RC32334/RC32332 Differences Between Z and Y Revisions

Application Note AN-350

Notes

By Paul Snell

Revision History

February 1, 2002: Initial publication.

Background

The RC32334/RC32332 are integrated processors that combine a 32-bit MIPS instruction set architecture (ISA) CPU core with a number of on-chip peripherals to enable direct connection to boot memory, main memory, IO, and PCI. The RC32334/RC32332 also includes system logic for DMA, reset, interrupts, timers, and UARTs. The RC32334/RC32332 components integrate all of the peripherals commonly associated with an embedded system to reduce board real estate, design time, and system cost.

IDT has modified the internal logic of the RC32334 and RC32332 to enhance their functionality in several respects. This application note describes the major differences between the original parts (Revision Z) and the modified parts (Revision Y). It also attempts to analyze the areas where potential compatibility problems might arise in transitioning from the old revision to the new one. The revision on a particular part can be identified by examining the date code on top of the unit. The date code consists of a seven character alphanumeric value. Example: *Z B 0 1 3 9 C*

The first letter identifies the major revision, the second letter identifies the minor revision, the next two digits are the year in which the part was manufactured, the next two digits are the work week, and the final letter identifies the IDT facility where the part was assembled (C for Santa Clara, CA and P for Penang, Malaysia). A date code that begins with Y identifies the part as a new revision; a date code that begins with Z identifies the part as an old revision.

Description

TheY-revision of the RC32334/RC32332 silicon is intended to address a number of performance limitations. Although the Y-revision includes a number of modifications, the primary enhancement is with the on-chip PCI controller.

The Z-revision of the silicon contains a PCI 2.1 compliant PCI core. This core supports PCI target reads and writes as well as PCI master reads and writes. There are separate eight-word-deep FIFOs to accommodate each of these transaction types for a total of four eight-word FIFOs. Since the vast majority of PCI devices and nearly all of the customer systems IDT has seen utilize target reads and target writes to move data, IDT has focused the re-engineering effort on these two modes of operation. In the Y-revision, the FIFO size for both target read and target writes is now 16 words each. The master read FIFO remains 8 words deep and the master write FIFO is now 16 words deep. In addition to increasing the target access buffer sizes, numerous other architectural changes were made.

General PCI Target Access Issues

In the Z-revision, only the four upper bits of the base address register (BAR) are programmable, resulting in the RC32334/RC32332 occupying a PCI memory space of 2^{28} or 256 MBytes for each BAR activated. This method is an inefficient use of memory space and can cause memory allocation problems in systems with multiple PCI devices.

In the Y-revision, the upper 24 bits are programmable with the result that minimum memory size is now 2^{8} or 256 Bytes. Additionally, 2 more BAR registers have been added for a total of 4 BAR registers. These additional BAR registers, coupled with the minimum memory size reduction, allow the user to set the BARs

Notes

such that some BARs point to SDRAM while other BARs point to internal system controller registers (for example, the PCI interrupt mailbox registers). This prevents the user from having to change the BAR value in order to access the interrupts as was required in the Z-revision. The availability of additional BAR registers resolves issues in multitasking systems where one process might be trying to access memory while another is accessing internal system controller registers.

PCI Target Write Changes

In the Z-revision, target PCI writes are limited to a maximum local bus burst size of one word. In an SDRAM based system, there is always some amount of latency involved in setting up a transaction and initiating a burst to SDRAM. IDT's SDRAM controller achieves real-world SDRAM burst accesses of 7-1-1-1, which is extremely fast. However, since the PCI target write never performs burst transactions across the local bus, the 1-1-1 is not taken advantage of. As a result, all PCI accesses incur arbitration overhead followed by a 7 cycle SDRAM access hit for each word. This means that even if a system performs an eight word burst into the PCI target write FIFO, data is written out of the FIFO into the SDRAM in eight separate transactions. Each of those transactions consumes a minimum of ten system clock cycles. Since the CPU had higher priority, it often interjects itself between transfers and further lengthens the total transfer time. The net result is that an eight word target write to the RC32334/RC32332 incurs a minimum of 10 x 8 = 80 system clock cycles. This substantially impacts the target write performance.

In the Y-revision, IDT has increased the maximum target write burst size from one word to eight words. In addition, the write FIFO size is increased from eight words to sixteen words. The FIFO incorporates a hysteresis threshold feature. It will not accept a PCI access unless there is available space for at least eight words in the FIFO. This guarantees that if a PCI device attempts to do an eight word burst, there will always be space in the target write FIFO to accommodate that burst. The eight word PCI burst will subsequently be burst to the SDRAM in one contiguous eight word local bus burst, provided that the part is configured to utilize eight word burst sizes.

There is a new register (PCI Target Control) that controls the maximum target write burst capability. It can be configured as either four or eight words, but it defaults to four words in order to somewhat minimize the potential impact of placing the new part into existing systems. (Refer to the PCI Changes section later in this application note for a discussion of the PCI Target Control Register.)

PCI Target Read Changes

The PCI target read functionality has been significantly altered as well. The Z-revision part favors PCI target writes over target reads. Multiple target writes can be buffered, but target reads cannot. Furthermore, there is a requirement for the write FIFO to be flushed before a target read is allowed to complete. So the Z-revision will retry the target read if there is anything remaining in the write FIFO. However, the write FIFO will continue to accept new writes. The net result is that in the case of sustained target write traffic, the target read is unable to acquire data for long periods of time. This can cause problems in systems with multiple PCI devices where one device continually dominates the PCI bus with sustained target writes. Even in systems where both devices are predominantly doing target writes, one of the devices will eventually have to do a read to acquire a new DMA descriptor chain. This read will be starved, and the second device will eventually time out because of excessive target read retrys.

Y-revision addresses this issue in several ways. First, once a target read attempt is made, the user has the option of having target writes automatically retried. Once the write FIFO empties, the requested target read data is fetched from memory. Only then are target writes re-enabled. When the PCI device that requested the target read comes back for the data, it will be available in the read FIFO. In order to further enhance target read throughput, new control bits have been added which, when enabled, map all target read attempts into read line commands. This results in an eight word fetch of data for one word read request, rather than just the one word requested. Therefore, if the PCI device attempts to read the next word beyond the one it just requested, it will already be in the target read FIFO. Note that the downside to enabling this mode is that if the requesting PCI device really does need only one word, bus bandwidth is wasted fetching the other seven words.

Notes

There is also another bit referred to as the “eager prefetch” bit. When this bit is set, the target read PCI block will continue to fetch data until it fills its FIFO. Since the new FIFO is now 16 words, this means that a single word fetch can result in 16 words, divided into two bursts, being fetched across the local bus. This will result in substantial throughput improvements in cases of long block reads. However, this mode should be used with caution. If the system is not moving blocks of data, but rather doing isolated reads from specific locations, enabling these features will degrade system performance rather than improve it.

Another feature which has been added to the PCI target read block is a variable TRDY timeout counter. The user can set this counter such that it actually violates the PCI spec. The following is one reason why a user might want to violate the spec. The PCI spec requires that any device which does not have the data ready in 16 cycles to retry the transaction. The Z-revision does not realize this ahead of time and holds the PCI bus for 16 cycles hoping to get the data, instead of automatically retrying the transaction. In many cases, this approach wastes 16 PCI cycles.

The Y-revision can be programmed to hold the bus until the data arrives, even if it is longer than 16 cycles. Our analysis of a number of RC32334/RC32332-based systems indicates that data arrives within 20 cycles, so that holding the bus can result in a substantial performance gain. Alternatively, the new PCI retry timeout counter can be programmed to give up the bus as quickly as possible. However, it should be noted that even when set to the minimum value, the target read transaction will still end up holding the PCI bus for eight to ten cycles before being retried.

PCI Master Write Changes

The master write interface has also been modified slightly. In most systems, data is delivered to the PCI master write interface via a memory-to-PCI DMA. In the Z-revision, the RC32334/RC32332 would not check the state of the master write FIFO prior to beginning the burst. As a result, in cases where the write FIFO did not have enough space to accommodate all four words of the DMA transfer, the DMA would sit on the local bus until space became available. In systems with slow PCI devices, this caused a significant performance impact. And it was particularly problematic in systems with real time components that required periodic servicing.

In the Y-revision, the internal bus arbiter now checks the state of the master write FIFO prior to granting the bus to the DMA. The DMA will not be granted the local bus unless there is space to completely accommodate the four word burst within the master write FIFO. This prevents the DMA from unnecessarily holding the bus for long periods of time and should substantially improve performance in systems which rely heavily on this transfer mode, particularly when the target of the transaction is a slow PCI device.

PCI Master Read

In most PCI devices, the PCI master read transaction is the most inefficient. Consequently, nearly all real-world systems are architected in such a way that they make very little use of it. The RC32334/RC32332 is no exception. The master read interface is substantially less efficient than either of the other three interfaces. Once a read is initiated, the DMA or CPU holds the local bus until the read is completed and the data becomes available. This is true in both the Z and Y revisions of the part. In the case of slow PCI devices, this can result in long delays. As noted earlier, however, this usually does not present a problem because PCI systems are architected in such a way that they make spare use of this transaction.

Device Errata

For a list of items that have been fixed in the Y-revision, refer to the RC32334/RC32332 Device Errata on IDT’s web site (http://www.idt.com/products/pages/Integrated_Processors-79RC32334.html). Some items have not been fixed. For example, items 2 and 3 in the Device Errata relate to EJTAG reset and would require extensive modifications to the CPU core, which was not touched in this revision. These items do not directly affect customers, as they only affect EJTAG and ICE probe development. IDT has worked with a number of 3rd party suppliers of emulator equipment to resolve these issues in software in order to provide a fully functional debug environment to the user.

Items 8 and 9 are related to the UART. They pose no functional problem and are related to technical minutia relative to having perfect 16550 compatibility.

Notes

Backwards Compatibility

Throughout these changes, constant attention was paid to minimizing possible compatibility issues between Z and Y revisions. All of the PCI changes discussed previously, with the exception of the local bus PCI target burst write of four words, default to the Z-revision legacy behavior. The PCI enhancements must be enabled via two new registers that have been added before they become active. This should prevent existing software from inadvertently enabling the features. In-depth risk assessment is included in the feature-by-feature discussion which follows.

Specific Changes & Impact Assessment

This section will discuss on a feature by feature basis all changes that were made to the part. The PCI changes are included in this discussion only in terms of risk assessment, as they were discussed more thoroughly in previous sections of the document.

Errata Item #4 DMA Arbitration Fix

This modification was made in order to improve the fairness of the round robin internal arbitration protocol. The type of arbitration used by the internal arbiter is selected via the arbiter control register located at 0x1800_0008.

Existing System Impact: This might potentially impact an existing design if round robin arbitration is being used. However, since this is a documented errata, and the workaround consists of disabling it, it is unlikely any customer will be affected.

PCI vs. DMA Arbitration Priority

In addition to this change, a mode was added to lower the priority of PCI from highest to lowest. This only affects the behavior of the device when it is running in fixed priority mode. It should be noted that at one point, the user manual incorrectly stated that DMA had higher priority than PCI. The legacy behavior of the Z-revision is to give the PCI higher priority than the DMA. In Y-revision this is now selectable via a bit (bit 1) in the Arbitration Register located at 0x1800_0008.

Existing System Impact: Z-revision uses only bit zero of this register. The other 21 bits were labeled as “reserved”. No default value for these unused bits are specified. Therefore, it is reasonable to assume that the no customer attempted to program these bits one way or another. In the case where the customer wanted to set bit zero, it is quite likely that 0xFFFF_FFFF was written to memory. However, this setting has no negative effect. In order for the user to create an unexpected arbitration mode that might impact system behavior, the user would need to lower PCI priority for fixed priority arbitration configurations. This would require a setting of 0xb0010 instead of 0xb0000, 0xb1111, or 0xb0001 that one would expect the customer to use. Therefore, the risk of software incompatibility due to this change is negligible.

SysID Register

The system Identification register was modified to indicate that this is a new revision. The major revision field was changed from a value of zero to one (bit 4 of the sysID register located at 0x1800_0018).

Existing System Impact: This may break software which checks this register prior to beginning execution. However, this change is unavoidable since there has to be a software method of determining which revision of the device is being accessed.

IPBus Timeout Change

On a DMA access that results in an IPBus Error, such as a master write to a non-existent PCI Target, the BIU Arbiter behavior is changed to generate an IPBus Timeout rather than a Watchdog Timeout.

Existing System Impact: This should have no effect on existing systems because this situation should not arise in a functional system. This is an error condition which should only occur during debug. However, in the event that a customer has a system which for some reason generates accesses to non-existent devices, the system behavior will change.

Notes

Synchronous DRAM Controller

Previously, the Z-revision of the part was not specifically designed to support any x32 SDRAMs or 256 Mbit or 512 Mbit devices. The Y-revision has been modified to provide support for these. In the course of doing this, it was not possible to maintain legacy SDRAM bank address (BA) to mem_addr line mapping. Therefore, new devices cannot be intermixed with older devices in the same system. Refer to the RC32334/RC32332 User Manual for a list of newly supported devices and address line mappings. Z-revision only uses mem_addr signals up to mem_addr[15] while Y-revision includes mem_addr[16].

It should be noted that the chip selects beginning with CS2 cannot be relocated because they are hard-coded into the RC32334/RC32332 memory space beginning at address 0x1000_0000. Therefore, at most, SDRAM can occupy the memory space from 0x0 to 0x0FFF_FFFF, which corresponds to 256MByte of SDRAM. This creates a practical limit of 256 MByte of SDRAM for any RC23334/RC32332 based system. For example, if a customer uses 16Mb x 8 x 4 SDRAMs (512 Mbit SDRAM chips), he will need four of these in the system (necessary to create a x32 datapath). This will create 256 MBytes of SDRAM in the system, which is the effective maximum.

Note that it is possible to utilize more memory using an “exotic” TLB mapped memory scheme. But the upper memory would only be accessible in Kernel mode. In most applications, this is probably not feasible.

In order to access 256 Mbit and 512 Mbit devices, the behavior of the SDRAM controller has been modified. In Z-revision, the behavior of the SDRAM controller is solely controlled by the contents of the SDRAM control register located at 0x1800_0300. In Y-revision, a new register has been added which compliments the existing one. This new register is located at 0x1800_0304. Refer to the RC32334/RC32332 User Manual for detailed information regarding the function of the bits in this register.

Existing System Impact: The new register defaults to settings which maintain compatibility with the legacy Z-revision part. There is some risk associated with this register since previously there was no register mapped into this address space. An existing application that performs an inadvertent, or possibly even intentional, burst fill operation which rolls through this previously unused address location will probably have an adverse affect on the system’s SDRAM controller behavior.

There were some other slight enhancements added to the SDRAM controller’s behavior. All of them are accessible via the register at 0x1800_0304. They all default to the Z-step legacy behavior. Users who are writing new code or modifying existing code to use the new part should enable these features. All of these features default to disabled in order to guarantee maximum compatibility between Y-revision and the Z-revision legacy systems.

PCI Changes

Changes made to the PCI section are discussed above. This section concentrates on changes to the configuration register and possible compatibility issues. All the new features, except the new target read control functions, are accessed via the new feature register located at 0x1800_20A0. The new PCI target transaction control functions are enabled or disabled via a new register located at memory address 0x1800_20A4, as shown in Table 1. Refer to the user manual for further information regarding these registers.

Address	Register
0x1800_20A0	New Feature Register
0x1800_20A4	PCI Target Control Register

Table 1 Additional PCI Control Registers

PCI Master Write FIFO Back Pressure

The feedback mechanism from the PCI master write FIFO back to the DMA, discussed previously in the PCI Master Write Changes section, is the default behavior for the Y-revision of the part. There is no provision to disable it.

Notes

Existing System Impact: This should be wholly transparent to the user. If the user is using quad-word DMA bursts to write to the output FIFO, throughput will be improved. Note that this mode may cause problems in the Z-revision (item #10 in the Device Errata) but not in the Y-revision. It is unlikely that any legacy designs will be using quad-word DMA to move data from memory to PCI.

PCI Target Write Four Word Default Burst

As noted earlier, the PCI target writes will now default to four word bursts.

Existing System Impact: This will cause an increase in system throughput for legacy systems where external devices are doing PCI burst writes with the RC32334/RC32332 as the target. This increase in throughput could potentially change the timing of various transactions within the system relative to each other and bring out a flaw that was not noted previously. But the risk is low.

PCI Config Register Accesses

The PCI specification states that when the host attempts a config cycle to unmapped address space, a value of 0xFFFF_FFFF should be returned. The Z-revision complies with this requirement. However, it also generates a bus error, which it should not. The Y-revision will continue to return 0xFFFF_FFFF for unmapped config read accesses. However, bit 1 of the new feature register at address location 0x1800_20A0 can, if set, suppress the associated bus error. The bit defaults to zero, forcing the Y-revision to behave in a similar manner as the Z-revision, thereby minimizing potential software differences.

Existing System Impact: Since the behavior defaults to legacy Z-revision behavior, there is no apparent risk associated with this feature.

PCI Arbiter Errata

Items #5 and #6 in the RC32334/RC32332 Device Errata are associated with the PCI arbiter's handling of grant assertion and deassertions. These issues exist in the Z-revision of the part but are corrected in the Y-revision.

Existing System Impact: Neither of these corrections is expected to have any impact on existing systems.

PCI Arbiter Idle and Bus Parking

Several new features that control idle bus behavior have also been added. These are accessed via new bits in the existing PCI arbitration register located at 0x1800_20E0. Bits 3 and 4 of this register, which were previously reserved with no default value specified, now control these functions. One of these functions changes the number of cycles that a master is allowed to hold the bus without initiating a transfer. The other function allows parking of the PCI bus on the last master that was granted the bus in the event that no other master requests the bus after the last master is done with it.

Existing System Impact: Even if any these functions is inadvertently enabled by the user, it will not have an appreciable affect on an existing system. The features have been added to further facilitate fairness between PCI devices and should augment, rather than detract from, the performance of existing systems.

PCI Configuration Register Device ID

Since the PCI core has substantially changed between the Z and Y revisions of the RC32334/RC32332 silicon, the value in the Device ID configuration register was changed from zero to one. This bit was changed so that software could differentiate between Z and Y revisions.

Existing System Impact: If existing software reads this value, there is some risk that it will no longer recognize the IDT part as being the correct part or that it will mis-report the device it finds in the slot. The device ID had to be changed so that external devices could recognize the difference between the Z and Y revisions of the part. There is some risk associated with it in systems where the RC32334/RC32332 is not the host device.

Notes

IPCI DSEL Decode Response Time

The Z-revision of the RC32334/RC32332 could, under some circumstances, respond with a fast IDSEL decode response. The Y-revision of the part responds with a medium decode speed under all circumstances. This may cost an extra PCI cycle in some target read and write situations.

Existing System Impact: This is not expected to have any impact on existing systems. The nominal change in PCI performance is insignificant from a system perspective.

PCI BAR Cacheable Register Change

In the Z-revision, the prefetchable field in the BAR registers is hardwired to enabled. As a result, it is impossible to mark any memory area as non-cacheable. In the Y-revision, this bit is configurable.

Existing System Impact: This is not expected to impact existing systems. The only case where this could be a problem is if the memory space is labeled as non-cacheable and then inadvertently changed to cacheable. In this case, there could be data coherency issues. However, since in the Z-revision the prefetchable field is hardwired to cacheable, this case cannot arise.

DMA Controller

A new bit (29) has been added to the DMA configuration register. This bit was previously reserved with no indication of what value it should default to. In Y-revision, setting this bit allows the user to read the DMA status register while the associated DMA channel is busy. In Z-revision, this is not possible.

Existing System Impact: Most likely, existing software will have written a value of zero to this bit. This will cause the part to behave in the legacy fashion. But since the field was labeled as reserved with no default value, there is nothing which mandates that existing software must have set the field to zero. If this feature is inadvertently enabled and a user is reading the status register and waiting for a non-zero value to indicate that the channel is idle, then the software will malfunction. Therefore, there is some risk associated with this change.

PIO Controller Changes

A double registering option has been added to the Y-revision to better handle asynchronous inputs. This feature is enabled by a new register that maps into a previously unused address in the PIO control register space. The new registers are depicted in Table 2. The only programmable bit in these registers is located at bit 0. When this bit is set to zero, the default setting, the PIO pins operate in Z-revision legacy mode. But when the value in these new registers is set to one, double registering is enabled.

Address	Register Name	Descriptions
1800_060C	PIO New Feature Register 0	New Feature Register
1800_061C	PIO New Feature Register 1	New Feature Register

Table 2 Additions to PIO Register Address Map

Existing System Impact: Enabling this function will delay the PIO input timing by one cycle because of the extra register added to the input path. But since existing software will not know to enable this feature, it should have no effect on existing applications.

Synchronous Reset Requirement

In the Z-revision, errata item #7 is associated with the need for synching the cold reset signal with the system clock. This has been corrected and is no longer necessary in the Y-revision.

Existing System Impact: This should have no impact on existing designs. Synching the reset, while no longer necessary, will not have any detrimental effects on the system.

Notes

Tri-state Mode Bit Addition

A new mode bit has been added which tri-states the memory bus. This feature allows the board designer to add a jumper to control this. By enabling the mode bit, the board manufacturer can tri-state the memory bus pins, allowing a fixture to be clipped onto the flash memory device. This allows the flash to be programmed, while it is soldered on the board, without running into bus contention issues with the processor. This mode is accessed via a previously unused bit combination on the mem_addr[22:21] lines. The new setting is shown in Table 3.

mem_addr[22:21]	mem_addr[20]	Description
1 1	x	Tri-state memory bus during coldreset_n assertion.

Table 3 Additional reset_boot_mode Initialization Settings

Existing System Impact: Since this tri-state mode can only be accessed via a currently invalid mode bit combination and inadvertently setting this combination would have broken any existing design, there is no risk to existing systems.

JTAG Device ID

The JTAG device ID value returned by the IDCODE JTAG instruction has been changed. The value in the Z-revision of the device is 0x0001_8067. The new value in the Y-revision of the device is 0x1001_8067. Bits 31 through 28 are reserved for the device revision number per the JTAG specification. These bits have been changed from a value of 0b0000 to 0b0001.

Existing System Impact: Existing software should recognize that this is a new revision of the existing part. But if the software is comparing all of the bits of the returned value, including the revision number, the software may not like the new value and refuse to work with the new part. There is no way to avoid this risk because the device ID revision number has to be changed.

Measured Performance Impact

This assessment is based on numbers generated in simulation. Therefore, it relies on a comparison of the throughput capability of Y-revision as a percentage of the throughput of Z-revision. Since the data was taken via simulation, rather than from a real-world application, this is currently the only viable way to correlate it. IDT intends to perform additional performance tests once Y-revision silicon is available.

Numerous options have been added to the PCI control registers. In order to make correct use of these, the user needs to tune them for the specific system. Incorrectly configuring these options could impede performance rather than improve it. It should also be noted that most of the features added to the Y-revision pertain to burst access improvements. Since any application which requires substantial performance is probably moving large blocks of data into contiguous memory locations, this should dramatically improve the RC32334/RC32332's throughput where it matters.

However, since the part has been optimized for bursts, and not for single accesses, the amount of throughput improvement seen in actual applications will be heavily dependent on the amount of data being moved. Larger packets with larger PCI burst sizes (up to eight words) will result in larger performance gains over Z-revision, provided the device is properly configured. There is no gain for single word transfers.

Data taken from the simulations is summarized in Table 4. The main point to note is the delta value. The PCI target write throughput has been improved by up to 4x. The actual amount of gain will depend on traffic composition, burst sizes, and packet sizes. The PCI target read has increased by up to 2x. However, this can only be achieved by heavily tuning the PCI target read transactions via the newly implemented control registers. The bulk of these new control bits are exclusively applicable to target read transactions.

Master reads were not changed in Y-revision. The master writes were changed to include feedback from the write FIFO back to the DMA. However, this gain can only be seen in the case of slow PCI devices. And the simulation environment was designed to stress the part with high-speed PCI accesses. As a result, the

Notes

effect could not be seen in simulation. It should be noted that the issue of the master writes holding the bus has only been seen in relatively few actual systems. In most systems, PCI devices are fast enough to absorb the data so that local bus throughput is not impaired.

Transfer Type	Transfer Size	RC32334/RC32332		Delta
		Z-revision	Y-revision	
Target Write	Various	14 MB/s	14-60 MB/s	1x - 4x
Target Reads	Various	10-30 MB/s	10-60 MB/s	1x - 2x
Master Writes	Various	8-30 MB/s	8-30 MB/s	1x
Master Reads	Various	5-20 MB/s	5-20 MB/s	1x

Table 4 PCI Throughput Comparison — Revision Z vs. Revision Y

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.