V850E2/MN4

DMA Control

R01AN0925EJ0100
Rev.1.00
Jan 30, 2012

Introduction

This application note explains how to set up the DMA (Direct Memory Access) and also gives an outline of the operation and describes the procedures for using a sample program. The sample program supports DMA transfer between locations in internal memory and between the internal memory and the internal peripheral I/O.

Target Device

V850E2/MN4 Microcontrollers

Contents

1. Overview 2
2. Usage Environment 5
3. Software 6
4. Sample Application 7

1. Overview

This sample program illustrates the usage examples of the DMAC (Direct Memory Access Controller) and DTFR (DMA Trigger Factor Register).

The parameters required for the transfer of data are stored in the DMAC, which transfers data in response to DMA transfer requests. As an example of software DMA transfer requests, the main points in the operation of the software to transfer data between locations in internal memory are illustrated below.

See section 4.1 “Flow Charts” for the details of the individual operations.

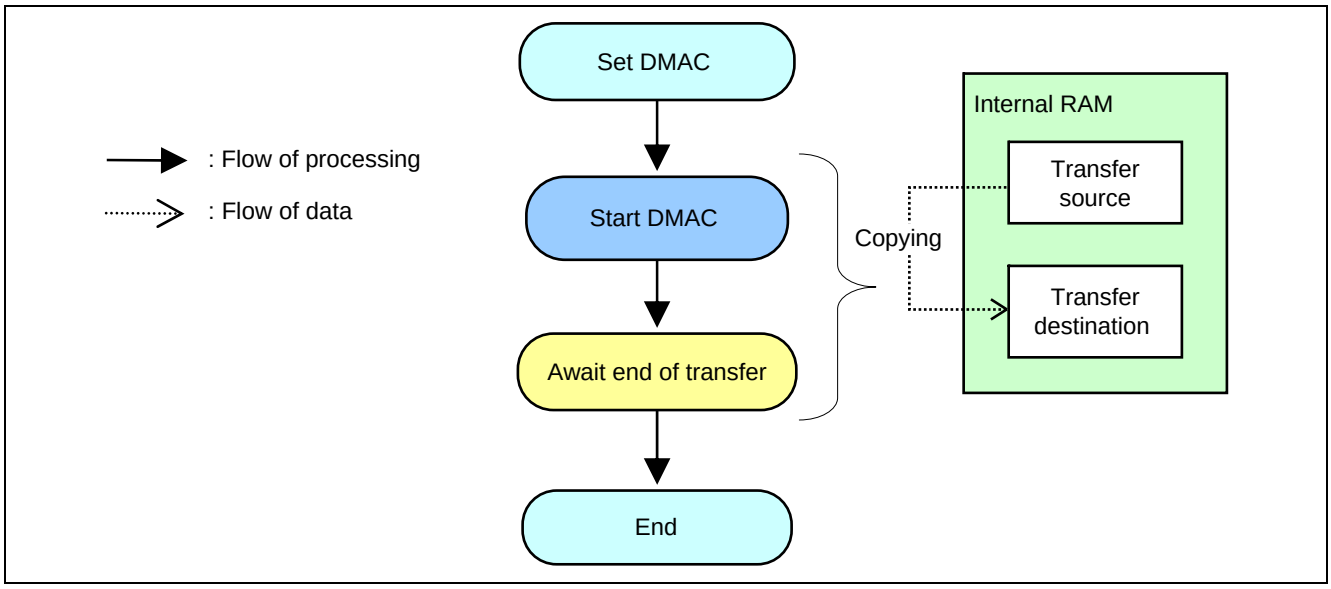

Figure 1.1 Example of Software DMA Transfer Request

The DTFR (DMA Trigger Factor Register) is used to select the interrupt signal which becomes the trigger for DMA from among all interrupt signals. Requests from the DTFR for the DMA transfer of data are handled by the DMAC. Specifically, the signal to be used as a DMA transfer request is selected from among the 128 input interrupt signals by the setting in DTFRn (n = 15 to 0). As an example of a hardware DMA transfer request, the main points in transferring data with a timer interrupt as the trigger are illustrated below. The data from internal RAM are output via port P0.

See section 4.1 “Flow Charts” for the details of the individual operations.

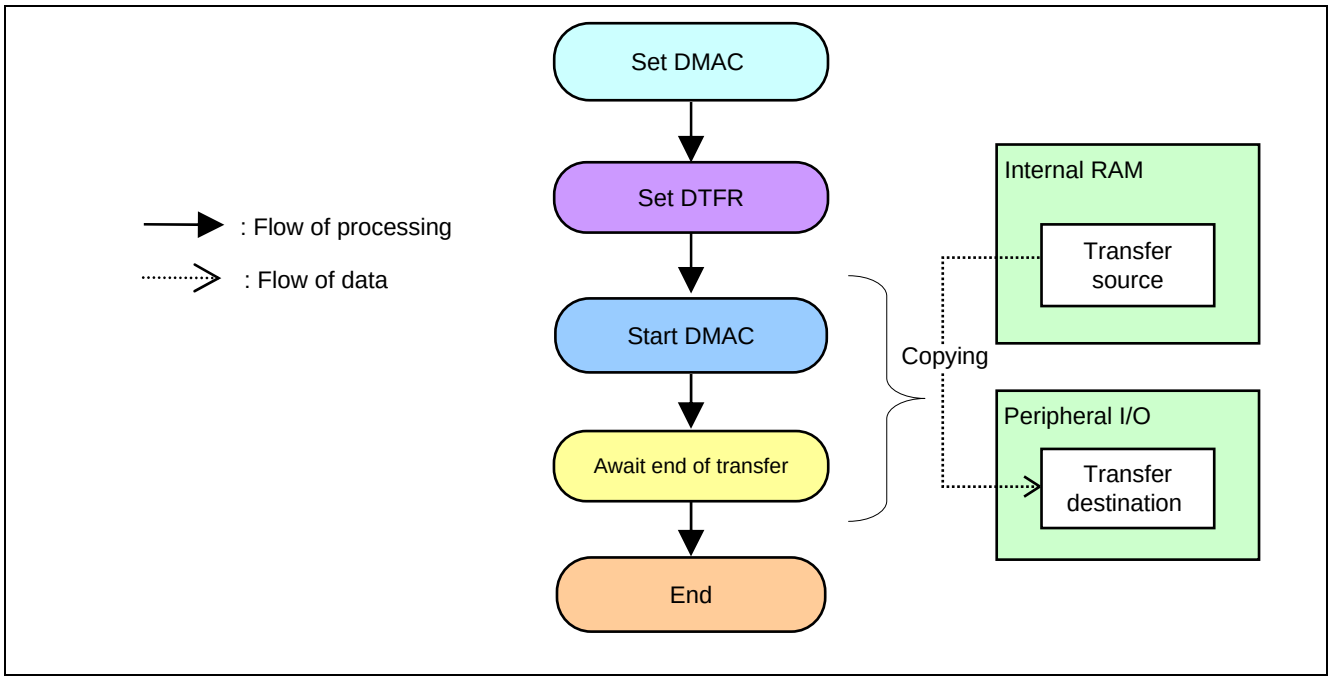

Figure 1.2 Example of Hardware DMA Transfer Request

1.1 Initialization

The general registers and functional pins are initialized.

<Port setup>

- Port n function control expansion registers (PFCEn)
- Port n function control registers (PFCn)
- Port n mode control registers (PMCn)
- Port n mode register (PMn)

1.2 DMAC Setup

The parameters required for the transfer of data are set in the DMAC. See section 4.2 for details.

- A transfer source address, a transfer destination address, and a transfer data size
- Chip select signals for a transfer source and a transfer destination
- A next transfer source address, a next transfer destination address, and a next transfer data size (as necessary)

1.3 DTFR Setup

The interrupt signal which will act as the DMA transfer factor is set.

1.4 DMAC Start

DMA transfer is enabled and the software request bit is set at the same time to start the DMAC transferring data in response to a software transfer request.

DMA transfer is enabled to start the DMAC transferring data in response to hardware transfer requests. Actual data transfer only starts when a request for transfer from the DTFR is triggered by an interrupt signal.

1.5 Wait for End of Transfer and End of Transfer

A transfer end interrupt generated after the DMA transfer indicates the end of transfer. Clearing the DMA transfer end status bit completes DMA transfer. The next address setting function can be used to automatically set the parameters for the next transfer.

2. Usage Environment

This section explains the circuit diagram and development environment to run this sample program.

2.1 Circuit Diagram

See “V850E2/MN4 Target Board User Manual: QB-V850E2MN4DUAL-TB (R20UT0683XJ)” for the details of the circuit diagram.

During data transfer between the internal memory and the internal peripheral I/O, the P0 port is set as the destination for transfer and acts as the output for the data.

The LEDs are connected to port 13. The P13_7 pin is used for LED1. The P13_6 pin is used for LED2.

2.2 Development Environment

It is necessary to install the tools that are listed below to run the sample program.

- CubeSuite+

The integrated development environment CubeSuite+ from Renesas Electronics provides various software development tools that are necessary for the user to develop applications. The user can use these tools seamlessly and easily in various development stages including coding, assembly, compilation, debugging using an emulator or simulator, and flash programming.

- MINICUBE

MINICUBE is a general-purpose in-circuit emulator from Renesas Electronics which adopts the JTAG interface system. It allows the user to debug an onboard real processor and provides highly transparent and stable emulation functionalities. An adapter is required to connect a TB board to MINICUBE.

- Multi

Green Hills software, Inc. integrated development tool suit.

- IAR Embedded Workbench

IAR Systems integrated development tool suit.

3. Software

This section describes the organization of the compressed files to be downloaded.

3.1 File Organization

The compressed files to be downloaded consist of the files that are listed below.

File Name (Tool Structure)	Description	Common Source File	CubeSuite+ File	Multi File
crtE.s	Hardware initialization processing		●	
startup.s				●
V850E2MN4.dir	Link directive file		●	
V850E2_MN4 DMA.ld				●
vector.s	Vector table			●
dma.h	Variable and function declarations	●		
main.c	Main processing	●		
initial.c	Software initialization processing	●		
dma_control.c	DMA control	●		
taua0_control.c	Timer control as DMA start trigger	●		
interrupt.c	Interrupt processing	●		

4. Sample Application

This section explains how to set up the DMAC and the DTFR.

4.1 Flow Charts

The flow charts of this sample program are given below.

4.1.1 Main Processing

The main processing transfers data between locations in internal memory and between the internal RAM and internal peripheral I/O, after which processing ends.

See sections 4.1.2 and 4.1.3 for the details of the individual transfer processing.

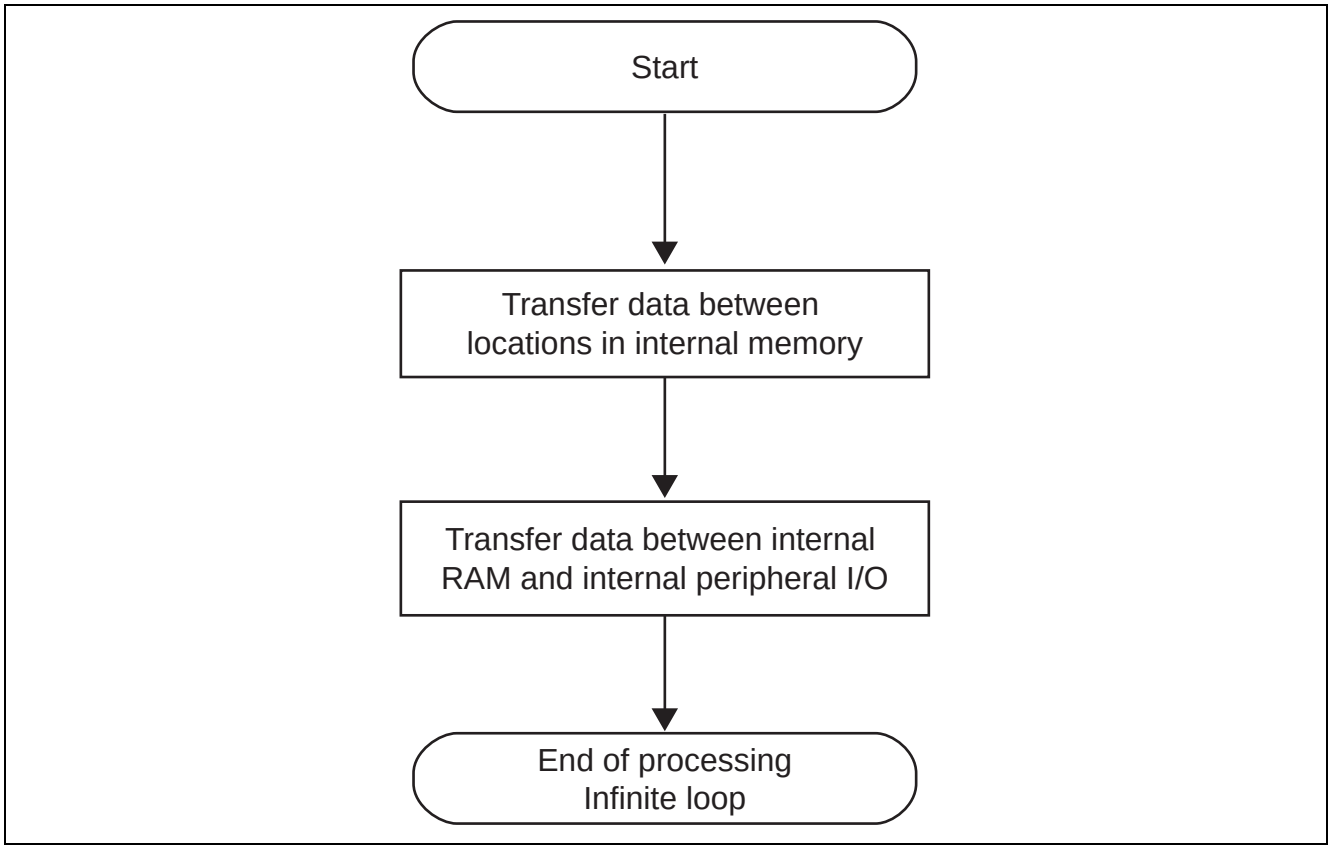

Figure 4.1 Main Processing Flowchart

4.1.2 Inter-Memory Transfer Processing

This inter-memory transfer processing transfers data of specified size from the specified transfer source address to the transfer destination address. In this example, DMA transfer is started by the software.

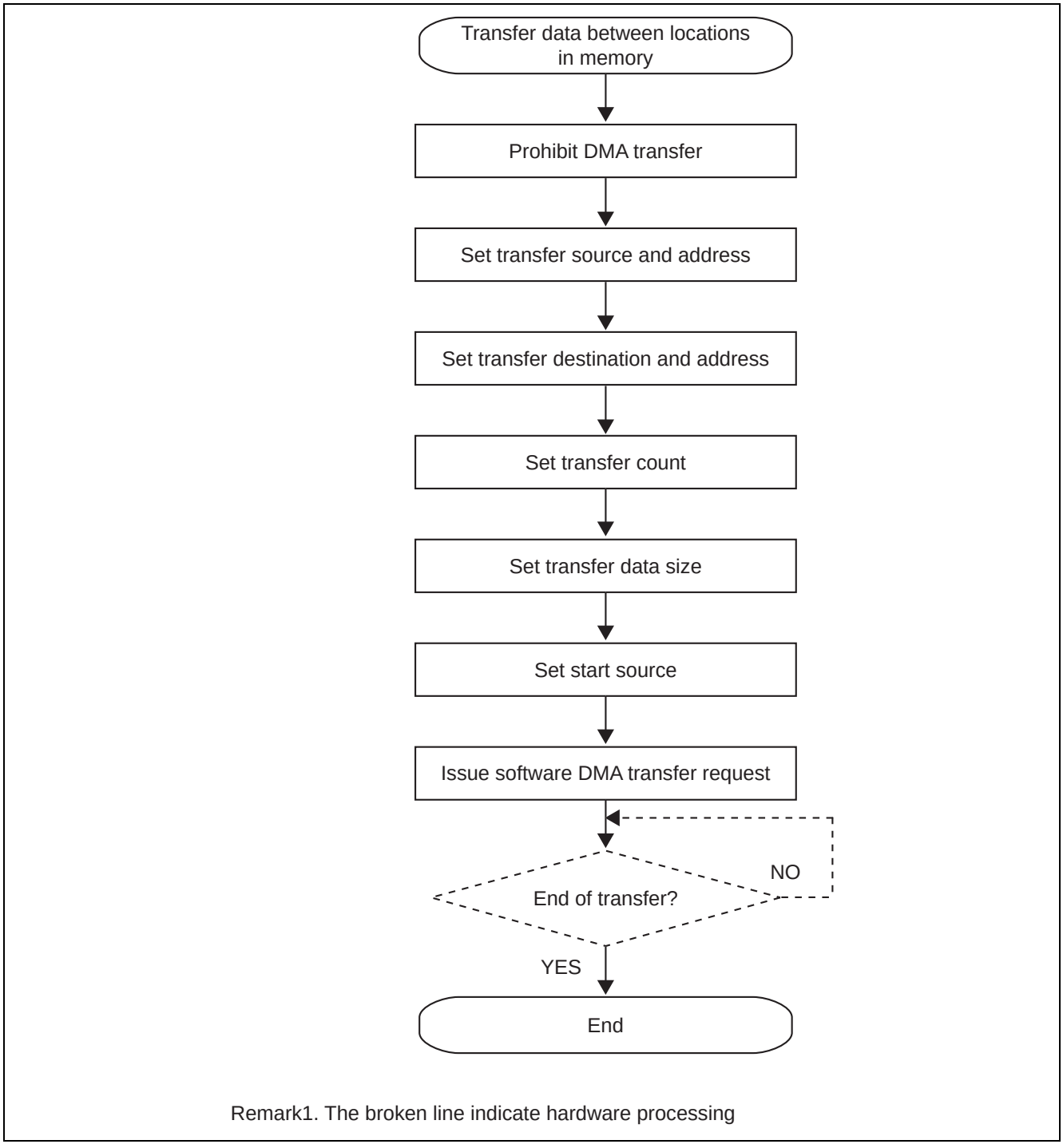

Figure 4.2 Inter–Memory Transfer Processing

4.1.3 Memory-to-Peripheral I/O Transfer Processing Flow

This memory-to-peripheral I/O transfer processing transfers data from the transfer source address to peripheral I/O. In this example, DMA transfer is started by a timer interrupt.

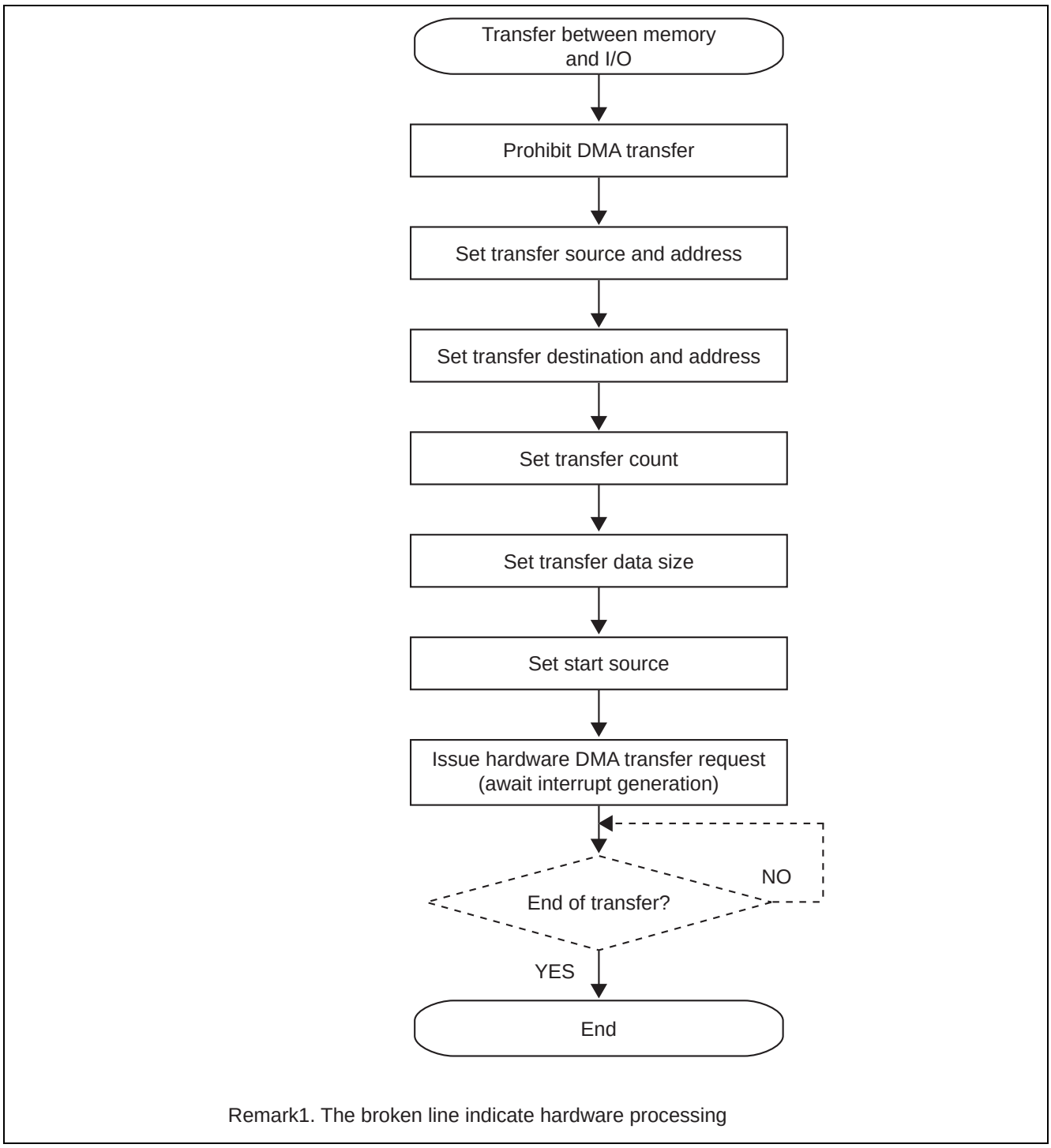

Figure 4.3 Memory–to–Peripheral I/O Transfer Processing Flow

4.1.4 Interrupt Processing

The V850E2/MN4 is able to generate an interrupt when DMA transfer ends or the transfer count matches.

In the case of data transfer between locations in internal RAM, an interrupt signal (INTDMACT0) is generated when the amount of data transferred corresponds to the transfer count setting in the transfer number compare register (DTCC). LED1 is then turned on by the transfer number match interrupt function, int_dmact0.

In the case of data transfer between the internal RAM and the internal peripheral I/O, a transfer completion interrupt signal (INTDMA1) is generated when DMA transfer has been executed the number of times specified by the transfer count register (DTC). The transfer end interrupt function, int_dma1, clears the transfer end flag and then turns LED2 on.

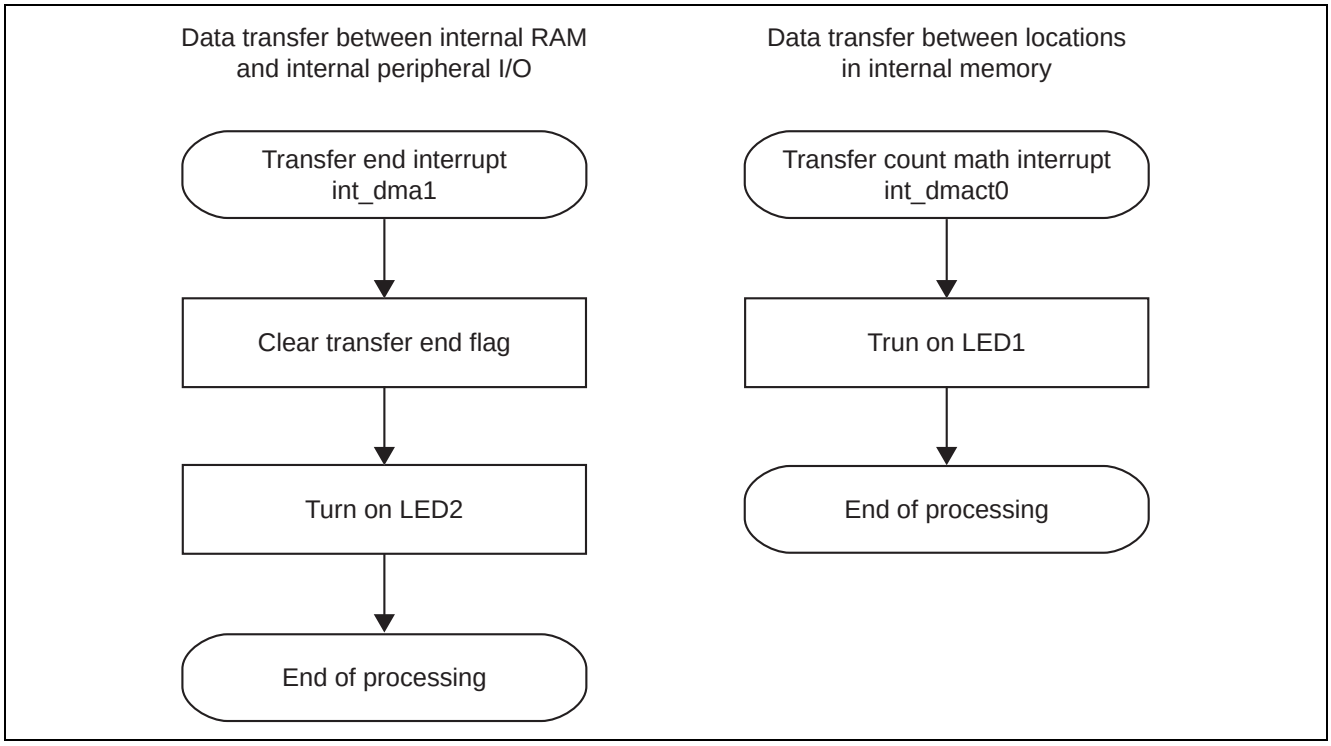

Figure 4.4 Interrupt Processing

4.2 Register Setup

This section explains how to set up the relevant registers according to the flow charts shown in section 4.1. The registers described below must be configured to control the DMAC.

4.2.1 Port Setup

In this sample program, during data transfer between the internal memory and the internal peripheral I/O, the P0 port is set as the destination for transfer and acts as the output for the data. The LEDs are connected to port 13. The pertinent control registers must be set up as summarized in the table below. The P13_7 pin is used for LED1. The P13_6 pin is used for LED2.

Macro	Pin	PMC	PFCE	PFC	PM	Corresponding function
PORT	P0_0 to 15	0	0	0	0	Port mode, output
	P13_6	0	0	0	0	Port mode, output
	P13_7	0	0	0	0	Port mode, output

Setting example:

```
/* P0: io,output */
 PFCE0  = 0x0000;
 PFC0   = 0x0000;
 PMC0   = 0x0000;
 PM0    = 0x0000;

 /* P13_6,7: LEDs,io,OUTPUT */
 PFCE13 = 0x0000;
 PFC13  = 0x0000;
 PMC13  = 0x0000;
 PM13   = 0x0000;
```

4.2.2 DMA Transfer Status Register (DTSn)

The DTSn register has the DMA transfer end status, the DMA transfer status, the DMA transfer error flag, and the hardware DMA transfer request flag. This register enables or disables DMA transfer by using the DTE bit. This register starts and stops software DMA transfer by using the SR bit.

Access This register can be read or written in 8- or 1-bit units.

Address DTS15: FFFF 768A$_H$, DTS14: FFFF 765A$_H$, DTS13: FFFF 762A$_H$, DTS12: FFFF 75FA$_H$, DTS11: FFFF 75CA$_H$, DTS10: FFFF 759A$_H$, DTS9: FFFF 756A$_H$, DTS8: FFFF 753A$_H$, DTS7: FFFF 748A$_H$, DTS6: FFFF 745A$_H$, DTS5: FFFF 742A$_H$, DTS4: FFFF 73FA$_H$, DTS3: FFFF 73CA$_H$, DTS2: FFFF 739A$_H$, DTS1: FFFF 736A$_H$, DTS0: FFFF 733A$_H$

Initial Value 00$_H$

7	6	5	4	3	2	1	0
DTSnTC	DTSnDT	0	0	DTSnER	DTSnDR	DTSnSR	DTSnDTE
R/W	R/W	R	R	R	R	R/W	R/W

Table 10-27 DTSn register contents (1/2)

Bit position	Bit name	Function
7	DTSnTC	DMA transfer end status This bit indicates that DMA transfer has been completed. Write "0" to this bit to clear it after reading "1" from it. It is recommended to write this bit using bit manipulation such as CLR1. 0: DMA transfer not completed 1: DMA transfer completed
6	DTSnDT	DT DMA transfer status This bit indicates that a DMA transfer request has been acknowledged and that DMA transfer is in progress. It is not set (to "1") when only a DMA transfer request is issued. This bit is cleared (to "0") when DMA transfer has been completed. If the DTSnDTE bit is "0", this bit can be cleared by the user. (It can also be written at the same time as the DTSnDTE bit.) 0: DMA transfer request acknowledged 1: DMA transfer in progress
3	DTSnER	DMA transfer error flag This bit indicates that a DMA transfer error has occurred in channel n. It is cleared (to "0") when the DTRCx.DTRCxERR bit is cleared. Note that this bit is read-only. 0: No DMA transfer error 1: DMA transfer error
2	DTSnDR	Hardware DMA transfer request flag This bit indicates that channel n has a hardware DMA transfer request. It is cleared (to "0") when the hardware DMA transfer request is deasserted. This bit operates regardless of the status of the DTSnDTE bit. It is not set (to "1") by a software DMA transfer request, or by a hardware DMA transfer request when a software DMA transfer request is selected by the DMA transfer request select register. Note that this bit is read-only. 0: No hardware DMA transfer request 1: Hardware DMA transfer request
1	DTSnSR	Software DMA transfer request This bit selects a software DMA transfer request. If a software DMA transfer request is selected by the DMA transfer request select register, writing "1" to this bit and the DTSnDTE bit starts DMA transfer. This bit is automatically cleared (to "0") when DMA transfer has been completed. Writing "0" to this bit aborts DMA transfer. 0: No software DMA transfer request 1: Software DMA transfer request

Figure 4.5 DTSn Register Format (1/2)

Table 10-27 DTSn register contents (2/2)

Bit position	Bit name	Function
0	DTSnDTE	DMA transfer enable This bit enables or disables DMA transfer. DMA transfer is executed if "1" is written to this bit and a DMA transfer request is issued. This bit is automatically cleared (to "0") if the MLE bit is "0" when DMA transfer has been completed. DMA transfer is aborted if "0" is written to this bit during DMA transfer. 0: Disables DMA transfer 1: Enables DMA transfer

Figure 4.6 DTSn Register Format (2/2)

Setting examples

```
DTS0DTE = 0x0;          /* prohibit DMA trasnfer */
DTS0DTE = 0x1;          /* permit DMA transfer */
DTS0SR = 0x1;           /* transfer start */
```

4.2.3 DMA Source Address Registers (DSAnL and DSAnH)

These registers set a DMA transfer source address.

Access This register can be read or written in 16-bit units.

Address DSA15L: FFFF 7664$_H$, DSA14L: FFFF 7634$_H$, DSA13L: FFFF 7604$_H$, DSA12L: FFFF 75D4$_H$, DSA11L: FFFF 75A4$_H$, DSA10L: FFFF 7574$_H$, DSA9L: FFFF 7544$_H$, DSA8L: FFFF 7514$_H$, DSA7L: FFFF 7464$_H$, DSA6L: FFFF 7434$_H$, DSA5L: FFFF 7404$_H$, DSA4L: FFFF 73D4$_H$, DSA3L: FFFF 73A4$_H$, DSA2L: FFFF 7374$_H$, DSA1L: FFFF 7344$_H$, DSA0L: FFFF 7314$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-11 DSAnL register contents

Bit position	Bit name	Function
15:0	SA15 to SA0	DMA source address These bits set the lower 16 bits of the transfer source address of channel n. If this register is referenced during DMA transfer, the address from which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DSAnH in 32-bit units. If the NSAV bit of the DNSAnH register is not set (to "1") when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is "0" in order to avoid data being transferred from an address that has not been completely set.
3. DMA transfer of misaligned data is not supported. The lower 4 bits of an address corresponding to the transfer data size are as follows (x indicates any bit).

   The operation is not guaranteed if a setting other than the following is made.

Data size	SA3	SA2	SA1	SA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
128 bits	0	0	0	0

Figure 4.7 DSAnL Register Format

Access This register can be read or written in 16-bit units.

Address DSA15H: FFFF 7666$_H$, DSA14H: FFFF 7636$_H$, DSA13H: FFFF 7606$_H$, DSA12H: FFFF 75D6$_H$, DSA11H: FFFF 75A6$_H$, DSA10H: FFFF 7576$_H$, DSA9H: FFFF 7546$_H$, DSA8H: FFFF 7516$_H$, DSA7H: FFFF 7466$_H$, DSA6H: FFFF 7436$_H$, DSA5H: FFFF 7406$_H$, DSA4H: FFFF 73D6$_H$, DSA3H: FFFF 73A6$_H$, DSA2H: FFFF 7376$_H$, DSA1H: FFFF 7346$_H$, DSA0H: FFFF 7316$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
0	0	0	SA28	SA27	SA26	SA25	SA24
R	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-12 DSAnH register contents

Bit position	Bit name	Function
12:0	SA28 to SA16	DMA source address These bits set the higher 13 bits of the transfer source address of channel n. If this register is referenced during DMA transfer, the address from which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DSAnL in 32-bit units. If the NSAV bit of the DNSAnH register is not set (to "1") when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is "0" in order to avoid data being transferred from an address that has not been completely set.

Figure 4.8 DSAnH Register Format

Setting example

```
DSA0    = 0x1EDFA000;   /* set transfer source address */
```

4.2.4 DMA Destination Address Register (DDAnL DDAnH)

These registers set a DMA transfer destination address.

Access This register can be read or written in 16-bit units.

Address DDA15L: FFFF 7674_H, DDA14L: FFFF 7644_H, DDA13L: FFFF 7614_H, DDA12L: FFFF $75E4_H$, DDA11L: FFFF $75B4_H$, DDA10L: FFFF 7584_H, DDA9L: FFFF 7554_H, DDA8L: FFFF 7524_H, DDA7L: FFFF 7474_H, DDA6L: FFFF 7444_H, DDA5L: FFFF 7414_H, DDA4L: FFFF $73E4_H$, DDA3L: FFFF $73B4_H$, DDA2L: FFFF 7384_H, DDA1L: FFFF 7354_H, DDA0L: FFFF 7324_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-17 DDAnL register contents

Bit position	Bit name	Function
15:0	DA15 to DA0	DMA destination address These bits specify the lower 16 bits of the transfer destination address of channel n. If this register is referenced during DMA transfer, the address to which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DDAnH in 32-bit units. If the NDAV bit of the DNDAnH register is not set (to "1") when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is "0" in order to avoid data being transferred from an address that has not been completely set.
3. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.
4. DMA transfer of misaligned data is not supported. The lower 4 bits of an address corresponding to the transfer data size are as follows (× indicates any bit). The operation is not guaranteed if a setting other than the following is made.

Data size	DA3	DA2	DA1	DA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
128 bits	0	0	0	0

Figure 4.9 DDAnL Register Format

Access This register can be read or written in 16-bit units.

Address DDA15H: FFFF 7676$_H$, DDA14H: FFFF 7646$_H$, DDA13H: FFFF 7616$_H$, DDA12H: FFFF 75E6$_H$, DDA11H: FFFF 75B6$_H$, DDA10H: FFFF 7586$_H$, DDA9H: FFFF 7556$_H$, DDA8H: FFFF 7526$_H$, DDA7H: FFFF 7476$_H$, DDA6H: FFFF 7446$_H$, DDA5H: FFFF 7416$_H$, DDA4H: FFFF 73E6$_H$, DDA3H: FFFF 73B6$_H$, DDA2H: FFFF 7386$_H$, DDA1H: FFFF 7356$_H$, DDA0H: FFFF 7326$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
0	0	0	DA28	DA27	DA26	DA25	DA24
R	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-18 DDAnH register contents

Bit position	Bit name	Function
12:0	DA28 to DA16	DMA destination address These bits specify the higher 13 bits of the transfer destination address of channel n. If this register is referenced during DMA transfer, the address to which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DDAnL in 32-bit units. If the NDAV bit of the DNDAnH register is not set (to "1") when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is "0" in order to avoid data being transferred from an address that has not been completely set.
3. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.

Figure 4.10 DDAnH Register Format

Setting example

```
DDA0    = 0x1EDFB000;   /* set transfer destination address */
```

4.2.5 DMA Source Chip Select Register (DSCn)

This register specifies an area to be selected as a transfer source of channel n.

Access This register can be read or written in 16-bit units.

Address DSC15: FFFF 7668_H, DSC14: FFFF 7638_H, DSC13: FFFF 7608_H, DSC12: FFFF $75D8_H$, DSC11: FFFF $75A8_H$, DSC10: FFFF 7578_H, DSC9: FFFF 7548_H, DSC8: FFFF 7518_H, DSC7: FFFF 7468_H, DSC6: FFFF 7438_H, DSC5: FFFF 7408_H, DSC4: FFFF $73D8_H$, DSC3: FFFF $73A8_H$, DSC2: FFFF 7378_H, DSC1: FFFF 7348_H, DSC0: FFFF 7318_H

Initial Value 0001_H

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	SCS1	SCS0	SCSE
R	R	R	R	R	R/W	R/W	R/W

Table 10-13 DSCn register contents

Bit position	Bit name	Function
2 1 0	SCS1 SCS0 SCSE	DMA source chip select These bits specify an area to be selected as the transfer source of channel n.

SCS1	SCS0	SCSE	Selected area
0	0	1	External memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area
0	1	0	Internal flash memory and internal RAM
Other than above			Setting prohibited

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set the SCS0 and SCSE bits so that only one of them is "1". If both of these bits are set to 1, the operation is not guaranteed.
3. Be sure to set the SCS1 bit to "0".

Figure 4.11 DSCn Register Format

Setting example

```
DSC0    = 0x0002;       /* set DMA source is inner RAM */
```

4.2.6 DMA Destination Chip Select register (DDCn)

This register specifies an area to be selected as a transfer destination of channel n.

Access This register can be read or written in 16-bit units.

Address DDC15: FFFF 7678_H, DDC14: FFFF 7648_H, DDC13: FFFF 7618_H, DDC12: FFFF $75E8_H$, DDC11: FFFF $75B8_H$, DDC10: FFFF 7588_H, DDC9: FFFF 7558_H, DDC8: FFFF 7528_H, DDC7: FFFF 7478_H, DDC6: FFFF 7448_H, DDC5: FFFF 7418_H, DDC4: FFFF $73E8_H$, DDC3: FFFF $73B8_H$, DDC2: FFFF 7388_H, DDC1: FFFF 7358_H, DDC0: FFFF 7328_H

Initial Value 0001_H

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	DCS1	DCS0	DCSE
R	R	R	R	R	R/W	R/W	R/W

Table 10-19 DDCn register contents

Bit position	Bit name	Function
2 1 0	DCS1 DCS0 DCSE	DMA destination chip select These bits specify an area to be selected as the transfer destination of channel n.

DCS1	DCS0	DCSE	Selected area
0	0	1	External memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area
0	1	0	Internal flash memory and internal RAM
Other than above			Setting prohibited

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set the DCS0 and DCSE bits so that only one of them is "1". If both of these bits are set to 1, the operation is not guaranteed.
3. Be sure to set the DCS1 bit to "0".

Figure 4.12 DDCn Register Format

Setting examples

```
DDC0    = 0x0002;      /* set DMA destination is inner RAM */
DDC1    = 0x0001;      /* set DMA destination is IO */
```

4.2.7 DMA Transfer Count Register (DTCn)

This register specifies the number of times of DMA transfers (DMA transfer count). When this register is referenced during DMA transfer, the remaining number of times of DMA transfer to be executed can be read.

Access This register can be read or written in 16-bit units.

Address DCT15: FFFF 7682_H, DCT14: FFFF 7652_H, DCT13: FFFF 7622_H, DCT12: FFFF $75F2_H$, DCT11: FFFF $75C2_H$, DCT10: FFFF 7592_H, DCT9: FFFF 7562_H, DCT8: FFFF 7532_H, DCT7: FFFF 7482_H, DCT6: FFFF 7452_H, DCT5: FFFF 7422_H, DCT4: FFFF $73F2_H$, DCT3: FFFF $73C2_H$, DCT2: FFFF 7392_H, DCT1: FFFF 7362_H, DCT0: FFFF 7332_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	DTC14	DTC13	DTC12	DTC11	DTC10	DTC9	DTC8
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTC7	DTC6	DTC5	DTC4	DTC3	DTC2	DTC1	DTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-23 DTCn register contents

Bit position	Bit name	Function
14:0	DTC14 to DTC0	DMA transfer count These bits specify the number of times of DMA transfers (DMA transfer count) for channel n. When this register is referenced during DMA transfer, the remaining number of times DMA transfer to be executed can be read. If the NTCV bit of the DNTCn register is not set (to "1"), these bits hold the values when DMA transfer has been completed (0000H).

DTC[14:0]	The operation
0000H	Transfer executed 32,768 times or until completion of transfer
0001H	Transfer executed once or transfer to be executed once
...	...
7FFFH	Transfer executed 32,767 times or 32,767 times of transfer to be executed

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.

Figure 4.13 DTCn Register Format

Setting examples

```
DTC0    = 0x0004;      /* set times of DMA transfer */
DTC1    = 0x0001;      /* set times of DMA transfer */
```

4.2.8 DMA Transfer Control Register (DTCTn)

This register specifies parameters, such as DMA transfer data size.

Access This register can be read or written in 16-bit units.

Address DTCT15: FFFF 7688_H, DTCT14: FFFF 7658_H, DTCT13: FFFF 7628_H, DTCT12: FFFF $75F8_H$, DTCT11: FFFF $75C8_H$, DTCT10: FFFF 7598_H, DTCT9: FFFF 7568_H, DTCT8: FFFF 7538_H, DTCT7: FFFF 7488_H, DTCT6: FFFF 7458_H, DTCT5: FFFF 7428_H, DTCT4: FFFF $73F8_H$, DTCT3: FFFF $73C8_H$, DTCT2: FFFF 7398_H, DTCT1: FFFF 7368_H, DTCT0: FFFF 7338_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	DS1	DS0	MLE	0	0	0	0
R	R/W	R/W	R/W	R	R	R	R

7	6	5	4	3	2	1	0
SACM1	SACM0	DACM1	DACM0	0	0	0	DSM
R/W	R/W	R/W	R/W	R	R	R	R/W

Table 10-26 DTCTn register contents (1/2)

Bit position	Bit name	Function
14 13	DS1 DS0	DMA transfer data size These bits specify the DMA transfer data size of channel n. DS1 / DS0 / Transfer data size 0 / 0 / 8 bits 0 / 1 / 16 bits 1 / 0 / 32 bits 1 / 1 / 128 bits
12	MLE	Multi-link enable This bit specifies whether to acknowledge the next DMA transfer request, even if the DTSnTC bit is not cleared (to "0") after DMA transfer has been completed. If this bit is set (to "1"), the DTSn.DTSnDTE bit is not cleared upon completion of DMA transfer. Even if the DTSnTC bit is not cleared, DMA transfer is executed if a DMA transfer request is issued. 0: Clears DTSnDTE bit upon completion of DMA transfer. 1: Does not clear DTSnDTE bit upon completion of DMA transfer.
7 6	SACM1 SACM0	DMA transfer source address counting direction These bits specify the direction in which the transfer source address of channel n is to be counted. SACM1 / SACM0 / Counting direction 0 / 0 / Incremented 0 / 1 / Decremented 1 / 0 / Fixed 1 / 1 / Setting prohibited

Figure 4.14 DTCTn Register Format (1/2)

Table 10-26 DTCTn register contents (2/2)

Bit position	Bit name	Function
5 4	DACM1 DACM0	DMA transfer destination address counting direction These bits specify the direction in which the transfer destination address of channel n is to be counted. DACM1 / DACM0 / Counting direction: 0 / 0 / Incremented 0 / 1 / Decremented 1 / 0 / Fixed 1 / 1 / Setting prohibited
0	DSM	DMA signal mode This bit specifies the output timing for the $\overline{\text{DMAAK[0:5]}}$ and $\overline{\text{DMATC[0:5]}}$ output pins. 0: Read cycle 1: Write cycle The DSM bit is provided only for the DTCD0 to DTCD5 registers. The DTCD6 to DTCD15 registers do not have this bit.

Cautions
1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. The operation cannot be guaranteed if the SACM[1:0] and DACM[1:0] bits are set to a prohibited status
3. Be sure to set bits 11 and 0 to "0".

Figure 4.15 DTCTn Register Format (2/2)

Setting examples

```
DTCT0   = 0x1000;       /* 8bit data size;donot clear DTE;
                        increment at tranfer source;
                        increment at tranfer destination;
                        output at read cycle */
DTCT1   = 0x3000;       /* 16bit data size;donot clear DTE;
                        increment at tranfer source;
                        increment at tranfer destination;
                        output at read cycle */
```

4.2.9 DMA Transfer Request Select Register (DTRSn)

This register specifies assignment of a DMA transfer request. A software DMA transfer request or a hardware DMA transfer request can be selected as a DMA transfer request.

Access This register can be read or written in 16-bit units.

Address DTRS15: FFFF 7660_H, DTRS14: FFFF 7630_H, DTRS13: FFFF 7600_H, DTRS12: FFFF $75D0_H$, DTRS11: FFFF $75A0_H$, DTRS10: FFFF 7570_H, DTRS9: FFFF 7540_H, DTRS8: FFFF 7510_H, DTRS7: FFFF 7460_H, DTRS6: FFFF 7430_H, DTRS5: FFFF 7400_H, DTRS4: FFFF $73D0_H$, DTRS3: FFFF $73A0_H$, DTRS2: FFFF 7370_H, DTRS1: FFFF 7340_H, DTRS0: FFFF 7310_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	DTR3	DTR2	DTR1	DTR0
R	R	R	R	R/W	R/W	R/W	R/W

Table 10-10 DTRSn register contents

Bit position	Bit name	Function
3:0	DTR3 to DTR0	DMA transfer request assignment These bits specify assignment of a DMA transfer request to channel n.

DTR3	DTR2	DTR1	DTR0	DMA transfer request
0	0	0	0	Software DMA transfer request
0	0	0	1	Hardware DMA transfer request
Other than above				Setting prohibited

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. The operation is not guaranteed if DTR[3:0] are set to a prohibited status.

Figure 4.16 DTRSn Register Format

Setting examples

```
DTRS0 = 0x0000;          /* software transfer request */
DTRS1 = 0x0001;          /* hardware transfer request */
```

4.2.10 DTFR Control Register (DTFRn)

This register enables or disables the operation of the DMA source selector of channel n and selects the transfer source.

Access This register can be read or written in 16-bit units.

Address DTFR0: FFFF 7B00$_H$, DTFR1: FFFF 7B02$_H$, DTFR2: FFFF 7B04$_H$,
DTFR3: FFFF 7B06$_H$, DTFR4: FFFF 7B08$_H$, DTFR5: FFFF 7B0A$_H$,
DTFR6: FFFF7B0C$_H$, DTFR7: FFFF 7B0E$_H$, DTFR8: FFFF 7B10$_H$,
DTFR9: FFFF 7B12$_H$, DTFR10: FFFF 7B14$_H$, DTFR11: FFFF 7B16$_H$,
DTFR12: FFFF 7B18$_H$, DTFR13: FFFF 7B1A$_H$, DTFR14: FFFF 7B1C$_H$,
DTFR15: FFFF 7B1E$_H$

Initial Value 0000_H

15	14	13	12	11	10	9	8
REQEN	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	IFCn6-0						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-29 DTFRn register contents

Bit position	Bit name	Function
15	REQEN	This bit enables or disables operation of the DMA source selector of channel n. 1: Enables operation of source selector 0: Stops operation of source selector. Does not issue DMA transfer request (DMARQ). The settings of IFCn6 to IFCn0 are valid. Requests are always sampled.
6:0	IFCn6 to IFCn0	These bits select the transfer source. The set values are shown in *Table 10-2 "DMA start sources (0 to 63)"*.

Figure 4.17 DTFRn Register Format

Setting example

```
DTFR0 = 0x0000;          /* software transfer request */
DTFR1 = 0x8020;          /* set hardware transfer request of dmac1 is taua0i0*/
```

Table 4.1 DMA Start Sources (1/2)

DTFRn.IFCn[6:0]	Interrupt to start DMA
0	INTP0
1	INTP1
2	INTP2
3	INTP3
4	INTP4
5	INTP5
6	INTP6
7	INTP7
8	INTP8
9	INTP9
10	INTP10
11	INTP11
12	INTP12
13	INTP13
14	INTP14
15	INTP15
16	INTP16
17	INTP17
18	INTP18
19	INTP19
20	INTP20
21	INTP21
22	INTP22
23	INTP23
24	INTP24
25	INTP25
26	INTP26
27	INTP27
28	INTADCA0I0
29	INTADCA0I1
30	INTADCA0I2
31	INTADCA0ILLT

DTFRn.IFCn[6:0]	Interrupt to start DMA
32	INTTAUA0I0
33	INTTAUA0I1
34	INTTAUA0I2
35	INTTAUA0I3
36	INTTAUA0I4
37	INTTAUA0I5
38	INTTAUA0I6
39	INTTAUA0I7
40	INTTAUA0I8
41	INTTAUA0I9
42	INTTAUA0I10
43	INTTAUA0I11
44	INTTAUA0I12
45	INTTAUA0I13
46	INTTAUA0I14
47	INTTAUA0I15
48	INTTAUA1I0
49	INTTAUA1I1
50	INTTAUA1I2
51	INTTAUA1I3
52	INTTAUA1I4
53	INTTAUA1I5
54	INTTAUA1I6
55	INTTAUA1I7
56	INTTAUA1I8
57	INTTAUA1I9
58	INTTAUA1I10
59	INTTAUA1I11
60	INTTAUA1I12
61	INTTAUA1I13
62	INTTAUA1I14
63	INTTAUA1I15

Table 4.2 DMA Start Sources (2/2)

DTFRn.IFCn[6:0]	Interrupt to start DMA
64	INTTAUA2I2
65	INTTAUA2I3
66	INTTAUA2I4
67	INTTAUA2I5
68	INTTAUA3I0
69	INTTAUA3I1
70	INTTAUA3I2
71	INTTAUA3I3
72	INTTAUA3I4
73	INTTAUA3I5
74	INTTAUA3I6
75	INTTAUA3I7
76	INTTAUA3I8
77	INTTAUA3I9
78	INTTAUA3I10
79	INTTAUA3I11
80	INTTAUA3I12
81	INTTAUA3I13
82	INTTAUA3I14
83	INTTAUA3I15
84	INTTAUJ0I0
85	INTTAUJ0I1
86	INTTAUJ0I2
87	INTTAUJ0I3
88	INTENCA0IOV
89	INTENCA0IUD
90	INTENCA0I0
91	INTENCA0I1
92	INTENCA0IEC
93	INTENCA1OV
94	INTENCA1IUD
95	INTENCA1I0
96	INTENCA1I1
97	INTENCA1IEC
98	INTTAPA0IPEK0
99	INTTAPA0IVLY0
100	INTTAPA2ADOUT0
101	INTTAPA0ADOUT0
102	INTTAPA0ADOUT1
103	INTTAPA1IPEK0
104	INTTAPA1IVLY0
105	INTTAPA3ADOUT0
106	INTTAPA1ADOUT0
107	INTTAPA1ADOUT1
108	INTCSIH0IR
109	INTCSIH0IC
110	INTCSIH1IR
111	INTCSIH1IC
112	INTCSIH2IR
113	INTCSIH2IC
114	INTCSIH3IR
115	INTCSIH3IC
116	INTCSIG0IR
117	INTCSIG0IC
118	INTCSIG1IR
119	INTCSIG1IC
120	INTCSIG2IR
121	INTCSIG2IC
122	INTCSIG3IR
123	INTCSIG3IC
124	INTCSIG4IR
125	INTCSIG4IC
126	INTCSIG5IR
127	INTCSIG5IC

4.3 Function Specifications

This section describes the specifications for the functions that are used by the sample program.

4.3.1 Main Processing (main.c)

[Function Name]	main ()
[Function]	Calls necessary initialization functions before entering an infinite loop.
[Arguments]	None
[Return Value]	None
[Startup Method]	Enters the main function after hardware initialization.
[SFRs Used]	DTS0SR, TAUA0TS
[Calling Function]	None
[Variables]	None
[File name]	main.c
[Notes]	None

4.3.2 Software Initialization Processing (initial.c)

[Function Name]	port_initial()
[Function]	Sets up ports and their mode.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	PFCE0, PFC0, PMC0, PM0, PFCE13, PFC13, PMC13, PM13
[Calling Function]	main()
[Variables]	None
[File Name]	initial.c
[Notes]	None

[Function Name]	cg_initial()
[Function]	Initializes the special clock frequency control register.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	SFRCTL3
[Calling Function]	main()
[Variables]	None
[File Name]	initial.c
[Notes]	None

[Function Name]	hbus_initial()
[Function]	Initializes the AHB bus.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	ETARCFG0, ETARADRS0, ETARMASK0
[Calling Function]	main()
[Variables]	None
[File Name]	initial.c
[Notes]	None

[Function Name]	board_initial()
[Function]	Sets up the initial state of the LEDs.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	P13
[Calling Function]	main()
[Variables]	None
[File Name]	initial.c
[Notes]	None

[Function Name]	ram_initial()
[Function]	Sets up the initial state of the internal RAM.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	None
[Calling Function]	main()
[Variables]	DMA_source, DMA_IO
[File Name]	initial.c
[Notes]	None

4.3.3 DMA Control Processing (dma_control.c)

[Function Name]	dma0_initial()
[Function]	Sets up the operation of the DMA.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	ICDMA0, ICDMACT0, DTS0, DSA0, DDA0, DSC0, DDC0, DTC0, DTCC0, DTCT0, DTRS0
[Calling Function]	main()
[Variables]	None
[File Name]	dma_control.c
[Notes]	None

[Function Name]	dma1_initial()
[Function]	Sets up the operation of the ADC.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	ICDMA1, ICDMACT1, DTS1, DSA1, DDA1, DSC1, DDC1, DTC1, DTCT1, DTRS1, DTFR1
[Calling Function]	main()
[Variables]	None
[File Name]	dma_control.c
[Notes]	None

4.3.4 Interrupt Processing (interrupt.c)

[Function Name]	int_dma1()
[Function]	Processes DMA transfer end interrupt.
[Arguments]	None
[Return Value]	None
[Startup Method]	Request INTDMA1 is present in an unmasked state.
[SFRs Used]	DTS1TC, P13
[Calling Function]	None
[Variables]	None
[File Name]	interrupt.c
[Notes]	None

[Function Name]	int_dmact0()
[Function]	Processes DMA transfer count match interrupt.
[Arguments]	None
[Return Value]	None
[Startup Method]	Request INTDMACT0 is present in an unmasked state.
[SFRs Used]	P13
[Calling Function]	None
[Variables]	None
[File Name]	interrupt.c
[Notes]	None

4.3.5 Timer Control Processing (taua0_control.c)

[Function Name]	taua0_initial()
[Function]	Sets up the inverter function so that INTTAUAOIO becomes the trigger for DMA1.
[Arguments]	None
[Return Value]	None
[Startup Method]	Call
[SFRs Used]	TAUA0TPS, TAUA0BRS, TAUA0CMOR0, TAUA0CMUR0, TAUA0CDR0, TAUA0TOE, TAUA0TOM, TAUA0TOC, TAUA0TOL, TAUA0TDE, TAUA0TDM, TAUA0TDL, TAUA0TRE, TAUA0TRO, TAUA0TRC, TAUA0TME, TAUA0RDE, TAUA0RDS, TAUA0RDM, TAUA0RDC
[Calling Function]	main()
[Variables]	None
[File Name]	taua0_control.c
[Notes]	None

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

All trademarks and registered trademarks are the property of their respective owners.

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jan 30, 2012	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2012 Renesas Electronics Corporation. All rights reserved.

Colophon 1.1