Universe IID/IIB™ Debug and Initialization Notes

8091142_AN005_04

October 15, 2009

6024 Silver Creek Valley Road San Jose, California 95138
Telephone: (408) 284-8200 • FAX: (408) 284-3572
Printed in U.S.A.

GENERAL DISCLAIMER

Integrated Device Technology, Inc. ("IDT") reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance. IDT does not assume responsibility for use of any circuitry described herein other than the circuitry embodied in an IDT product. Disclosure of the information herein does not convey a license or any other right, by implication or otherwise, in any patent, trademark, or other intellectual property right of IDT. IDT products may contain errata which can affect product performance to a minor or immaterial degree. Current characterized errata will be made available upon request. Items identified herein as "reserved" or "undefined" are reserved for future definition. IDT does not assume responsibility for conflicts or incompatibilities arising from the future definition of such items. IDT products have not been designed, tested, or manufactured for use in, and thus are not warranted for, applications where the failure, malfunction, or any inaccuracy in the application carries a risk of death, serious bodily injury, or damage to tangible property. Code examples provided herein by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of such code examples shall be at the user's sole risk.

1. Universe IID/IIB Debug and Initialization Notes

This application note is intended to show common problems that may occur which can cause the Universe II not to operate as expected. This document discusses the following:

Revision History

8091142_AN005_04, Formal, October 2009

This version of the document was rebranded as IDT. It does not include any technical changes.

8091142_AN005_03, Formal, June 2009

Table 1 has been updated.

1.1 Overview

This application note is intended to show possible problems that may cause the Universe II not to operate as expected. This application note is intended to be used as a guide only.

This document is divided into two sections Hardware and Software. Each section outlines various issues within those areas which would cause the Universe II to not function as expected. As a reference, users may also wish to read the application note on Fast Initialization. (8091142_AN003).

1.2 Hardware

The following describes various hardware issues associated with the Universe II.

1.2.1 Power up Options

The Universe II may be automatically configured at power-up to operate in different functional modes. These power-up options allow the Universe II to be set in a particular mode independent of any local intelligence. The Universe II power-up options are listed in Table 1. The majority of the Universe II power-up options are loaded from the VMEbus address and data lines after any PWRRST#. There are two power-up options that are not initiated by PWRRST#. The first of these is PCI bus width (a power-up option required by the PCI bus specification), this is loaded on any RST# event from the REQ64# pin. The second special power-up option is VMEbus SYSCON enabling, required by the VMEbus specification. The SYSCON option is loaded during a SYSRST* event from the BG3IN* signal.

Power up options are described in more detail in more detail in “Power-up Option” on of the Universe II User’s Manual.

Table 1: Pin Level for Specific Power up Modes

Pin [a]	Level [b]	Power up Mode	Register	Field
VA[31]	low high	VMEbus Register Access Disable VMEbus Register Access Enable	VRAI_CTL	EN
VA[30:29]	low, low low, high high, low high, high	VMEbus Address space A16 VMEbus Address space A24 VMEbus Address space A32 reserved	VRAI_CTL	VAS
VA[28:21]		See Table 2: VRAI Base Address Power-up Options	VRAI_BS	BS
VA[20] [c]	low high	PCI Bus Memory Space PCI Bus I/O Space	VCSR_CTL	LAS[0]
VA[19:15]		VMEbus CSR Translation offset	VCSR_TO	TO
VD[30]	low high	DY4 Auto ID Disable DY4 Auto ID Enable	MISC_STAT	DY4AUTO
VD[29] [d]	low high	No effect Initiate VME64 Auto ID sequence	MISC_CTL	V64AUTO
	low high	VME Software Interrupt masked VME Software Interrupt enabled	VINT_EN	SW_INT
	low high	VME Software Interrupt inactive VME Software Interrupt active	VINT_STAT	SW_INT
	low high	VMEbus Software interrupt destination 0 (masked) VMEbus Software interrupt destination VIRQ*[2]	VINT_MAP1	SW_INT

Table 1: Pin Level for Specific Power up Modes

Pin [a]	Level [b]	Power up Mode	Register	Field
VD[28]	low high	Universe II is not in BI-Mode Universe II is in BI-Mode	MISC_CTL	BI
VD[27]	low high	No effect Assert VXSYSFAIL [e]	VCSR_SET	SYSFAIL
VBGIN[3]*	low high	Universe II is VMEbus System Controller Universe II is not VMEbus System Controller	MISC_CTL	SYSCON
VA[1]	low high	PCI_BS0 mapped to I/O, PCI_BS1 mapped to memory PCI_BS0 mapped to memory, PCI_BS1 mapped to I/O	PCI_BS0, PCI_BS1	SPACE
VA[5:2]		PCI Target Image 0 Bound Address Range[31:28] [f]	LSIO_BD	BD
VA[9:6]		PCI Target Image 0 Base Address [31:28] [f]	LSIO_BS	BS
VA[11:10]	low, low low, high high, low high, high	VMEbus Address space A16 VMEbus Address space A24 VMEbus Address space A32 reserved	LSIO_CTL	VAS
VA[12]	low high	Destination is PCI Bus Memory Space Destination is PCI Bus I/O Space	LSIO_CTL	LAS
VA[13]	low high	Image Disabled Image Enabled	LSIO_CTL	EN
VA[14]	low high	PCI Master Disabled PCI Master Enabled	PCI_CSR	BM
REQ64# [g]	low high	64 bit PCI Bus Size 32 bit PCI Bus Size	MISC_STAT	LCLSIZE

a. All power-up options are latched only at the rising-edge of PWRRST#. They are loaded when PWRRST#, SYSRST* and RST# are negated.

b. The default level for all pins in low due to internal pull-downs.

c. The LAS field will enable the PCI_CSR register's MS or IOS field if the EN FIELD of the LSIO_CTL register is set.

d. Pin VA[29] determines whether V64Auto ID will be used and sets up required registers

e. This power-up option is over-ridden if VME64 Auto-ID has been enabled. This option would be used when extensive on-board diagnostics need to be performed before release of SYSFAIL*. After completion of diagnostics, SYSFAIL* may be released through software or through initiation of the VME64 Auto-ID sequence if that mechanism is to be used.

f. The lower bits ([27:12]) are set to 0

g. The PCI Bus Size is loaded on any RST# event, as per the PCI 2.1 Specification

Table 2: VRAI Base Address Power-up Options

VRAI_CTL: VAS	BS[31:24]	BS[23:16]	BS[15:12]
A16	0	0	Power-up Option VA[28:21]
A24	0	Power-up Option VA[28:21]	0
A32	Power-up Option VA[28:21]	0	0

1.2.2 Test Mode

The Universe II has I/O capabilities that are specific to manufacturing test functions. These pins are not required in a non-manufacturing test setting. Table 3 below shows how these pins should be terminated for normal operation.

Test Mode is described in more detail “Auxiliary Test Modes” of the Universe II Manual.

Table 3: Test Mode Pin Settings for Normal Operation

Pin Name	Pin Value
TMODE[2]	V_{SS}
TMODE[1]	V_{SS}
TMODE[0]	V_{SS}
PLL_TESTSEL	V_{SS}
ENID	V_{SS}
PLL_TESTOUT	V_{SS}
VOCTL	V_{SS}

1.2.3 JTAG

The Universe II includes dedicated user-accessible test logic that is fully compatible with the IEEE 1149.1 Standard Test Access Port (TAP) and Boundary Scan Architecture. This standard was developed by the Test Technology Technical Committee of IEEE Computer Society and the Joint Test Action Group (JTAG). The JTAG pins are not required for normal operation, see Table 4 for JTAG Pin Connections for Normal Operating Mode.

JTAG is described in more detail “JTAG Support” of the Universe II Manual.

Table 4: JTAG Pin Connections for Normal Operating Mode

Pin Name	Pin Value
TRST#	V_{SS}
TCK	V_{dd}
TDI	V_{dd}
TMS	V_{dd}
TDO	V_{SS}

1.2.4 Clocks

1.2.4.1 CLK64

CLK64 is a 64 MHz clock that is required by the Universe II in order to synchronize internal Universe II state machines and to produce the VMEbus system clock (VSYSCLK) when the Universe II is system controller (SYSCON). This clock is specified to have a minimum 60-40 duty cycle (±20%) with a maximum rise time of 5 ns. Using a different frequency is not recommended as it will alter various internal timers and change some VME timing.

1.2.4.2 SYSCLK

The Universe II provides a 16 MHz VMEbus system clock (SYSCLK) derived from CLK64 when configured as SYSCON.

1.2.4.3 LCLK

The PCI Clock provides timing for all transactions on the PCI bus. PCI signals are sampled on the rising edge of LCLK, and all timing parameters are defined relative to this signal. The PCI clock frequency of the Universe II must be between 25 and 33MHz. Lower frequencies will result in invalid PCI and VME timing.

1.2.5 Resets

1.2.5.1 Overview of Reset Support

The Universe II provides a number of pins and registers for reset support. Pin support is summarized in Table 5. Register support is summarized in Table 7 Software Initialization Summary.

Table 5: Hardware Reset Mechanisms

Interface and Direction	Pin Name	Long Name	Effects [a]
VMEbus Input	VRSYSRST#	VMEbus Reset Input	Asserts LRST# on the local bus, resets the Universe II, and reconfigures power-up options.
VMEbus Output	VXSYSRST	VMEbus System Reset	Universe II output for SYSRST* (resets the VMEbus)
PCI Input	PWRRST#	Power-up Reset	Resets the Universe II and reconfigures power-up options.
	RST#	PCI Reset Input	Resets the Universe II from the PCI bus.
	VME_RESET#	VMEbus Reset Initiator	Causes Universe II to assert VXSYSRST
PCI Output	LRST#	PCI Bus Reset Output	Resets PCI resources
JTAG Input	TRST#	JTAG Test Reset	Provides asynchronous initialization of the TAP controller in the Universe II.

a. A more detailed account of the effects of reset signals is provided in “Reset Implementation Cautions” in the Universe II User’s Manual

The Universe II can only be reset through hardware, software can only cause the Universe II to assert its reset outputs. In order to reset the Universe II through software, the Universe II reset outputs must be connected to the Universe II reset inputs. For example, the SW_LRST bit in the MISC_CTL register, which asserts the LRST# output, which will not reset the Universe II itself unless LRST# is looped back to RST#. As described in “Reset Implementation Cautions” of the users manual, there are potential loopback configurations resulting in permanent reset.

Table 6: Software Reset Mechanism

Register	Name	Type	Function
MISC_CTL	SW_LRST	W	Software PCI Reset 0=No effect, 1=Initiate LRST# A read always returns 0.
	SW_SYSRST	W	Software VMEbus SYSRESET 0=No effect, 1=Initiate SYSRST* A read always returns 0.
VCSR_SET	RESET	R/W	Board Reset Reads: 0=LRST# not asserted, 1=LRST# asserted Writes: 0=no effect, 1=assert LRST#
	SYSFAIL	R/W	VMEbus SYSFAIL Reads: 0=VXSYSFAIL not asserted, 1=VXSYSFAIL asserted Writes:0=no effect, 1=assert VXSYSFAIL
VCSR_CLR	RESET	R/W	Board Reset Reads: 0=LRST# not asserted, 1=LRST# asserted Writes: 0=no effect, 1=negate LRST#
	SYSFAIL	R/W	VMEbus SYSFAIL Reads: 0=VXSYSFAIL not asserted, 1=VXSYSFAIL asserted Writes:0=no effect, 1=negate VXSYSFAIL

1.2.5.2 Universe II Reset Circuitry

For Information on the Reset Circuitry and the effects of various reset events of the Universe II, see the *Universe II User’s Manual*.

1.3 Software and Registers

The following describes various software and register issues that may cause the Universe II not to operate as expected.

1.3.1 Software Initialization Summary

The following describes various software issues for the programmer during initialization and while debugging.

Table 7: Software Initialization Summary

Functional Group	Register	Bit	Description
PCI memory Map	PCI_CSR	BM	Master Enable
		MS	Target Memory Enable
		IOS	Target IO Enable
	PCI_BS0	BS	Base Address
		SPACE	PCI Bus Address Space
	PCI_BS1	BS	Base Address
		SPACE	PCI Bus Address Space
VME memory Map	VCSR_BS	-	VMEbus CSR Base Address
PCI Slave Images	LSx_CTL	EN	Image Enable
	LSx_BS	-	Base Address
	LSx_BD	-	Bound Address
	LSx_TO	-	Translation Offset
VME Slave Images	VSIx_CTL	EN	Image Enable
	VSIx_BS	-	Base Address
	VSIx_BD	-	Bound Address
	VSIx_TO	-	Translation Offset
VME Requests	MAST_CTL	VRL	VMEbus Request Level
		VRM	VMEbus Request Mode
		VREL	VMEbus Release Mode

Table 7: Software Initialization Summary

Functional Group	Register	Bit	Description
Interrupts	LINT_EN	-	PCI Interrupt Enables
	LINT_STAT	-	PCI Interrupt Status
	LINT_MAP0	-	PCI Interrupt Map
	LINT_MAP1	-	PCI Interrupt Map
	LINT_MAP2	-	PCI Interrupt Map
	VINT_EN	-	VMEbus Interrupt Enables
	VINT_STAT	-	VMEbus Interrupt Status
	VINT_MAP0	-	VME Interrupt Map
	VINT_MAP1	-	VME Interrupt Map
	VINT_MAP2	-	VME Interrupt Map
	VINT_STAT	-	VME Interrupt Status
	STATID	-	VME STATUS/ID output
	Vx_STATID	-	VME STATUS/ID input
DMA	DCTL	L2V	DMA Transfer Direction
	DTBC	-	DMA Transfer Byte Count Register
	DLA	-	DMA PCI Bus Address
	DVA	-	DMA VME Bus Address
	DGCS	Go	DMA Go bit
		CHAIN	DMA Direct or Link List Mode
	DCPP (linked list)	-	DMA Command Packet Pointer
	D_LLUE	UPDATE	DMA link list Update Enable
Error Logging	L_CMDERR	L_STAT	PCI Error Log Status
		CMDERR	PCI Command Error Log
	LAERR	-	PCI Address Log
	V_AMERR	AMERR	VMEbus AM Code Error Log
		V_STAT	VMEbus Error Log Status
	VAERR	-	VMEbus Address Error Log
BI-Mode	MISC_CTL	ENGBI	BI-Mode bit

Table 7: Software Initialization Summary

Functional Group	Register	Bit	Description
Timers	LMISC	CWT	Coupled Window Timer
VME Arbiter	MISC_CTL	VARB	VMEbus Arbitration Mode
		VARBTO	VMEbus Arbitration Time-out
SYSFAIL	VCSR_SET	SYSFAIL	VMEbus SYSFAIL

1.3.2 Registers

The following will describe in greater detail those registers shown in Table 7 on page 10. For more Information concerning these registers see the Universe II User's Manual.

1.3.2.1 PCI Memory Map

PCI_CSR

The PCI_CSR register contains the PCI configuration space control and status registers. The BM bit determines whether a VME slave image will respond to a request. The MS and IOS bits determine whether the target IO and memory are enabled.

PCI_BSx

The first registers that must be programmed in the Universe II are the PCI_BS0 and PCI_BS1 registers. These registers specifies the base address of the Universe II register space as seen from the PCI bus.

One of these two registers determines the mapping into I/O space and the other into Memory, this is determined by the SPACE bit set during Power-up (See Power-up Options of the Universe II User Manual).

1.3.2.2 VME Memory Map

1.3.2.3 VCSR_BS

The VCSR_BS register determines the base address for register access to the universe II on the VMEbus.

1.3.2.4 Images

PCI Slave Images

The following registers must be programmed for an image to function properly, and not overlap with each other or other cards: base address LSx_BS, Bound Address LSx_BD, and the Translation offset LSx_TO (this register may be ignored if no translation offset is desired)

The last bit that should be programmed is the EN bit in the LSx_CTL register, which enables the image.

1.3.2.5 VME Slave Images

The following registers must be programmed for an image to function properly, and not overlap with each other or other cards: Base Address (VSIx_BS), Bound Address (VSIx_BD), and the Translation Offset (VSIx_TO) - this register may be ignored if no translation offset is desired.

The last bit that should be programmed is the EN bit in the VSIx_CTL register, which enables the image.

VME Requests

The following Bits in the MAST_CTL register determine how the Universe II will request and release the VMEbus: VRL, VRM, VREL. VRL determines on which request level the Universe II will request the bus, the default is level 3. VRM determines the request mode that the Universe II will use, the default is demand mode. VREL determines went the Universe II will release the bus, release when done or release on request, the default is release when done.

1.3.2.6 Interrupts

Interrupts are enabled through the following registers: LINT_EN and VINT_EN. Once enabled the interrupts should be mapped through LINT_MAPx and VINT_MAPx respectively. The Status of interrupts can be determined from the LINT_STAT and VINT_STAT registers respectively. On the VMEbus there are additional registers associated with interrupts: STATID and Vx_STATID. The STATID register is the value that the Universe II will return on an IACK cycle and Vx_STATID is the STATUS/ID that was returned to the Universe II on an IACK cycle (when the Universe II is acting as an interrupt handler).

When software interrupt bits are set (in LINT_EN and VINT_EN) they will automatically cause an interrupt of the appropriate level to be generated.

1.3.2.7 DMA transfers

DMA transfers can be programmed into two modes of operation: Direct Mode and Link list mode.

In direct mode the user must reprogram the source and destination address registers (DMA, DLA) before each transfer. These registers are not updated in direct mode. In linked-list mode, these registers are updated by the DMA when (and only when) the DMA is stopped, halted, or at the completion of processing a command packet. If read during DMA activity, they will return the number of bytes remaining to transfer on the PCI side. All of the DMA registers are locked against any changes by the user while the DMA is active.

The source and destination addresses for the DMA reside in two registers: the DMA PCI bus Address Register (DLA register), and the DMA VMEbus Address Register (DVA register). The determination of which is the source address, and which is the destination is made by the L2V bit in the DCTL register. When set, the DMA transfers data from the PCI to the VMEbus. When cleared, the DMA transfers data from the VMEbus to PCI bus and DLA becomes the PCI destination register.

The DMA Command Packet Pointer points to a 32-byte aligned address location in PCI Memory space that contains the next command packet to be loaded once the transfer currently programmed into the DMA registers has been successfully completed. When it has been completed (or the DTBC register is zero when the GO bit is set) the DMA reads the 32-byte command packet from PCI memory and executes the transfer it describes.

Once all the parameters associated with the transfer have been programmed (source/destination addresses, transfer length and data widths, and if desired, linked lists enabled), the DMA transfer is started by setting the GO bit in the DGCS register. If DTBC=0, it checks the CHAIN bit in the DGCS register and if that bit is cleared it assumes the transfer to have completed and stops. Otherwise, if the CHAIN bit is set, it loads into the DMA registers the command packet pointed to by the DCPP register and initiates the transfer describe there.

1.3.2.8 Error Logging

The Universe II has a number of register dedicated to Error detection and logging, for errors on the PCI bus L_CMDERR and LAERR are used while V_AMERR and VAERR are used for the VMEbus.

The bits in the L_CMDERR register are used for the following:

- CMDERR logs command information for the transaction that caused the error
- L_STAT - error is qualified with this bit. (status of error log)

LAERR is used to log the address of the errored transaction logged into the L_CMDERR register.

The bits in the V_AMERR register are used for the following:

- AMERR - VME bus AM Code Error Log
- V_STAT - VME log status

VAERR is used to log the VMEbus address of the errored transaction logged into the V_AMERR register.

For further information on when data is logged to the error registers see the Universe II manual.

1.3.2.9 BI-Mode

When in BI-Mode the Universe II is logically isolated from the VMEbus and will allow only register accesses. A side effect of this is that all PCI coupled transactions will be retried. No VME interrupts will be serviced by the Universe II, Once the DMA FIFO fills/empties no more DMA activity will occur.

BI-Mode is set in the BI bit MISC_CTL register - (0=Universe II not in BI-Mode, 1=Universe II in BI-Mode). The only traffic permitted is to Universe II registers either through configuration cycles, the PCI register image, the VMEbus register image, or CR/CSR space.

There are four ways to cause the Universe II to enter BI-Mode. The Universe II is put into BI-Mode:

1. If the BI-Mode power-up option is selected
2. When SYSRST* or RST# is asserted any time after the Universe II has been powered-up in BI-Mode,

3. When VRIRQ# [1] is asserted, provided that the ENGBI bit in the MISC_CTL register has been set
4. When the BI bit in the MISC_CTL register is set.

There are two ways to remove the Universe II from BI-Mode:

1. power-up the Universe II with the BI-Mode option off
2. clear the BI bit in the MISC_CTL register, which will be effective only if the source of the BI-Mode is no longer active.

1.3.2.10 Timers

The Coupled Window time in the LMISC register determines how long the Universe II will hold VMEbus after performing a coupled read or write.

1.3.2.11 VME Arbiter

The Universe II can act as VMEbus arbiter when it is in Slot 1 as SYSCON. The bits associated with the arbiter are in the MISC_CTL. The VARB bits determine the Arbitration mode: Round Robin or Priority And the VARBTO bit determine the VMEbus Arbitration Time-out.

1.3.2.12 SYSFAIL

If the SYSFAIL bit in the VCSR_SET register is read it return a 1 if VXSYSFAIL is asserted and 0 otherwise. If a 1 is written to this bit then VXSYSFAIL is asserted.

1.4 Generating PCI Configuration Cycles

PCI Configuration cycles can be generated by accessing a VMEbus slave image within the Universe II.

1.4.1 Generating Configuration Type 0 Cycles

The Universe II asserts one of AD[31:11] on the PCI bus to select a device during a configuration Type 0 access. To perform a configuration Type 0 cycle on the PCI bus:

- Program the LAS field of VSIx_CTL for Configuration Space,
- Program the VSIx_BS, VSIx_BD registers to a suitable value,
- Program the VSIx_TO register to 0, and
- Program the BUS_NO field of the MAST_CTL register to a value.
- Perform a VMEbus access where:
  - VA[7:2] identifies the PCI Register Number (mapped directly to AD[7:2])
  - VA[10:8] identifies the PCI Function Number (mapped directly to AD[10:8])
  - VA[15:11] selects the device on the PCI bus (mapped to AD[31:12] according to Table 2.2,
  - VA[23:16] matches the BUS_NO in MAST_CTL register, and
  - Other address bits are irrelevant - they are not mapped to the PCI bus.

Table 8: PCI Address Line Asserted as a Function of VA[15:11]

VA[15:11][a]	PCI Address Line Asserted[bb]
00000	11
00001	12
00010	13
00011	14
00100	15
00101	16
00110	17
00111	18
01000	19
01001	20
01010	21
01011	22
01100	23
01101	24
01110	25
01111	26
10000	27
10001	28
10010	29
10011	30
10100	31

a. The other values of VA[15:11] are not defined and must not be used.

b. Only one of AD[31:11] is asserted; the other address lines in AD[31:11] are negated

1.4.2 Generating Configuration Type 1 Cycles

To generate a configuration Type 1 cycle on the VMEbus:

- Program LAS field of VSIx_CTL to Configuration Space,
- Program the VSIx_BS, VSIx_BD registers to some suitable value,

- Program the VSIx_TO register to 0 and
- Program the BUS_NO field of the MAST_CTL register to some value.
- Perform a VMEbus access where:
  - VMEbus Address[7:2] identifies the PCI Register Number,
  - VMEbus Address[10:8] identifies the PCI Function Number,
  - VMEbus Address[15:11] identifies the PCI Device Number,
  - VMEbus Address[23:16] does not match the BUS_NO in MAST_CTL register, and
  - VMEbus Address[31:24] are mapped directly through to the PCI bus.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.