Renesas Electronics America Inc - UPD78F0474GC-GAD-AX 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	Active
Core Processor	78K/0
Core Size	8-Bit
Speed	10MHz
Connectivity	3-Wire SIO, LINbus, UART/USART
Peripherals	LCD, LVD, POR, PWM, WDT
Number of I/O	62
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0474gc-gad-ax

CHAPTER 25 POWER-ON-CLEAR CIRCUIT 674

25.1 Functions of Power-on-Clear Circuit 674
25.2 Configuration of Power-on-Clear Circuit 675
25.3 Operation of Power-on-Clear Circuit 675
25.4 Cautions for Power-on-Clear Circuit 678

CHAPTER 26 LOW-VOLTAGE DETECTOR 680

26.1 Functions of Low-Voltage Detector 680
26.2 Configuration of Low-Voltage Detector 681
26.3 Registers Controlling Low-Voltage Detector 681
26.4 Operation of Low-Voltage Detector 684
26.4.1 When used as reset 685
26.4.2 When used as interrupt 690
26.5 Cautions for Low-Voltage Detector 695

CHAPTER 27 OPTION BYTE 698

27.1 Functions of Option Bytes 698
27.2 Format of Option Byte 700

CHAPTER 28 FLASH MEMORY 703

28.1 Internal Memory Size Switching Register 703
28.2 Internal Expansion RAM Size Switching Register 704
28.3 Writing with Flash memory programmer 705
28.4 Programming Environment 708
28.5 Communication Mode 708
28.6 Connection of Pins on Board 710
28.6.1 FLMD0 pin 710
28.6.2 Serial interface pins 710
28.6.3 $\overline{\text{RESET}}$ pin 712
28.6.4 Port pins 712
28.6.5 REGC pin 712
28.6.6 Other signal pins 712
28.6.7 Power supply 712
28.7 Programming Method 713
28.7.1 Controlling flash memory 713
28.7.2 Flash memory programming mode 713
28.7.3 Selecting communication mode 714
28.7.4 Communication commands 715
28.8 Security Settings 716
28.9 Processing Time for Each Command When PG-FP5 Is Used (Reference) 718
28.10 Flash Memory Programming by Self-Programming 721
28.10.1 Boot swap function 729

(2/2)

Item		μPD78F0471	μPD78F0472	μPD78F0473	μPD78F0474	μPD78F0475
10-bit successive approximation type A/D converter		–				
16-bit ΔΣ type A/D converter		–				
Serial interface		• UART supporting LIN-bus[Note 1]: 1 channel • 3-wire serial I/O/UART[Note 2]: 1 channel • Automatic transmit/receive 3-wire CSI: 1 channel				
LCD controller/driver		• External resistance division and internal resistance division are switchable. • Segment signal outputs: 40 (36) [36 (32)] [Note 3, 4] • Common signal outputs: 4 (8)[Note 3]				
Remote controller receiver		Provided				
Manchester code generator		Provided				
Vectored interrupt sources	Internal	20				
	External	7				
<R> Segment key source signal output		Segment key source signal outputs: 8 (SEG24(KS0)-SEG31(KS7))				
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of key input pins (KR0 to KR7).				
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector				
On-chip debug function		Provided				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = −40 to +85°C				
Package		• 80-pin plastic LQFP (14 × 14) • 80-pin plastic LQFP (fine pitch) (12 × 12)				

Notes 1. The LIN-bus supporting UART pins can be changed to the automatic transmit/receive 3-wire CSI pins (pin numbers 75 and 76).

2. Select either of the functions of these alternate-function pins.

3. The values in parentheses are the number of signal outputs when 8com is used.

<R> **4.** The values in square brackets are the number of signal outputs when using the UART6 pins (RxD6, TxD6) on the bottom side.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are three types of pin I/O buffer power supplies: AV_{REF}[Note 1], V_{LC0}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}[Note 1]	P20 to P27
V_{LC0}	COM0 to COM7, SEG0 to SEG31, SEG32 to SEG39[Note 2], V_{LC0} to V_{LC3}
V_{DD}	Pins other than above

Notes 1. μPD78F048x and 78F049x only. The power supply is V_{DD} with μPD78F047x.

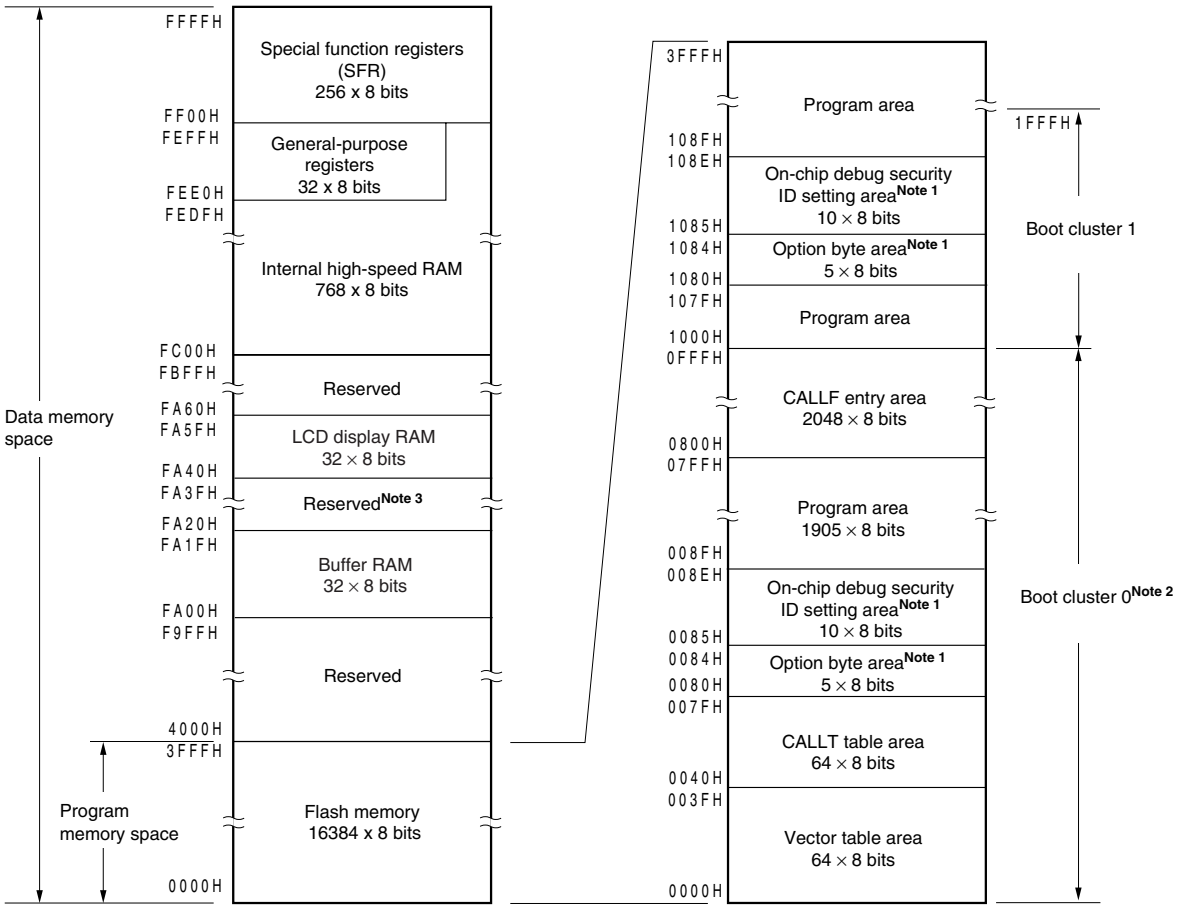
2. μPD78F047x and 78F048x only.

(1) Port pins **(1/3)**

Function Name	I/O	Function	After Reset	Alternate Function
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCL
P11				$\overline{SCK10}$
P12				SI10/RxD0
P13				SO10/TxD0
P14				$\overline{SCKA0}$/INTP4
P15				SIA0/<RxD6>
P16				SOA0/<TxD6>
P17				–

Remark The functions within arrowheads (< >) can be assigned by setting the input switch control register (ISC).

Figure 3-2. Memory Map (μPD78F0491)



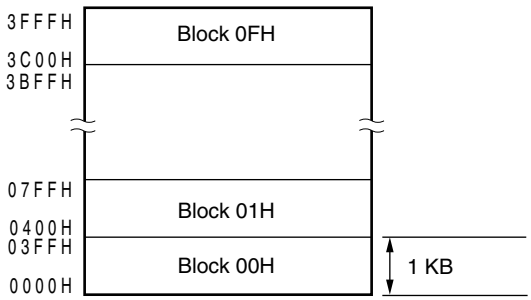
Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

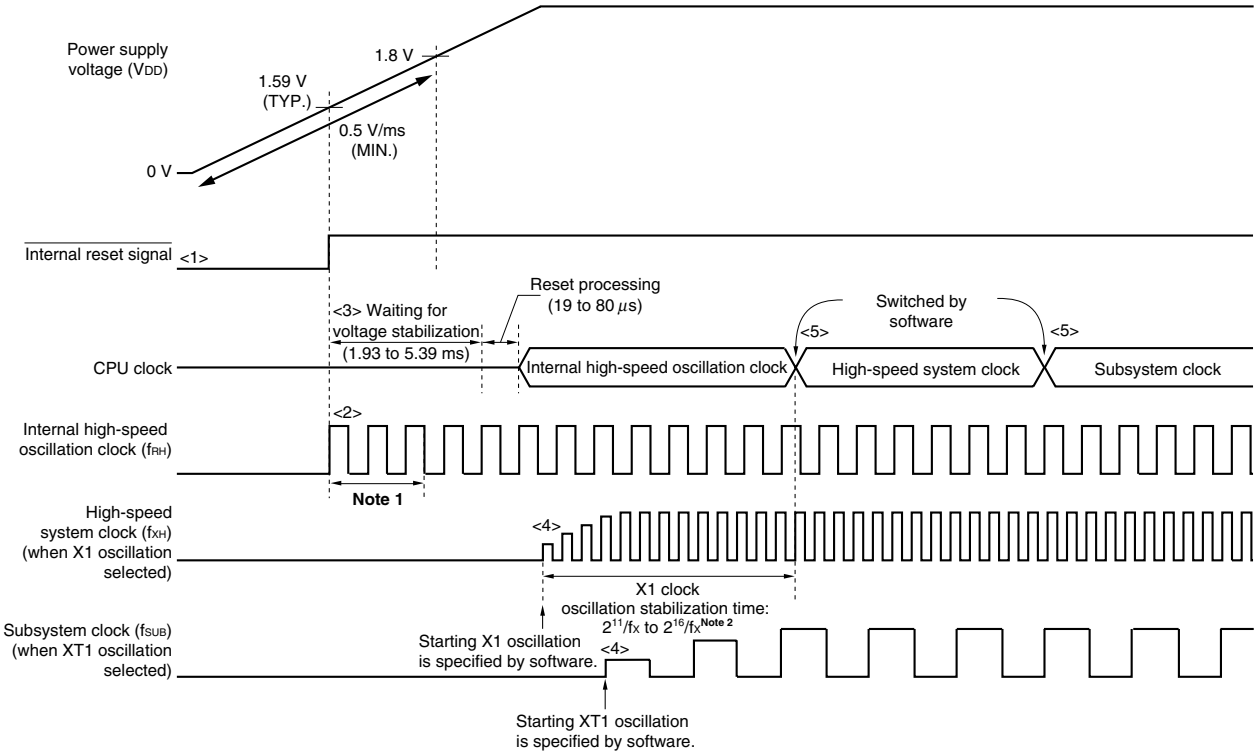
2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **28.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.



<R> **Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When 1.59 V POC Mode Is Set (Option Byte: POCMODE = 0))**



<1> When the power is turned on, an internal reset signal is generated by the power-on-clear (POC) circuit.

<2> When the power supply voltage exceeds 1.59 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

<3> When the power supply voltage rises with a slope of 0.5 V/ms (MIN.), the CPU starts operation on the internal high-speed oscillation clock after the reset is released and after the stabilization times for the voltage of the power supply and regulator have elapsed, and then reset processing is performed.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock)**.

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Notes 1. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 1.8 V, or set the 2.7 V/1.59 V POC mode by using the option byte (POCMODE = 1) (see Figure 5-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-13 after reset release by the $\overline{\text{RESET}}$ pin.

2. It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

(4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped in the following two ways.

- Executing the STOP instruction to set the STOP mode
- Setting MSTOP to 1 and stopping the X1 oscillation (disabling clock input if the external clock is used)

(a) To execute a STOP instruction

<1> Setting to stop peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 23 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after standby release

When the CPU is operating on the X1 clock, set the value of the OSTS register before the STOP instruction is executed.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1

<1> Confirming the CPU clock status (PCC and MCM registers)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock.

When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the high-speed system clock (MOC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

Caution Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

5.6.2 Example of controlling internal high-speed oscillation clock

The following describes examples of clock setting procedures for the following cases.

(1) When restarting oscillation of the internal high-speed oscillation clock
(2) When using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as peripheral hardware clock
(3) When stopping the internal high-speed oscillation clock

Figure 6-9. Format of Prescaler Mode Register 00 (PRM00)

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	PRM002	PRM001	PRM000

ES101	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES001	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM002	PRM001	PRM000	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz
0	1	1	$f_{PRS}/2^4$	1.25 MHz	2.5 MHz	625 kHz
1	0	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	0	1	f_{SUB}	32.768 kHz		
1	1	0	TI000 valid edge[Note 3]			
1	1	1	TM52 output			

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.
- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of PRM002 = PRM001 = PRM000 = 0 (count clock: f_{PRS}) is prohibited.

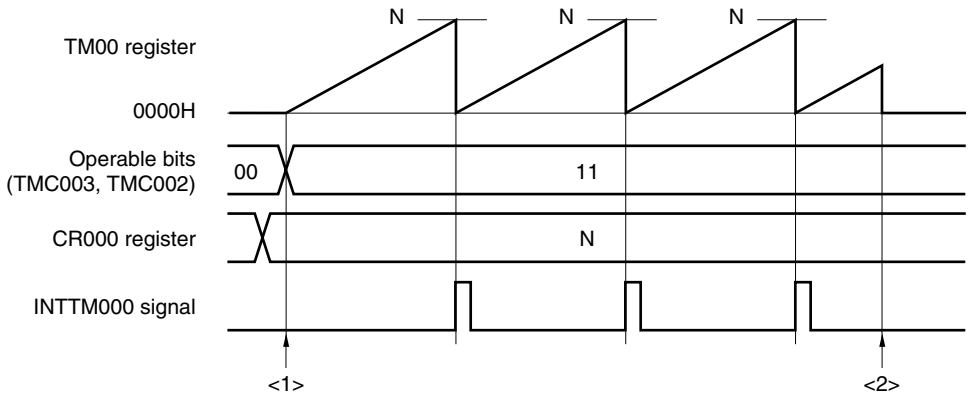
3. The external clock from the TI000 pin requires a pulse longer than twice the cycle of the peripheral hardware clock (f_{PRS}).

Caution Do not select the valid edge of TI000 as the count clock during the pulse width measurement.

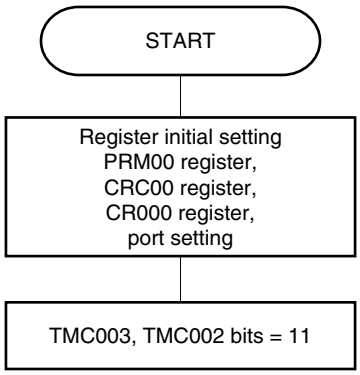
Remarks 1. 8-bit timer/event counter 52 (TM52) output can be selected as the TM00 count clock by setting PRM002, PRM001, PRM000 = 1, 1, 1. Any frequency can be set as the 16-bit timer (TM00) count clock, depending on the TM52 count clock and compare register setting values.

2. f_{PRS}: Peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency

Figure 6-15. Example of Software Processing for Interval Timer Function



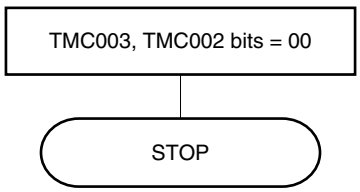
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Figure 7-8. Format of Timer Clock Selection Register 52 (TCL52)

Address: FF5BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520

TCL522	TCL521	TCL520	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	Falling edge of clock selected by ISC2			
0	0	1	Rising edge of clock selected by ISC2			
0	1	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of TCL522, TCL521, TCL520 = 0, 1, 0 (count clock: f_{PRS}) is prohibited.

Cautions 1. When rewriting TCL52 to other data, stop the timer operation beforehand.

2. Be sure to clear bits 3 to 7 to 0.

Remark f_{PRS}: Peripheral hardware clock frequency

(4) Real-time counter control register 2 (RTCC2)

The RTCC2 register is an 8-bit register that is used to control the interval interrupt function and the RTCDIV pin.

RTCC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-5. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: FF8BH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
RTCC2	RINTE	RCLOE2	RCKDIV	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	Interval interrupt (INTRTCI) selection
0	×	×	×	Interval interrupt is not generated.
1	0	0	0	$2^6/f_{RTC}$ (1.953125 ms)
1	0	0	1	$2^7/f_{RTC}$ (3.90625 ms)
1	0	1	0	$2^8/f_{RTC}$ (7.8125 ms)
1	0	1	1	$2^9/f_{RTC}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{RTC}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{RTC}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{RTC}$ (125 ms)

RCLOE2[Note]	RTCDIV pin output control
0	Output of RTCDIV pin is disabled.
1	Output of RTCDIV pin is enabled.

RCKDIV	Selection of RTCDIV pin output frequency
0	RTCDIV pin outputs 512 Hz (1.95 ms).
1	RTCDIV pin outputs 16.384 kHz (0.061 ms).

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Cautions 1. Change ICT2, ICT1, and ICT0 when RINTE = 0.

2. When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of f_{RTC} and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of f_{XT} may be generated.

14.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the BRGC0 register (see **Figure 14-4**).
<2> Set bits 1 to 4 (SL0, CL0, PS00, and PS01) of the ASIM0 register (see **Figure 14-2**).
<3> Set bit 7 (POWER0) of the ASIM0 register to 1.
<4> Set bit 6 (TXE0) of the ASIM0 register to 1. → Transmission is enabled.
Set bit 5 (RXE0) of the ASIM0 register to 1. → Reception is enabled.
<5> Write data to the TXS0 register. → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 14-2. Relationship Between Register Settings and Pins

POWER0	TXE0	RXE0	PM13	P13	PM12	P12	UART0 Operation	Pin Function	
								TxD0/SO10/P13	RxD0/SI10/P12
0	0	0	×Note	×Note	×Note	×Note	Stop	SO10/P13	SI10/P12
1	0	1	×Note	×Note	1	×	Reception	SO10/P13	RxD0
	1	0	0	×	×Note	×Note	Transmission	TxD0	SI10/P12
	1	1	0	×	1	×	Transmission/ reception	TxD0	RxD0

Note Can be set as port function or serial interface CSI10.

Remark ×: don't care
POWER0: Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
PM1×: Port mode register
P1×: Port output latch

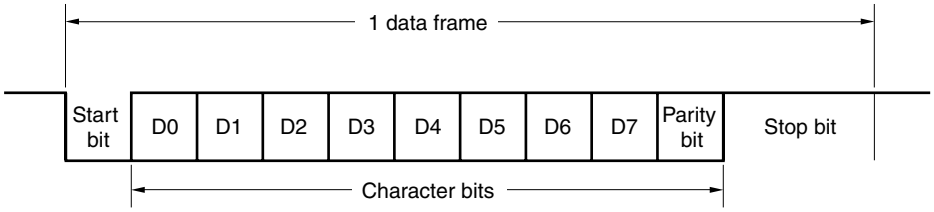
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

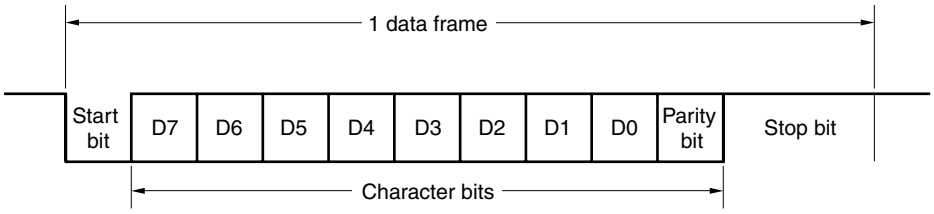
Figures 15-15 and 15-16 show the format and waveform example of the normal transmit/receive data.

Figure 15-15. Format of Normal UART Transmit/Receive Data

1. LSB-first transmission/reception



2. MSB-first transmission/reception



One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

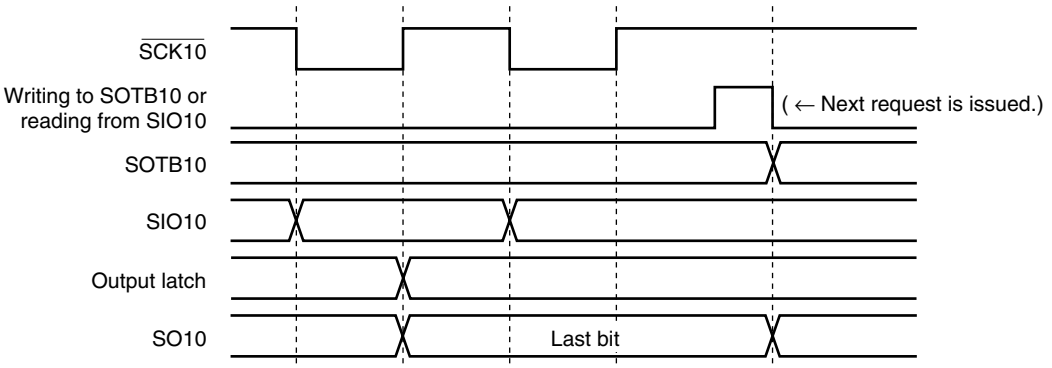
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6 (ASIM6).

Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).

Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 16-9. Output Value of SO10 Pin (Last Bit) (2/2)

(c) Type 2: CKP10 = 0, DAP10 = 1



(d) Type 4: CKP10 = 1, DAP10 = 1

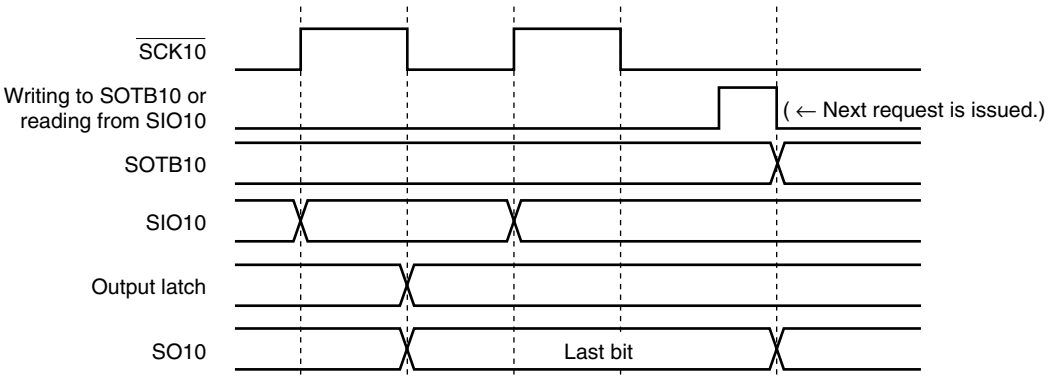
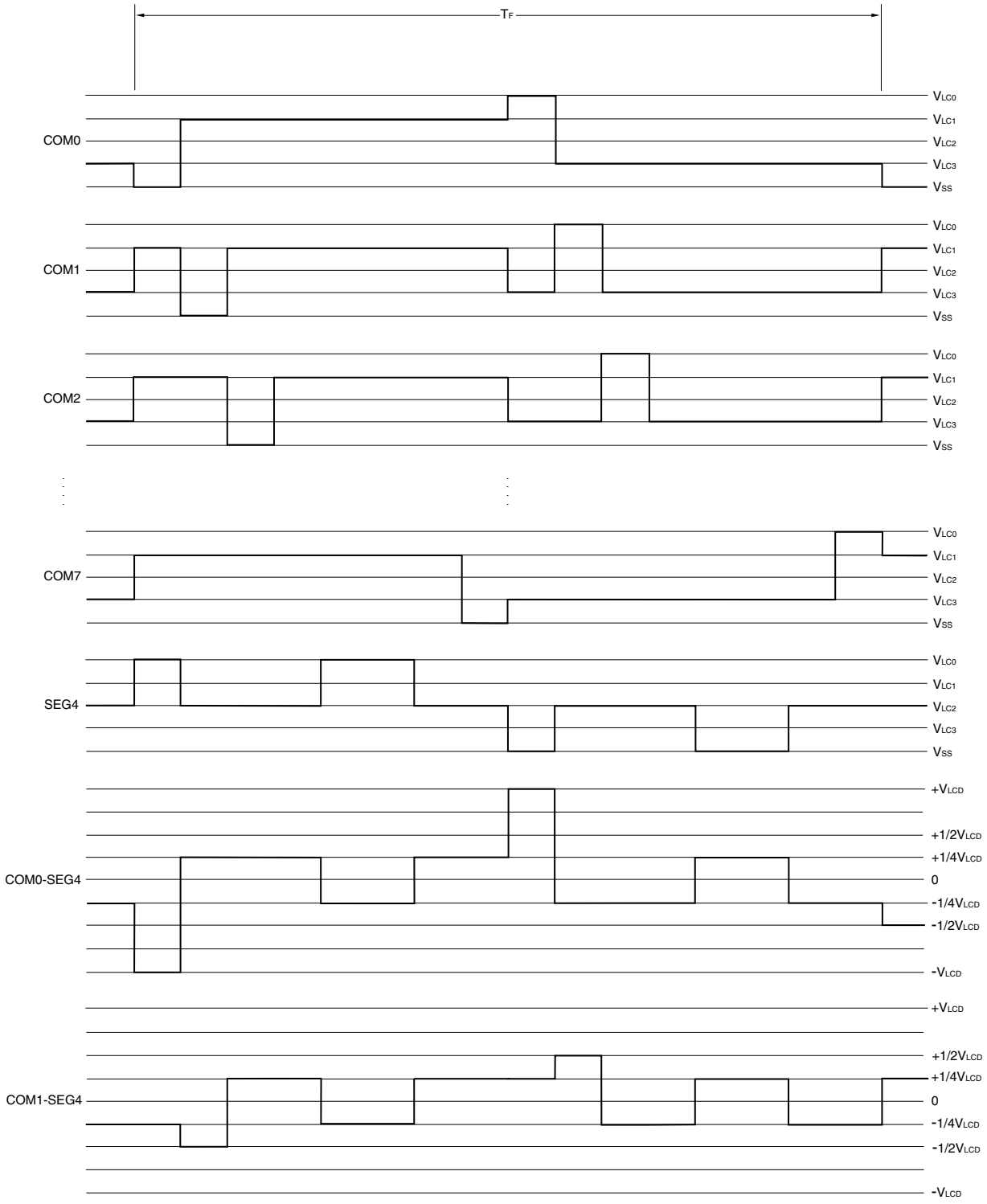


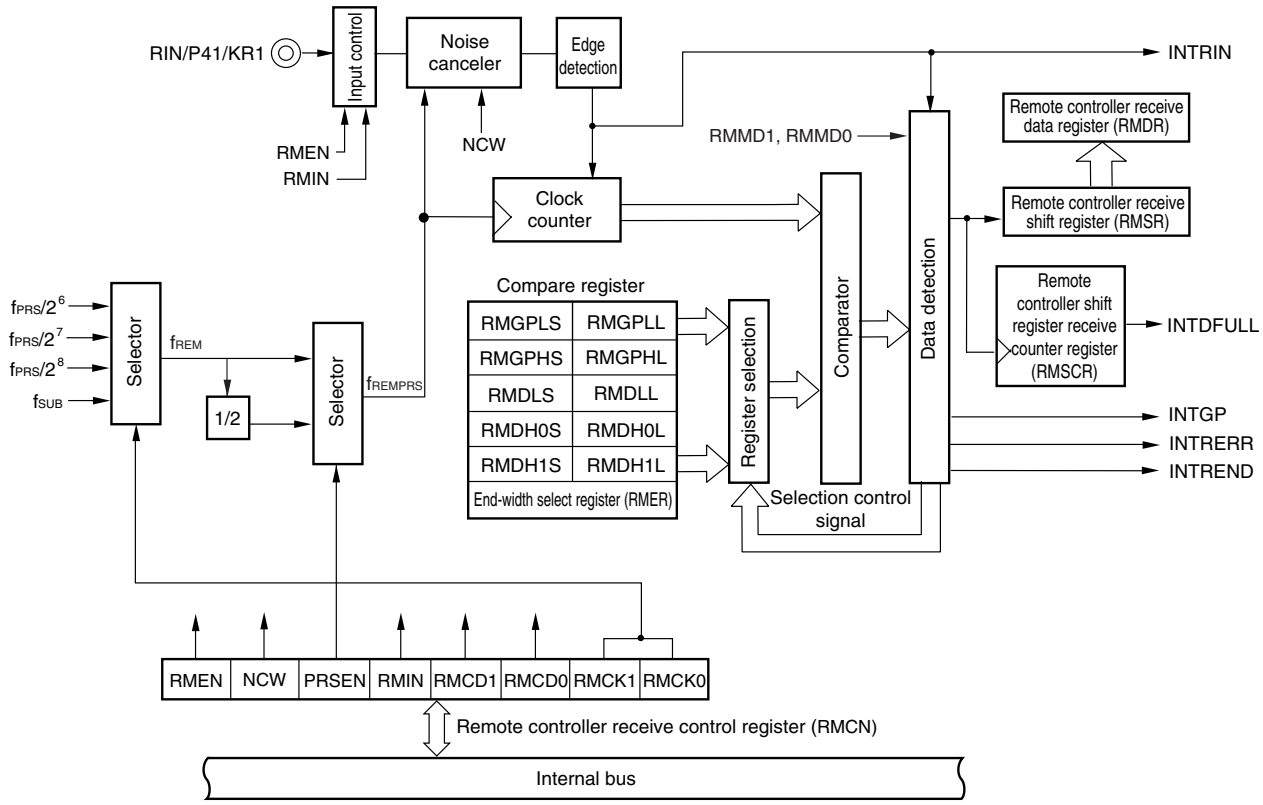
Figure 18-32. Eight-Time-Slice LCD Drive Waveform Examples (1/4 Bias Method)

(a) When segment key scan function is not used (KSON = 0)



Remark The waveforms for COM3 to COM6, COM2 to SEG4 and COM7 to SEG4 are omitted.

Figure 20-1. Block Diagram of Remote Controller Receiver



(1) Remote controller receive shift register (RMSR)

This is an 8-bit register for reception of remote controller data.

Data is stored in bit 7 first. Each time new data is stored, the stored data is shifted to the lower bits. Therefore, the latest data is stored in bit 7, and the first data is stored in bit 0.

RMSR is read with an 8-bit memory manipulation instruction.

Reset signal generation sets RMSR to 00H.

Also, RMSR is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMEN = 0).
- Error is detected (INTRERR is generated).
- INTDFULL is generated.
- RMSR is read after INTREND has been generated.

Caution Reading RMSR is disabled during remote controller reception. Complete reception, then read RMSR. When the reading operation is complete, RMSR is cleared. Therefore, values once read are not guaranteed.

(4) Remote controller receive GPLS compare register (RMGPLS) (Type B reception mode)

This register is used to detect the low level of a remote controller guide pulse (short side).

RMGPLS is set with an 8-bit memory manipulation instruction.

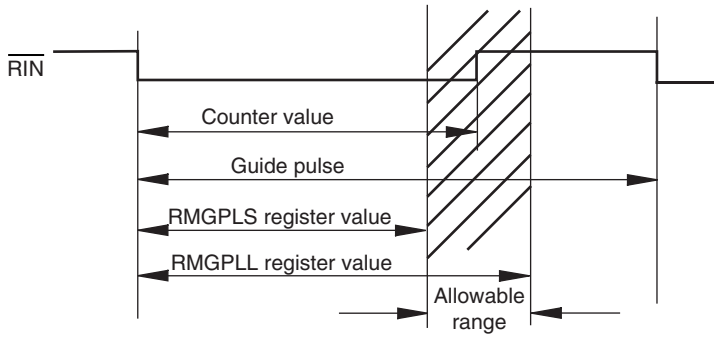
Reset signal generation sets RMGPLS to 00H.

(5) Remote controller receive GPLL compare register (RMGPLL) (Type B reception mode)

This register is used to detect the low level of a remote controller guide pulse (long side).

RMGPLL is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPLL to 00H.



If RMGPLS ≤ counter value < RMGPLL is satisfied, it is assumed that the low level of the guide pulse has been successfully received.

21.4 Interrupt Servicing Operations

21.4.1 Maskable interrupt acknowledgment

A maskable interrupt becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0).

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 21-4 below.

For the interrupt request acknowledgment timing, see **Figures 21-8** and **21-9**.

Table 21-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: $1/f_{CPU}$ (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 21-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

26.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

(1) When used as reset

The system may be repeatedly reset and released from the reset status.

In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take (b) of action (2) below.

<Action>

(1) When used as reset

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 26-9**).

(2) When used as interrupt

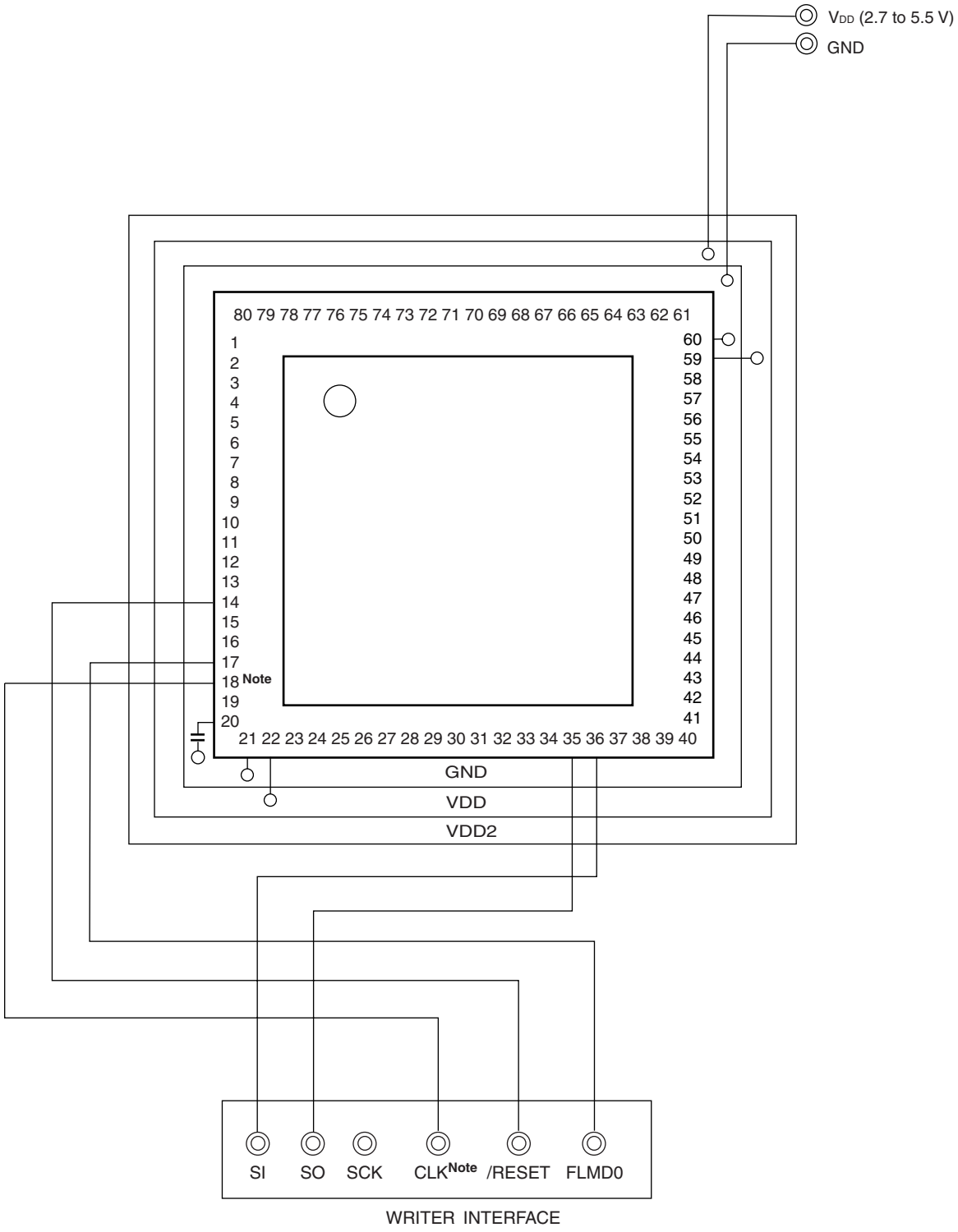
(a) Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 0 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

(b) In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, using the LVIF flag, and clear the LVIIF flag to 0.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 28-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART6) Mode



Note The above figure illustrates an example of wiring when using the clock output from the PG-FP5 or FL-PR5.

<R> **16-bit ΔΣ type A/D Converter Characteristics (μPD78F049x only)**

(T_A = –40 to +85°C, 2.7 V ≤ AV_{REF} ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Resolution	R_{ES2}				8		16	bit
Sampling clock[Note 1]	f_{VP}	At differential input		3.5 V ≤ AV_{REF} ≤ 5.5 V	0.016		1.25	MHz
				2.7 V ≤ AV_{REF} < 3.5 V	0.016		0.625	MHz
		At single input		2.85 V ≤ AV_{REF} ≤ 5.5 V	0.016		0.625	MHz
				2.7 V ≤ AV_{REF} < 2.85 V	0.016		0.525	MHz
Integral non-linearity error (relative accuracy)	I_{LE2}	At differential input[Note 2]	14-bit resolution[Note 3]	AV_{REF} = 5.0 V		±1.0		LSB
				3.5 V ≤ AV_{REF} ≤ 5.5 V		±1.7		LSB
				2.7 V ≤ AV_{REF} < 3.5 V		±2.6		LSB
		At single input[Note 2]	12-bit resolution[Note 3]			±2.8		LSB
Differential non-linearity error (relative accuracy)	D_{LE2}	At differential input[Note 2]	14-bit resolution[Note 3]	AV_{REF} = 5.0 V		±1.0		LSB
				3.5 V ≤ AV_{REF} ≤ 5.5 V		±1.7		LSB
				2.7 V ≤ AV_{REF} < 3.5 V		±2.6		LSB
		At single input[Note 2]	12-bit resolution[Note 3]			±2.8		LSB
Offset	EOS	At differential input				±0.032		%FSR
		At single input				±0.16		%FSR
Gain error	GE	At differential input				±0.09		%
		At single input				±0.1		%
Reference voltage	REF+					AV_{REF}		V
	REF–					AV_{SS}		V
Analog input voltage	V_{AIN2}	In high-accuracy mode OFF			0		REF+	V
		In high-accuracy mode ON			0.1REF+		0.9REF+	V

Notes 1. The conversion time can be calculated by using the following expression, based on the sampling clock (f_{VP}) and set resolution (N bits).

Conversion time = 2^N / f_{VP}

2. These values apply when the high-accuracy mode is set to be on during differential input, or when the high-accuracy mode is set to be off during single input.

3. The characteristics of resolutions (N bits) other than those stated as conditions in the integral linearity error (I_{LE2}) and differential linearity error (D_{LE2}) columns can be calculated by using the following expressions.

- During differential input

  I_{LE2} in N-bit resolution = I_{LE2} in 14-bit resolution × $2^{(N-14)}$

  D_{LE2} in N-bit resolution = D_{LE2} in 14-bit resolution × $2^{(N-14)}$

- During single input

  I_{LE2} in N-bit resolution = I_{LE2} in 12-bit resolution × $2^{(N-12)}$

  D_{LE2} in N-bit resolution = D_{LE2} in 12-bit resolution × $2^{(N-12)}$

Remark In the 16-bit ΔΣ type A/D converter characteristics, the approximation line is defined by the least-squares method.