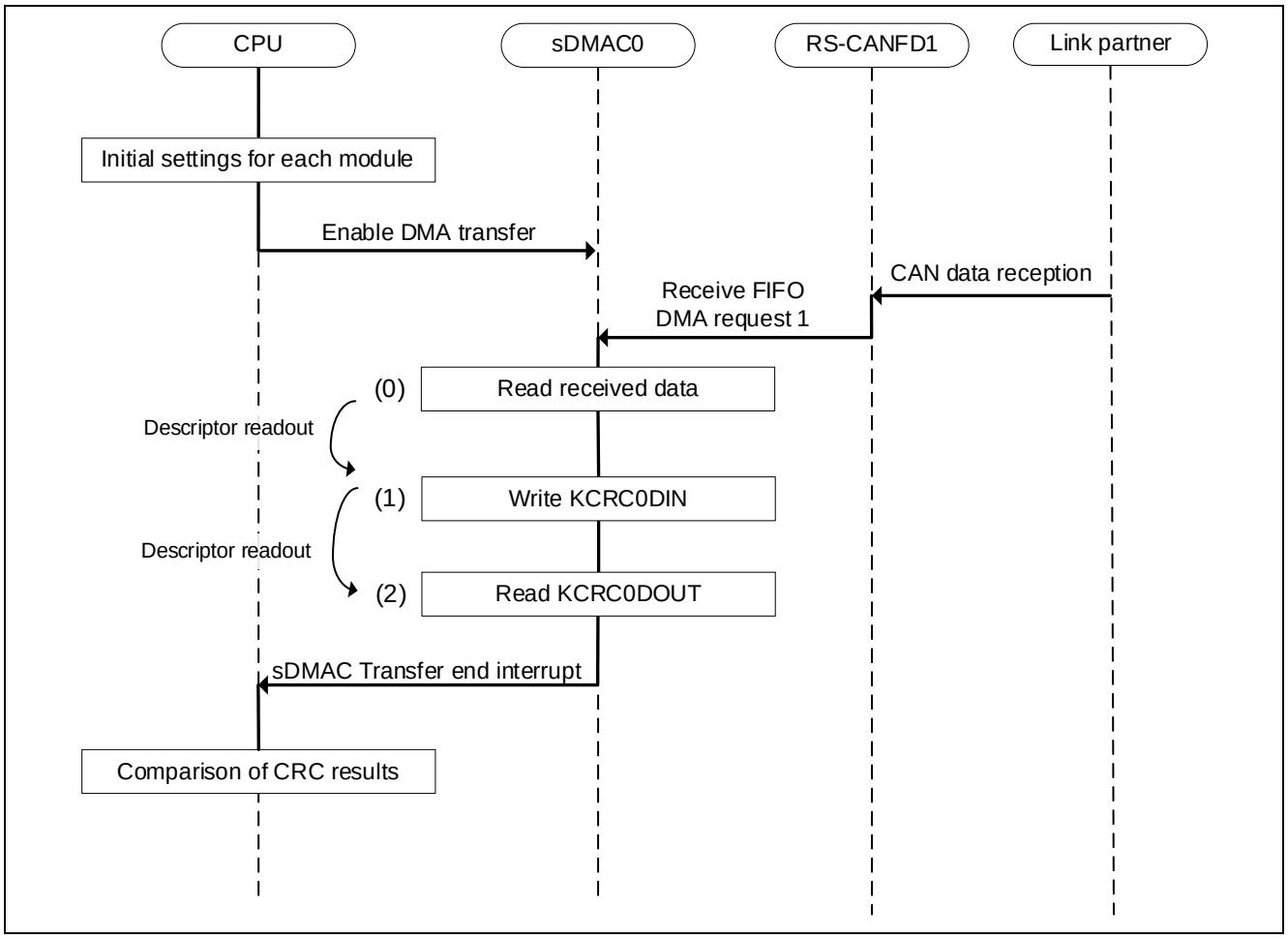

Figure 3-7 Operation Sequence of Reception

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug 31, 2023	-	Issued the 1st edition.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.