To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

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.

32192/32195/32196 Group

Application of the Direct RAM Interface (Special Mode)

1. Overview

This documentation presents examples of sample programs which use the Direct RAM Interface (DRI) module incorporated in the 32192/32195/32196 Group Microcomputer.

This sample task demonstrates a capturing video data from the camera module.

2. Introduction

The sample task described in this documentation applies to the use of the following microcomputers, under the respective conditions.

- Microcomputer: 32192 Group (M32192F8VFP, M32192F8UFP, and M32192F8TFP)
  32195 Group (M32195F4VFP, M32195F4UFP, and M32195F4TFP)
  32196 Group (M32196F8VFP, M32196F8UFP, and M32196F8TFP)
- Operating Frequency: 128 to 160 MHz (The sample programs are compiled assuming a frequency of 160 MHz.)
- Operation Board: 32192 μT-Engine R0P3219TR001MRK

3. Explanation of the Technology Applied

3.1 Outline of the Direct RAM Interface (DRI)

The Direct RAM Interface (DRI) is a parallel interface used to take in parallel data into the internal RAM as it is input to the microcomputer synchronously with the clock. Since a dedicated bus provided separately from the M32R-FPU is used to write data from the DRI to the internal RAM, data can be taken in without having to stop operation of the M32R-FPU. Furthermore, a selective data capture function is supported that makes use of the internal event counter of the DRI.

Please refer to the 32192/32196 Group hardware manual for details of the DRI functions.

4. DRI Sample Program

4.1 Outline of the Sample Program

In this sample task, an image (240 × 320 dots) from the area shown in figure 4.1.1 is captured from one frame of camera video image (640 × 480 dots).

The capture starting point X, Y is defined by the program.

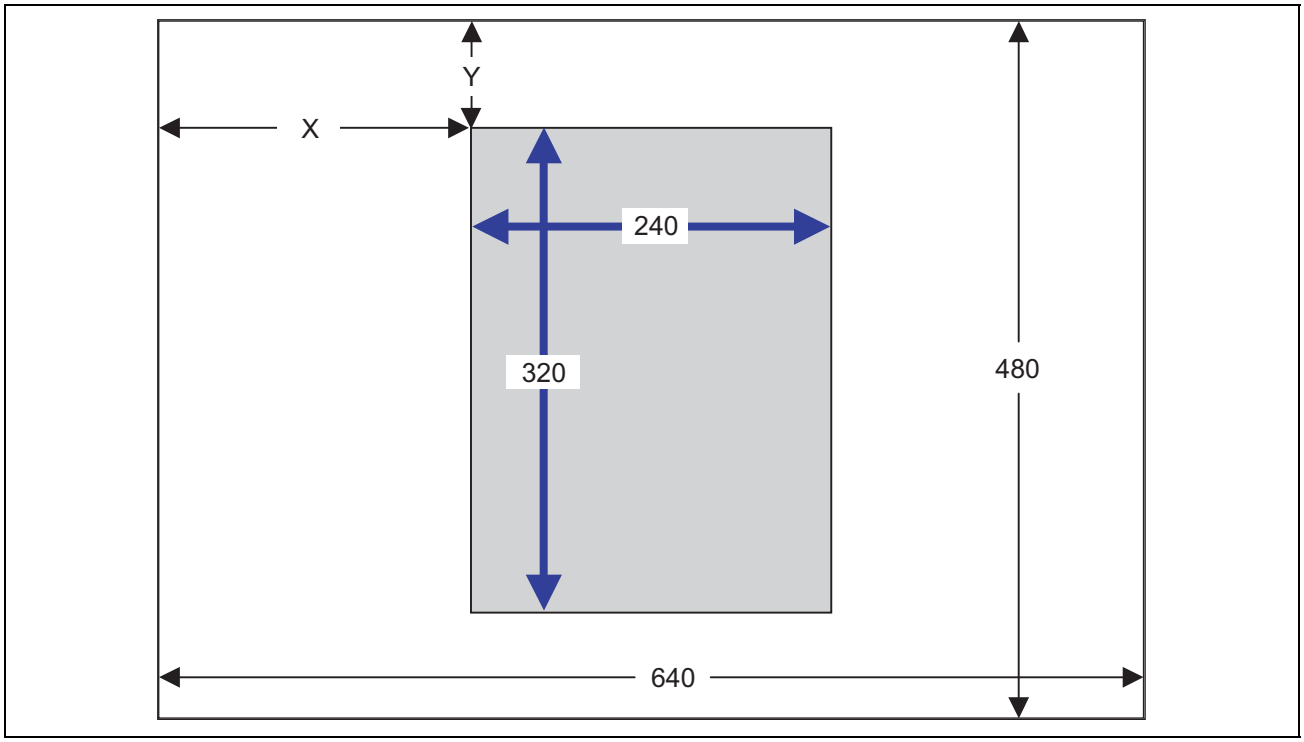


Figure 4.1.1 The Image Data Capture Range

4.2 Connecting the Camera Module

4.2.1 Signal Output from the Camera Module

The waveform output from the camera module is shown below.

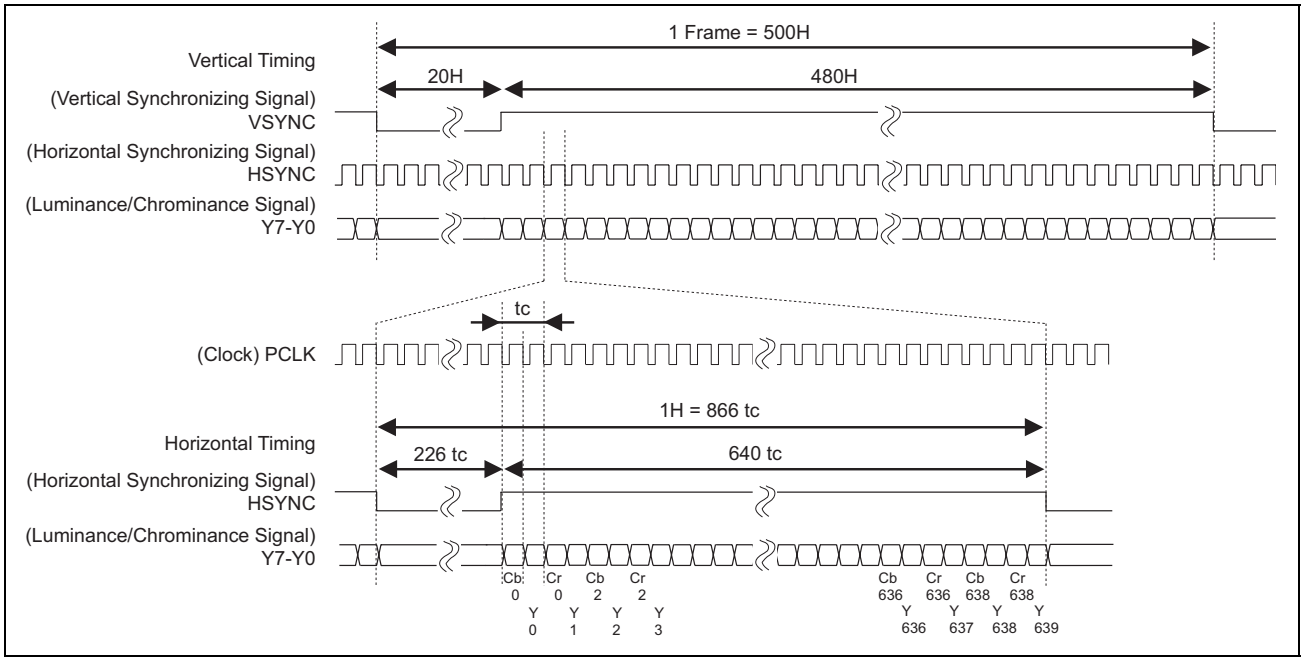


Figure 4.2.1 Camera Module Signal Diagram

4.2.2 Camera Module Connection Diagram

The camera module image data, shown in figure 4.2.1, is output synchronized to the vertical synchronizing signal and horizontal synchronizing signal. The camera module and the microcomputer are connected as shown in figure 4.2.2.

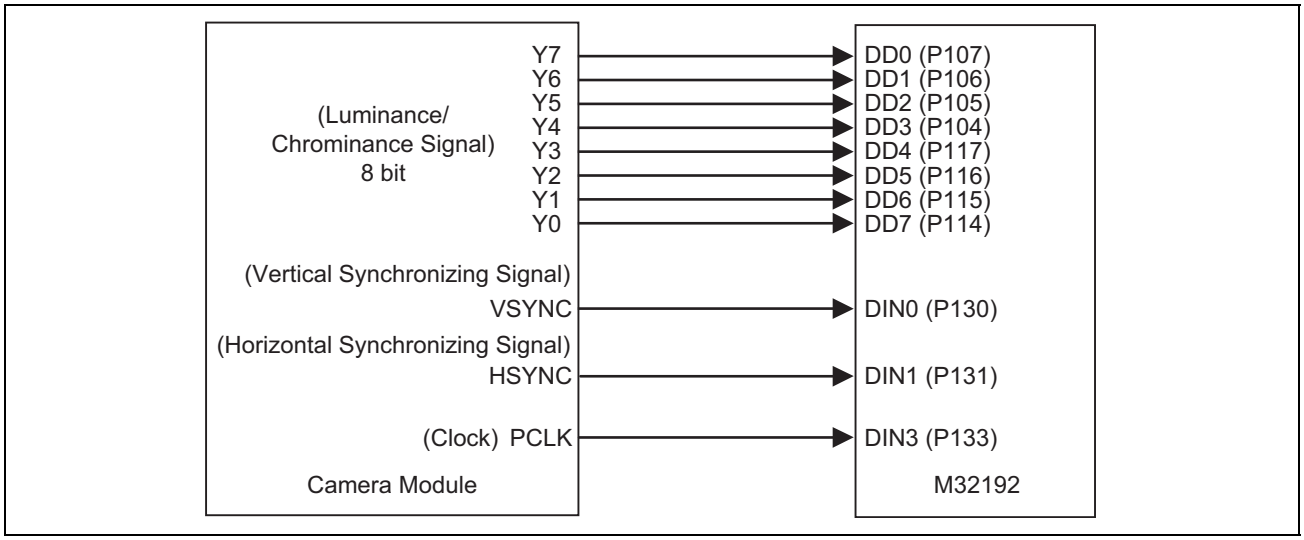


Figure 4.2.2 Camera Module Connections

4.3 Operation of the Sample Program

In the sample task, in order to capture data with high speed DRI special mode is used. In this case the minimum period of data capture is 50ns (when data width is 8 bit). If DRI special mode is not used, it is 87.5ns (when data width is 8 bit). Furthermore, using the selective data capturing function in DEC counter, the range of data specified by the program is captured from one frame of video image.
The operation timing of this sample task is shown below.

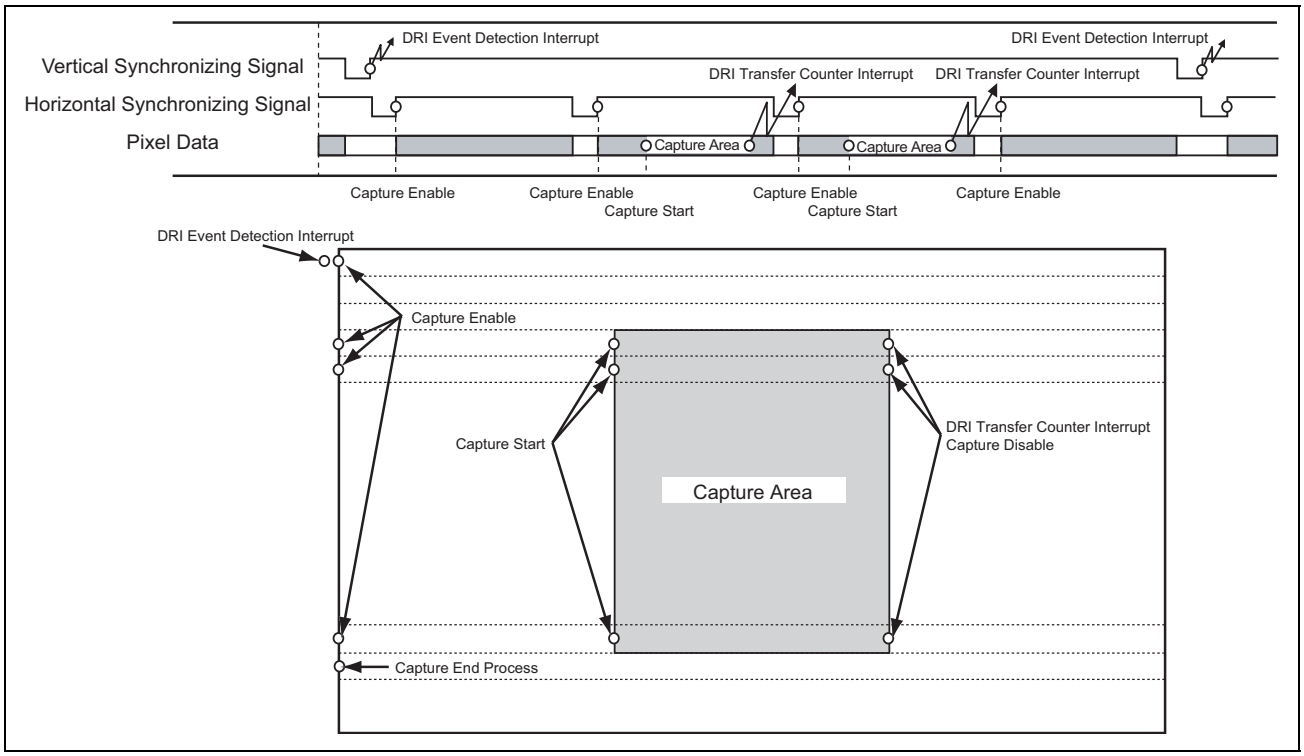


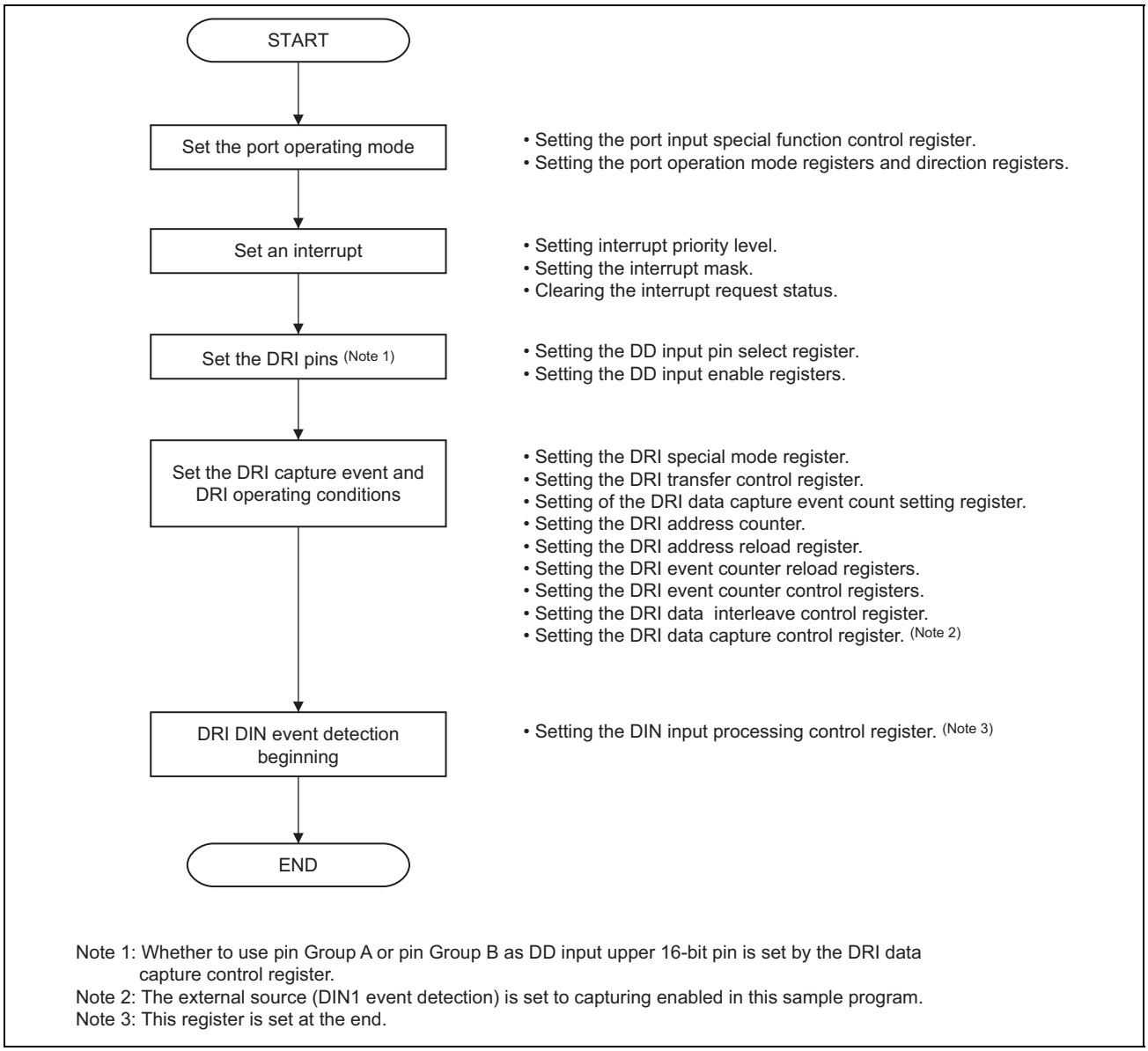
Figure 4.3.1 Operation Timing

(1) The rising DIN0 generates DRI event detection interrupt and the DEC0 counter is enabled.
Within the DRI event detection interrupt handler the DRI transfer counter interrupt is enabled.
(2) With rising DIN1, DEC1 and DEC2 counters are enabled.
The DEC3 counter is enabled by the underflow of the DEC2 counter.
(3) The DEC0 counter counts the number of DIN1 inputs, an underflow occurs when the capture area line is reached.
DEC1 counter counts the number of pixels in the range of the capture area within one line, under flows at the starting point of the capture area. The DEC3 counter is initialized with underflow value.
(4) The DEC0 counter, DEC1 counter, and DEC3 counter are set to selective data capturing enabled. Therefore, after the underflow of DEC1 counter a capture event is generated and the DEC2 counter starts counting and then DRI transfer starts.
(5) DEC2 counter counts the number of pixels in the remaining areas within the line.
At the end of the capture area, the DEC2 counter and DRI transfer counter underflow, and DRI transfer counter interrupt occurs.
(6) The DEC3 counts the number of remaining pixels in one line and underflows at the end of the line.
(7) After capturing data in the capturing area, DRI data capture is disabled. Then, capturing processing ends.

For the timings of DIN0, DIN1 and the individual event counters, please refer figure 4.7.2, DRI Data Capture Timing.

4.4 Processing Procedure

Figure 4.4.1 shows the processing flow for setting the DRI.



Note 1: Whether to use pin Group A or pin Group B as DD input upper 16-bit pin is set by the DRI data capture control register.
Note 2: The external source (DIN1 event detection) is set to capturing enabled in this sample program.
Note 3: This register is set at the end.

Figure 4.4.1 DRI Initialization Flow

4.5 Interpretation of the Sample Program

Note The registers used are indicated as (register name: bit name).

4.5.1 Main Function (main ())

(1) Calling the function for initializing the DRI.

(2) Setting the video image data capture starting point.
- Setting value X to variable "usHsyncCtrlCnt" and value Y to variable "usVsyncCtrlCnt" in figure 4.1.1.

(3) Calling the function for DRI frame data input processing.

4.5.2 DRI Initialization Function (dri_init ())

(1) Calling the interrupt disable function.

(2) Input port initial setting.
- Set the port input enable bit of the port input special function control register to enable inputs. (PICNT: PIEN0)
- Set P10, P11, P13 direction registers to input mode. (P10DIR, P11DIR, P13DIR)
- Set P10, P11, P13 operation mode registers to DRI pin function. (P10MOD, P11MOD, P13MOD)

(3) Setting the DRI interrupt function register.
- Set the DIN interrupt request status register to interrupts not requested. (DRIDINIST)
- Set DRI transfer interrupt request status register to interrupts not requested. (DRITRMIST)
- Set DRI transfer interrupt request enable register to interrupt request disabled. (DRITRMIEN)
- Set DRI interrupt control register to interrupt priority level 2. (IDRIEVCR)
- Set DRI transfer interrupt control register to interrupt priority level 2. (IDRITRCR)
- Set DIN interrupt request enable register to DIN0 interrupt request enabled. (DRIDINIEN)

(4) Setting DRI terminals.
- Select DD0 to DD3 input terminal when using pin group A. (DDSEL: DD03SEL)
- Set DD0 to DD7 to input enabled. (DRIDDEN0)
- Set DD8 to DD15 to input disabled. (DRIDDEN1)
- Set DD16 to DD23to input disabled. (DRIDDEN2)
- Set DD24 to DD31to input disabled. (DRIDDEN3)

(5) Calling the interrupt enable function.

4.5.3 DRI Frame Data Input Processing Function (AR_GetPict ())

(1) Calling the function to DRI capture start setting.

(2) Loops until the capture of the set number of lines is complete.

4.5.4 DRI Capture Start Setting Function (dri_start ())

(1) Setting operating conditions.

- Set the number of lines captured to variable "usHsyncCnt".
- Set DIN interrupt request enable register to DIN0 interrupt request enabled. (DRIDINIEN: DIN0IEN)
- Set DRI special mode register (Note 1) as below. (DRISPMOD: SPSSL, SPISL, SPMEN)
  DIN3 sampled on the rising edge.
  Special mode control unit initialization DIN1 level is a low level.
  Special mode enable bit to special mode ON.
- Set DRI transfer control register to operation enabled. (DRITRMCNT: DRST)
- Set the number of captured pixels in the DRI data capture event count register. (DRIDCAPNUM)
- Set transfer destination address to DRI address counter 0. (DRIADR0CT)
- Set transfer destination address to DRI address reload register 0. (DRIADR0RLD)

(2) Setting the event counter control registers

- Set the initial values to the DEC0 and DEC1 counters. (DEC0CT, DEC1CT)
- Initialize DEC3 counter in the underflow state. (DEC3CT)
- Set the next initial value to the DRI event counter reload register as follows.
  Unnecessary number of lines from the starting point of the vertical synchronizing signal to the DEC0 reload register. (DEC0RLD)
  Unnecessary number of pixels from the starting point of the horizontal synchronizing signal to the DEC1 reload register. (DEC1RLD)
  Number of captured pixels in one line to the DEC2 reload register. (DEC2RLD)
  Unnecessary number of captured pixels after capturing one line in the DEC3 reload register. (DEC3RLD)
- Set DEC0 count enable source to DIN0, and counted events to DIN1. (DEC0CNT)
- Set DEC1 count enable source to DIN1, and counted events to DIN3. (DEC1CNT)
- Set DEC2 count enable source to DIN1, and counted events for capture. (DEC2CNT)
- Set DEC3 count enable source to DEC2, and counted events to DIN3. (DEC3CNT)

(3) Setting the capture condition

- Set event counters 0, 1, and 3 of the DRI data interleave control register to enable selective data capturing function. (DRIDSELCNT: DSD0, DSD1, DSD3)
- Set DRI data capture control register as below. (DRIDCAPCNT: DCPEN, DEXSL, DDSSL, DWDSL, DCPSL, DDSL, DWEPR, DTMSL)
  Data capture disabled.
  Capture enabled external source is DIN1 event detection.
  Capture external control disable source to disable source unselected.
  Input data bus width to 16-bit. (Note 2)
  Capture event to DIN3 event. (Note 2)
  DD input upper 16-bit pins to pin group A selected.
  Capture control WR protection bit to WR enabled.
  Capture timing to default.
- Set how to detect the event of the input signal by the DIN input processing control registers. (DINCNT: DIN0ED, DIN1ED, DIN2ED, DIN3ED)
  DIN0, DIN1 to the rising edge.
  DIN2 to input invalid.
  DIN3 falling edge. (Note 2)

Notes 1: For details of the special mode, refer to section 14.2.4, DRI Special Mode Control Register, of the 32192/32196 Group Hardware Manual.

Notes 2: Limitations or restrictions in using the special mode.

4.5.5 DRI Event Detect Interrupt Process Function (dri_vsync_int ())

When the vertical synchronizing signal (DIN0) is input, an interrupt is generated and this function is called.

(1) Clearing the DIN interrupt request status register. (DRIDINIST: DIN0IS, DIN1IS, DIN2IS)
- Clears the DIN0 to DIN2 interrupt request status bits.

(2) DRI transfer counter interrupt request enable bit of the DRI transfer interrupt request enable register is set to interrupt request enabled. (DRITRMIEN: DTRFIEN)

4.5.6 DRI Transfer Interrupt Processing Function (dri_hsync_int ())

When one line of data transmission is complete (DRI transfer counter is in the underflow state) an interrupt is generated and this function is called.

(1) Clearing the DRI transfer interrupt request status register. (DRITRMIST)

(2) Decrement the remaining number of lines for capturing data.

(3) When the remaining number of lines for capturing data becomes “0” (capturing data in the specified range is complete) the capturing operation ends.
- Set the event detection control bits of the DIN input processing control register to disabled. (DINCNT)
- Set the DRI data capture control register to disable capturing of data. (DRIDCAPCNT)
- Set the DRI transfer interrupt request enable register to disable interrupt requests. (DRITRMIEN)
- Set the DIN interrupt request enable register to disable interrupt requests. (DRIDINIEN)

4.5.7 Function for Calculating the Number of Lines for which Data Capturing is Complete (chk_dri_h ())

(1) Calculating the number of lines for which data capturing is complete.

4.5.8 Startup Routine (startup.ms)

(1) Setting the interrupt.
- Set the start address of the DRI transfer interrupt processing function (dri_hsync_int ()) to the ICU vector table interrupt source and the DRI transfer interrupt address (H'0000 0114), and set the start address of the DRI event detection interrupt processing function (dri_vsync_int ()) to DRI event detection interrupt address (H'0000 011C).

4.6 Sample Programs

Sample programs for the application of Direct RAM Interface (in the special mode) is shown below.
Note that the sample programs below require the SFR definition file.
The latest SFR definition file can be downloaded from Renesas Technology website.
When using the SFR definitions file, adjust the path setting to match the operating computer environment.

4.6.1 dri.c

```
/*""FILE COMMENT""*****************************************************
 *      M32R C Programming              Rev. 1.01
 *              < Sample Program for 32192 >
 *              < Direct RAM Interface (DRI) >
 *
 *      Copyright (c) 2005 Renesas Technology Corporation
 *                         All Rights Reserved
 ***********************************************************************/

/************************************************************************/
/*              Include file                                            */
/************************************************************************/

#include                "..\inc\sfr32192_pragma.h"

/************************************************************************/
/*              Function prototype declaration                          */
/************************************************************************/

extern  void            DisInt( void );                 /* Interrupt disable function */
extern  void            EnInt( void );                  /* Interrupt enable function */

        void            main(void);
        void            dri_init(void);
        void            AR_GetPict(ULONG *dst);
        void            dri_start(ULONG *dst);
        void            dri_vsync_int(void);
        void            dri_hsync_int(void);
        USHORT          chk_dri_h(void);

/************************************************************************/
/*              Externally referenced variable                          */
/************************************************************************/


/************************************************************************/
/*              Define macro                                            */
/************************************************************************/

#define                 SENSOR_WIDTH    640ul           /* Number of effective pixels X         */
#define                 SENSOR_HEIGHT   480u            /* Number of effective pixels Y         */
#define                 PICT_WIDTH      240ul           /* Number of taking pixels X            */
#define                 PICT_HEIGHT     320ul           /* Number of taking pixels Y            */
#define                 VDEL_ENABLE     0x80u           /* DEC0 Data Interleave                 */
#define                 HDEL_ENABLE     0x40u           /* DEC1 Data Interleave                 */
#define                 DEL_ENABLE      0x10u           /* DEC3 Data Interleave                 */
#define                 VSYNC_CTRLSET   0x44u           /* DEC0 Setting                         */
                                                        /* 0100 0100B */
                                                        /* |||| |||+--- Single-shot mode        */
                                                        /* |||| ||+---- don't care              */
                                                        /* |||| ++----- DIN1 event detection    */
                                                        /* |+++-------- DIN0 event detection    */
                                                        /* +------------ Disable count          */
#define                 HSYNC_CTRLSET   0x58u           /* DEC1 Setting                         */
                                                        /* 0101 1000B */
                                                        /* |||| |||+--- Single-shot mode        */
                                                        /* |||| ||+---- don't care              */
                                                        /* |||| ++----- DIN3 event detection    */
                                                        /* |+++-------- DIN1 event detection    */
                                                        /* +------------ Disable count          */
#define                 HDATA_CTRLSET   0x5Cu           /* DEC2 Setting                         */
                                                        /* 0101 1100B */
                                                        /* |||| |||+--- Single-shot mode        */
                                                        /* |||| ||+---- don't care              */
                                                        /* |||| ++----- Capture event           */
                                                        /* |+++-------- DIN1 event detection    */
                                                        /* +------------ Disable count          */
```

```
#define                 DELDATA_CTRLSET 0x64u           /* DEC3 Setting                         */
                                                        /* 0110 0100B */
                                                        /* |||| |||+--- Single-shot mode       */
                                                        /* |||| ||+---- don't care             */
                                                        /* |||| ++------ DIN3 event detection   */
                                                        /* |+++-------- DEC2 event detection   */
                                                        /* +----------- Disable count          */
#define                 GET_WIDTH       PICT_WIDTH
#define                 GET_TRANS       GET_WIDTH/2ul

/************************************************************************/
/*              Global variable                                         */
/************************************************************************/
        USHORT          usHsyncCnt;                     /* VSYNC Interrupt counter              */
        USHORT          usHsyncCtrlCnt;                 /* Hsync counter                        */
        USHORT          usVsyncCtrlCnt;                 /* Vsync counter                        */
        USHORT          usDec1CtrlCnt;                  /* Hsync counter                        */
        USHORT          usDec0CtrlCnt;                  /* Vsync counter                        */
        ULONG           arpict[(240*320)/2];            /* data buffer                          */


/*""FUNC COMMENT""*************************************************************
 * Function name: main()
 *------------------------------------------------------------------------------
 * Description  :
 *------------------------------------------------------------------------------
 * Argument     : -
 *------------------------------------------------------------------------------
 * Returns      : -
 *------------------------------------------------------------------------------
 * Notes        : -
 *""FUNC COMMENT END""*********************************************************/
void main(void)
{
        dri_init();                                     /* DRI Initialization */

        usVsyncCtrlCnt = 80;                            /* Vertical Unnecessary part */
        usHsyncCtrlCnt = 50;                            /* Horizontal Unnecessary part */

        while(1) {

                AR_GetPict( arpict );                   /* DRI start */

        }
}

/*""FUNC COMMENT""*************************************************************
 * Function name: dri_init()
 *------------------------------------------------------------------------------
 * Description  : DRI Initialization
 *------------------------------------------------------------------------------
 * Argument     : -
 *------------------------------------------------------------------------------
 * Returns      : -
 *------------------------------------------------------------------------------
 * Notes        : -
 *""FUNC COMMENT END""*********************************************************/
void     dri_init(void)
{
        DisInt();                                       /* Disable interrupt */

        PICNT = 0x01;                                   /* Enable port input */

        P10DIR &= 0xf0u;                                /* Input data (must be set prior to mode) */
        P11DIR = 0x00;                                  /* Input data (must be set prior to mode) */
        P13DIR &= 0x2fu;                                /* Input data (must be set prior to mode) */
        P10MOD &= 0xf0u;                                /* DD0-DD3 select */
        P11MOD = 0x00;                                  /* DD4-DD7 select */
        P13MOD |= 0xd0u;                                /* DIN0,DIN1,DIN3 select */

        DRIDINIST = 0;                                  /* DIN Interrupt not requested */
        DRITRMIST = 0;                                  /* TRM Interrupt not requested */
        DRITRMIEN = 0x00;                               /* Disable TRM interrupt        */

        IDRIEVCR = 0x02;                                /* Enable Event Detection Interrupt */
        IDRITRCR = 0x02;                                /* Enable Transfer Interrupt */
        DRIDINIEN = 0x80;                               /* DIN0(Vsync) Enable interrupt request */
```

```
145
146         DDSEL = 0x01;                                   /* input pin select P10 */
147         DRIDDEN0 = 0xff;                                /* DD0-DD7 Enable input */
148         DRIDDEN1 = 0;                                   /* DD8-DD15 Disable input */
149         DRIDDEN2 = 0;                                   /* DD16-DD23 Disable input */
150         DRIDDEN3 = 0;                                   /* DD24-DD31 Disable input */
151
152         EnInt();                                        /* Enable interrupt */
153         return ;
154 }
155
156 /*""FUNC COMMENT""*****************************************************
157  * Function name: AR_GetPict()
158  *----------------------------------------------------------------------
159  * Description  : It stores in the domain to which dst shows the data picturized at the end.
160  *----------------------------------------------------------------------
161  * Argument     : ULONG *dst    data save area
162  *----------------------------------------------------------------------
163  * Returns      : -
164  *----------------------------------------------------------------------
165  * Notes        : The area in 640×480×2 bytes is necessary for the area that dst shows.
166  *""FUNC COMMENT END""*************************************************/
167 void AR_GetPict(ULONG *dst)
168 {
169         USHORT tv_length;
170
171         dri_start(dst);
172         do{
173                 tv_length = chk_dri_h();
174         }while(tv_length < PICT_HEIGHT);
175
176         return;
177 }
178
179 /*""FUNC COMMENT""*****************************************************
180  * Function name: dri_start()
181  *----------------------------------------------------------------------
182  * Description  : DRI data taking beginning
183  *----------------------------------------------------------------------
184  * Argument     : ULONG *dst    data save area
185  *----------------------------------------------------------------------
186  * Returns      : -
187  *----------------------------------------------------------------------
188  * Notes        : -
189  *""FUNC COMMENT END""*************************************************/
190 void    dri_start(ULONG  *dst)
191 {
192         ULONG uladdr = (ULONG)dst & 0x0003fffcul;
193
194         usHsyncCnt = PICT_HEIGHT;                       /* QVGA Vertical */
195
196         DRIDINIEN = 0x80;                               /* DIN0(Vsync) Enable interrupt request */
197
198         DRISPMOD = 0x00u | 0x10u;                       /* Special mode on */
199
200         DRITRMCNT = 0x80u | 0x00u;                      /* Enable operation      */
201
202         DRIDCAPNUM = GET_WIDTH;                         /* QVGA Horizontal */
203
204         DRIADR0CT = uladdr;
205         DRIADR0RLD = uladdr;                            /* DRI Destination address */
206
207         usDec0CtrlCnt = usVsyncCtrlCnt;
208         usDec1CtrlCnt = usHsyncCtrlCnt;
209         DEC0CT = usDec0CtrlCnt;
210         DEC1CT = usDec1CtrlCnt;
211         DEC3CT = 0xffff;
212
213         DEC0RLD = usDec0CtrlCnt;
214         DEC1RLD = usDec1CtrlCnt;
215         DEC2RLD = PICT_WIDTH - 1ul;                     /* Number of effective data */
216         DEC3RLD = (SENSOR_WIDTH - PICT_WIDTH) - usDec1CtrlCnt - 1ul;
217
218         DEC0CNT = VSYNC_CTRLSET;
219         DEC1CNT = HSYNC_CTRLSET;
220         DEC2CNT = HDATA_CTRLSET;
221         DEC3CNT = DELDATA_CTRLSET;
```

```
222
223         DRIDSELCNT = HDEL_ENABLE | VDEL_ENABLE | DEL_ENABLE;    /* DEC0,1,3 Interleave Enable */
224
225         DRIDCAPCNT = ( 0x5000u | 0x0140u );             /* HSYNC Enable capturing data */
226
227         DINCNT = 0x5200;                                /* select Event Detection */
228
229         return;
230 }
231
232 /*""FUNC COMMENT""*************************************************************
233  * Function name: dri_vsync_int()
234  *------------------------------------------------------------------------------
235  * Description  : VSYNC Interrupt
236  *------------------------------------------------------------------------------
237  * Argument     : -
238  *------------------------------------------------------------------------------
239  * Returns      : -
240  *------------------------------------------------------------------------------
241  * Notes        : -
242  *""FUNC COMMENT END""*********************************************************/
243 void dri_vsync_int(void)
244 {
245         DRIDINIST = 0x1f;                               /* DIN Interrupt not requested */
246         DRITRMIEN |= 0x08u;                             /* DRI transfer counter interrupt request enable */
247
248         return;
249 }
250
251 /*""FUNC COMMENT""*************************************************************
252  * Function name: dri_hsync_int()
253  *------------------------------------------------------------------------------
254  * Description  : HSYNC Interrupt
255  *------------------------------------------------------------------------------
256  * Argument     : -
257  *------------------------------------------------------------------------------
258  * Returns      : -
259  *------------------------------------------------------------------------------
260  * Notes        : -
261  *""FUNC COMMENT END""*********************************************************/
262 void dri_hsync_int(void){
263         DRITRMIST = 0;                                  /* Interrupt request clear*/
264         usHsyncCnt--;
265         if( usHsyncCnt == 0 ) {
266                 DINCNT &= ~0x5200u;                     /* Input has no effect */
267                 DRIDCAPCNT = 0x0000;                    /* Disable capturing data */
268                 DRITRMIEN = 0;
269                 DRIDINIEN = 0;
270         }
271         return;
272 }
273
274 /*""FUNC COMMENT""*************************************************************
275  * Function name: chk_dri_h()
276  *------------------------------------------------------------------------------
277  * Description  : The number of remainder reception lines is returned.
278  *------------------------------------------------------------------------------
279  * Argument     : -
280  *------------------------------------------------------------------------------
281  * Returns      : line
282  *------------------------------------------------------------------------------
283  * Notes        : -
284  *""FUNC COMMENT END""*********************************************************/
285 USHORT chk_dri_h(void){
286
287         USHORT retvalue = PICT_HEIGHT - usHsyncCnt;     /* Number of remainder lines */
288         return retvalue;
289 }
```

4.6.2 startup.ms (an extract)

(Omitted)

```
;****************************************
; ICU Vector Table
;****************************************
;
        .SECTION        ICUVECT, DATA, ALIGN=4
;
        .IMPORT         $dri_vsync_int, $dri_hsync_int
;
vectbl:
        .DATA.W         EIT_reset                   ; H'0000 0094   MJT Input Interrupt 4:TIN3-TIN6
        .DATA.W         EIT_reset                   ; H'0000 0098   MJT Input Interrupt 3:TIN20-TIN27
        .DATA.W         EIT_reset                   ; H'0000 009C   MJT Input Interrupt 2:TIN16-TIN19
        .DATA.W         EIT_reset                   ; H'0000 00A0   MJT Input Interrupt 1:TIN0
        .DATA.W         EIT_reset                   ; H'0000 00A4   MJT Input Interrupt 0:TIN7-TIN10
        .DATA.W         EIT_reset                   ; H'0000 00A8   MJT Output Interrupt 7:TMS0,TMS1
        .DATA.W         EIT_reset                   ; H'0000 00AC   MJT Output Interrupt 6:TOP8,TOP9
        .DATA.W         EIT_reset                   ; H'0000 00B0   MJT Output Interrupt 5:TOP10
        .DATA.W         EIT_reset                   ; H'0000 00B4   MJT Output Interrupt 4:TIO4-TIO7
        .DATA.W         EIT_reset                   ; H'0000 00B8   MJT Output Interrupt 3:TIO8,TIO9
        .DATA.W         EIT_reset                   ; H'0000 00BC   MJT Output Interrupt 2:TOP0-TOP5
        .DATA.W         EIT_reset                   ; H'0000 00C0   MJT Output Interrupt 1:TOP6,TOP7
        .DATA.W         EIT_reset                   ; H'0000 00C4   MJT Output Interrupt 0:TIO0-TIO3
        .DATA.W         EIT_reset                   ; H'0000 00C8   DMAC0-4 Interrupt:DMA0-DMA4
        .DATA.W         EIT_reset                   ; H'0000 00CC   SIO1 Receive Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00D0   SIO1 Transmit Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00D4   SIO0 Receive Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00D8   SIO0 Transmit Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00DC   A-D0 Conversion Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00E0   TID0 Output Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00E4   TOD0 Output Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00E8   DMAC5-9 Interrupt:DMA5-DMA9
        .DATA.W         EIT_reset                   ; H'0000 00EC   SIO2,3 Transmit/Receive Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00F0   RTD Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00F4   TID1 Output Interrupt
        .DATA.W         EIT_reset                   ; H'0000 00F8   TOU1 Output Interrupt:TOU1_0-TOU1_7
        .DATA.W         EIT_reset                   ; H'0000 00FC   SIO4,5 Transmit/Receive Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0100   Reserved
        .DATA.W         EIT_reset                   ; H'0000 0104   Reserved
        .DATA.W         EIT_reset                   ; H'0000 0108   TML1 Input Interrupt:TIN30-TIN33
        .DATA.W         EIT_reset                   ; H'0000 010C   CAN0 Transmit/Receive & Error Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0110   CAN1 Transmit/Receive & Error Interrupt
        .DATA.W         $dri_hsync_int              ; H'0000 0114   DRI Transfer Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0118   DRI Counter Interrupt:DEC0-DEC4
        .DATA.W         $dri_vsync_int              ; H'0000 011C   DRI Event Detection Interrupt:DIN0-DIN5
        .DATA.W         EIT_reset                   ; H'0000 0120   CAN0 Transmit/Receive Completion Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0124   CAN0 Single-Shot Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0128   CAN0 Error Interrupt
        .DATA.W         EIT_reset                   ; H'0000 012C   CAN1 Transmit/Receive Completion Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0130   CAN1 Single-Shot Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0134   CAN1 Error Interrupt
        .DATA.W         EIT_reset                   ; H'0000 0138   RAM Write Monitor Interrupt
```

(Omitted)

4.7 Operation Timing

4.7.1 Data Capture Conditions

The count enable sources and counted events of each event counter are shown in table 4.7.1.

Table 4.7.1 Event Counter Count Enable Sources and Counted Events

Event Counter	Count Enabled Source	Count Event
Event counter 0 (DEC0)	DIN0 (vertical synchronizing signal) (start of one frame)	DIN1 (horizontal synchronizing signal) (to point of data capture)
Event counter 1 (DEC1)	DIN1 (horizontal synchronizing signal) (start of one column)	DIN3 (data synchronizing signal) (to data capture start point)
Event counter 2 (DEC2)	DIN1 (horizontal synchronizing signal) (start of one column)	DIN3 (captured event) (data capture range)
Event counter 3 (DEC3)	Underflow of event counter 2 (DEC2CT)	DIN3 (data synchronizing signal) (from data capture completed point to the end)

4.7.2 Data Capture Timing

The operation timing in this sample program is as shown below.

In the sample program, DEC0, DEC1, DEC3 are set to enable selective data capturing function. Therefore, when these three counters are all in an underflow state, the next event becomes enabled as the capture event.

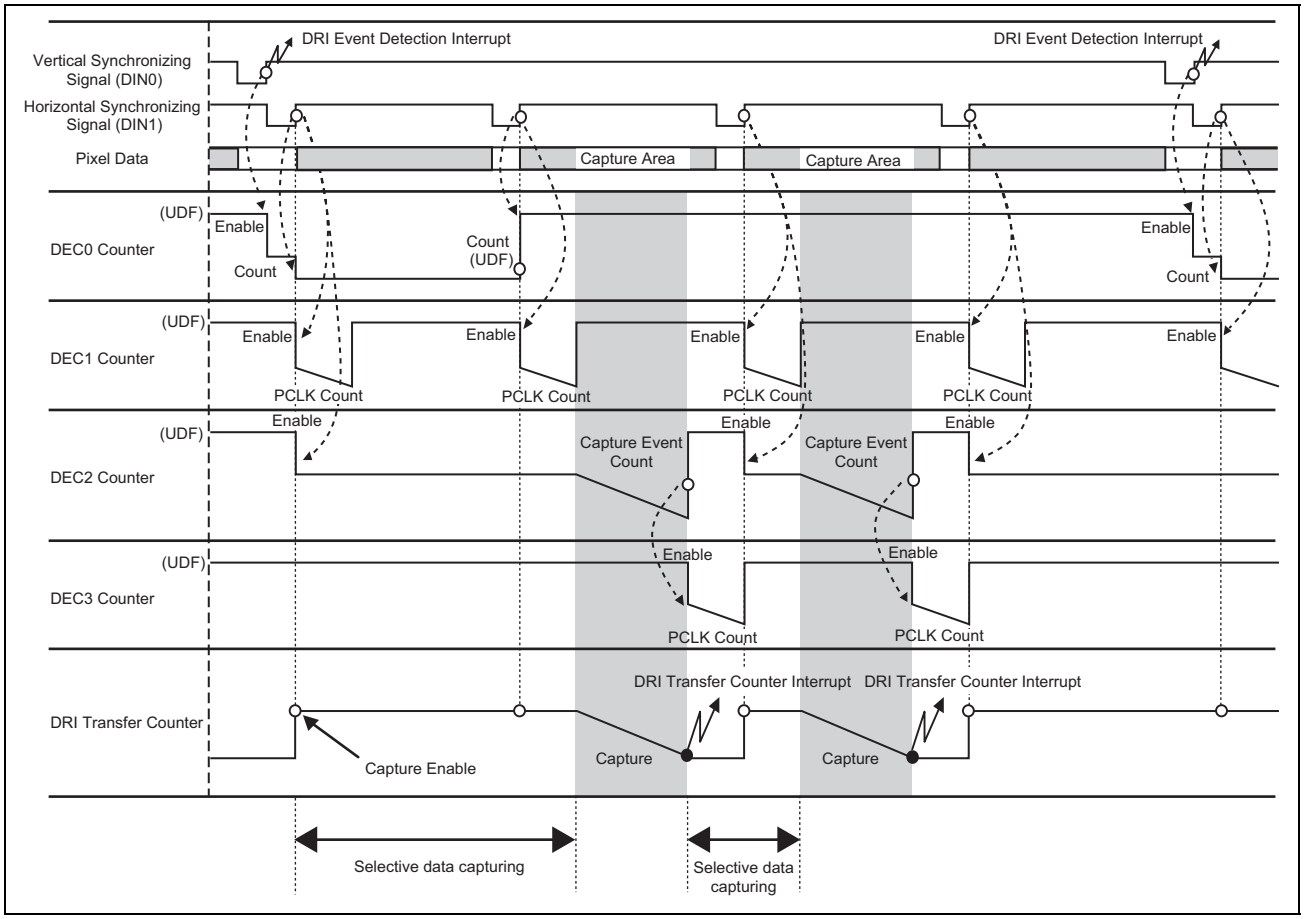
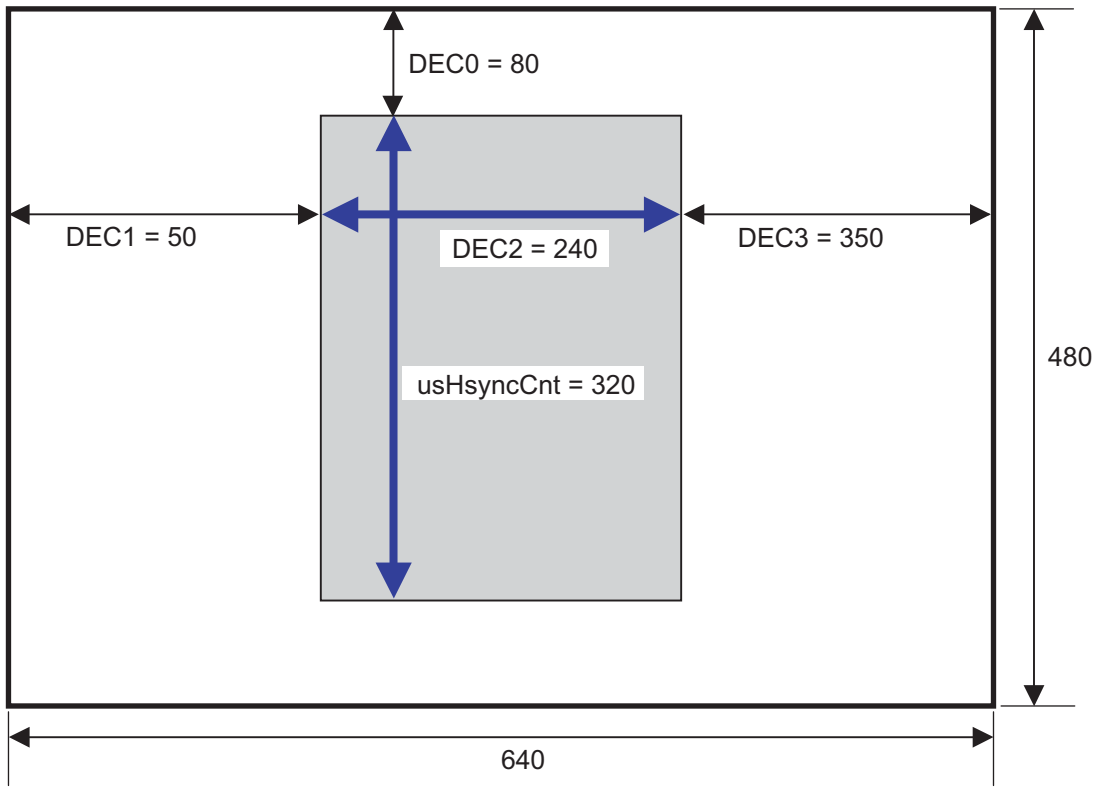


Figure 4.7.2 DRI Data Capture Timing

4.7.3 Operating Conditions

The figure below shows the counter setting values when this sample program is executed.



DEC0: Unnecessary line count starting from the vertical synchronizing signal.
DEC1: Unnecessary line count starting from the horizontal synchronizing signal.
DEC2: Number of valid pixels while capturing one line
DEC3: Number of unnecessary pixels after capturing valid pixels.
usHsyncCnt: The number of captured lines.

Figure 4.7.3 Program Operation Counter Setting

5. Reference Documents

- 32192/32196 Group Hardware Manual (Rev.1.01)
- 32195 Group Datasheet (Rev.1.00)
- M3T-CC32R Version 5.00 User's Manual (Compiler)
- M3T-AS32R Version 5.00 User's Manual (Assembler)
- M32R-FPU Software Manual (Rev.1.01)

(Please get the latest one from Renesas Technology Corp. website.)

6. Homepage and Support Center

- Renesas Technology Corp. Website:
  http://www.renesas.com/

- Inquires for all Renesas products and technical inquiries for the M32R Family products:
  Customer Support Center: csc@renesas.com

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Jul.28.05	—	First edition issued.
1.01	Mar.03.06	5	Explanation change.
		6	The order of setting the registers is changed. (Source program is updated to revision 1.01)

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.