V850E2/ML4

DMA Control

R01AN1223JJ0100
Rev.1.00
Jun. 22, 2012

Abstract

This document describes 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 features of the operation are described below:

Transfer inside of inner RAM.
Transfer between inner RAM and peripheral I/O.
DMAC (Direct Memory Access Controller) transfers by software trigger.
DTFR (DMA trigger factor register) transfers by interrupt signal trigger.

Products

V850E2/ML4

Integrated development environments

CubeSuite+, GHS MULTI V5.1.7D, and IAR for V850 Kickstart V3.80.

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

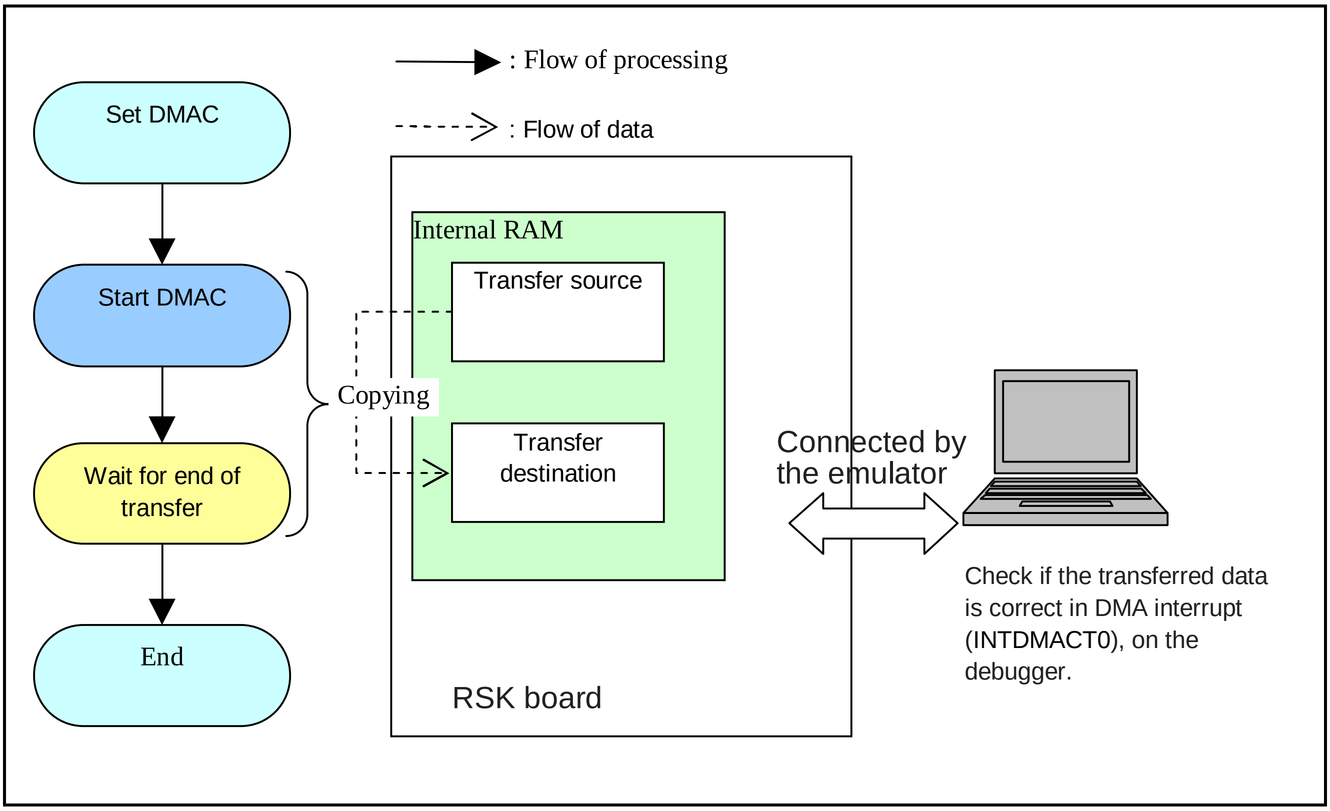
This application note explains examples using DMAC (Direct Memory Access Controller) and DTFR (DMA Trigger Factor Register).

Table 1.1 lists the Peripheral Functions and their Applications and Fugure1.1 shows the Example1: transfer inside of inner RAM by DMAC.

Table 1.1 Peripheral Functions and their Applications

Peripheral Function	Application
Ports(P1_4, P1_5, P4_3, P4_4)	Connected to LEDs, and light on or off LEDs.
Ports(P0_0..P0_15)	Destination of DTFR transfer.

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.



Fugure1.1 Example1: transfer inside of inner RAM by DMAC

[1] Mass production of RSK board will start in August, 2012.

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.

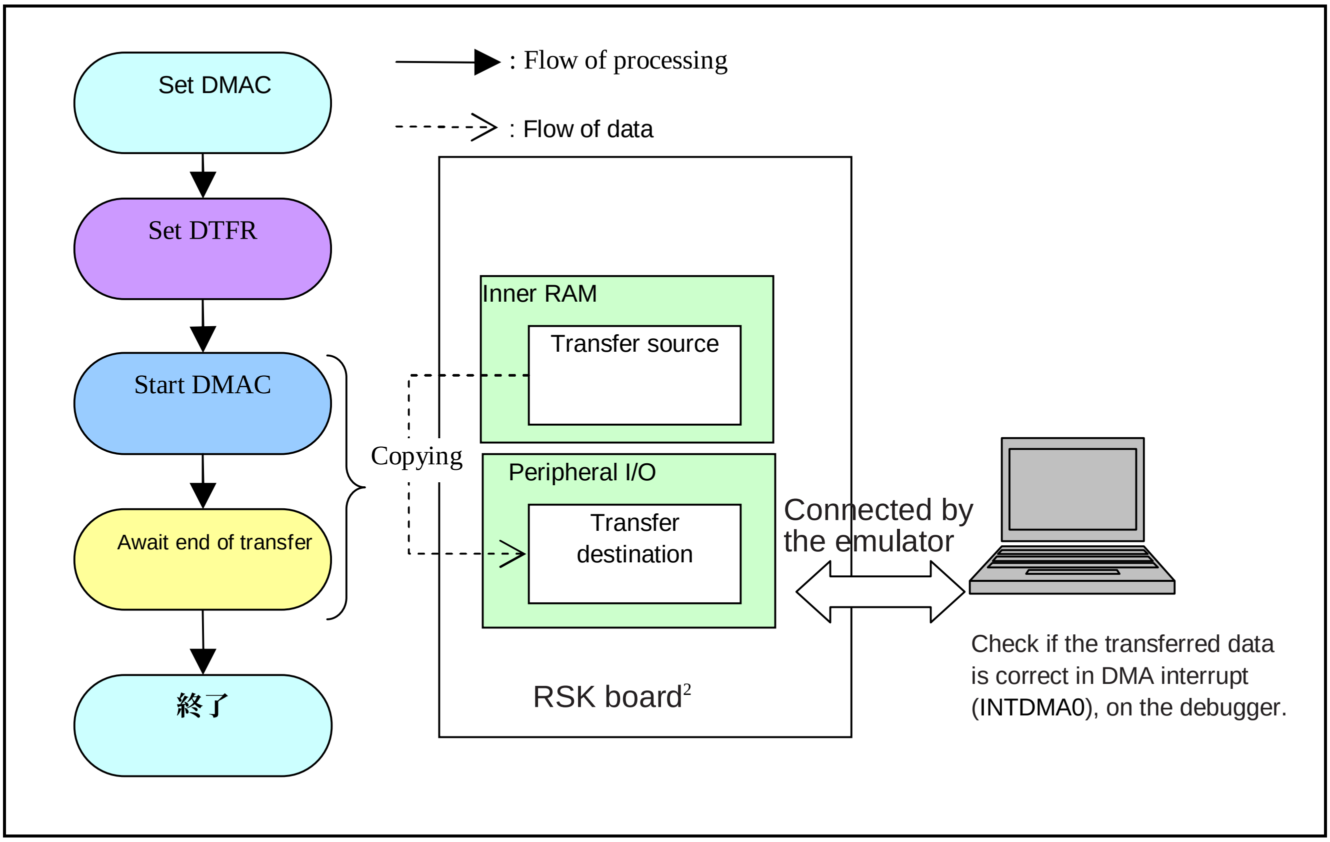


Figure 1.2 Example2: transfer between inner RAM and peripheral I/O by DTFR

[2] Mass production of RSK board will start in August, 2012.

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	V850E2/ML4
Operating frequency	200MHz (PLL multiplies the oscillator input frequency (f_X: 10MHz) by 20.)
Operating voltage	3.3V
Integrated development environment	CubeSuite+ V1.00
	GHS MULTI V5.1.7D
	IAR for V850 Kickstart V3.80.1
C compiler	CX V1.20(CubeSuite+), optimization: default
	C-V850E 5.1.7 RELEASE(GHS MULTI) , optimization: default
	IAR C/C++ Compiler for V850 3.80.1 [Kickstart] (3.80.1.30078), optimization: default
Operating mode	Normal operation mode
Sample code version	V1.00
Board used	RSK board
Device used	E1 emulator or MINICUBE
Tool used	none

2.1 Pin(s) Used

Table 2.2 lists the **Pins Used and Its Function**.

Table 2.2 Pins Used and Its Functions

Pin Name	I/O	Function
PORT P1_4	output	Port mode, output, LED0
PORT P1_5	output	Port mode, output, LED1
PORT P4_3	output	Port mode, output, LED2
PORT P4_4	output	Port mode, output, LED3
PORT P0_0..P0_15	output	The destination of DTFR transfer

3. Software

3.1 Operation Overview

Operation Overview is described in the following figure. The main() function initializes each functions, and waits interrupt. When DMA transfer completion or transfer count match interrupt has occurred, this sample lights on or off LED, or clears transfer completion flag.

Figure3.1 shows the Sequence.

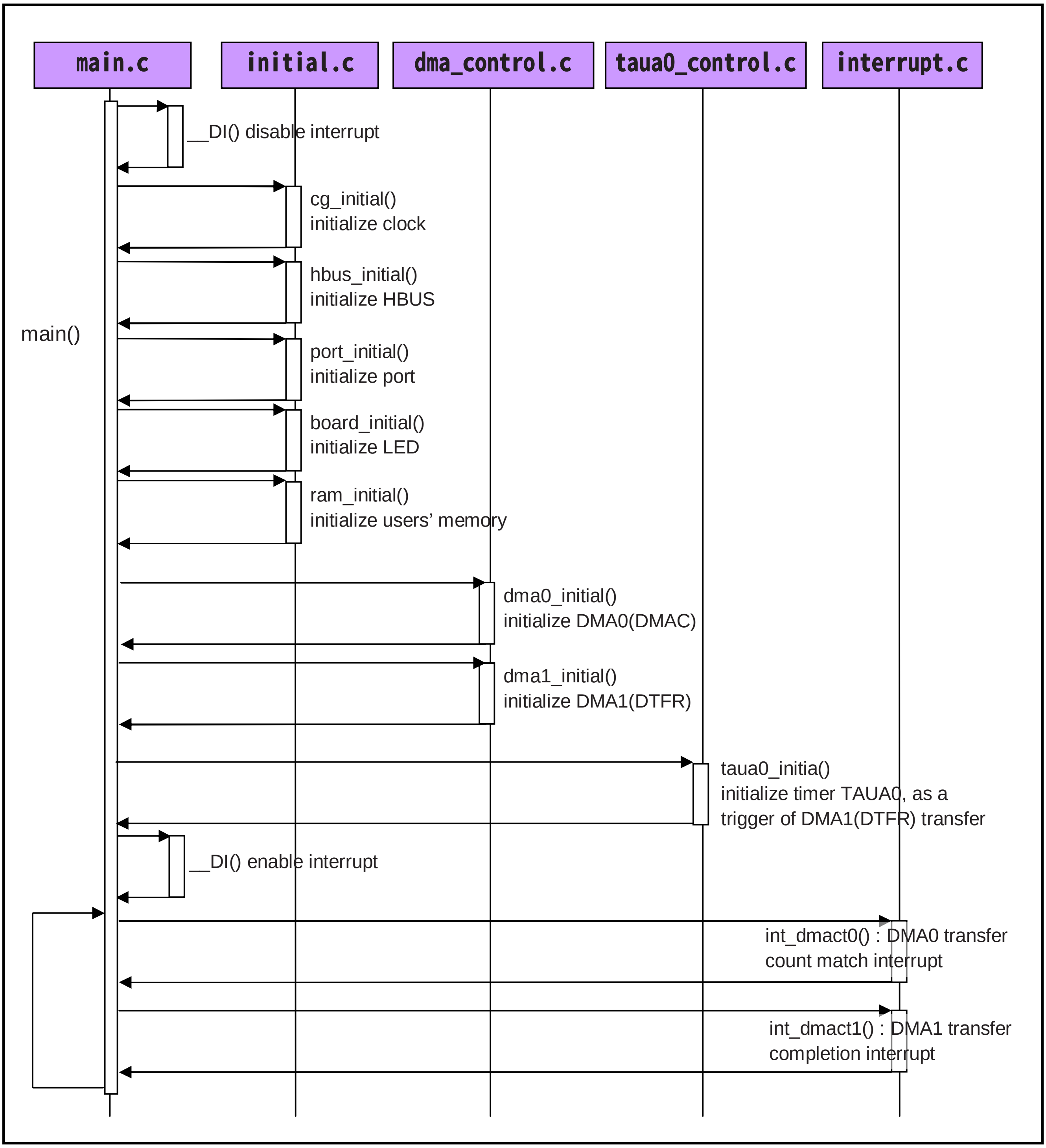


Figure3.1 Sequence diagram

3.2 Required Memory Size

Table 3.1 lists the Required Memory Size. (CubeSuite+, optimization=default)

Table 3.1 Required Memory Size

Memory Used	Size	Remarks
ROM	5100	Shown as ROM area size in map file
RAM	4108	Shown as RAM area size in map file
Maximum user stack usage	4	CubeSuite+ stack estimation tool calculated.
Maximum interrupt stack usage	60	The same as above.

Note: The required memory size varies depending on the C compiler version and its options.

3.3 File Composition

Table 3.2 lists the File(s) Used in the Sample Code. Files not generated by the integrated development environment should not be listed in this table.

Table 3.2 File(s) Used in the Sample Code

File Name	Outline	Remarks
crtE.s	Initialize hardware	Only in the project for CubeSuite+
startup.s		Only in the project for GHS MULTI
V850E2ML4.dir	Linker directive file	Only in the project for CubeSuite+
V850E2_ML4 DMA.ld		Only in the project for GHS MULTI
vector.s	Vector table	Only in the project for GHS MULTI
dma.h	Declare variables and functions.	
df4022_800.h	Declare register macros for V850E2/ML4	Only in the project for GHS MULTI
main.c	Main routine	
initial.c	Initialize software	
dma_control.c	Initialize DMA	
taua0_control.c	Initialize timer TAUA0	
interrupt.c	Interrupt routines	

3.4 Option-Setting Memory

This sample does not specify any option-bytes. Specify them if necessary.

3.5 Function(s)

Table 3.3 lists the Function(s).

Table 3.3 Function(s)

Function Name	Outline
void port_initial(void)	Sets up ports and their mode.
void cg_initial(void)	Initializes the special clock frequency control register.
void hbus_initial(void)	Initializes the AHB bus
void board_initial(void)	Initializes the LEDs
void ram_initial(void)	Sets up the initial state of the user RAM
void dma0_initial(void)	Initializes DMA0(DMAC)
void dma1_initial(void)	Initializes DMA1(DTFR)
void taua0_initial(void)	Initialize timer TAUA0 as a trigger of DTFR
interrupt void int_dmact0(void)	DMA0 transfer count match interrupt.
interrupt void int_dma1(void)	DMA1 transfer completion interrupt.
void main(void)	Calls necessary initialization functions and waits interrupt in infinite loop.

3.6 Function Specification(s)

The following tables list the sample code function specification(s).

main()	
Outline	Main routine
Header	—
Declaration	void main(void)
Description	Calls necessary initialization functions and waits interrupt in infinite loop.
Arguments	— —
Return Value	—

port_initial()	
Outline	Sets up ports and their mode.
Header	csih.h
Declaration	void port_initial (void)
Description	Sets up ports in port-mode, output, for controlling LEDs.
Arguments	none -
Return Value	none

cg_initial()	
Outline	Initialize clock
Header	csih.h
Declaration	void cg_initial(void)
Description	Initializes the special clock frequency control register.
Arguments	none -
Return Value	none

hbus_initial()	
Outline	Initialize H-bus
Header	csih.h
Declaration	void hbus_initial(void)
Description	Initializes AHB-bus
Arguments	none -
Return Value	none

board_initial()	
Outline	Initialize board
Header	csih.h
Declaration	void board_initial(void)
Description	Initialize LED on the board.
Arguments	— —
Return Value	—

ram_initial()		
Outline	Initialize users' memory.	
Header	csih.h	
Declaration	void ram_initial(void)	
Description	Initialize LED on the board.	
Arguments	—	—
Return Value	—	

dma0_initial()		
Outline	Initialize DMA0	
Header	dma.h	
Declaration	void dma0_initial(void)	
Description	Initialize DMA0 as software-triggered DMAC.	
Arguments	—	—
Return Value	—	

dma1_initial()		
Outline	Initialize DMA1	
Header	dma.h	
Declaration	void dma1_initial(void)	
Description	Initialize DMA0 as hardware-triggered DTFR.	
Arguments	—	—
Return Value	—	

taua0_initial ()		
Outline	Initialize timer TAUA0	
Header	dma.h	
Declaration	void taua0_initial(void)	
Description	Initialize TAUA0 as a trigger of DTFR.	
Arguments	—	—
Return Value	—	

int_dmact0()		
Outline	DMA0 transfer count match interrupt	
Header	—	
Declaration	__interrupt void int_dmact0(void)	
Description	DMA0 transfer count match interrupt, inverts LED.	
Arguments	—	—
Return Value	—	

int_dma1()		
Outline	DMA1 transfer completion interrupt	
Header	—	
Declaration	__interrupt void int_dma1(void)	
Description	DMA1 transfer completion interrupt, inverts LED, and clears transfer completion flag.	
Arguments	—	—
Return Value	—	

3.7 Flowchart(s)

3.7.1 Main Processing

Figure3.2 shows the Main Processing.

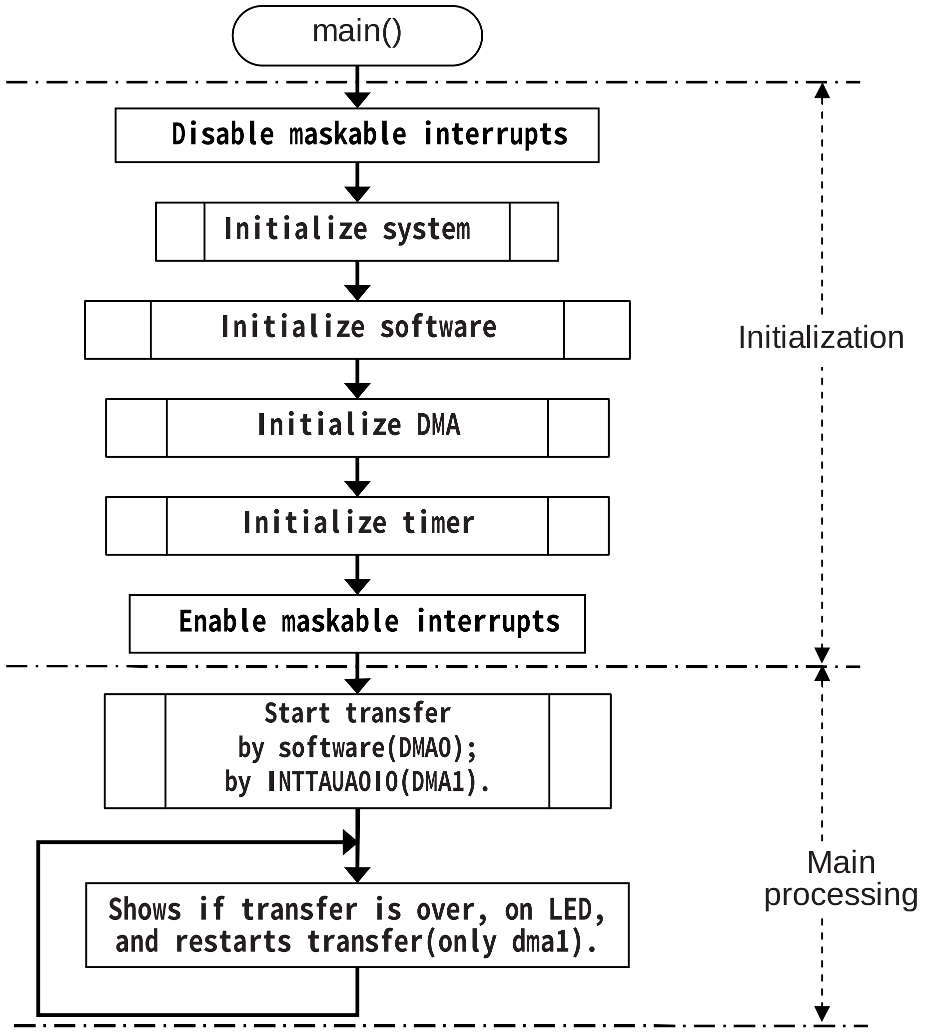


Figure3.2 Main Processing

3.7.2 Initialize DMA

Figure3.3 shows flowchart of Initialize DMA0(DMAC).

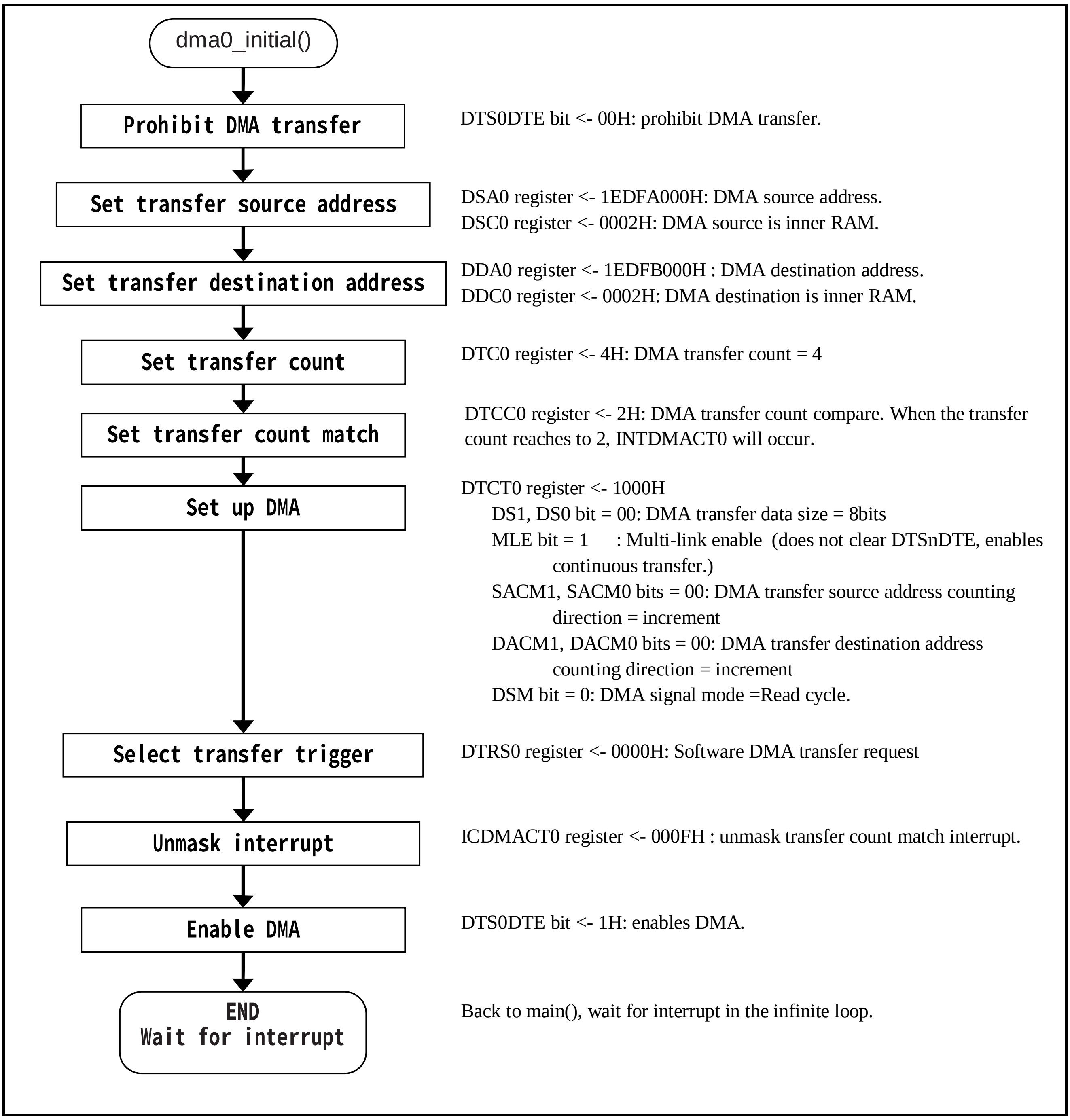
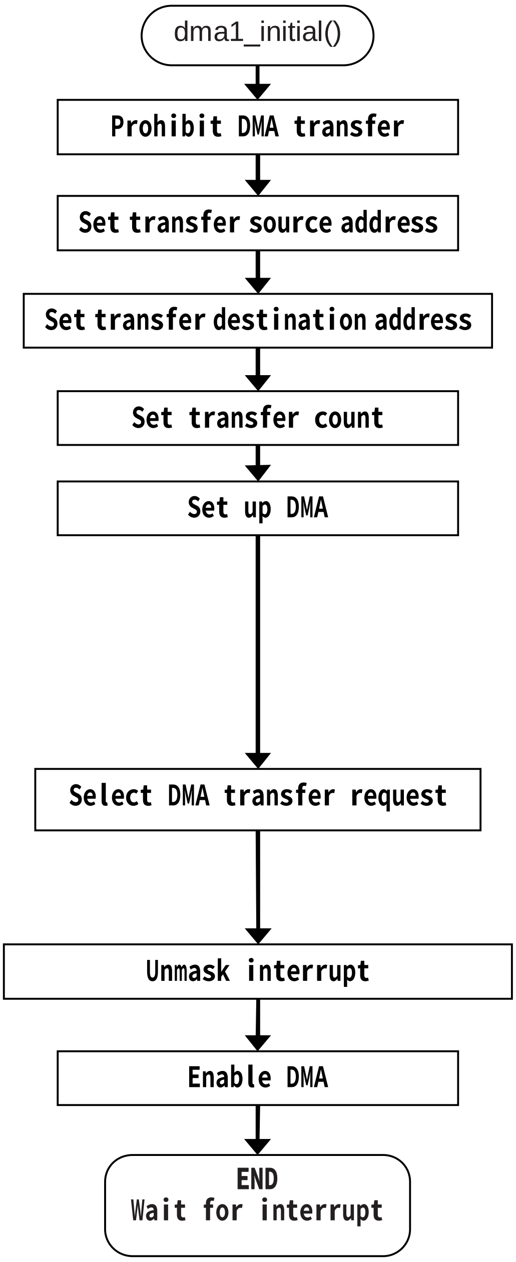


Figure3.3 Initialize DMA0(DMAC)

Figure3.4 shows flowchart of Initialize DMA1(DTFR).



DTS1DTE bit <- 00H: prohibit DMA transfer.

DSA1 register <- 1EDFC000H: DMA source address.
DSC1 register <- 0002H: DMA source is inner RAM.

DDA1 register <- 1FFF8000H: DMA destination address.
DDC1 register <- 0001H: DMA destination is peripheral I/O.

DTC0 register <- 1H: DMA transfer count = 1

DTCT0 register <- 3000H
- DS1, DS0 bit = 01B :DMA transfer data size = 16bits
- MLE bit = 1 : Multi-link enable (does not clear DTSnDTE, enables continuous transfer.)
- SACM1, SACM0 bits = 00: DMA transfer source address counting direction = increment
- DACM1, DACM0 bits = 00: DMA transfer destination address counting direction = increment
- DSM bit = 0: DMA signal mode =Read cycle.

DTRS1 register <- 0001H: hardware DMA transfer request
DTFR1 register <- 8020H:
- REQEN bit = 1: enables DMA source selector.
- IFCn6- IFCn0 bits = 20H：DMA start source = INTTAUA0I0.

ICDMA1 register <- 000FH : unmask transfer completion interrupt.

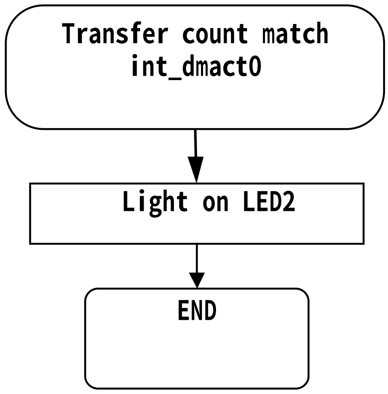
DTS0DTE bit <- 1H: enables DMA.

Back to main(), wait for interrupt in the infinite loop.

Figure3.4 Initialize DMA1(DTFR)

3.7.3 Interrupt Processing

Figure3.5 shows the flowchart of Transfer count match interrupt.

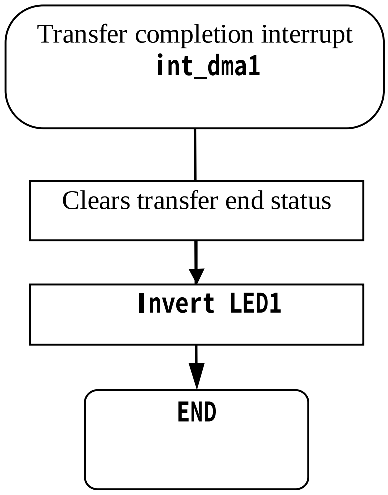


An interrupt INTDMACT0 occurs when the transferred count matches to the number specified in DTCC0 register.

Light on LED2, shows the transfer is over.

Figure3.5 Transfer count match interrupt

Figure3.6 shows the flowchart of Transfer completion interrupt.



An interrupt INTDMA1 occurs when the transfer of DMA1 has completed.

DTS1TC bit =0: clears transfer end status. Wait for the next transfer request (INTTAUA0I0).

Invert LED1, shows the transfer is over.

Figure3.6 Transfer completion interrupt

4. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

5. Reference Documents

User's Manual: Hardware

V850E2/ML4 User's Manual: Hardware (R01UH0262EJ)

The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website

http://www.renesas.com

Inquiries

http://www.renesas.com/contact/

REVISION HISTORY	V850E2/ML4 Application Note　DMA Controller

Rev.	Date	Description	
		Page	Summary
1.00	Jun. 22, 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.

Copying

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

Colophon 2.0