## Notes

By Ernest Jih

## Revision History

**April 19, 2001**: Initial publication.

**December 18, 2001**: Added RC32351.

## Objectives

In this application note, the operations of ATM VC Cache and Frame Mode Output will be explained, and an example will be given to illustrate how to set up the DMA Descriptor for both ATM VC Cache and Frame Mode Output.

## Background Information

### ATM Background

ATM is a connection-oriented technology. Every ATM Cell has a VPI/VCI header identity code (not address). The RC32355/RC32351 has an ATM interface that can receive and transmit ATM cells. This means a VPI/VCI table should be set up inside the RC32355/RC32351 before it can recognize different ATM cells. ATM Cell format is shown in Figure 1.

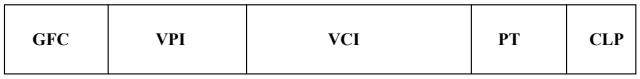



**Figure 1 VPI/VCI Header**

PDU stands for Protocol Data Unit. The format of AAL5 PDU is shown in Figure 2.

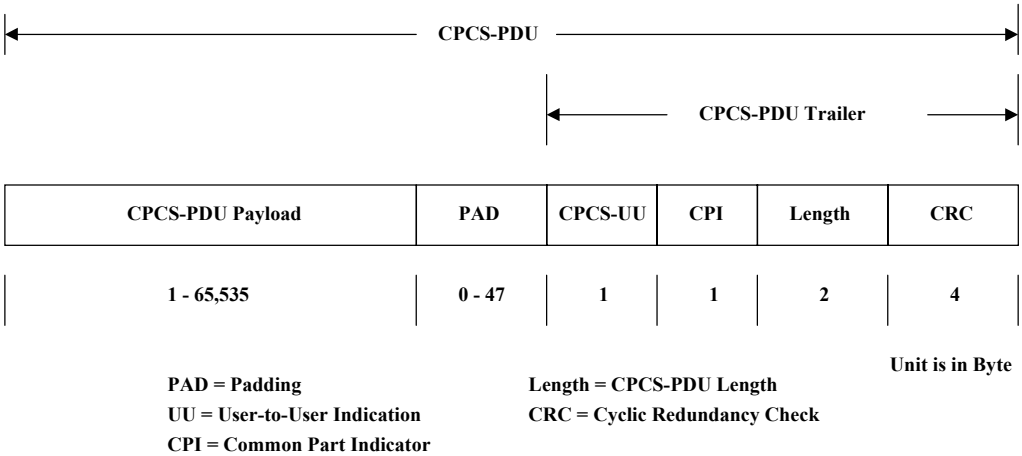



**Figure 2 AAL5 Common Part Convergence Sublayer (CPCS)**

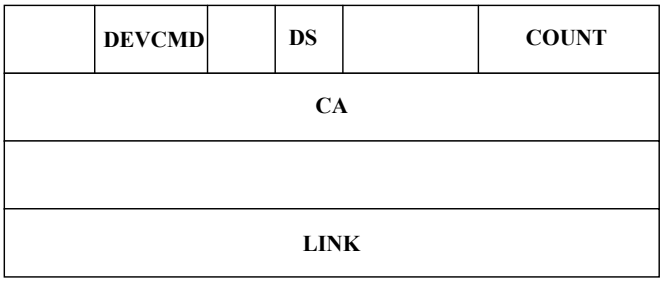
## Notes

## DMA Operations

DMA stands for direct memory access and performs data transfer between main memory and peripheral devices without CPU intervention. In order to perform a DMA operation, proper initialization should be done. The most common parameters needed to initialize the DMA controllers are as follows:

1. Source address
2. Destination address
3. Number of bytes transfer

In the RC32355/RC32351, these parameters are grouped together to form a data structure called ***DMA Descriptor*** (as shown in Figure 3).




**DS : Device Select**
**COUNT: Byte Count**
**CA: Current Address**
**LINK: Address of next descriptor in the list**
**DEVCMD: Device Command**

**Figure 3 The DMA Descriptor**

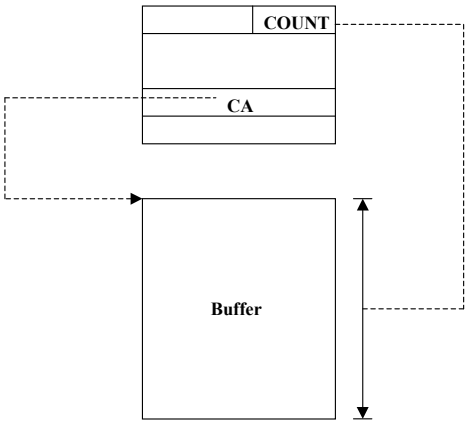
DS (Device Select) can be a source or destination address

CA (Current Address) can also be a source or destination address

COUNT (Byte Count) is the number of bytes transfered

DEVCMD and LINK will be explained later.

As shown in Figure 4, a DMA Descriptor is a data structure that describes a block of memory or a memory buffer which will be used in data transfer.




**COUNT : Size of Buffer**
**CA: Current Address**

**Figure 4 Physical Meanings of DMA Descriptor**

Notes

## DMA Descriptor Lists and Chaining

DMA Descriptors can be linked together by using the LINK field of DMA Descriptor to form a descriptor list (by doing so, buffers can be linked together to form a bigger buffer). The purpose of using the DMA Descriptor list is to allocate buffer space dynamically (i.e., the DMA Descriptor can be added to or removed from the DMA Descriptor list as required to accommodate different buffer sizes).

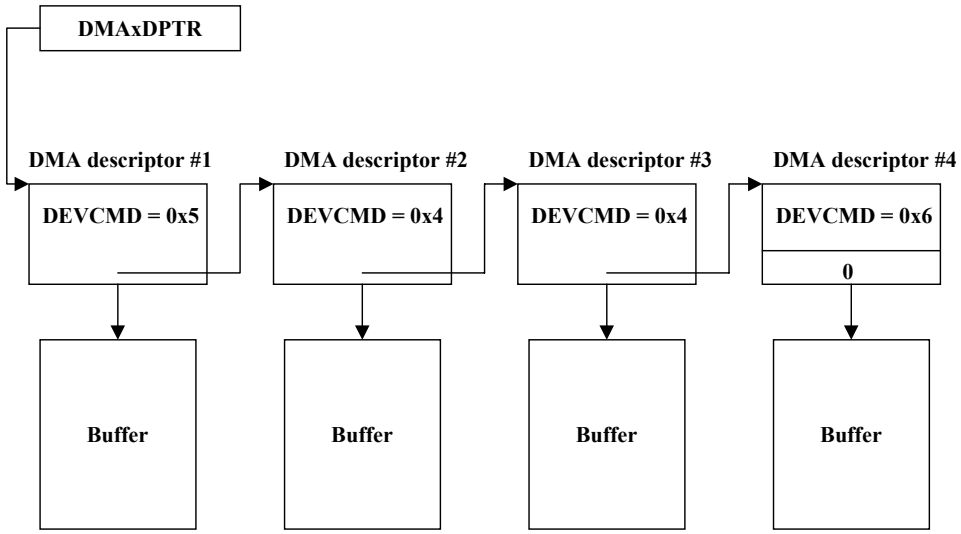



**Figure 5 DMA Descriptor List**

In DMA Descriptor link list, DEVCMD field of DMA Descriptor, is used to indicate the type of DMA Descriptor. A detail description of DEVCMD field is stated as follows:

| DEVCMD | Description |
|---|---|
| 0x0 | No CRC |
| 0x1 | CRC-10 |
| 0x2 | Reserved |
| 0x3 | Reserved |
| 0x4 | CRC-32 |
| 0x5 | CRC-32 – first descriptor |
| 0x6 | CRC-32 – last descriptor |
| 0x7 | CRC-32 – first and last descriptor |

As shown in Figure 5, four DMA Descriptors are linked together to form a DMA Descriptor link list. The DMA Descriptor contains the address of the next descriptor in its LINK field. The value of DEVCMD field of the DMA Descriptor indicates the position of the DMA Descriptor in the descriptor list: 0x5 is the first, 0x4 is the middle, and 0x6 is the last. A single DMA Descriptor should have DEVCMD set to 0x7. The LINK field of the last DMA Descriptor is set to zero which indicates the end of descriptor list.

Notes

# ATM VC Cell Transmission and Reception Using DMA in the RC32355/RC32351

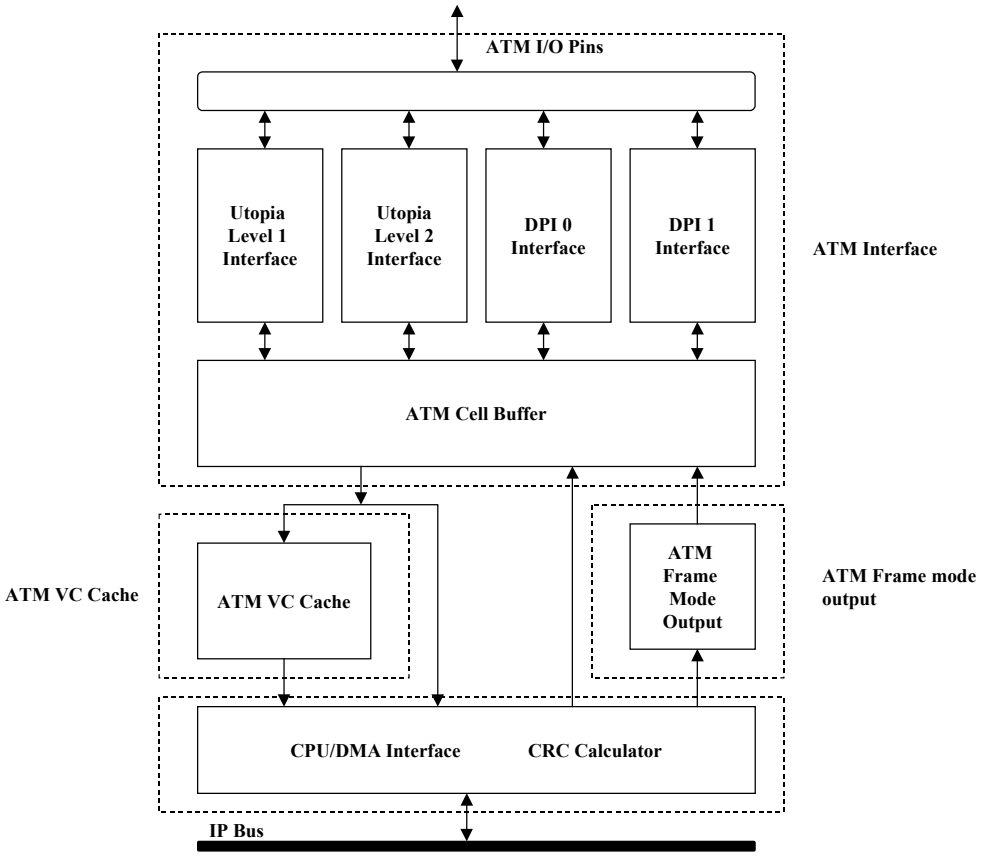



Figure 6

## ATM Interface

The ATM interface is used for receiving or transmitting ATM cells, as shown in Figure 6. The RC32355/RC32351 has an ATM Interface that contains 16 Cell Buffers, UTOPIA1, UTOPIA2, DPI0(Single DPI mode) and DPI1 (Dual DPI mode) interface. ATMINTFC (ATM Interface Control Register) is used for choosing different interfaces.

## ATM Cell Reception Using VC Cache

As mentioned in the *ATM Background* section, a VC table should be set up inside the RC32355/RC32351 so that the RC32355/RC32351 can find a valid connection by comparing the incoming VC to each VC entry in the table. VC Cache is designed for speeding up this kind of VC comparison. If there is a match (cache hit), the corresponding DMA channel will perform the data transfer.

## Notes

## ATM VC Cache Operations

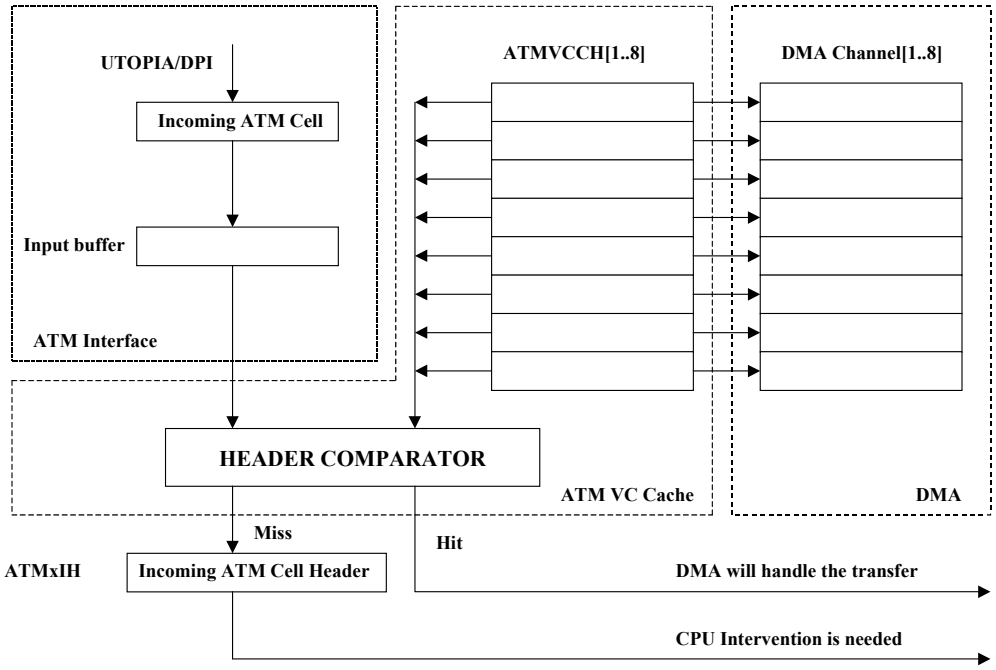



**Figure 7 ATM VC Cache Operations**

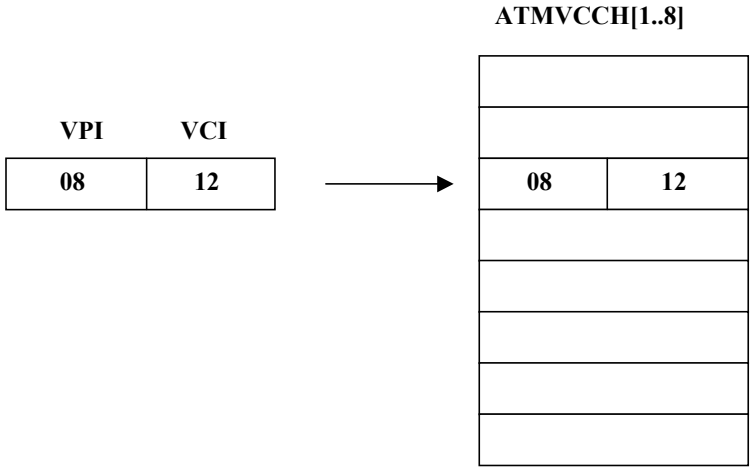
As shown in Figure 7, when an input cell arrives, its VPI and VCI fields are simultaneously compared to all active VC Cache Entry registers. A VC Cache hit occurs when the VPI and VCI fields match a VC Cache Header register. A DMA request is sent to the DMA Controller and the cell payload is DMAed.

If the input cell header fails to match any of the active entries in the VC Cache Entry registers, then CPU processing is required. The input cell header is written to the corresponding ATM Input Header (ATM[0|1]H) register. The Input ready bit will be set in the corresponding ATM Status (ATM[0|1]S) register. The CPU can access the data through either the ATMID register or transferring data through DMA Channel 0

## ATM Cache Entry Initialization

Step 1: A VC Cache entry is initialized by writing the VPI and VCI of the VC to be cached into the corresponding ATM VC Cache Header (ATMVCCH[1..8]) register.

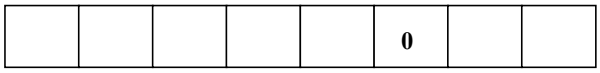
Example: VPI=0x08, VCI=0x12 for ATMVCCH[3]



## Notes

Step 2: The ATM VC Cache entry is enabled by clearing the corresponding disable bit in the ATM Cache Disable (ATMCCD) register.

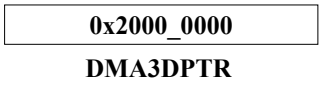
Example: Enable VC Cache entry 3 for VPI/VCI header comparison, ATMVCCD[3] = 0



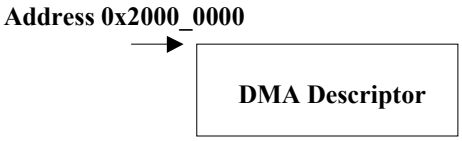

**ATMVCCD (ATM VC Cache Disable Register)**

Step 3: A DMA Descriptor address is written to DMAxPTR to complete the initialization.

Example: Enable DMA channel 3. The DMA Descriptor is stored at memory address 0x2000_0000.




**DMA3DPTR**




### Adding a New ATM Cache Entry

A new ATM Cache entry can be added by reading the ATM VC Cache free entry (ATMVCCFE) register. Reading this register returns the index of lowest VC Cache entry which is free. If the index is non-zero, the VC Cache entry should be initialized as described in the *ATM Cache Entry Initialization* section.

### DMA Descriptor Setup for VC Cache Entry

The following example illustrates how to set up a DMA Descriptor for a VC Cache entry, for transferring 32 ATM Cells from the VC Cache Entry 3 to memory location 0x8000_0000. The address of the DMA Descriptor is stored at memory location 0x2000_0000. The DMA Descriptor should be initialized as shown in Table 1.

| DMA Descriptor Parameter | Value | Explanations |
|---|---|---|
| DEVCMD | 0x7 | This descriptor is the only descriptor in the descriptor list (the first and the last descriptor) |
| DS | 0x0 | DMA Channel 3, DS (Device Select) 0 is for the VC Cache Entry 3 |
| COUNT | 0x600 | COUNT = 32*48 =1536(0x600) |
| CA | 0x8000_0000 | Starting address of the memory buffer |

**Table 1**

DMA3DPTR = 0x2000_0000 (the location of the DMA Descriptor in memory)

The sample DMA Descriptor is shown in Figure 8.

## Notes

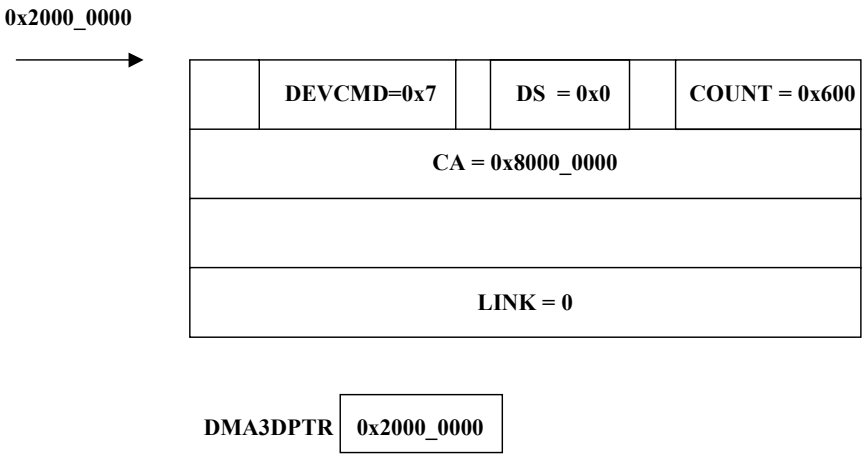



**Figure 8 Sample DMA Descriptor for the VC Cache Entry**

# ATM Cell Transmission Using Frame Mode Output

## Frame Mode Output Operations

.

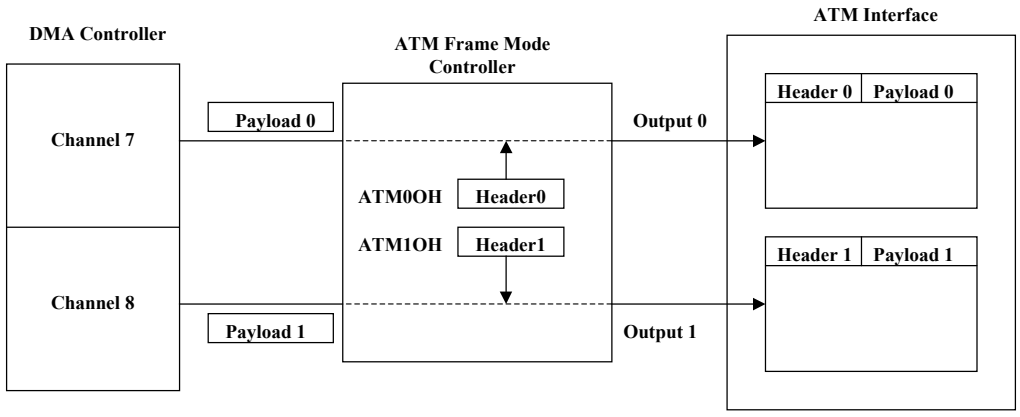



**Figure 9 Frame Mode Output Operations**

The ATM Frame Mode Output controller can be used to transmit contiguous AAL5 CPCS-PDUs with very little overhead. The Frame Mode Output Controller consists of two channels. Output Channel 0 may be used to transmit contiguous AAL5 CPCS-PDUs to ATM cell buffer zero, while Channel One may be used to transmit contiguous AAL5 CPCS-PDUs to ATM cell buffer one.

The ATM Output Header (ATM[0|1]OH) register holds the ATM header to be used when cells are transmitted. The ATM-user-to-user indication bit (CPCS-UU, referring to Figure 2) is automatically set by the ATM frame mode output when the last cell of an AAL5 CPCS-PDU is transmitted.

## DMA Setup in Frame Mode Output

The following example illustrates how to set up a DMA Descriptor for Frame Mode Output, we are going to set up a DMA Descriptor for transferring 32 ATM Cells from the memory location 0x8000_0000 to ATM Cell buffer one. The address of the DMA Descriptor is stored at memory location 0x2000_0000. The DMA Descriptor should be initialized as shown in Table 2.

## Notes

| DMA Descriptor Parameter | Value | Explanations |
|---|---|---|
| DEVCMD | 0x7 | This descriptor is the only descriptor in the descriptor list (the first and the last descriptor) |
| DS | 0x1 | DMA Channel 8, DS (Device Select) 1 is for the ATM Frame Mode Buffer one Output |
| COUNT | 0x600 | COUNT = 32*48 =1536(0x600) |
| CA | 0x8000_0000 | Starting address of the memory buffer |

**Table 2**

DMA8DPTR = 0x2000_0000 (the location of the DMA Descriptor in memory)

A sample DMA Descriptor is shown in Figure 10.

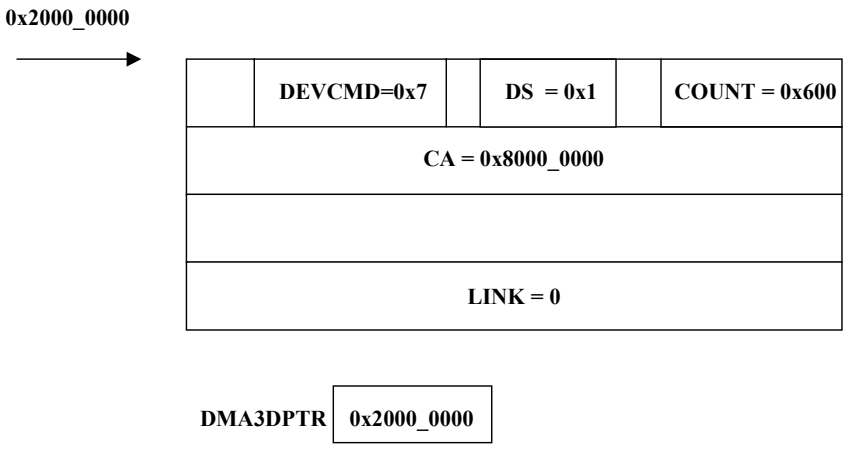



**Figure 10 Sample Descriptor for ATM Frame Mode Output**

## Conclusions

Both ATM VC Cache and Frame Mode Output hardware modules are designed to reduce RISCore 32300 overhead when receiving or transmitting contiguous AAL5 CPCS-PDU. In addition the, DMA Controller is designed to work with these two hardware modules to handle the data transfer without CPU intervention during ATM cell reception or transmission.

## IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)


### Corporate Headquarters

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


### 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-us/.

### Trademarks

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