Renesas Electronics America Inc - R5S72671W144FP#V0 Datasheet

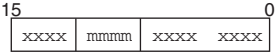
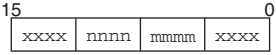
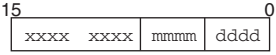
Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	SH2A-FPU
Core Size	32-Bit Single-Core
Speed	144MHz
Connectivity	CANbus, EBI/EMI, I²C, IEBus, SCI, SIO, SPI, USB
Peripherals	DMA, POR, PWM, WDT
Number of I/O	104
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	1.5M x 8
Voltage - Supply (Vcc/Vdd)	1.15V ~ 3.6V
Data Converters	A/D 6x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	176-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5s72671w144fp-v0

Instruction Formats	Source Operand	Destination Operand	Example
m format 	mmmm: Register direct	Control register or system register	LDC Rm,SR
	mmmm: Register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
	mmmm: Register indirect	—	JMP @Rm
	mmmm: Register indirect with pre-decrement	R0 (Register direct)	MOV.L @-Rm,R0
	mmmm: PC relative using Rm	—	BRAF Rm
nm format 	mmmm: Register direct	nnnn: Register direct	ADD Rm,Rn
	mmmm: Register direct	nnnn: Register indirect	MOV.L Rm,@Rn
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Register indirect with post-increment	nnnn: Register direct	MOV.L @Rm+,Rn
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	MOV.L Rm,@-Rn
	mmmm: Register direct	nnnn: Indexed register indirect	MOV.L Rm,@(R0,Rn)
md format 	mmmmdddd: Register indirect with displacement	R0 (Register direct)	MOV.B @(disp,Rm),R0

(2) Data Array Write

The longword data specified by the data is written to the position specified by the L bit in the address from the entry address specified by the address and the entry corresponding to the way.

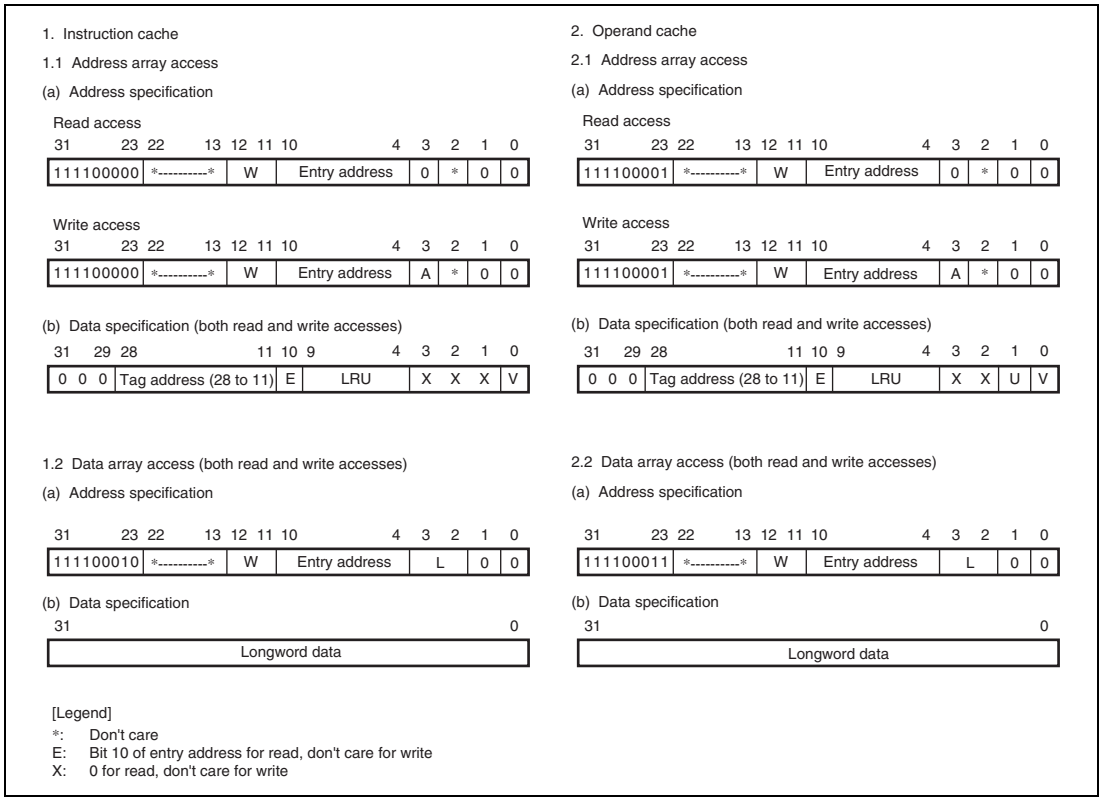


Figure 8.4 Specifying Address and Data for Memory-Mapped Cache Access

9.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	A2ROW[1:0]		-	A2COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DEEP	-	RFSH	RMODE	PDOWN	BACTV	-	-	-	A3ROW[1:0]		-	A3COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20, 19	A2ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 2 Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17, 16	A2COL[1:0]	00	R/W	Number of Bits of Column Address for Area 2 Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 11.20 TIORL_0 (Channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOC0C Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode

Figure 11.115 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

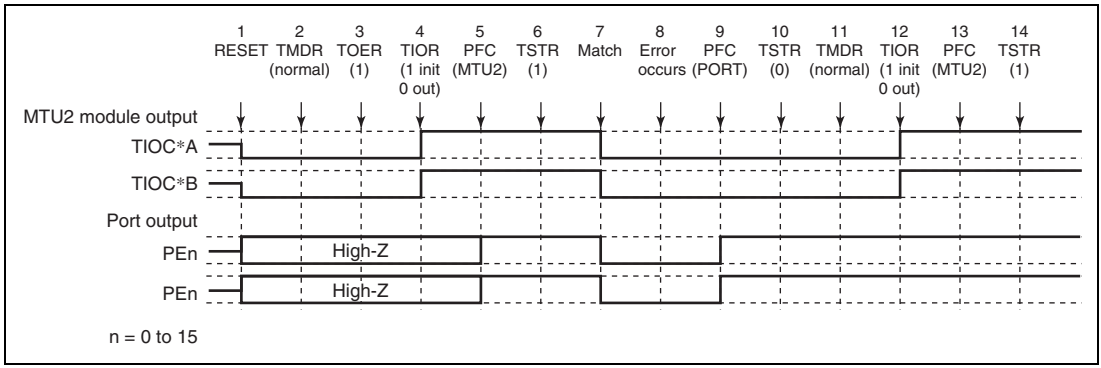


Figure 11.115 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, the module output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set the multi-function timer pulse unit 2 output with the general I/O port.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the general I/O port and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

(2) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 11.116 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

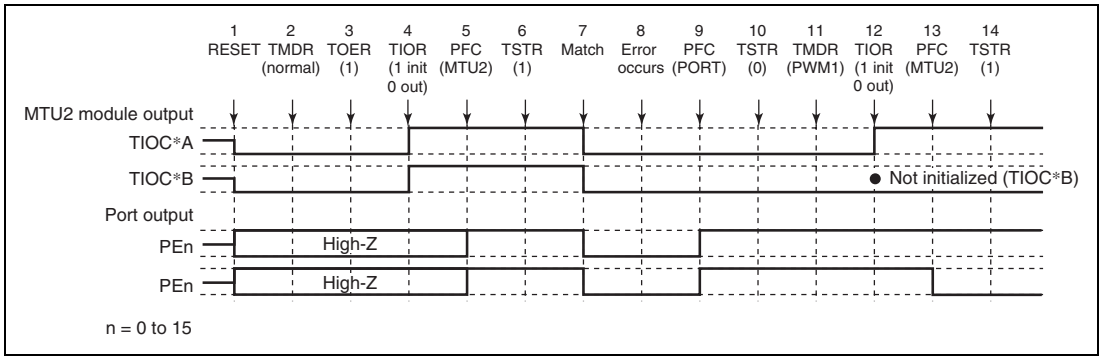


Figure 11.116 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.115.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. If initialization is required, initialize in normal mode, and then switch to PWM mode 1.)
13. Set the multi-function timer pulse unit 2 output with the general I/O port.
14. Operation is restarted by TSTR.

16.4.6 Error Detection

In the normal serial transfer, the data written from the data register (SPDR) to the transmit buffer is serially transmitted, and the serially received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit buffer/receive buffer or the status at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, this module detects the event as an overrun error or a mode fault error. Table 16.7 shows the relationship between non-normal transfer operations and the error detection function.

Table 16.7 Relationship between Non-Normal Transfer Operations and Error Detection Function

	Occurrence Condition	Operation	Error Detection
A	SPDR is written when the transmit buffer is full.	Missing write data.	None
B	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is serially transmitted.	None
C	SPDR is read when the receive buffer is empty.	The output data is undefined.	None
D	Serial transfer terminates when the receive buffer is full.	Missing serial receive data.	Overrun error (only in slave mode)
E	The SSL input signal is negated during serial transfer in slave mode.	Serial transfer suspended. Missing send/receive data. Operation disabled.	Mode fault error

(1) MSB First Transfer (32-Bit Data)

Figure 17.5 shows the operation of the transmit buffer and transmit shift register, and the receive shift register and receive buffer when this module performs a 32-bit MSB-first data transfer.

For data transmission, the CPU or direct memory access controller writes 32-bit transmit data to the transmit buffer (SPTXB). If the transmit shift register is empty, this module copies the data with MSB-aligned in the transmit buffer to the transmit shift register, and fills the transmit shift register. When data transmission is started, this module outputs data beginning at the MSB (bit 31) of the transmit shift register, and when the QSPCLK clock cycle required for the serial transfer of 32 bits has passed, the transmit shift register becomes empty.

For data reception, data received from the data pin is stored in the receive shift register beginning at the LSB (bit 0). When the QSPCLK clock cycle required for the serial transfer of 32 bits has passed, the receive shift register becomes full. If the receive buffer (SPRXB) has a space for 32 bits or more, this module copies the 32-bit data beginning at the LSB from the receive shift register to the receive buffer, and empties the receive shift register. If the receive buffer does not have a space for 32 bits or more, data reception is not carried out. In order to start reception, the specified length of data should be read from the receive buffer to secure the space for 32 bits or more in the receive buffer.

In actual transfer, this operation is repeated for the number of times defined by SPBMUL0 to SPBMUL3.

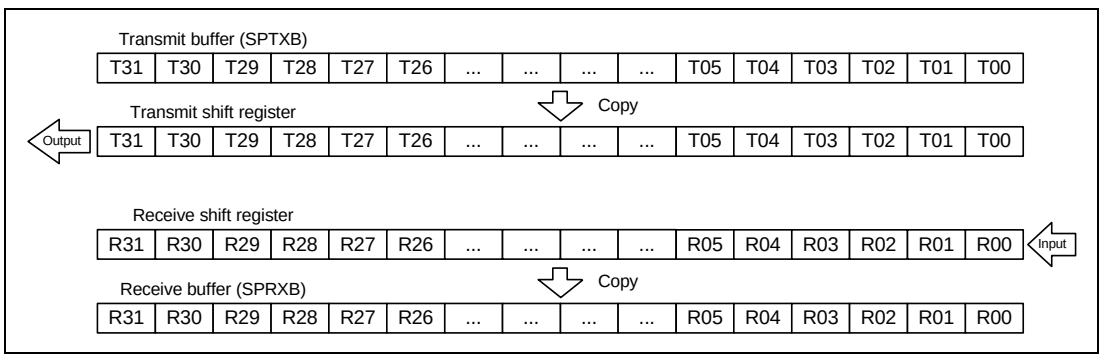


Figure 17.5 MSB First Transfer (32-Bit Data)

(2) Reception in Master Mode

Figure 20.7 shows an example of reception settings and operation when this module is used as a master.

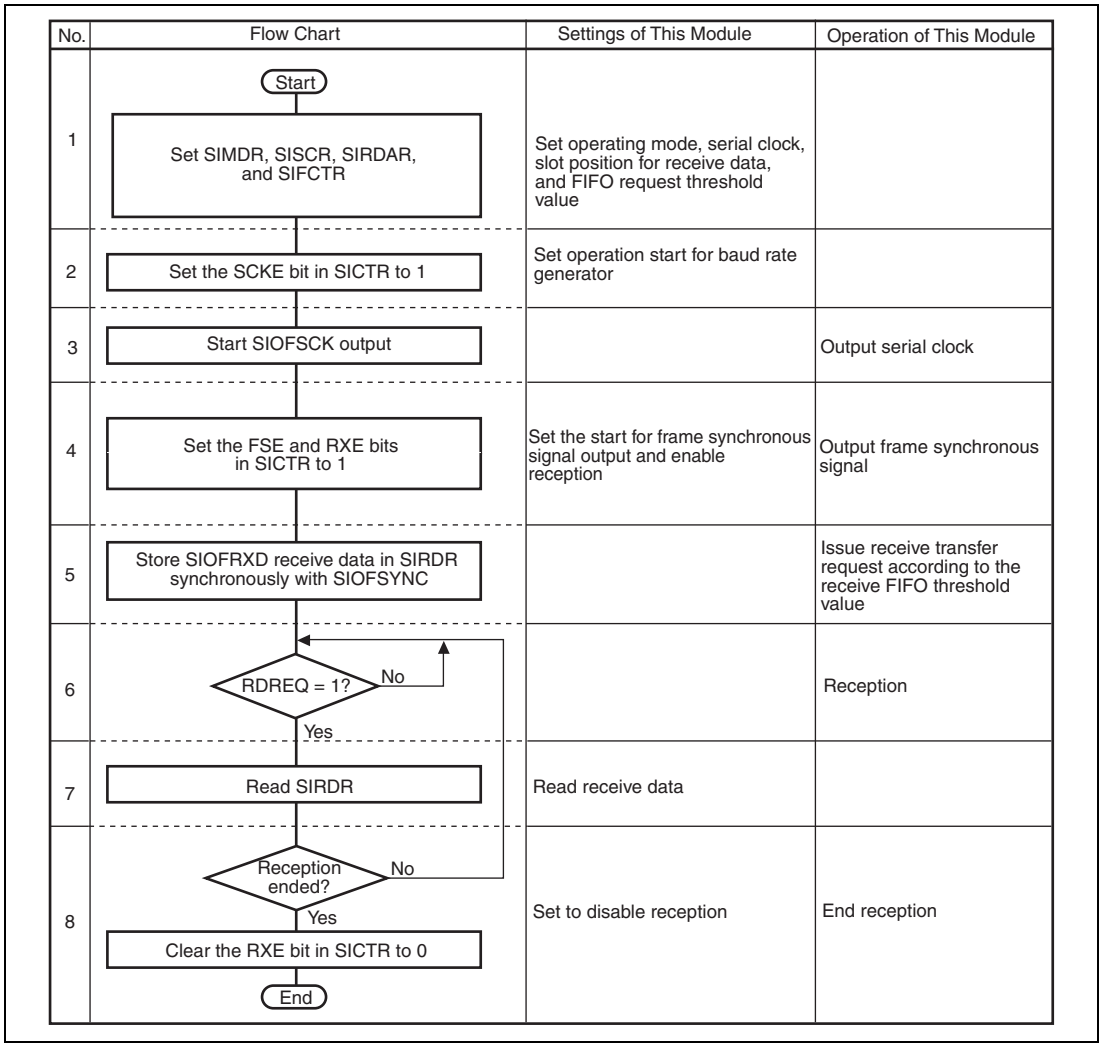


Figure 20.7 Example of Receive Operation in Master Mode

(2) Data Command Transfer (Control Bits: Read (H'3, H'7), Write (H'A, H'B, H'E, H'F))

In the case of data read (H'3, H'7), data in the data buffer of the slave unit is read in the master unit. In the case of data write (H'B or H'F) or command write (H'A or H'E), data received in the slave unit is processed in accordance with the operation specification of the slave unit.

Notes: 1. The user can select data and commands freely in accordance with the system.
2. H'3, H'A, or H'B may lock depending on the communications condition and status.

(3) Locked Address Read (Control Bits: H'4, H'5)

In the case of the locked address read (H'4 or H'5), the address (12 bits) of the master unit, which issues the lock instruction, is configured in bytes as shown in figure 22.3.

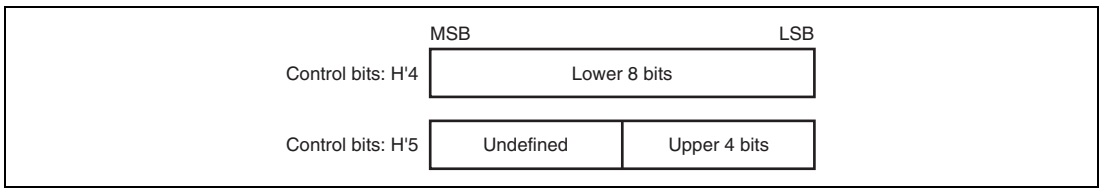


Figure 22.3 Locked Address Configuration

(4) Locking/Unlocking (Control Bits: Setting (H'3, H'A, H'B), Cancellation: (H'6))

The lock function is used for message transfer over multiple communications frames. A locked unit receives data only from the unit which locked it.

Locking and unlocking are described below.

(a) Locking

When an acknowledge bit of 0 in the message length field is transmitted/received with the control bits (H'3, H'A, H'B) indicating the lock operation, and then the communications frame is completed before completion of data transmission/reception for the number of bytes specified by the message length bits, the slave unit is locked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte data indicating the slave status is set to 1.

Lock is set only when the number of data exceeds the maximum number of transfer bytes in one frame. Lock is not set by other error terminations.

24.3.10 Post-ECC Subheader Error Status Register (CROMST3)

The post-ECC subheader error status register (CROMST3) indicates error status in the post-ECC subheader.

Bit:	7	6	5	4	3	2	1	0
	ER2_ SHEAD0	ER2_ SHEAD1	ER2_ SHEAD2	ER2_ SHEAD3	ER2_ SHEAD4	ER2_ HEAD5	ER2_ HEAD6	ER2_ HEAD7
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	ER2_ SHEAD0	0	R	Indicates that the subheader (file number) still has an error after ECC correction. Indicates the error of the SHEAD20 register.
6	ER2_ SHEAD1	0	R	Indicates that the subheader (channel number) still has an error after ECC correction. Indicates the error of the SHEAD21 register.
5	ER2_ SHEAD2	0	R	Indicates that the subheader (sub-mode) still has an error after ECC correction. Indicates the error of the SHEAD22 register.
4	ER2_ SHEAD3	0	R	Indicates that the subheader (data type) still has an error after ECC correction. Indicates the error of the SHEAD23 register.
3	ER2_ SHEAD4	0	R	Indicates that the subheader (file number) still has an error after ECC correction. Indicates the error of the SHEAD24 register.
2	ER2_ SHEAD5	0	R	Indicates that the subheader (channel number) still has an error after ECC correction. Indicates the error of the SHEAD25 register.
1	ER2_ SHEAD6	0	R	Indicates that the subheader (sub-mode) still has an error after ECC correction. Indicates the error of the SHEAD26 register.
0	ER2_ SHEAD7	0	R	Indicates that the subheader (data type) still has an error after ECC correction. Indicates the error of the SHEAD27 register.

Bit	Bit Name	Initial Value	R/W	Description
14	REW	0	R/W*	Buffer Pointer Rewind Specifies whether or not to rewind the buffer pointer. 0: The buffer pointer is not rewound. 1: The buffer pointer is rewound. When the selected pipe is in the receiving direction, setting this bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.
13, 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11, 10	MBW[1:0]	00	R/W	CFIFO Port Access Bit Width Specifies the bit width for accessing the CFIFO port. 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited Once reading data is started after setting these bits, these bits should not be modified until all the data has been read. When the selected pipe is in the receiving direction, these bits should be set in the following timing: • Set the CURPIPE and MBW bits simultaneously. • When the DCP is selected (CURPIPE = B'000), set the ISEL and MBW bits simultaneously. For details, see section 27.4.4, FIFO Buffer Memory. When the selected pipe is in the transmitting direction, the bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.

Bit	Bit Name	Initial Value	R/W	Description
15	BVAL	0	R/W*[2]	Buffer Memory Valid Flag This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE bits (selected pipe). 0: Invalid 1: Writing ended When the selected pipe is in the transmitting direction, set this bit to 1 in the following cases. Then, this module switches the FIFO buffer from the CPU side to the SIE side, enabling transmission. • To transmit a short packet, set this bit to 1 after data has been written. • To transmit a zero-length packet, set this bit to 1 before data is written to the FIFO buffer. • Set this bit to 1 after the number of data bytes has been written for the pipe in continuous transfer mode, where the number is a natural integer multiple of the maximum packet size and less than the buffer size. When the data of the maximum packet size has been written for the pipe in non-continuous transfer mode, this module sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission. When the selected pipe is in the transmitting direction, if 1 is written to BVAL and BCLR bits simultaneously, this module clears the data that has been written before it, enabling transmission of a zero-length packet. Writing 1 to this bit should be done while FRDY indicates 1 (set by this module). When the selected pipe is in the receiving direction, do not set this bit to 1.

27.3.23 USB Address Register (USBADDR)

USBADDR is a register that indicates the USB address. This register is valid only when the function controller function is selected. When the host controller function is selected, peripheral device addresses should be set using the DEVSEL bits in PIPEMAXP.

This register is initialized by a power-on reset or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	USBADDR [6:0]	H'00	R	USB Address When the function controller function is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.

(b) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing using the current pipe once again. The REW bit in C/DnFIFOSEL is used for this.

If a pipe is selected when the REW bit is set to 1 and at the same time the CURPIPE bit in C/DnFIFOSEL is set, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. Also, if a pipe is selected with 0 set for the REW bit, data can be read and written in continuation of the previous selection, without the pointer used for reading from and writing to the buffer memory being reset.

To access the FIFO port, FRDY = 1 must be ensured after selecting a pipe.

(c) Accessing FIFO Port for Odd Data

For reading data from the FIFO port, when the number of data bits to be read is smaller than the access width specified by the MBW bits in the FIFO port select registers, read the data with the specified width and discard the unnecessary bits through software.

For writing data to the FIFO port, when the number of data bits to be written is smaller than the access width specified by the MBW bits in the FIFO port select registers, access the registers as shown in the following examples. In the examples, the FIFO port access width is 32 bits (MBW = 10) and 24-bit data is written to the FIFO port.

(6) Controlling Video and Graphics Image Display Positions

This module provides functions for generating the Vsync and Hsync signals for video, graphics images, and output to the panel with desired timings with respect to the reference Vsync signal. The positions of the video and graphics image displayed on the panel are shown in figure 28.8.

The timing of the Vsync signal for video has a restriction because the period for buffering the input video in the large-capacity on-chip RAM should be taken into account. For details, refer to section 28.8, Usage Notes.

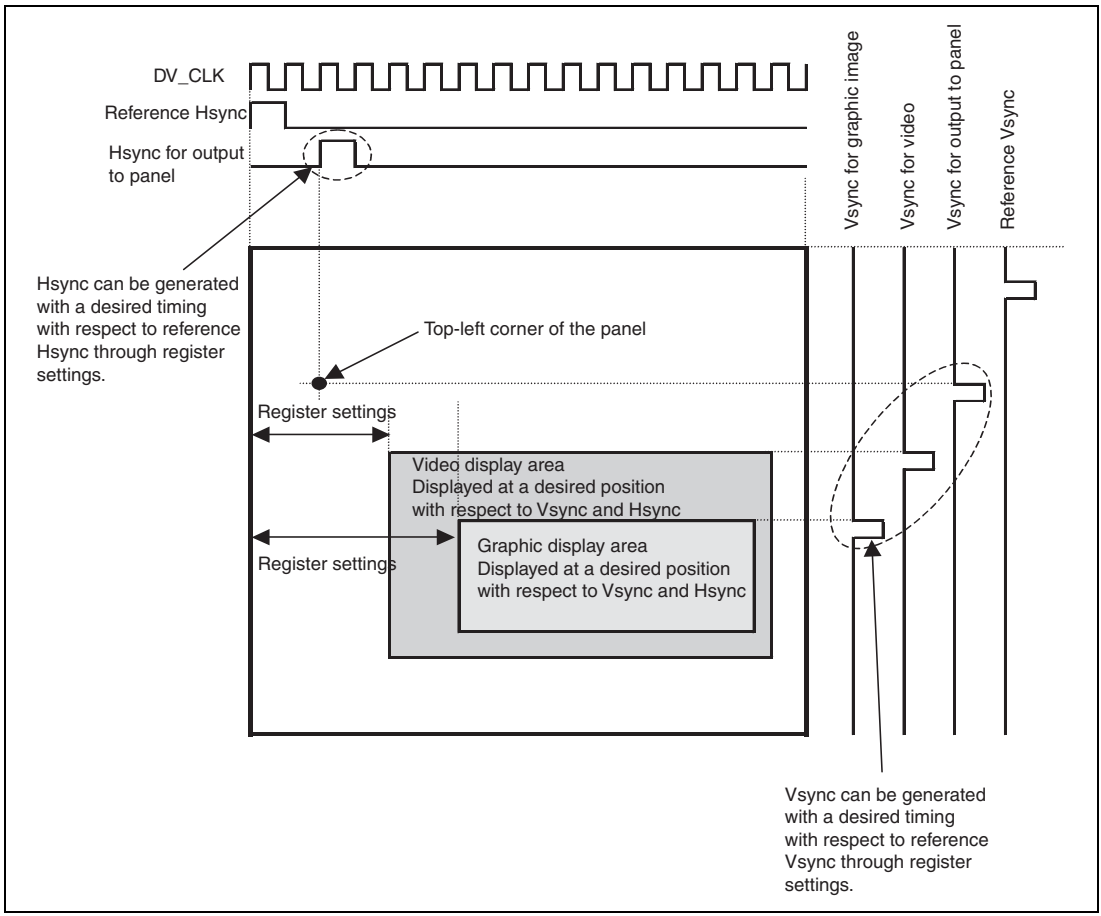
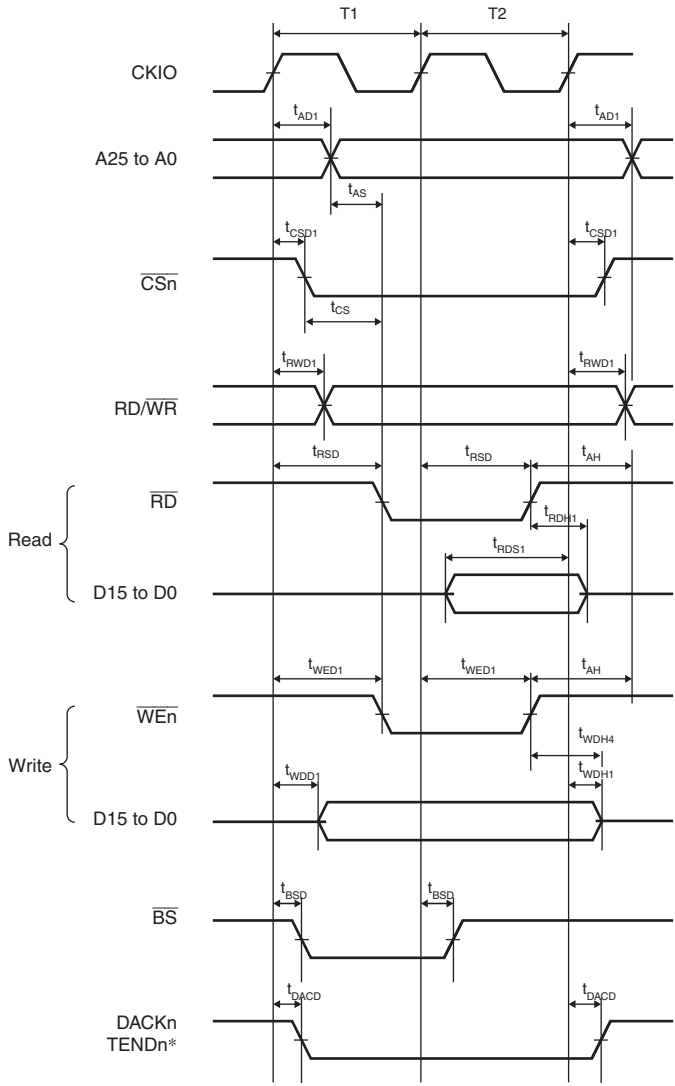


Figure 28.8 Video and Graphics Image Display Positions on Panel

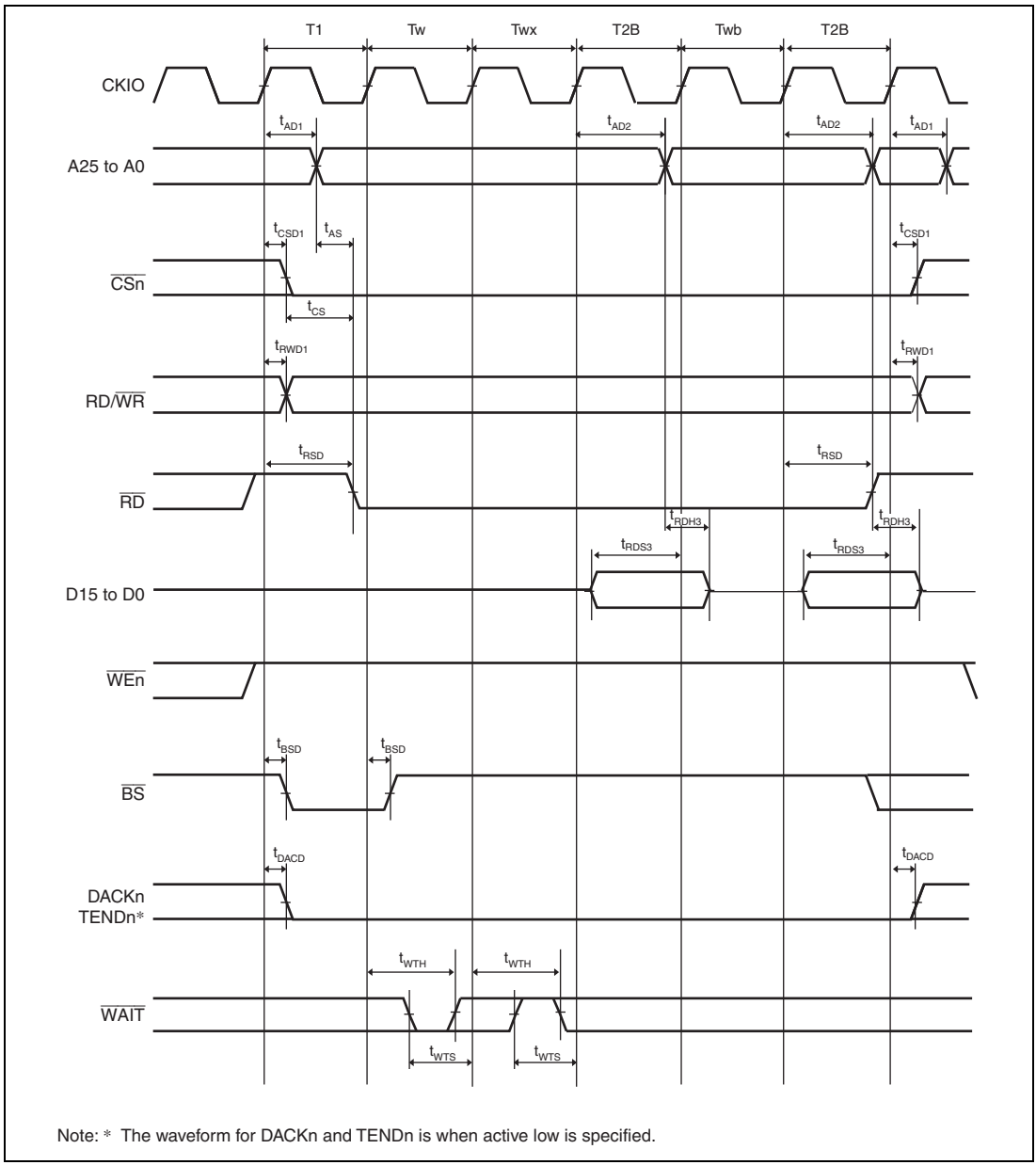
Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
Direct memory access controller	RDMATCR_7	—	—	—	—	—	—	—	—
	SAR_8								
	DAR_8								
	DMATCR_8	—	—	—	—	—	—	—	—
	CHCR_8	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR_8								
	RDAR_8								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
USB 2.0 host/function module	PIPE1TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE1TRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE2TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE2TRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE3TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	IPE3TRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE4TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE4TRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	PIPE5TRE	—	—	—	—	—	—	TRENB	TRCLR
		—	—	—	—	—	—	—	—
	PIPE5TRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	USBACSWR1	—	—	—	—	—	—	—	—
		UAC23	—	—	—	—	—	—	—
	DEVADD0	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD1	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD2	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD3	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—
	DEVADD4	—	UPPHUB[3]	UPPHUB[2]	UPPHUB[1]	UPPHUB[0]	HUBPORT[2]	HUBPORT[1]	HUBPORT[0]
		USBSPD[1]	USBSPD[0]	—	—	—	—	—	—



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 37.9 Basic Bus Timing for Normal Space (No Wait)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 37.16 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two Burst)