Renesas Electronics America Inc - UPD703103AGJ-UEN-A 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	Obsolete
Core Processor	V850E1
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	-
Data Converters	A/D 8x10b
Oscillator Type	-
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-FLQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd703103agj-uen-a

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

16.6.3 Selection of communication mode 540
16.6.4 Communication commands 541
16.7 Flash Memory Programming by Self-Programming 542
16.7.1 Outline of self-programming 542
16.7.2 Self-programming function 543
16.7.3 Outline of self-programming interface 543
16.7.4 Hardware environment 544
16.7.5 Software environment 546
16.7.6 Self-programming function number 547
16.7.7 Calling parameters 548
16.7.8 Contents of RAM parameters 549
16.7.9 Errors during self-programming 550
16.7.10 Flash information 550
16.7.11 Area number 551
16.7.12 Flash programming mode control register (FLPMC) 552
16.7.13 Calling device internal processing 554
16.7.14 Erasing flash memory flow 557
16.7.15 Successive writing flow 558
16.7.16 Internal verify flow 559
16.7.17 Acquiring flash information flow 560
16.7.18 Self-programming library 561
16.8 How to Distinguish Flash Memory and Mask ROM Versions 563

CHAPTER 17 ELECTRICAL SPECIFICATIONS 564
17.1 Normal Operation Mode 564
17.2 Flash Memory Programming Mode (μPD70F3107A and 70F3107A(A) Only) 620

CHAPTER 18 PACKAGE DRAWINGS 623

CHAPTER 19 RECOMMENDED SOLDERING CONDITIONS 625

APPENDIX A NOTES ON TARGET SYSTEM DESIGN 629

APPENDIX B CAUTIONS 632
B.1 Restriction on Page ROM Access 632
B.1.1 Description 632
B.1.2 Countermeasures 633

APPENDIX C REGISTER INDEX 634

APPENDIX D INSTRUCTION SET LIST 642
D.1 Conventions 642
D.2 Instruction Set (in Alphabetical Order) 645

APPENDIX E REVISION HISTORY 652

CHAPTER 2 PIN FUNCTIONS

The names and functions of the pins in the V850E/MA1 are listed below. These pins can be divided into port pins and non-port pins according to their functions.

2.1 List of Pin Functions

(1) Port pins

(1/3)

Pin Name	I/O	Function	Alternate Function
P00	I/O	Port 0 8-bit I/O port Input/output can be specified in 1-bit units.	PWM0
P01			TI000/INTP000
P02			INTP001
P03			TO00
P04			$\overline{\text{DMARQ0}}$/$\overline{\text{INTP100}}$
P05			$\overline{\text{DMARQ1}}$/$\overline{\text{INTP101}}$
P06			$\overline{\text{DMARQ2}}$/$\overline{\text{INTP102}}$
P07			$\overline{\text{DMARQ3}}$/$\overline{\text{INTP103}}$
P10	I/O	Port 1 4-bit I/O port Input/output can be specified in 1-bit units.	PWM1
P11			INTP010/TI010
P12			INTP011
P13			TO01
P20	Input	Port 2 P20 is an input port dedicated to checking the NMI input status. If a valid edge is input, it operates as an NMI input. P21 to P27 are a 7-bit I/O port. Input/output can be specified in 1-bit units.	NMI
P21	I/O		INTP020/TI020
P22			INTP021
P23			TO02
P24			$\overline{\text{TC0}}$/$\overline{\text{INTP110}}$
P25			$\overline{\text{TC1}}$/$\overline{\text{INTP111}}$
P26			$\overline{\text{TC2}}$/$\overline{\text{INTP112}}$
P27			$\overline{\text{TC3}}$/$\overline{\text{INTP113}}$
P30	I/O	Port 3 8-bit I/O port Input/output can be specified in 1-bit units.	SO2/$\overline{\text{INTP130}}$
P31			SI2/$\overline{\text{INTP131}}$
P32			$\overline{\text{SCK2}}$/$\overline{\text{INTP132}}$
P33			TXD2/$\overline{\text{INTP133}}$
P34			RXD2/$\overline{\text{INTP120}}$
P35			$\overline{\text{INTP121}}$
P36			$\overline{\text{INTP122}}$
P37			ADTRG/$\overline{\text{INTP123}}$

4.9 Bus Priority Order

There are five external bus cycles: bus hold, instruction fetch, operand data access, DMA cycle, and refresh cycle.

In order of priority, bus hold is the highest, followed by the refresh cycle, DMA cycle, operand data access, and instruction fetch, in that order.

An instruction fetch may be inserted between a read access and write access during a read modify write access. Also, an instruction fetch may be inserted between bus accesses when the CPU bus is locked.

Table 4-2. Bus Priority Order

Priority Order	External Bus Cycle	Bus Master
High ↕ Low	Bus hold	External device
	Refresh cycle	DRAM controller
	DMA cycle	DMA controller
	Operand data access	CPU
	Instruction fetch	CPU

4.10 Boundary Operation Conditions

4.10.1 Program space

(1) Branching to the on-chip peripheral I/O area or successive fetches from the internal RAM area to the on-chip peripheral I/O area are prohibited. If the above is performed (branching or successive fetch), undefined data is fetched, and fetching from the external memory is not performed.

(2) If a branch instruction exists at the upper limit of the internal RAM area, a prefetch operation (invalid fetch) that straddles over the on-chip peripheral I/O area does not occur.

4.10.2 Data space

The V850E/MA1 is provided with an address misalign function.

Through this function, regardless of the data format (word or halfword), data can be allocated to all addresses. However, in the case of word data and halfword data, if the data is not subject to boundary alignment, the bus cycle will be generated at least 2 times and bus efficiency will drop.

(1) In the case of halfword-length data access

When the address's LSB is 1, a byte-length bus cycle will be generated 2 times.

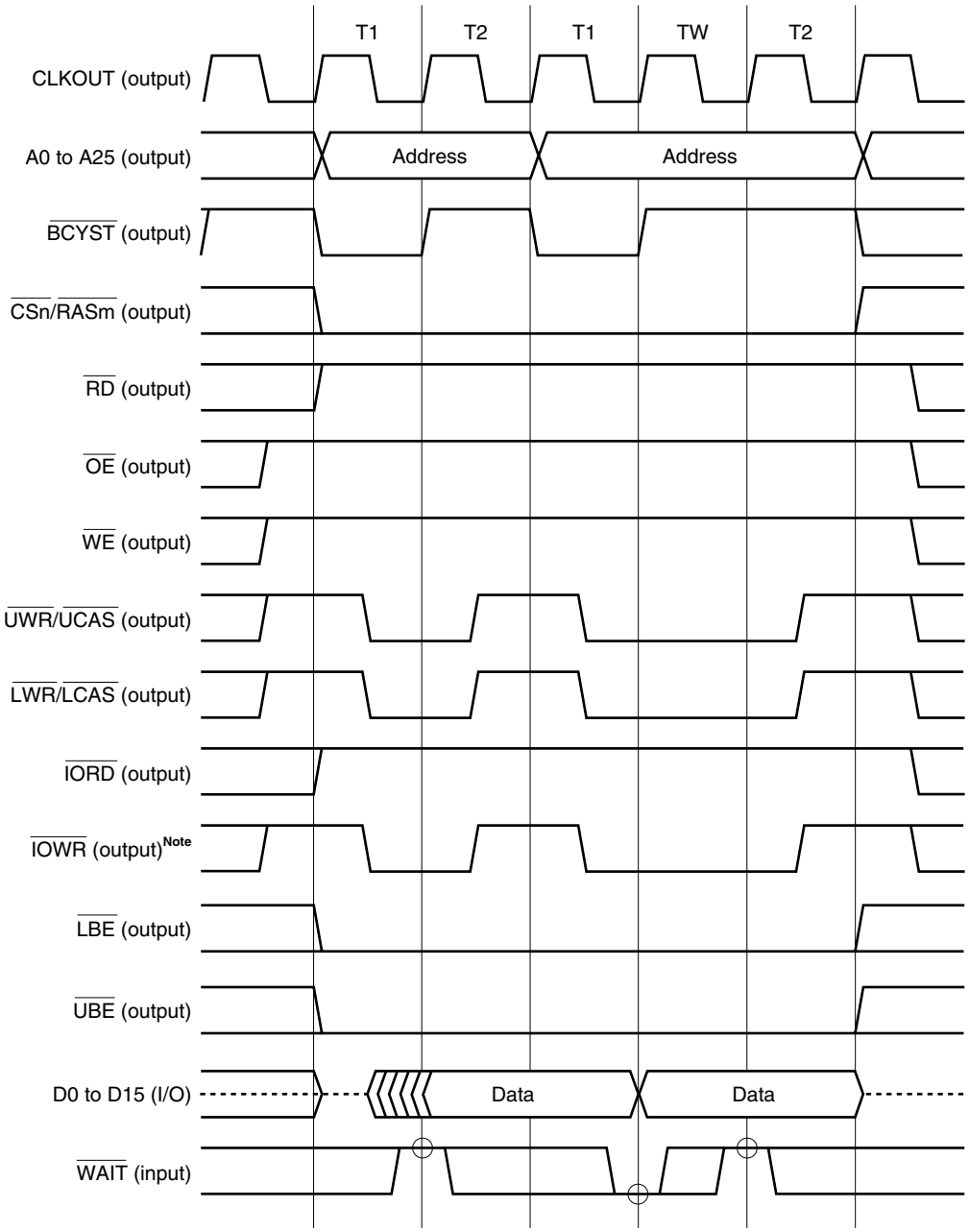
(2) In the case of word-length data access

(a) When the address's LSB is 1, bus cycles will be generated in the order of byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle.

(b) When the address's lower 2 bits are 10, a halfword-length bus cycle will be generated 2 times.

Figure 5-2. SRAM, External ROM, External I/O Access Timing (3/6)

(c) When written



<R>

Note When the IOEN bit of the BCP register is set to 1.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

3. n = 0 to 7, m = 1, 3, 4, 6

6.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)

These registers are used to set the DMA destination address (28 bits) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DDAnH and DDAnL.

Also, since these registers are configured as 2-stage FIFO buffer registers consisting of a master register and a slave register, a new transfer destination address for DMA transfer can be specified during DMA transfer. (Refer to **6.9 Next Address Setting Function**.) In this case, the newly set value of the DDAn register is transferred to the slave register and becomes valid only when DMA transfer has been completed normally and the TCn bit of the DCHCn register is set to 1, or when the INITn bit of the DCHCn register is set to 1 (n = 0 to 3).

When flyby transfer is specified with bit TTYP of DMA addressing control register n (DADCn), regardless of the transfer direction, the setting of DMA destination address register n (DDAn) is ignored (n = 0 to 3).

(1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)

These registers can be read/written in 16-bit units.

Be sure to clear bits 14 to 12 to 0. If they are set to 1, the operation is not guaranteed.

Cautions 1. When setting an address of an on-chip peripheral I/O register for the destination address, be sure to specify an address between FFFF000H and FFFFFFFH. An address of the on-chip peripheral I/O register image (3FFF000H to 3FFFFFFH) must not be specified.

2. Do not set the DDAnH register while DMA is suspended.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DDA0H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF086H	Undefined
DDA1H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF08EH	Undefined
DDA2H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF096H	Undefined
DDA3H	IR	0	0	0	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF09EH	Undefined

Bit position	Bit name	Function
15	IR	Internal RAM Select Specifies the DMA destination address. 0: External memory, on-chip peripheral I/O 1: Internal RAM
11 to 0	DA27 to DA16	Destination Address Sets the DMA destination address (A27 to A16). During DMA transfer, it stores the next DMA transfer destination address. This setting is ignored during flyby transfer.

6.5.3 Block transfer mode

In the block transfer mode, once transfer starts, the DMAC continues the transfer operation without releasing the bus until a terminal count occurs. No other DMA requests are acknowledged during block transfer.

After the block transfer ends and the DMAC releases the bus, another DMA transfer can be acknowledged. The bus cycle of the CPU is not inserted during block transfer, but bus hold and refresh cycles are inserted in between DMA transfer operations.

The following shows an example of block transfer in which a higher priority DMA request is issued. DMA channels 2 and 3 are in the block transfer mode.

Figure 6-8. Block Transfer Example

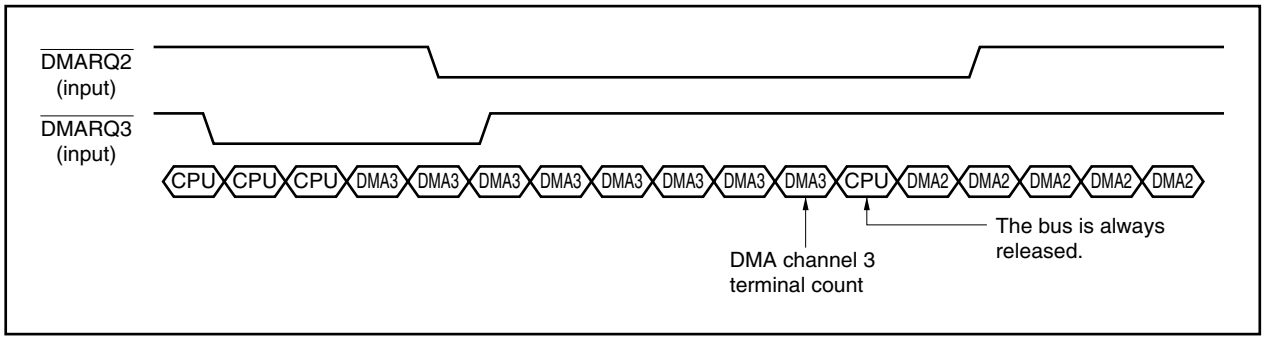
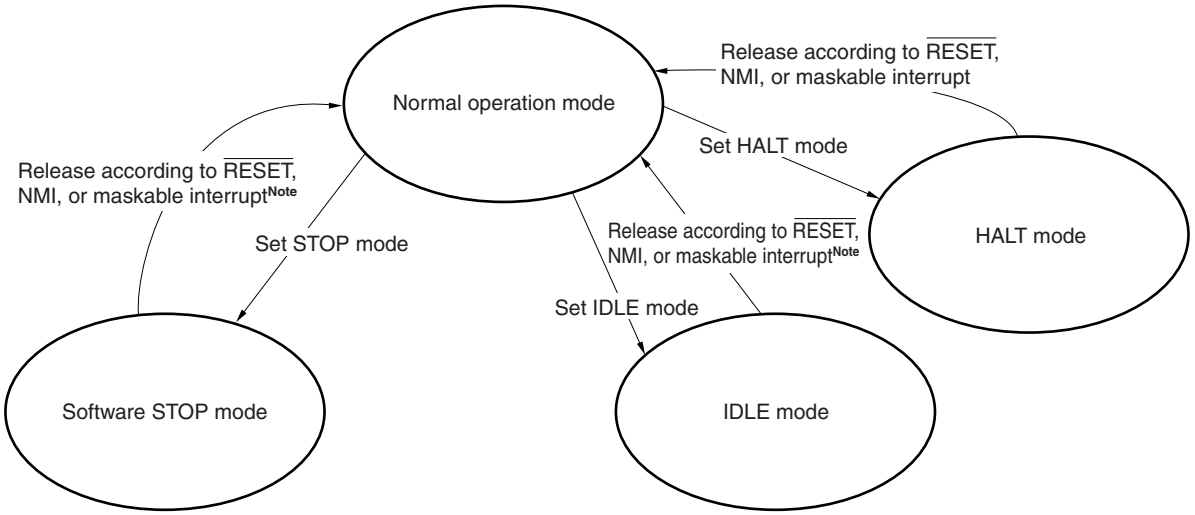


Figure 9-1 shows the operation of the clock generator in normal operation mode, HALT mode, IDLE mode, and software STOP mode.

An effective low power consumption system can be realized by combining these modes and switching modes according to the required use.

Figure 9-1. Power-Save Mode State Transition Diagram



Note INTP1nn (n = 0 to 3)
When level detection is specified for the $\overline{\text{INTP1nn}}$ pin, software STOP mode and IDLE mode cannot be released.

Table 9-1. Clock Generator Operation Using Power-Save Control

Clock Source		Power-Save Mode	Oscillator	PLL Synthesizer	Clock Supply to Peripheral I/O	Clock Supply to CPU
PLL mode	Oscillation with resonator	Normal operation	√	√	√	√
		HALT mode	√	√	√	–
		IDLE mode	√	√	–	–
		Software STOP mode	–	–	–	–
	External clock	Normal operation	–	√	√	√
		HALT mode	–	√	√	–
		IDLE mode	–	√	–	–
		Software STOP mode	–	–	–	–
Direct mode	External clock	Normal operation	–	–	√	√
		HALT mode	–	–	√	–
		IDLE mode	–	–	–	–
		Software STOP mode	–	–	–	–

Remark √: Operating
–: Stopped

(2) Release of software STOP mode

Software STOP mode is released by a non-maskable interrupt request, an unmasked maskable interrupt request (INTP1nm), or $\overline{\text{RESET}}$ pin input. Also, to release software STOP mode when PLL mode (CKSEL pin = low level) and resonator connection mode (CESEL bit of CKC register = 0) are used, the oscillator's oscillation stabilization time must be secured (n = 0 to 3, m = 0 to 3).

Moreover, the oscillation stabilization time must be secured even when an external clock is connected (CESEL bit = 1). See **9.4 PLL Lockup** for details.

(a) Release according to a non-maskable interrupt request or an unmasked maskable interrupt request

The software STOP mode can be released by an interrupt request only when it has been set with the INTM and NMIM bits of the PCS register cleared to 0. The IDLE mode cannot be released if it is specified that the level of the $\overline{\text{INTP1nm}}$ pin is detected.

Software STOP mode is released by a non-maskable interrupt request or by an unmasked maskable interrupt request (INTP1nm) regardless of the priority (n = 0 to 3, m = 0 to 3). The operation after release is as follows.

<R> **Caution When the NMIM and INTM bits of the PSC register = 1, the software STOP mode cannot be released by the non-maskable interrupt request signal and unmasked maskable interrupt request signal (INTPnm) (n = 0 to 3, m = 0 to 3).**

Table 9-7. Operation After Software STOP Mode Is Released by Interrupt Request

Release Source	Enable Interrupt (EI) Status	Disable Interrupt (DI) Status
Non-maskable interrupt request	Branch to handler address	
Maskable interrupt request	Branch to handler address or execute next instruction	Execute next instruction

If the system is set to software STOP mode during a maskable interrupt servicing routine, operation will differ as follows.

(i) If an interrupt request is generated with a lower priority than that of the interrupt request that is currently being serviced, software STOP mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request is held pending.

(ii) If an interrupt request (including non-maskable interrupt requests) is generated with a higher priority than that of the interrupt request that is currently being serviced, software STOP mode is released and the newly generated interrupt request is acknowledged.

If the system is set to software STOP mode during an NMI servicing routine, software STOP mode is released, but the interrupt is not acknowledged (interrupt is held pending).

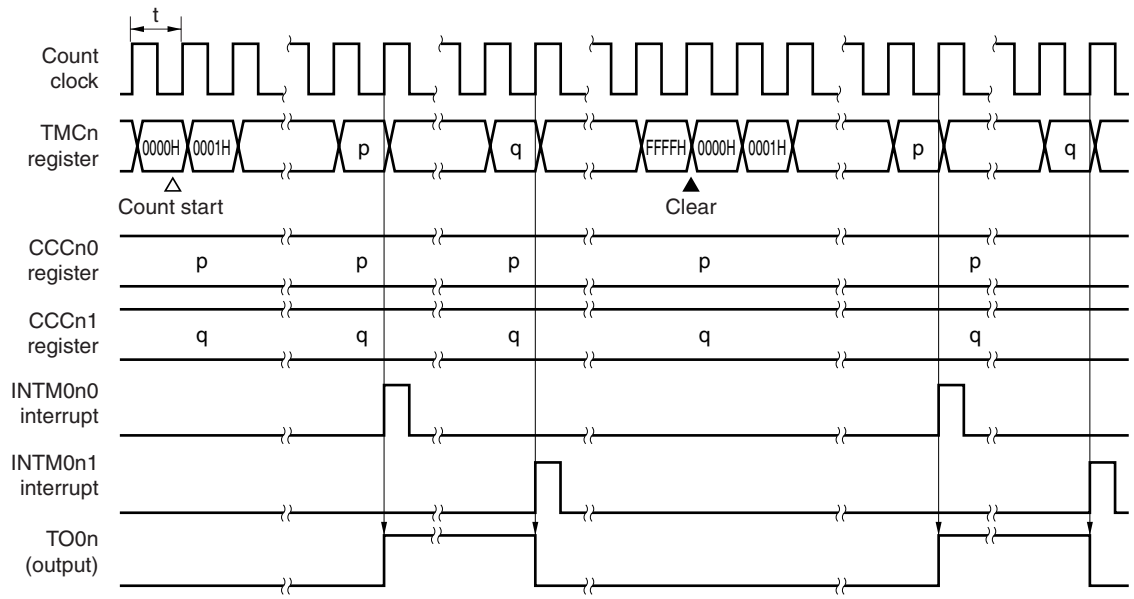
Interrupt servicing that is started when software STOP mode is released by NMI pin input is handled in the same way as normal NMI interrupt servicing that occurs during an emergency (because the NMI interrupt handler address is unique). Therefore, when a program must be able to distinguish between these two situations, a software status must be prepared in advance and that status must be set before setting the PSMR register using a store instruction or a bit manipulation instruction.

By checking for this status during NMI interrupt servicing, an ordinary NMI can be distinguished from the servicing that is started when software STOP mode is released by NMI pin input.

(b) Release according to $\overline{\text{RESET}}$ pin input

This is the same as a normal reset operation.

Figure 10-10. PWM Output Timing Example



Remarks 1. p: Setting value of CCCn0 register (0000H to FFFFH)

q: Setting value of CCCn1 register (0000H to FFFFH)

$p \neq q$

t: Count clock cycle

PWM cycle = 65,536 × t

$$\text{Duty} = \frac{q - p}{65{,}536}$$

2. In this example, the active level of the TO0n output is set to the high level.

3. n = 0 to 3

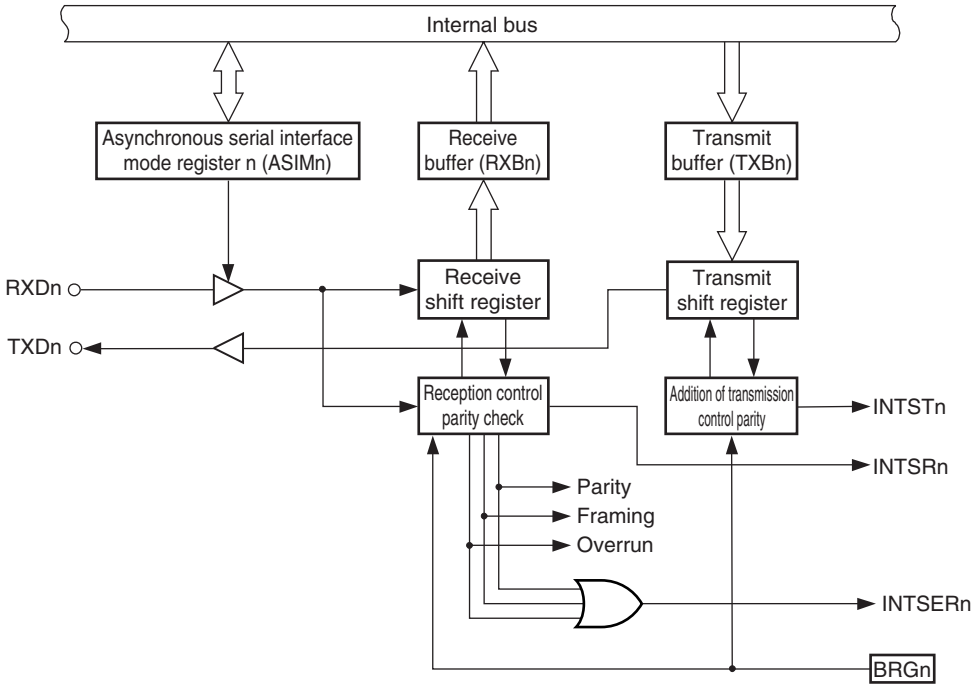
(8) Transmit buffer (TXBn)

TXBn is an 8-bit buffer for transmit data. A transmit operation is started by writing transmit data to TXBn.

(9) Addition of transmission control parity

Transmit operations are controlled by adding a start bit, parity bit, or stop bit to the data that is written to the TXBn register, according to the contents that were set in the ASIMn register.

Figure 11-1. Asynchronous Serial Interface Block Diagram



Remarks 1. For the configuration of the baud rate generator, refer to **Figure 11-12**.

2. n = 0 to 2

(3/3)

Bit position	Bit name	Function
4, 3	PSn1, PSn0 (n = 0 to 2)	• 0 parity During transmission, the parity bit is cleared (0) regardless of the transmit data. During reception, no parity error is generated because no parity bit is checked. • No parity No parity bit is added to transmit data. During reception, the receive data is considered to have no parity bit. No parityerror is generated because there is no parity bit.
2	CLn (n = 0 to 2)	Character Length Specifies the character length of the transmit/receive data. 0: 7 bits 1: 8 bits **Caution To overwrite the CLn bit, first clear (0) the TXEn and RXEn bits.**
1	SLn (n = 0 to 2)	Stop Bit Length Specifies the stop bit length of the transmit data. 0: 1 bit 1: 2 bits **Cautions 1. To overwrite the SLn bit, first clear (0) the TXEn bit.** **2. Since reception always operates by using a single stop bit length, the SLn bit setting does not affect receive operations.**
0	ISRMn (n = 0 to 2)	Interrupt Serial Receive Mode Specifies whether the generation of reception completion interrupt requests when an error occurs is enable or disabled. 0: A reception error interrupt request (INTSERn) is generated when an error occurs. In this case, no reception completion interrupt request (INTSRn) is generated. 1: A reception completion interrupt request (INTSRn) is generated when an error occurs. In this case, no reception error interrupt request (INTSERn) is generated. **Caution To overwrite the ISRMn bit, first clear (0) the RXEn bit.**

(2) Transmit operation

When UARTCAEn is set to 1 in the ASIMn register, a high level is output to the TXDn pin.

Then, when TXEn is set to 1 in the ASIMn register, transmission is enabled, and the transmit operation is started by writing transmit data to transmit buffer register n (TXBn) (n = 0 to 2).

(a) Transmission enabled state

This state is set by the TXEn bit in the ASIMn register (n = 0 to 2).

- TXEn = 1: Transmission enabled state
- TXEn = 0: Transmission disabled state

However, when the transmission enabled state is set, to use UART0 and UART1, which share pins with clocked serial interfaces 0 and 1 (CSI0 and CSI1), the CSICAEn bit of clocked serial interface mode registers 0 and 1 (CSIM0 and CSIM1) should be set to 0.

Since UARTn does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in a reception enabled state.

(b) Starting a transmit operation

In the transmission enabled state, a transmit operation is started by writing transmit data to transmit buffer register n (TXBn). When a transmit operation is started, the data in TXBn is transferred to transmit shift register n. Then, transmit shift register n outputs data to the TXDn pin sequentially beginning with the LSB (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically (n = 0 to 2).

(c) Transmission interrupt request

When the transmit shift register becomes empty, a transmission completion interrupt request (INTSTn) is generated. The timing for generating the INTSTn signal differs according to the specification of the stop bit length. The INTSTn signal is generated at the same time that the last stop bit is output (n = 0 to 2).

If the data to be transmitted next has not been written to the TXBn register, the transmit operation is suspended.

Caution Normally, when transmit shift register n becomes empty, a transmission completion interrupt (INTSTn) is generated. However, no transmission completion interrupt (INTSTn) is generated if transmit shift register n becomes empty due to the input of a $\overline{\text{RESET}}$.

(b) Baud rate generator control registers 0 to 2 (BRGC0 to BRGC2)

The BRGCn register is an 8-bit register that controls the baud rate (serial transfer speed) of UARTn.

These registers can be read or written in 8-bit units.

Caution If the BRGn7 to BRGn0 bits are to be overwritten, TXEn and RXEn should be set to 0 in the ASIMn register first (n = 0 to 2).

	7	6	5	4	3	2	1	0	Address	After reset
BRGC0	MDL07	MDL06	MDL05	MDL04	MDL03	MDL02	MDL01	MDL00	FFFFFA07H	FFH
BRGC1	MDL17	MDL16	MDL15	MDL14	MDL13	MDL12	MDL11	MDL10	FFFFFA17H	FFH
BRGC2	MDL27	MDL26	MDL25	MDL24	MDL23	MDL22	MDL21	MDL20	FFFFFA27H	FFH

Bit position	Bit name	Function
7 to 0	BRGn7 to BRGn0 (n = 0 to 2)	Specifies the 8-bit counter's divisor value.

BRGn7	BRGn6	BRGn5	BRGn4	BRGn3	BRGn2	BRGn1	BRGn0	Divisor value (k)	Serial clock
0	0	0	0	0	x	x	x	–	Setting prohibited
0	0	0	0	1	0	0	0	8	$f_{UCLK}/8$
0	0	0	0	1	0	0	1	9	$f_{UCLK}/9$
0	0	0	0	1	0	1	0	10	$f_{UCLK}/10$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	0	1	0	250	$f_{UCLK}/250$
1	1	1	1	1	0	1	1	251	$f_{UCLK}/251$
1	1	1	1	1	1	0	0	252	$f_{UCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{UCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{UCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{UCLK}/255$

Remarks 1. f_{UCLK}: Frequency [Hz] of basic clock according to TPSn3 to TPSn0 bits of CKSRn register.

2. k: Value set according to BRGn7 to BRGn0 bits (k = 8, 9, 10, ..., 255)

3. The baud rate is the output clock for the 8-bit counter divided by 2

4. x: don't care

14.3.13 Port CM

Port CM is a 6-bit I/O port that can be set to the input or output mode in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PCM	–	–	PCM5	PCM4	PCM3	PCM2	PCM1	PCM0	FFFFF00CH	Undefined

Bit position	Bit name	Function
5 to 0	PCMn (n = 5 to 0)	Port CM I/O port

In addition to their function as port pins, in the control mode, the port CM pins operate as the wait insertion signal input, internal system clock output/bus clock output, bus hold control signal output, and refresh request signal output from DRAM.

(1) Operation in control mode

Port		Alternate Function Pin Name	Remark	Block Type
Port CM	PCM0	$\overline{\text{WAIT}}$[Note]	Wait insertion signal input	D
	PCM1	CLKOUT/BUSCLK	Internal system clock output/bus clock output	K
	PCM2	$\overline{\text{HLDAK}}$	Bus hold acknowledge signal output	J
	PCM3	$\overline{\text{HLDRQ}}$[Note]	Bus hold request signal input	D
	PCM4	$\overline{\text{REFRQ}}$	Refresh request signal output	J
	PCM5	SELFREF[Note]	Self-refresh request signal input	E

Note The default assumption of the $\overline{\text{WAIT}}$, $\overline{\text{HLDRQ}}$, and SELFREF signals is the control mode in ROMless modes 0 and 1, and single-chip mode 1. Fix these pins to the inactive level when they are not used. When these pins are used as port pins, they function in the control mode until they are set in the port mode by the port CM mode control register (PMCCM). Therefore, be sure to set these pins to the inactive level until they are set in the port mode.

(2) I/O mode/control mode setting

The port CM I/O mode setting is performed by the port CM mode register (PMCM), and the control mode setting is performed by the port CM mode control register (PMCCM) and the port CM function control register (PFCCM).

(a) Port CM mode register (PMCM)

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

	7	6	5	4	3	2	1	0	Address	After reset
PMCM	1	1	PMCM5	PMCM4	PMCM3	PMCM2	PMCM1	PMCM0	FFFFF02CH	FFH

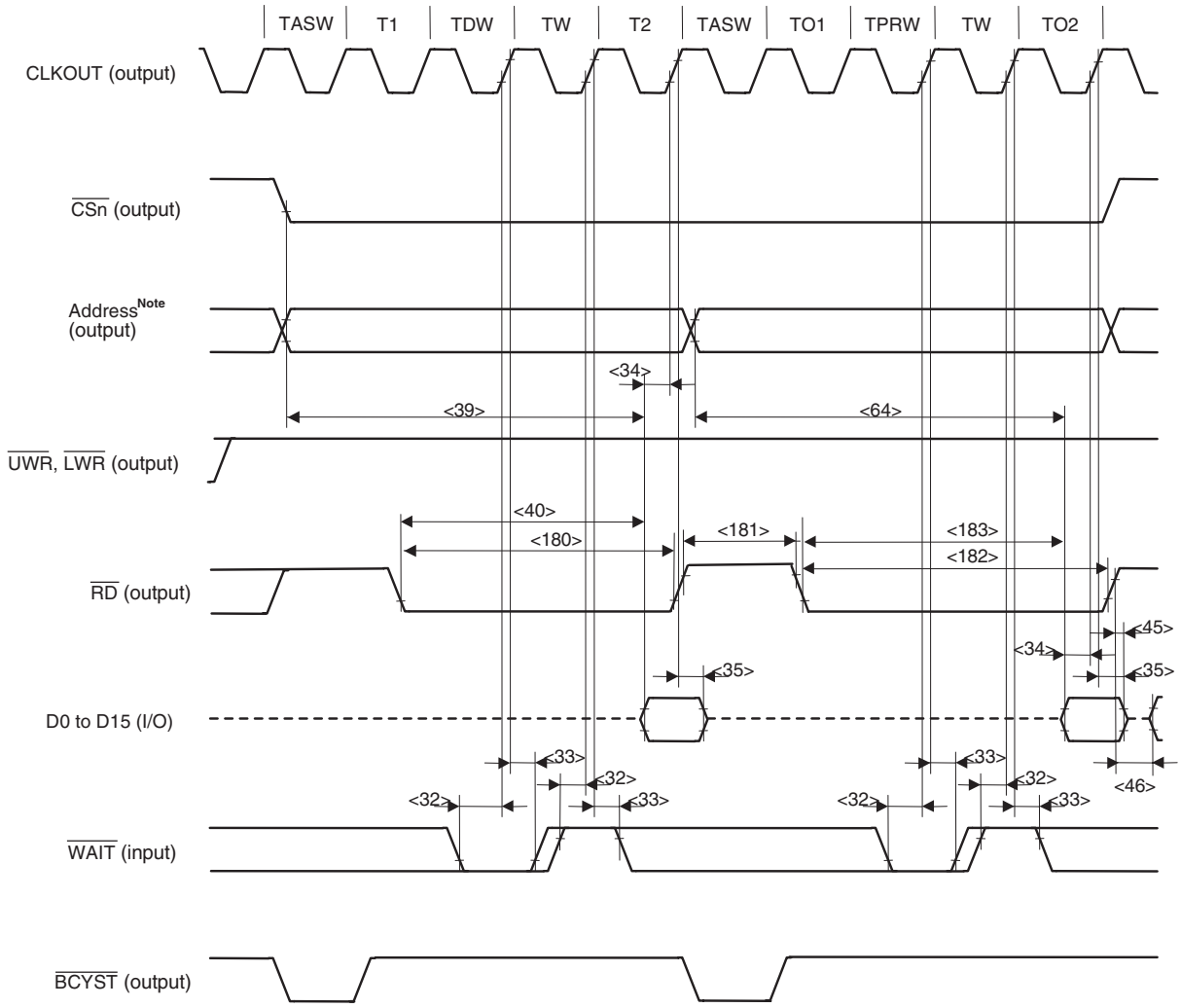
Bit position	Bit name	Function
5 to 0	PMCMn (n = 5 to 0)	Port Mode Specifies input/output mode for PCMn pin. 0: Output mode (output buffer on) 1: Input mode (output buffer off)

(5) Internal manipulation setup parameter

If the self-programming mode is switched to the normal operation mode, the μPD70F3107A must wait for 100 μs before it accesses the flash memory. In the program example in (4) above, the elapse of this wait time is ensured by setting ISETUP to "130" (@ 50 MHz operation). The total number of execution clocks in this example is 39 clocks (divh instruction (35 clocks) + add instruction (1 clock) + jne instruction (3 clocks)). Ensure that a wait time of 100 μs elapses by using the following expression.

39 clocks (total number of execution clocks) × 20 ns (@ 50 MHz operation) × 130 (ISETUP) = 101.4 μs (wait time)

(b) 8-bit bus width (byte access) and 16-bit bus width (byte/halfword access) (2/2)



Note On-page and off-page addresses are as follows.

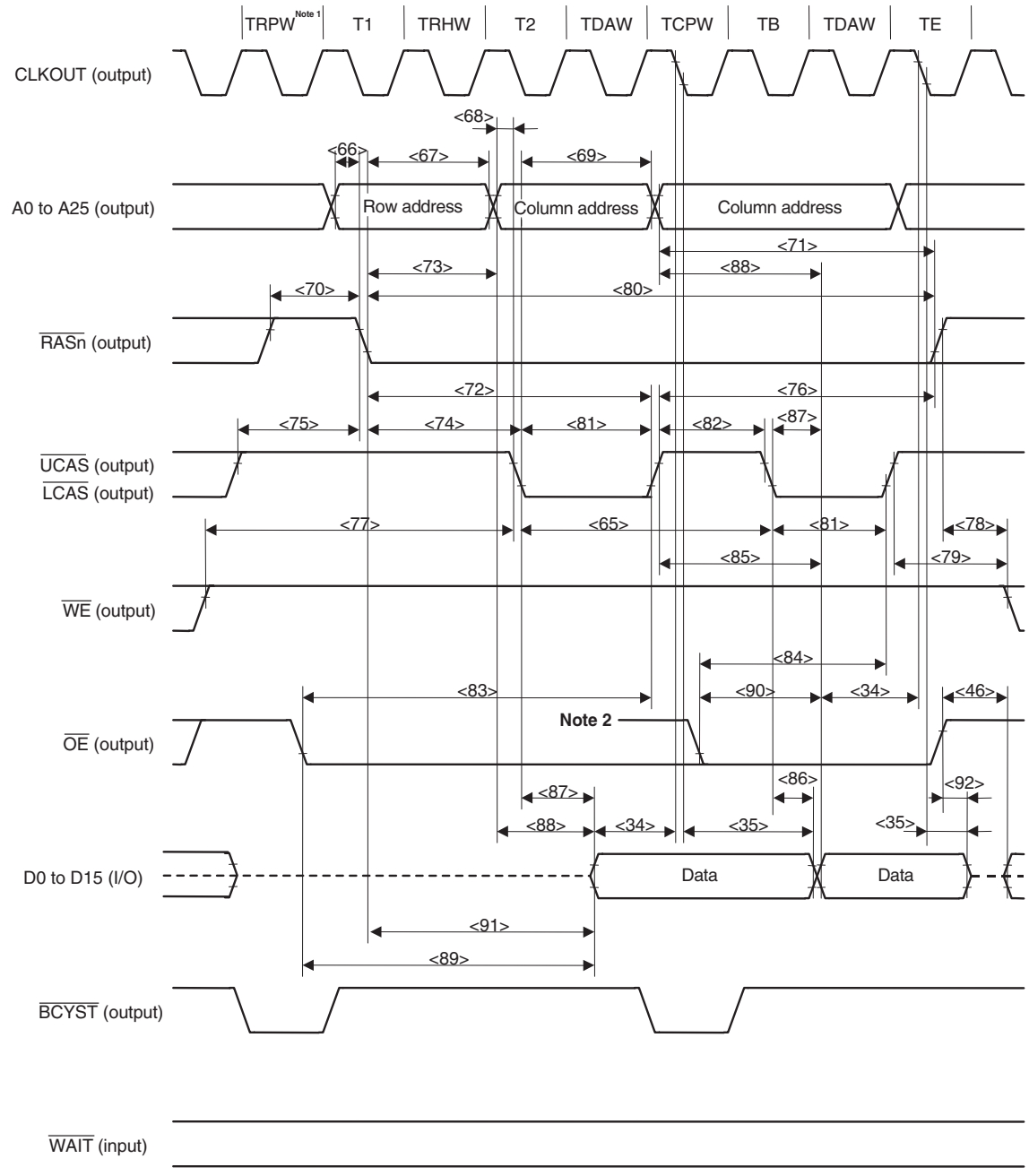
PRC Register				On-Page Address	Off-Page Address
MA6	MA5	MA4	MA3		
0	0	0	0	A0 to A2	A3 to A25
0	0	0	1	A0 to A3	A4 to A25
0	0	1	1	A0 to A4	A5 to A25
0	1	1	1	A0 to A5	A6 to A25
1	1	1	1	A0 to A6	A7 to A25

Remarks 1. This is the timing for the following case.
Wait count based on the DWC0 and DWC1 registers (TDW): 1
Wait count based on the PRC register (TPRW): 1
Wait count based on the ASC register (TASW): 1

2. Broken lines indicate high impedance.

3. n = 0 to 7

(a) Read timing (EDO DRAM) (3/3)



Notes 1. At least one clock is inserted in TRPW.

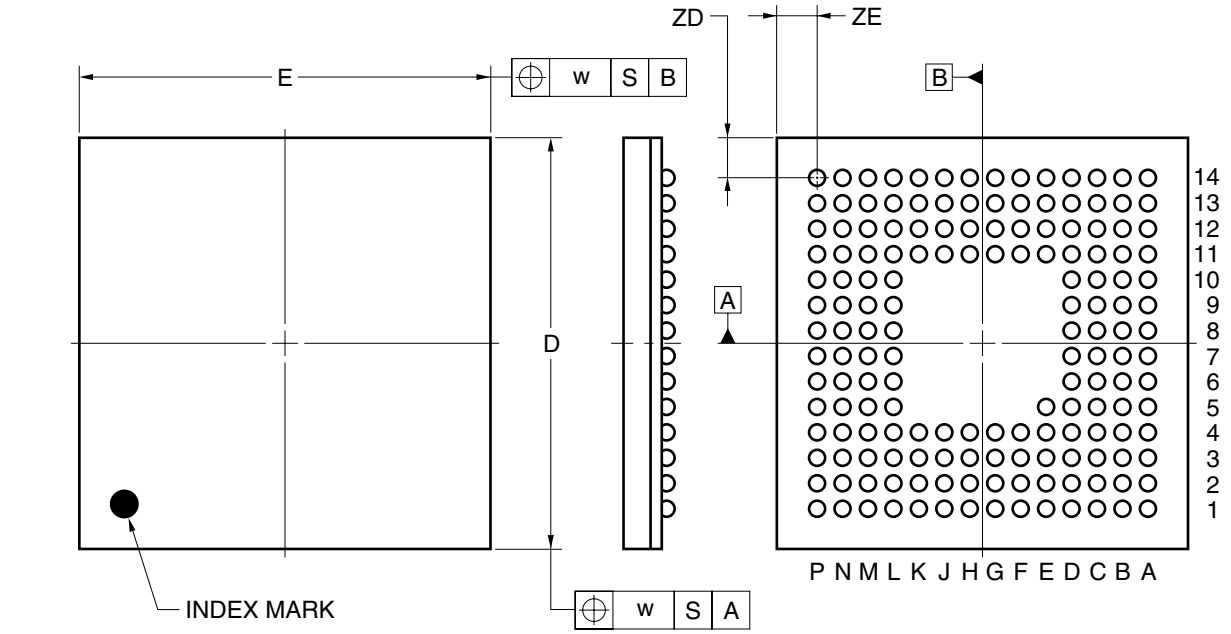
2. During on-page access from other cycles while $\overline{RAS}$ is low level.

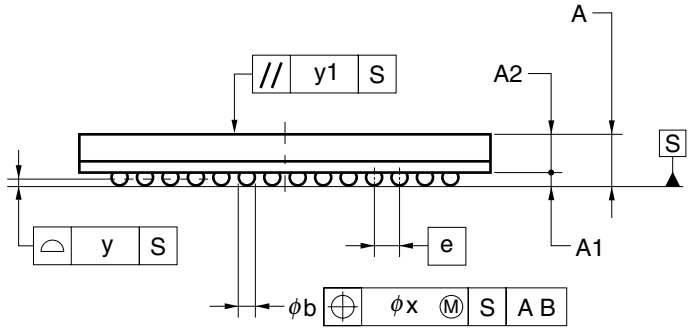
Remarks 1. This is the timing for the following case.
Wait count based on the RPC1n and RPC0n bits of the SCRn register (TRPW): 1
Wait count based on the RHC1n and RHC0n bits of the SCRn register (TRHW): 1
Wait count based on the DAC1n and DAC0n bits of the SCRn register (TDAW): 1
Wait count based on the CPC1n and CPC0n bits of the SCRn register (TCPW): 1

2. Broken lines indicate high impedance.

3. n = 1, 3, 4, 6

161-PIN PLASTIC FBGA (13x13)





ITEM	MILLIMETERS
D	13.00±0.10
E	13.00±0.10
w	0.20
A	1.48±0.10
A1	0.35±0.06
A2	1.13
e	0.80
b	$0.50^{+0.05}_{-0.10}$
x	0.08
y	0.10
y1	0.20
ZD	1.30
ZE	1.30

P161F1-80-EN4-1

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1, reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1, reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2)+(GR[reg1] logically shift left by 1) PC←(PC+2)+(sign-extend (Load-memory (adr, Halfword))) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7:0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15:0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 i i i i i 0000000100000000	EIPC ← PC + 4 (return PC) EIPSW ← PSW ECR.EICC ← exception code (40H to 4FH, 50H to 5FH) PSW.EP ← 1 PSW.ID ← 1 PC ← 00000040H (when vector is 00H to 0FH (exception code: 40H to 4FH)) 00000050H (when vector is 10H to 1FH (exception code: 50H to 5FH))	4	4	4					
TST	reg1, reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3, disp16[reg1]	11bbb111110RRRRR ddddddddddddddddd	adr←GR[reg1]+sign-extend (disp16) Z flag←Not (Load-memory-bit (adr, bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr, reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1, reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16, reg1, reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7:0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15:0))	1	1	1					

Notes 1. ddddddddd: Higher 8 bits of disp9.

2. 4 if there is an instruction that rewrites the contents of PSW immediately before

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. Same operation as when n = 1 if n = 0)

5. RRRRR: other than 00000.

6. The lower halfword data only is valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).