

Signal Integrity and Clock System Design

ALLAN LIU, APPLICATIONS ENGINEER, IDT

Introduction

Signal integrity is the art of getting a signal from point A to point B with minimum distortion to that signal. The recent attention on this subject stems from the necessity to build systems with ever increasing throughput and from silicon manufacturing advances in geometry scaling that are causing system-level effects (such as crosstalk, transmission line effects and voltage drops) to manifest themselves on the chip level in a detrimental way. The vast amount of literature on this subject almost exclusively references, and correctly so, high-speed signals that operate at 1 gigabit (Gb) per second or higher. These signals are the most sensitive to these limitations and thus should receive the most attention. However, what the industry neglects is that the quality of these high-speed signals is highly dependent on the quality of the input reference clock that is used to generate these high-speed signals. The design of the clock network is equally deserving of attention on the subject of signal integrity.

Clock management in today's 10 Gb/s systems cannot be left as the last design decision. Generating and distributing a 25 MHz Transistor-Transistor Logic (TTL) or Low-Voltage Transistor-Transistor Logic (LVTTTL) clock for systems with bit widths of 100 nanoseconds (ns) or 10 ns is a vastly different challenge than distributing a 161.13 MHz Low-Voltage Positive Emitter Coupled Logic (LVPECL) or Low-Voltage Differential Signaling (LVDS) clock for a system with bit widths of 96 pico seconds (ps). Clock management must be an early and integral part of the overall system design decision. It must be considered with the same technical depth as the data path and the elements on the data path. The buzz words used in clock management are the same buzz words used in data path design: low jitter, supply isolation, sharp edges, crosstalk, etc. Moving forward, the challenge for silicon manufacturers is how and what they will contribute to the system engineer's toolbox to help solve or alleviate these problems in clock tree design. In this white paper, we will look at the innovations that IDT is bringing to the clock management market and the problems they specifically address.

Crosstalk

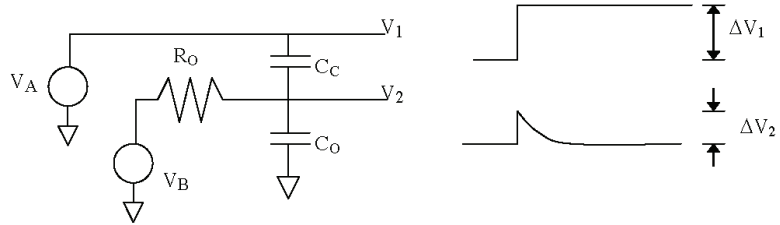
One of the biggest problems facing board designers is crosstalk. In essence, crosstalk is interference caused by capacitive and inductive coupling between two conductors. The term "aggressor" refers to the conductor that is inducing the interference and the term "victim" refers to the conductor that is the recipient of that interference. Crosstalk is a function of frequency and voltage because it is an effect caused by capacitive and inductive coupling between conductors. Consider the simple case of a driven line next to a floating line. Note that this is just a capacitive voltage divider:

In this simple scenario, the disturbance is only a function of the voltage. This scenario is common inside integrated circuits (ICs)

$$\Delta V_2 = \frac{C_c}{C_c + C_o} \Delta V_1$$

that use pre-charge logic. It also illustrates the importance of not leaving input pins floating if there is no DC path to ground. There is a danger of having that pin charged up, due to crosstalk or other means, to a state that changes the intended functionality of the chip.

The more complicated and more common scenario is having crosstalk between two conductors that are driven. Consider the diagram below, where V_A is the aggressor source and V_B is the victim source:



The unit step response is given by:

$$V_2(t) = \frac{C_c}{C_c + C_o} \exp\left(\frac{-t}{R_o(C_o + C_c)}\right) = k_c \exp\left(\frac{-t}{\tau}\right)$$

If the stimulus is not a step, but rather a unit ramp function with rise time, t_r , then the response is given by:

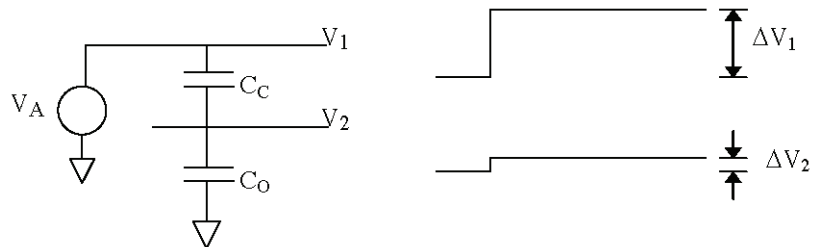
$$V_2(t) = \begin{cases} k_c \left(\frac{\tau}{t_r}\right) \left[1 - \exp\left(-\frac{t}{\tau}\right)\right] & \text{for } t < t_r \\ k_c \left(\frac{\tau}{t_r}\right) \left[\exp\left(-\frac{t-t_r}{\tau}\right) - \exp\left(-\frac{t}{\tau}\right)\right] & \text{for } t \geq t_r \end{cases}$$

Here we see that the interference on the victim signal is indeed a function of the rise time of the aggressor signal. To think of this in the frequency domain, recall that a signal with a low rise time has a wider bandwidth than a signal with a high rise time. As noted earlier, crosstalk, because it is a capacitive and inductive effect, is a function of frequency and voltage. Energy at high frequencies creates more significant interference than energy at low frequencies. As such, the higher bandwidth signal (low rise time) creates more interference than the low bandwidth signal (high rise time). This is consistent with our time domain analysis.

There are many board level techniques the engineer can use to reduce crosstalk. The above equations point to two simple countermeasures:

- 1) The amplitude of aggressor signals should be minimized. There are two ways to interpret this point. The first is that the engineer should not use full swing signals where it is not necessary. The second is that the engineer should not route full swing signals next to sensitive signals because the potential crosstalk can be large (due to large) and detrimental to the integrity of the system.
- 2) The rise/fall time should be made as long as possible and still meet timing requirements.

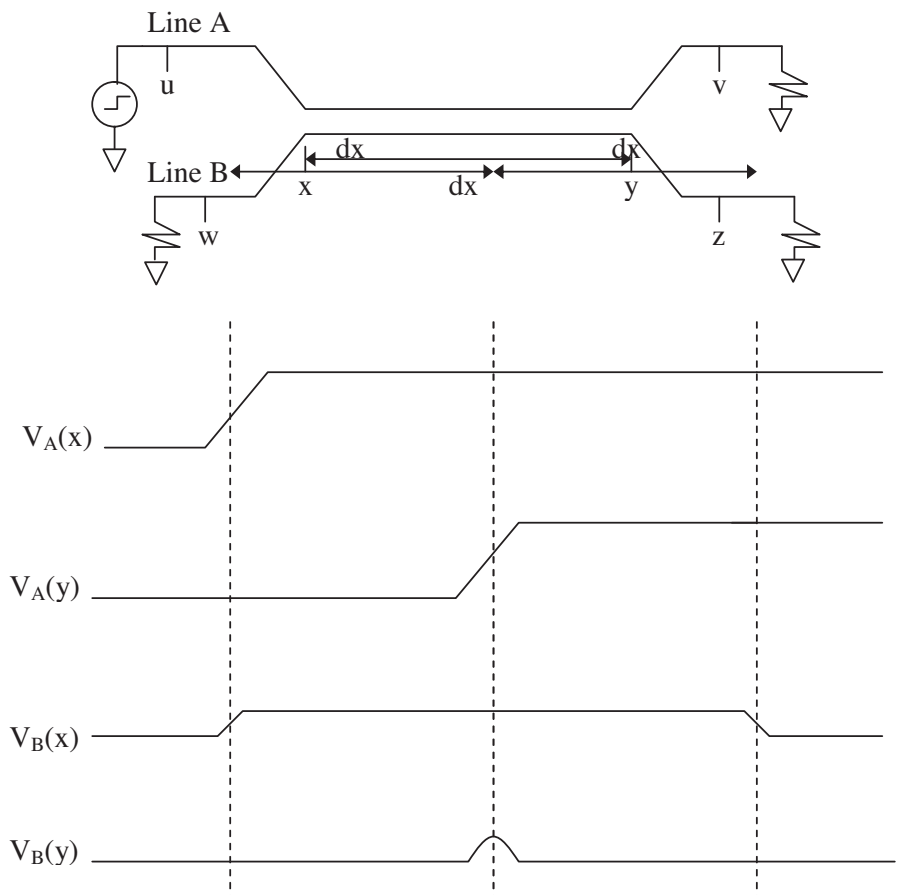
If component suppliers implement features that allow the board designer to control the amplitude and/or the rise/fall time of the output signal, the board designer would have two extra weapons against crosstalk.



IDT has done just that with its clock management products. Many of existing IDT products, such as the 5T9820 (programmable zero-delay buffer), the 5T9890 (programmable skew device) and the 5T907 (fanout buffer), offer options to limit the output signal swing. And, starting with the 5V9885 (programmable generator), designers will have the ability to control the output slew rate.

A point that was not discussed above is that differential signaling is also a very effective tool against crosstalk. Input buffers that receive differential signals reject common-mode noise—noise that is on the true and complement signals. If differential signals are routed close together, then crosstalk will impact both signals, making it common-mode noise and thus will be rejected at the receiver. IDT has a complete family of LVDS and LVPECL clock management devices, such as the 8737 (LVPECL fanout buffer) and the 5T9306 (LVDS fanout buffer), that will aid in reducing crosstalk. Clock networks can get extremely large with traces easily exceeding 10 inches. Longer traces have a higher potential of being affected by crosstalk from all its neighboring signals. It's even more imperative that these longer traces use differential signaling to minimize the effects of crosstalk and other forms of interference.

The above discussion avoided the topic of transmission lines. If we treat the aggressor and victim lines not as ideal conductors, but as transmission lines, we will add an additional dimension to the calculations above. While the mathematics is straight forward, it is nevertheless cumbersome to reproduce here; the reader can consult a good textbook on this subject for the details. We will take an intuitive approach here. Suppose we have two transmission lines that are coupled for some distance (dx). The capacitive and inductive coupling creates forward and reverse traveling waves along the transmission line. When V_A arrives at point x , it induces a reverse traveling wave on line B. This wave sets up the pulse on $V_B(x)$. The pulse width is exactly the round trip delay of the transmission line. The pulse does not go away until the last reverse traveling wave created by V_A at point y gets back to point x . This is known as near-end crosstalk. There is also far-end crosstalk. The edge of V_A creates a forward traveling wave. It arrives at point y after the transmission line delay. Note that point y is the last point of coupling between line A and Line A Line B. After the forward traveling wave moves past point y , line B is no longer affected by it. This is the reason for the “hump” on line B at point y rather than a pulse.

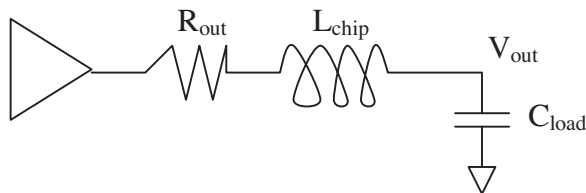


Far-End Crosstalk

In general, far-end crosstalk is seldom a problem. Far-end crosstalk is proportional to the difference between the capacitive and inductive coupling factors or ($k_{cx} - k_{lx}$). The physics works out that the far-end crosstalk coupling factor is negligible because k_{cx} and k_{lx} are almost identical for well-designed boards. Near-end crosstalk, however, is proportional to the sum of the capacitive and inductive coupling factors or ($k_{cx} + k_{lx}$). Near-end crosstalk is not a problem in itself. The problem arises when the near-end crosstalk is reflected back to the far-end (the receiver end). If the transmission lines are not terminated correctly, near-end crosstalk can certainly eat away portions of the noise margins. However, with careful board design, near-end crosstalk can be very well controlled. IDT clock management devices take this one step further. Several of the clock devices, such as the 5V2310 (1-to-10 fanout buffer) and 5V2305 (1-to-5 fanout buffer), have integrated damping resistors to correctly terminate a 50-ohm transmission line. This eliminates the need for external resistors and the guesswork associated with selecting that resistor. Most of the upcoming IDT clock devices will also have this feature. However, rather than having a fixed resistor value, these upcoming devices will have a range of resistor values that can be selected using a serial programming interface. This will improve the matching as well as allow for traces that are not 50ohms. The underlying value propositions are the same however: eliminate external components, enable easy termination of transmission lines, and eliminate the iterative process of soldering and desoldering various resistors to find the most optimized solution. The IDT devices look to solve hardware problems with the turnkey speed of software programming. Keep an eye on future IDT clock management devices for this and other innovative features.

Rise/Fall Time and Signal Integrity

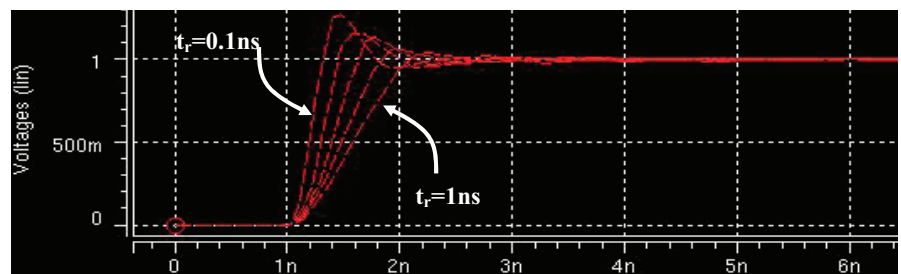
Controlling rise/fall time and termination resistance is not just useful for controlling crosstalk. It is useful in improving signal integrity in general. We will not belabor the importance of properly terminating transmission lines. The reader is probably tired of hearing about this in the available literature. Suffice it to say that correctly terminating a transmission line will not only improve the quality of the signal, but also will reduce EMI emissions. Having integrated termination resistors is certainly a major step forward in helping system engineers easily and cost effectively address this issue. So, instead of focusing on transmission lines and such, we will focus on rise/fall time control in this section. Consider the output driver driving some capacitive load on the board.



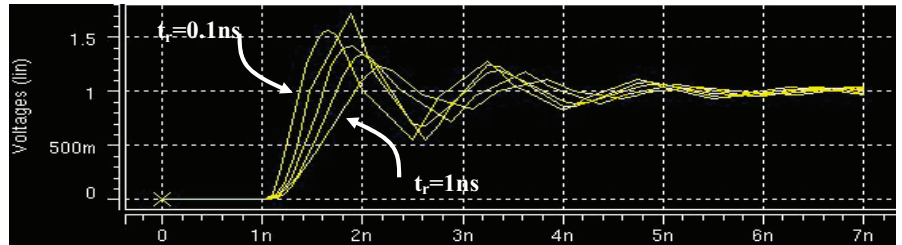
Here, R_{out} is the output resistance of the driver, L_{chip} is the parasitic inductance from the bond wire and the pin on the package, and C_{load} is the capacitive load on the board. Note that this configuration creates a tank circuit. The reader can go through the math by summing the current around the loop and solving for V_{out} . The reader will note that V_{out} is an exponentially decaying function with some oscillation frequency. The key factor, Q , gives the number of cycles before the waveform is attenuated by

$$\frac{1}{e}, \text{ where } Q = \frac{1}{\pi R_{out}} \sqrt{\frac{L_{chip}}{C_{load}}}$$

Let's look at some simulations. For simplicity, let's look at $L_{chip} = 3\text{nH}$, $C_{load} = 5\text{pF}$, and $R_{out} = 16\Omega$. We will vary the rise time from 0.1ns to 1ns.



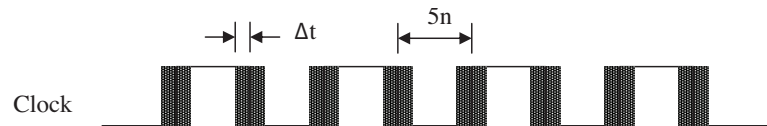
It's quite obvious how the rise time affects the quality of the waveform. The higher the Q factor, the worse the ringing gets. Let's look at another example. We'll vary the rise time from 0.1ns to 1ns again. This time, we'll set $L_{chip} = 7nH$, $C_{load} = 5pF$ and $R_{out} = 4\Omega$. The ringing is much more significant. Ringing not only contributes to intersymbol interference, but also can create sampling errors and high-frequency EMI radiation.



Having the ability to fine tune the rise/fall time via pins or a programmable interface opens up a new level of control to create and maintain quality signals. The new family of IDT programmable clocks, starting with the 5V9885 (programmable generator), is leading the industry in enabling techniques that allow system engineers to optimize their designs for high signal fidelity.

Jitter

We have not explored the issue of jitter thus far. Jitter is timing uncertainty in the position of signal transitions from one state (high) to another state (low). This definition applies to all types of signals, but for this white paper, we will strictly talk about clock signals. Clock signals are periodic signals. In the absence of jitter, the exact position of any clock edge can be deterministically deemed a priority. For example, given a 100MHz square wave, we can say without a doubt that there is a state transition every 5ns. In the presence of jitter, this level of certainty is no longer possible. We might be able to say that there is a state transition every $5ns \pm \Delta t$, where Δt represents the maximum jitter or maximum uncertainty in the position of the state transition.



Jitter can be caused by any numbers of issues, including the ones discussed above, namely crosstalk, improper termination of transmission lines and ringing. We have discussed how the innovative features on the IDT clock products help minimize the impact of those three problems. Power supply noise is also a major contributor to jitter. Let's look at some simple techniques that are implemented on IDT clock products that help deal with power supply noise and also minimize the amount of supply noise that originates from the clock devices. One technique is to use a voltage regulator for more sensitive circuits, such as Phase-Locked Loops (PLLs). In many of the IDT products, including the 5T2010 (zero-delay buffer), 5T9890 (programmable skew device) and 5T940 (precision clock generator), voltage regulators were used to minimize the impact of supply noise on the PLL. Linear voltage regulators come at the cost of power consumption. However, since the PLLs consume very little current anyway, the power penalty is minimal. This has helped tremendously to reduce the impact of supply noise on the chip's jitter performance. In addition, the use of differential signaling inside the chip makes the chip highly immune to supply noise because supply noise is common-mode noise, which is rejected by circuits using differential signaling. The 5T940 (precision clock generator), the 5T9306 (LVDS fanout buffer) the 8737 (LVPECL fanout buffer) and other devices use such techniques to further guard against supply noise.

To minimize the supply noise that is generated by the chip itself, several techniques are used. The first is to decouple the power supply of the core circuits from the power supply of the input/output (I/O) circuits. I/O circuits in general account for more than 60 percent of the total power consumption of a chip. Push-pull output drivers, such as LVTTTL or Complementary Metal-Oxide Semiconductor (CMOS) drivers, can put tremendous noise on the power supply rails inside a chip. This is created by the varying need to source and sink current from the supply rails. If the power supply is ideal, this would not be a problem. However, the power supply is not ideal and cannot source or sink infinite current in zero time. Parasitic inductance and resistance in the package and in the power distribution network on the chip limits the amount of current

that can be sourced or sunk by the supply rails. Consequently, you get supply noise (recall $V = L \cdot di/dt$ and $V = R \cdot I$). Limiting and isolating this type of noise will minimize the jitter that is generated by the chip. Many of the IDT clock products, including the 5T9306 (LVDS fanout buffer) and 5T9820 (programmable zero-delay buffer), implement this noise-limiting technique to maximize their jitter performance. Another technique is to de-synchronize the output drivers. In clock products, synchronizing the outputs or minimizing the phase difference between outputs is an important performance metric. However, not all applications require this feature. In fact, many systems do not need to have all of their outputs synchronized. Synchronizing outputs creates a “beat” on the power supply. All the output drivers (they account for more than half of the chip’s power consumption) drive high or drive low at the same time. This creates tremendous stress on the power supply. If we can somehow force some of the output drivers to drive high while other output drivers are driving low, then we would lower the demand on the supply to source or sink current. This ability to desynchronize the output drivers is implemented in some of the more recent IDT products, including the 5T2010 (zero-delay buffer), the 5T9820 (programmable zero-delay buffer), the 5T9890 (programmable skew device) and the 5V9885 (programmable generator).

Jitter is a real and serious challenge in today’s systems. Despite the best efforts to launch the highest fidelity signal, to prevent board level interference and to minimize jitter generation in the devices, jitter nevertheless creeps into the system and threatens to cripple it. Jitter is not an easy monster to tame. For example, for 10 Gb/s Ethernet, the physical layer devices must have their transmit jitter be lower than 0.3UI or 28.8 ps, where 1UI (unit interval or bit width) = 96 ps. Let’s assume that the physical layer devices generate 0.2UI. That means the total jitter from the clock, assuming all of it is transferred to the physical layer devices’ outputs, must be less than 0.1UI or 9.6 ps. Generating a low jitter clock may be easy, but what’s the best way to get it to the destinations without compromising the integrity of the clock? One solution, obviously, is to generate the clock exactly where it is needed. This will minimize trace length, thus minimizing any effects of crosstalk. The chip can drive the load with less current because the load is smaller and the trace length is shorter. This per our discussion above, also improves the fidelity of the signal. Of course, with today’s system’s ever-increasing port density, this is neither a cost-effective nor practical solution. Assuming we have a device that can drive four ports, in a high port density application, we can have as many as 8 or 12 low jitter crystal oscillators. That’s very expensive. The more reasonable approach is to have a central point of distribution. A clock (or a pair of redundant clocks) is generated and then distributed to where it is needed in a tree structure. Of course, with the limitation of 0.1UI of jitter at the clock input of the physical layer devices, this is not necessarily a simple proposition. The first order of business is to treat these clock traces just as one would treat any high-speed data trace and to use the best-possible board design techniques to protect them. Even so, this might not be enough to guarantee that the great looking clock that was generated at the source on the board will still look great when it gets to the destination, possibly 10 to 20 inches away. For the high-speed data lines, the channel or transmission medium is budgeted to cause about 0.3UI of jitter. This is obviously not an acceptable jitter budget for the clocks. If we generate a clock with 0.1UI of jitter, the channel must not cause any jitter. Obviously, this is impossible to implement. There are two solutions to this problem. The first is to generate a clock with much less jitter than 0.1UI and anticipate that the channel will create jitter such that the jitter measured at the destination is less than 0.1UI. It’s expensive to generate very low jitter outputs and it’s difficult to limit the jitter to less than 0.1UI. The better solution is to allow for some reasonable amount of jitter to be added by the channel and to allow a reasonable amount of jitter to be generated by the source. This minimizes the cost of generating the clocks and the cost of routing the clocks. The clocks are routed to their destination. However, right before they are used by the destination chip, these clocks are “cleaned up.” This “cleaning” function is performed by precision timing devices. The main function of these devices is to attenuate jitter. However, they are more versatile than that. They can not only attenuate jitter, but they can also perform I/O translation and frequency multiplication/division. These precision timing devices will perform the functions that are needed to guarantee that the clock meets all the requirements of the receiving device, be it frequency, signaling or jitter requirements. IDT currently has one such device in its family of precision timing devices, the 5T940 (precision clock generator). More products are currently being developed for this family. This device has a configurable PLL loop bandwidth and has very low jitter generation. The configurable loop bandwidth allows the user to fine tune the amount and the frequency of jitter that they want to attenuate. It’s another example of the innovative features that IDT brings to the clock management market to give system engineers an extra tool to solve their system level design challenges.

As system-level issues expand and become more challenging, semiconductor vendors must step up and contribute to the toolboxes of system engineers. Clock management has been a quiet market that generally gets ignored in the press, the literature, and even by system engineers. However, the clock network in a system does indeed play a major role in determining the overall performance of the system. Gone are the days of single function jelly-bean clock devices. Today's clock management devices must offer the features that contribute to the overall successful implementation of the system. And these functions must be configurable such that each engineer can fine tune the performance of the device on their system. IDT understands the signal integrity challenges facing the engineer. Its current and future products contain many innovations that can and will help overcome these challenges.

Glossary

CMOS	Complementary Metal-Oxide Semiconductor
Gb	Gigabit
IC	Integrated circuit
I/O	Input/output
LVDS	Low-Voltage Differential Signaling
LVPECL	Low-Voltage Positive Emitter Coupled Logic
LVTTL	Low-Voltage Transistor-Transistor Logic
NS	Nanosecond
PLL	Phase-Locked Loop
PS	Picosecond
TTL	Transistor-Transistor Logic

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)

Corporate Headquarters

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

Trademarks

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

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/.