8V97003

Performance Optimization Guidelines

Abstract

The 8V97003 is a high-performance wideband mmWave, RF Synthesizer using a Phase Lock Loop (PLL) that generates output frequencies up to 18GHz from an integrated Voltage Controlled Oscillator (VCO). The excellent VCO performance supports the generation of clock signals with low phase noise and RMS jitter.

With extensive programmability on all major functional blocks, 8V97003 is ideal for many applications including 5G mmWave wireless infrastructure, phase array beam forming, point-to-point and point-to-multipoint microwave links, satellites, clock generation, and high-speed RF converter sampling clock.

This document provides information on how to optimize the 8V97003 performance parameters such as output signal phase noise and spurious.

Contents

1. VCO Range and Output Frequency 3

2. Setting the Phase Detector Frequency (PFD) 3

3. Calculation of the Fractional Feedback Divider 5

4. VCO Calibration 5

5. Loop Filter Design 6

6. Using Bleeder Current in Fractional Mode 8

7. Output Configuration with OutDoubler_Freq Bit 8

8. Slew Rate Requirement for Low-Frequency Input Reference 9

9. Appendix A – Open Loop VCO 10

10. Appendix B – Python Code 15

11. Revision History 16

Figures

Figure 1. Loop Filter A - 200MHz < PFD Frequency 6

Figure 2. Loop Filter B – 80MHz ≤ PFD Frequency ≤ 200MHz 6

Figure 3. Loop Filter C – PFD Frequency < 80MHz 7

Figure 4. 8V97003 Performance with Input Reference Frequency of 10MHz 9

Figure 5. f_{VCO} = 6GHz at 25°C 10

Figure 6. f_{VCO} = 6GHz at -40°C 10

Figure 7. f_{VCO} = 6GHz at 90°C 11

Figure 8. f_{VCO} = 8GHz at 25°C 11

Figure 9. f_{VCO} = 8GHz at -40°C 12

Figure 10. f_{VCO} = 8GHz at 90°C 12

Figure 11. f_{VCO} = 11GHz at 25°C 13

Figure 12. f_{VCO} = 11GHz at -40°C 13
Figure 13. f_{VCO} = 11GHz at 90°C 14

Tables

Table 1. Desired Output Frequency and VCO Frequency Relationship 3
Table 2. VCO Sensitivity (KVCO) 7
Table 3. Output Doubler Frequency Bit Register Settings 8

1. VCO Range and Output Frequency

The frequency range of the integrated VCO of the device is from 5.5GHz to 11GHz. The output doubler can double the VCO frequency up to 18GHz (2 × f_{VCO}). A good signal integrity at output frequencies higher than 18GHz is not guaranteed. An output divider provides division to the VCO frequency as low as 171.875MHz. When using the output frequency divider, the doubler is not used and is bypassed. The relationship between the desired output frequency and VCO frequency is shown in Table 1.

Table 1. Desired Output Frequency and VCO Frequency Relationship

Desired Output Frequency	VCO Frequency	Output Doubler	Output Divider
5.5GHz ≤ f_{OUT} ≤ 11GHz	$f_{OUT} = f_{VCO}$	Bypassed	Bypassed
f_{OUT} > 11GHz	$f_{OUT} = 2 \times f_{VCO}$	Enabled	Bypassed
f_{OUT} < 5.5GHz	$f_{OUT} = f_{VCO} \div M0$	Bypassed	Enabled
f_{OUT} = 5.5GHz	f_{VCO} = 5.5GHz	Bypassed	Bypassed
	f_{VCO} = 11GHz	Bypassed	Enabled (÷2)
f_{OUT} = 11GHz	f_{VCO} = 5.5GHz	Enabled	Bypassed
	f_{VCO} = 11GHz	Bypassed	Bypassed

Note: M0 = Output divider

When programming the 8V97003, the VCO frequency must be known to calculate the feedback divider value.

2. Setting the Phase Detector Frequency (PFD)

While the device supports an input reference frequency from 10MHz to 1,000MHz, the phase detector is specified for frequencies from 10MHz to 250MHz (fractional synthesis mode) and from 10MHz to 500MHz (integer synthesis mode). This is the first criterion for selecting PFD frequency. A higher PFD frequency improves the device phase noise performance. Consider the additional criteria when setting the PFD frequency for 8V97003 operation:

Desired Synthesizer Channelization: Channelization is the resolution of output frequency. It is given by $\frac{fPFD}{MOD}$ where MOD is the denominator of the fractional part of the feedback divider.

- **Integer-Boundary Condition:** An integer-boundary condition is defined when the fraction feedback divider is near an integer, but not exact. The condition causes high spurious outputs and increased phase noise. Choose a PFD frequency such that an integer-boundary condition can be avoided. A solid baseline would be the integer value ±0.1 or a larger offset. In the following example, you can see how the 8V97003 input reference divider avoids an integer-boundary condition.
  - Example:
    - Input reference frequency f_{REF} = 368.64MHz
    - Target output frequency f_{OUT} = 12042.4MHz
  - Because f_{OUT} > 11000MHz, $f_{OUT} = 2 \times f_{VCO}$. Therefore, f_{VCO} = 6021.2MHz.
  - Because the VCO frequency value of 6021.2 is not an integer multiple of 368.64, the PLL operates in fractional mode. Therefore, PFD frequency must be ≤ 250MHz
  - To keep PFD frequency reasonably high for good phase noise performance results in two choices for the input reference frequency divider R:
    - PFD frequency = 368.64MHz ÷ 2 and
    - PFD frequency = 368.64MHz ÷ 3

- Selecting R = Input Frequency ÷ 3:
  - PFD frequency = 122.88MHz
  - Feedback divider = 6021.2 ÷ 122.88 = 49.00065104
  - This is an integer-boundary condition (fractional part is close to 0)
- Alternatively, selecting R = Input Frequency ÷ 2
  - PFD frequency = 184.32MHz
  - Feedback divider = 6021.2 ÷ 184.32 = 32.66710069
  - Good fractional operation avoids the integer boundary condition
  - Therefore, to fulfill the requirement defined above, set the input reference divider R to ÷2 to avoid an integer-boundary condition

More information about integer boundary spur in PLL is available in the *Integer Boundary Spurs in Fractional-Feedback Phase-Locked Loops (PLLs)* white paper.

- **Integer Versus Fractional Synthesis Mode:** Depending on the performance requirements of the application, the PFD frequency is selected so that the device operates in integer or fractional mode. The following example demonstrates how the PFD frequency selection determines the mode of operation discussed and the performance trade-offs associated with that selection:
  - Example:
    - Input reference frequency: f_{REF} = 100MHz
    - Target output frequency: f_{OUT} = 6.1GHz
  - Because 5.5GHz ≤ f_{OUT} ≤ 11GHz, $f_{OUT} = f_{VCO}$. Therefore, f_{VCO} = 6.1Hz.
  - The first strategy is to use a higher PFD frequency for better phase noise performance. Therefore, enabling the input doubler results in PFD frequency = 100MHz × 2 = 200MHz. The Feedback divider is 6100 ÷ 200 = 30.5, resulting in a fractional mode operation. This operation may suffer from performance degradation because of Delta-Sigma Modulator (DSM) noise and suffer from fractional spurs. The *Fractional Spurs Due to the Quantization Noise* table in the datasheet can provide more information.
  - Bypassing the input frequency doubler results in PFD frequency = 100MHz. The feedback is 6100 ÷ 100 = 61 which results in an integer mode operation. The trade-off with this configuration is the increase in noise because of a larger multiplication factor.

The previous example is only meant to demonstrate how PFD frequency selection can result in different modes of operation. In this example, phase noise performance is better with feedback divider of 30.5, because the PFD frequency is higher, and the fractional spur is far out (100MHz offset). However, when the input reference frequency is low (such as 10MHz), operating in fractional mode means having fractional spur close-in, which could degrade integrate noise performance. In such cases, you can choose to operate at lower PFD frequency in integer mode to avoid spur.

- **Minimum Valid Feedback Divider Setting in 8V97003:** The design of 8V97003 has a lower limit on the value of the feedback divider. This lower limit is 12. In the example below, you can see how this minimum feedback divider setting in the 8V97003 affects the selection of PFD frequency:
  - **Requirement:** Input reference frequency (REF_{CLK}) = 250MHz, desired output frequency (f_{OUT}) = 5500MHz
  - Because f_{OUT} = 5500MHz, the preferred solution is $f_{OUT} = f_{VCO}$. Therefore, f_{VCO} = 5500MHz
  - Typically, a higher PFD frequency is selected for better phase noise performance. Therefore, enabling the input doubler, gives PFD frequency = 250MHz × 2 = 500MHz.
    Feedback divider = 5500 ÷ 500 = 11, which violates the minimum valid feedback divider setting in 8V97003
  - Therefore, in this scenario, select PFD frequency = 250MHz.
    Feedback divider = 5500 ÷ 250 = 22.

3. Calculation of the Fractional Feedback Divider

The Fractional-N divider architecture is implemented using a cascaded programmable dual modulus pre-scaler, controlled by a DSM. The N divider offers a division ratio in the feedback path of the Phase Lock Loop (PLL), and is given by programming the values of INT, FRAC, and MOD in the following equation:

$N = INT + \frac{FRAC}{MOD}$ where:

INT is the integer divider ratio, a binary 16-bit counter

FRAC is the numerator value of the fractional divide ration. It is programmable from 0 to $2^{32} - 1$

MOD is the 32-bit modulus. It is programmable from 2 to $2^{32} - 1$

Calculating the feedback divider ration is straight forward:

- $f_{VCO} \div$ PFD frequency $= INT + \frac{FRAC}{MOD}$
- Frac = (($f_{VCO} \div$ PFD frequency) – INT) × MOD
  - 5.5GHz ÷ 40MHz = 137.5
  - 137.5 – 137 = 0.5
  - 0.5 × (2^{32}-1) = 2147483648 (Rounded to the nearest integer)
    - The rounded integer may shift the frequency accuracy by a few parts per trillion. To avoid this, scale down the Frac to Mod ratio until both are integer values that represent the desired fraction.

The Python code in Appendix B illustrates the calculation of the feedback dividers FRAC and MOD values:

- FRAC: 32-bit registers Frac at 0x0012 to 0x0015
- MOD: 32-bit registers Mod at 0x0016 to 0x0019 in a fractional mode operation.

When the feedback divider ratio (N) is a repeating decimal (such as, 11GHz ÷ 122.88MHz = 89.518229166666), it is critically necessary to modify the Frac value calculated from the above procedure to avoid spurious. Renesas recommends that in cases of repeating decimal N, the programmed Frac value should be incremented by a small enough value to offset the decimal.

4. VCO Calibration

The VCO consists of multiple VCO cores where one core is used by the PLL at a time. Each core selects one of multiple VCO frequency bands for operation. The process of selecting the best core and the frequency band for a given target PLL frequency is called VCO calibration or band selection.

This allows for a lower VCO gain (Kvco), which results in the best possible VCO phase noise and spurious performance. In normal operation, the VCO band select logic of the 8V97003 automatically selects the most suitable VCO band for operation when you complete the programming for the input reference path and the feedback divider. The following settings must be properly set to ensure the most accurate VCO calibration:

- **Band Select Divider:** The band select logic operates between 50kHz and 100kHz. Therefore, the band select clock divider (BndSelDiv[12:0] in the Band Select Clock Divider Control Registers) must be set to divide the PFD frequency down to the range of 50kHz and 100kHz. The following examples show how the band select divider setting is properly set according to the PFD frequency
  - Input reference frequency f_{REF} = 100MHz, Input Doubler = Enable, PFD frequency = 200MHz, BndSelDiv = 2560 (default) → Band Select Clock = 200MHz ÷ 2560 = 78.125kHz which is between 50kHz and 100kHz. Using the default BndSelDiv = 2560 is enough to ensure proper band select logic operation

- Input reference frequency f_{REF} = 245.76MHz, Input Doubler = Enable, PFD frequency = 491.52MHz, BndSelDiv = 2560 (default) → Band Select Clock = 491.52MHz ÷ 2560 = 192kHz which is greater 100kHz. Therefore, set the BndSelDiv to 8191 to ensure proper band select logic operation.

- **Band Select Resolution:** BandSelAcc[1:0] in register 0x0021 is defaulted to the finest available resolution. This default setting would ensure the most accurate VCO calibration. However, using the finest available resolution (BandSelAcc[1:0] = 3) causes longer VCO calibration time, therefore, a longer PLL lock time. If PLL lock time is of importance in your application, BandSelAcc[1:0] can be set to 2 to shorten lock time. Renesas does not recommend using a lower calibration resolution setting than 2.

For example, because there is overlap between VCO4 and VCO5, bit position 4 set preference for VCO4 and VCO5: value of 0 set preference for VCO4, value of 1 VCO1.

5. Loop Filter Design

The 8V97003 EVB has a 3rd order loop filter, which allows evaluation of the 8V97003 with a sufficiently high input reference frequency (f_{REF} > 100MHz). However, should phase noise performance specifications of your application require different loop filter design, there are three different settings recommended base on the PFD frequency.

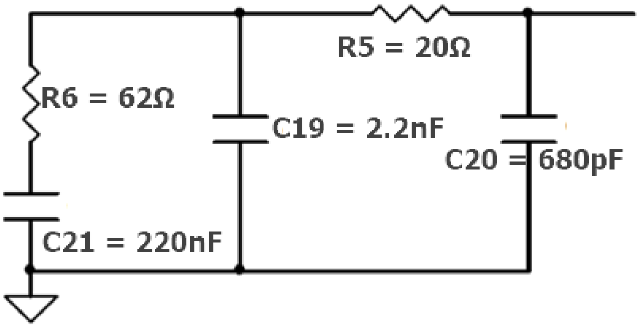


Figure 1. Loop Filter A - 200MHz < PFD Frequency

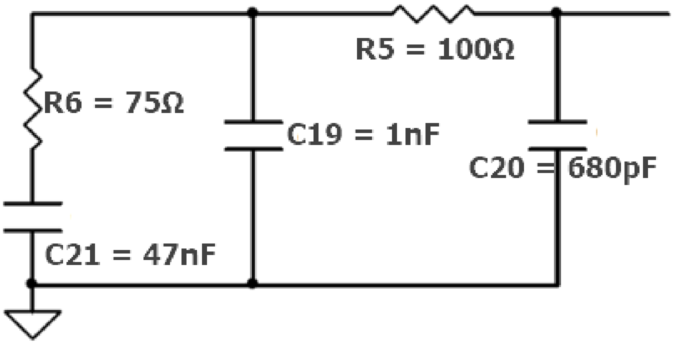


Figure 2. Loop Filter B – 80MHz ≤ PFD Frequency ≤ 200MHz

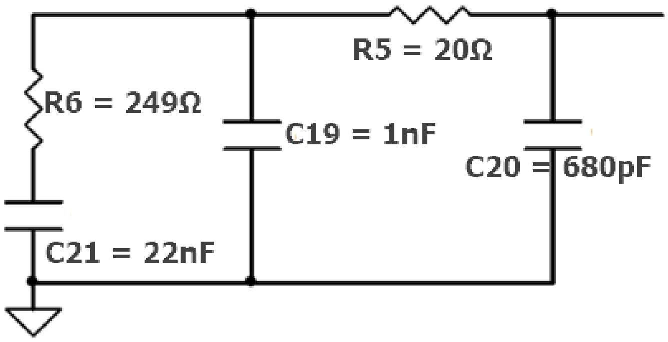


Figure 3. Loop Filter C – PFD Frequency < 80MHz

Changing the loop filter can shift the bandwidth to allow for adjusted charge pump and bleeder settings. This can be beneficial when attenuating spurs and lowering phase noise. Loop Filter calculation information can be found in the 8V97003 Datasheet.

The relevant data necessary for designing a loop filter that meets your performance specification are:

- VCO Gain (Kvco)
- Open Loop VCO phase noise

Table 2 represents the VCO gain at the center of the each VCO band. The device has overlapping VCO bands so the VCO gain at the midpoint between adjacent VCO frequencies listed can be either value. For example, at 6GHz, the VCO gain can be either 100MHz/V or 120MHz/V.

Table 2. VCO Sensitivity (KVCO)

VCO Frequency	Typical
5.625GHz	100MHz/V
6.23GHz	120MHz/V
6.975GHz	140MHz/V
7.8GHz	160MHz/V
8.65GHz	210MHz/V
9.585GHz	165MHz/V
10.3GHz	155MHz/V
10.9GHz	170MHz/V

See Appendix A for Open Loop VCO phase noise.

6. Using Bleeder Current in Fractional Mode

The charge pump current setting as well as the programmable bleeder current, provide more flexibility and control settings that allow you to offset the charge pump gain and account for non-linearity in the PLL. The non-linearity can show up in fractional mode as higher close-in noise. The charge pump current settings, and the programmable bleeder current, can be used in certain cases to reduce the non-linearity and improve the close-in noise performance.

For any fractional measurement, the bleeder charge pump current should be increased and adjusted to optimize the noise. For low charge pump current settings, the bleeder might not need to be large. For higher charge pump current settings, the bleeder needs to be larger.

The objective is to get nearly the same phase noise in fractional mode as can be achieved in integer mode. The bleeder is only needed in fractional mode to improve the close-in noise. The bleeder should improve the close-in noise in fractional mode to the close-in noise performance of integer mode. When using the bleeder, there is a trade-off between this close-in noise and the reference spur. Such that, as the bleeder current is increased, the reference spur becomes larger.

The basic rule of thumb for the bleeder is to increase the bleeder current to a point where the close-in noise is equivalent to the close-in noise in integer mode. Increasing the bleeder current past this point does nothing for the noise and only makes the reference spur worse.

7. Output Configuration with OutDoubler_Freq Bit

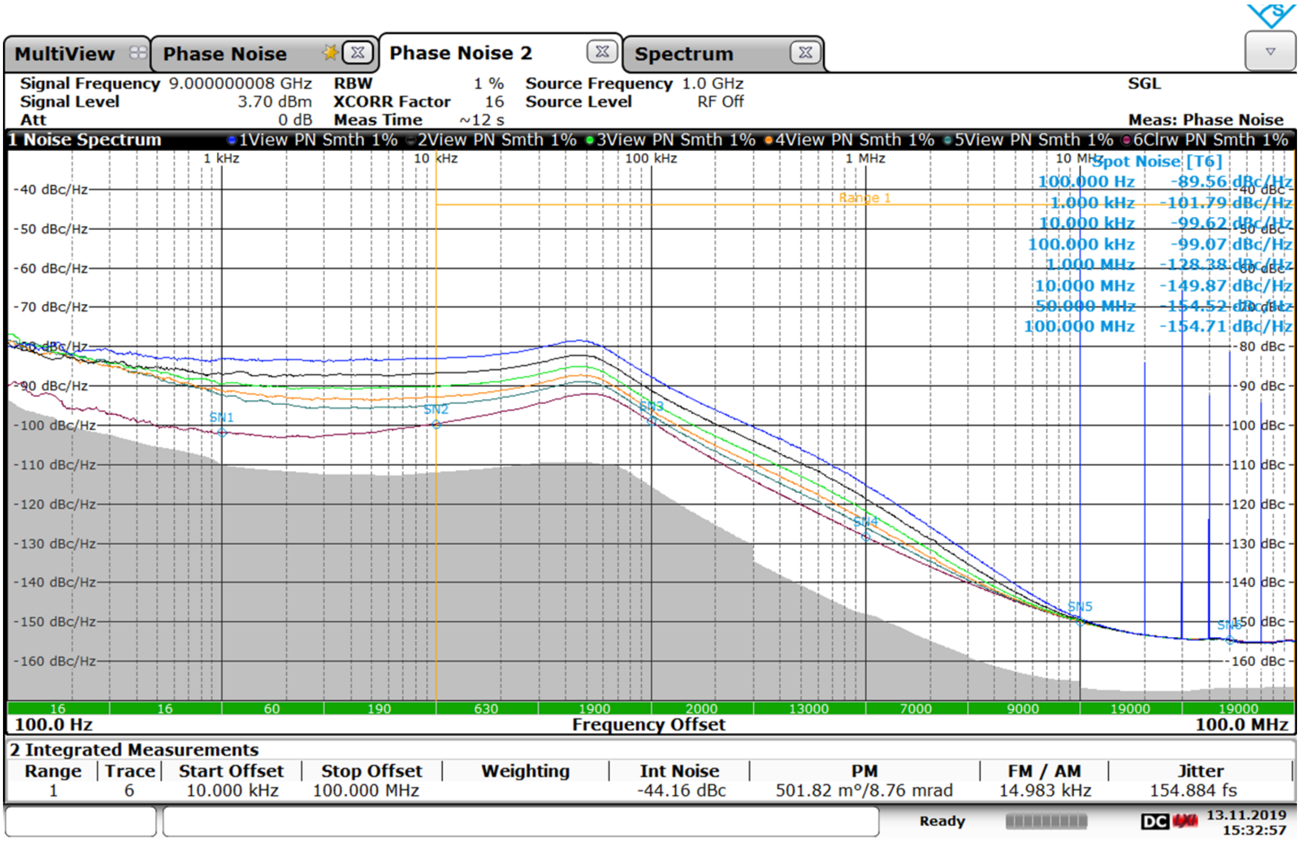
When the Output Doubler is enabled, it is important that the OutDoubler_Freq bit (Register 0x003B[5]) is correctly set, as followed:

Table 3. Output Doubler Frequency Bit Register Settings

VCO Frequency	OutDoubler_Freq (Register 0x003B[5])
7.5GHz ≤ f_{VCO} ≤ 9GHz	0
5.5GHz ≤ f_{VCO} ≤ 7.5GHz	1

8. Slew Rate Requirement for Low-Frequency Input Reference

For configurations with input reference frequencies below 40MHz, testing/evaluating the performance of the 8V97003 using the RF output of the signal generator results in non-optimal phase noise performance. The RF output port of the signal generator provides sinewave signal with slow slew rate. The phase noise plot in Figure 4 shows performance of the 8V97003 with input reference frequency of 10MHz signals of various slew rates.



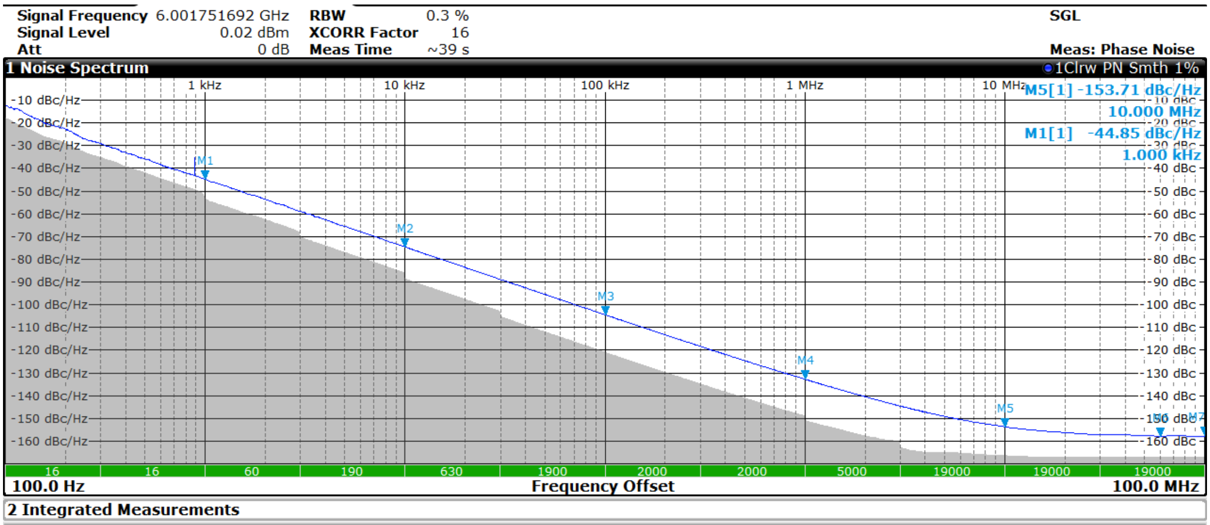
15:32:58 13.11.2019

Slew Rate [V/µs]	Curve Color
16.67	Blue
26.40	Black
41.60	Green
66.89	Orange
111.11	Turquoise
> 580	Red

Figure 4. 8V97003 Performance with Input Reference Frequency of 10MHz

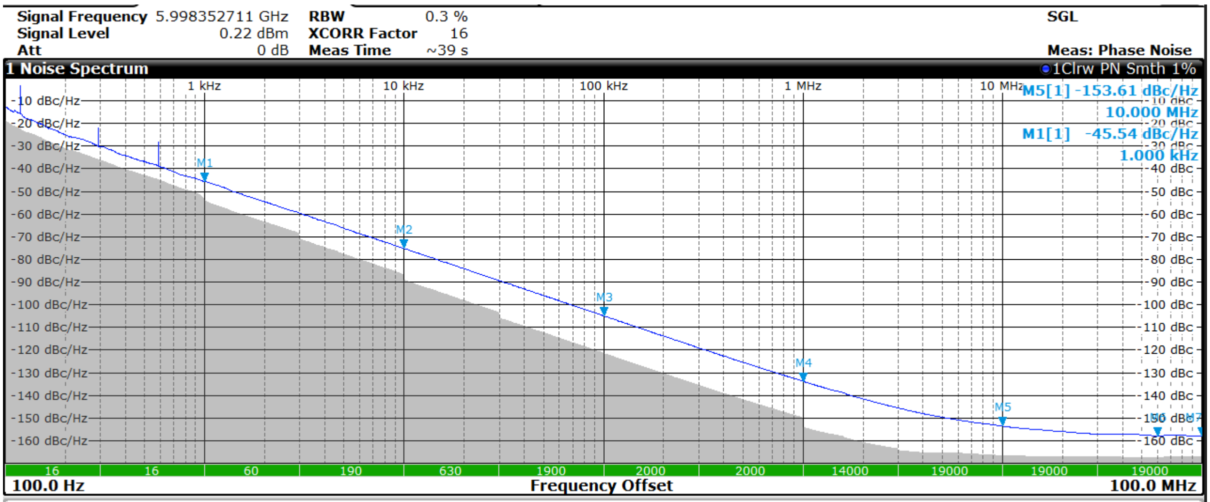
With slew rate of greater than 580V/µV, 8V97003 achieves the best performance. When testing/evaluating or designing in the 8V97003, Renesas recommends input reference clock with a minimum slew rate of 580V/µV for optimal performance.

9. Appendix A – Open Loop VCO



Type	Ref	Trace	X-value	Y-value
M1		1	1.0 kHz	-44.85 dBc/Hz
M2		1	10.0 kHz	-74.66 dBc/Hz
M3		1	100.0 kHz	-104.37 dBc/Hz
M4		1	1.0 MHz	-132.78 dBc/Hz
M5		1	10.0 MHz	-153.71 dBc/Hz
M6		1	60.0 MHz	-157.84 dBc/Hz
M7		1	100.0 MHz	-157.75 dBc/Hz

Figure 5. f_{VCO} = 6GHz at 25°C



Type	Ref	Trace	X-value	Y-value
M1		1	1.0 kHz	-45.54 dBc/Hz
M2		1	10.0 kHz	-75.25 dBc/Hz
M3		1	100.0 kHz	-104.93 dBc/Hz
M4		1	1.0 MHz	-133.84 dBc/Hz
M5		1	10.0 MHz	-153.61 dBc/Hz
M6		1	60.0 MHz	-157.82 dBc/Hz
M7		1	100.0 MHz	-157.8 dBc/Hz

Figure 6. f_{VCO} = 6GHz at -40°C

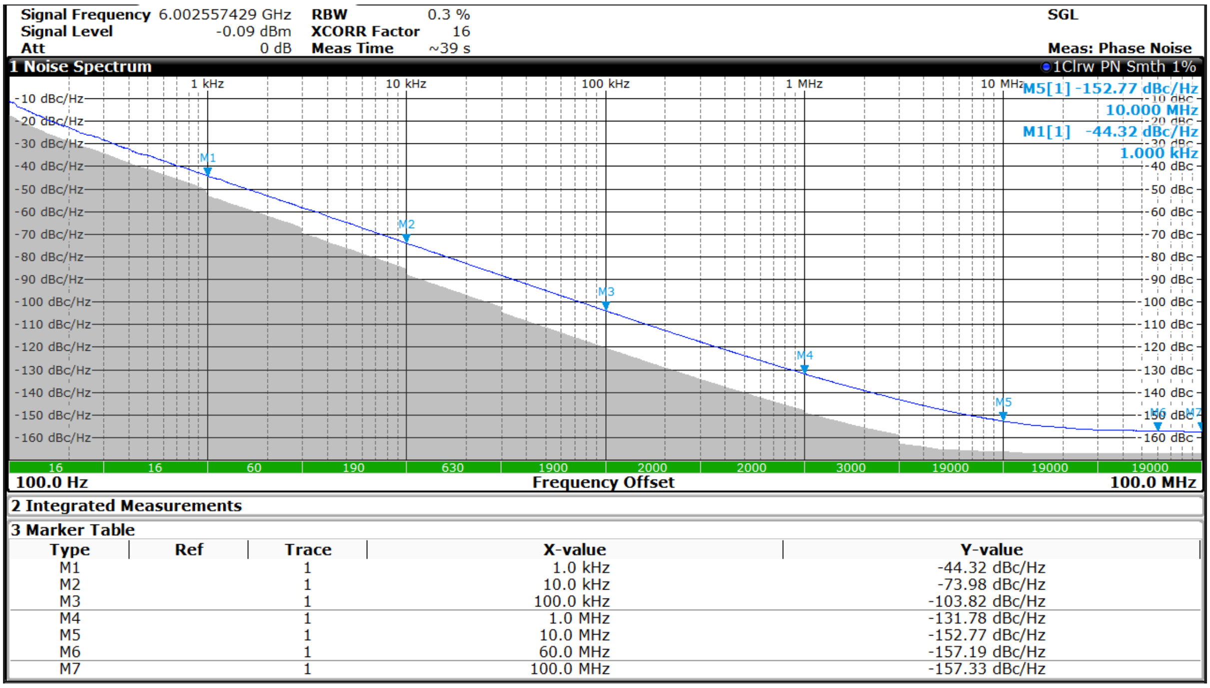


Figure 7. f_{VCO} = 6GHz at 90°C

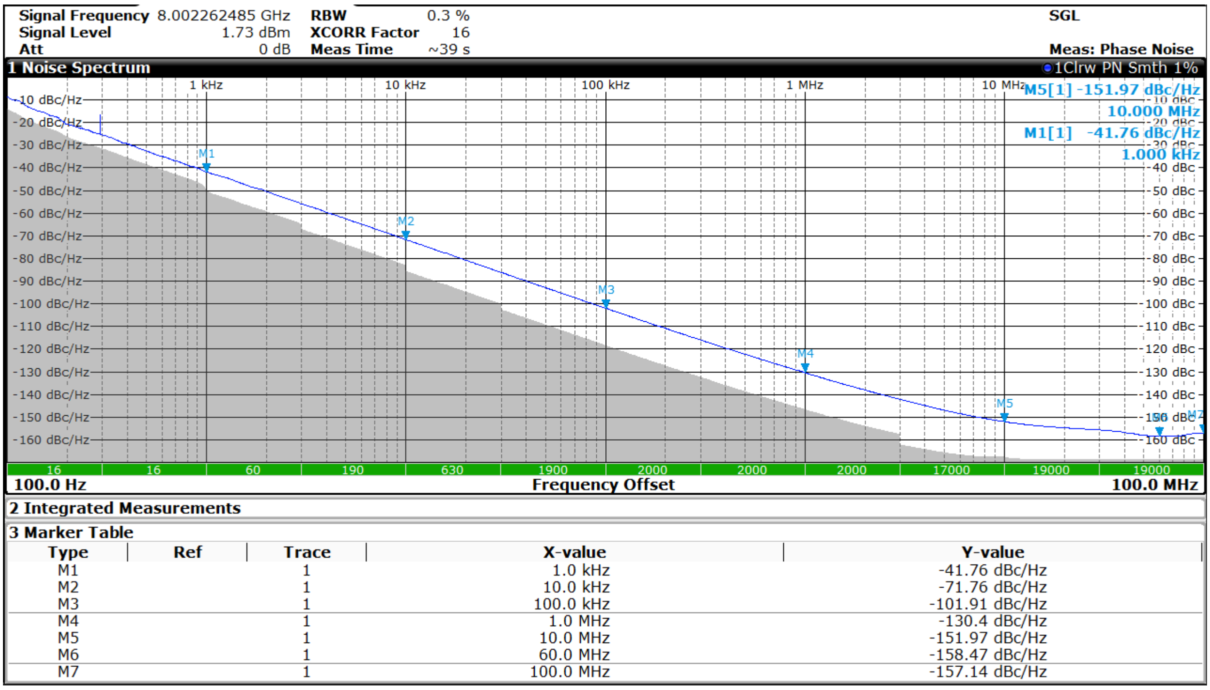


Figure 8. f_{VCO} = 8GHz at 25°C

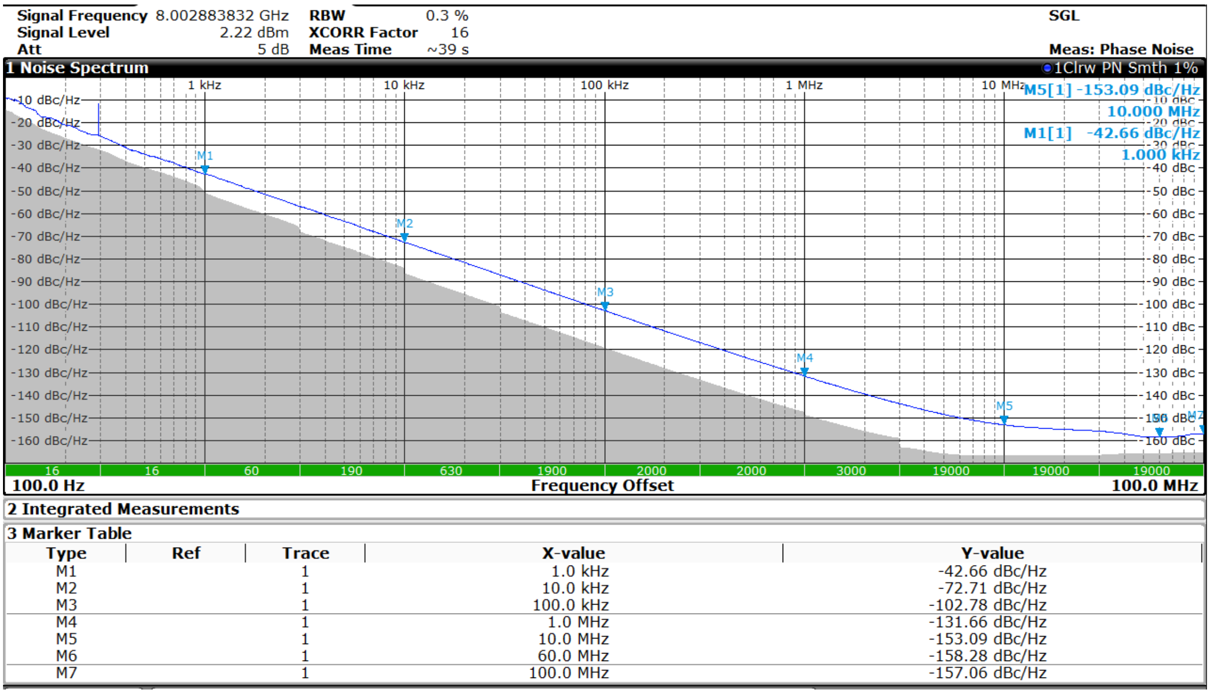


Figure 9. f_{VCO} = 8GHz at -40°C

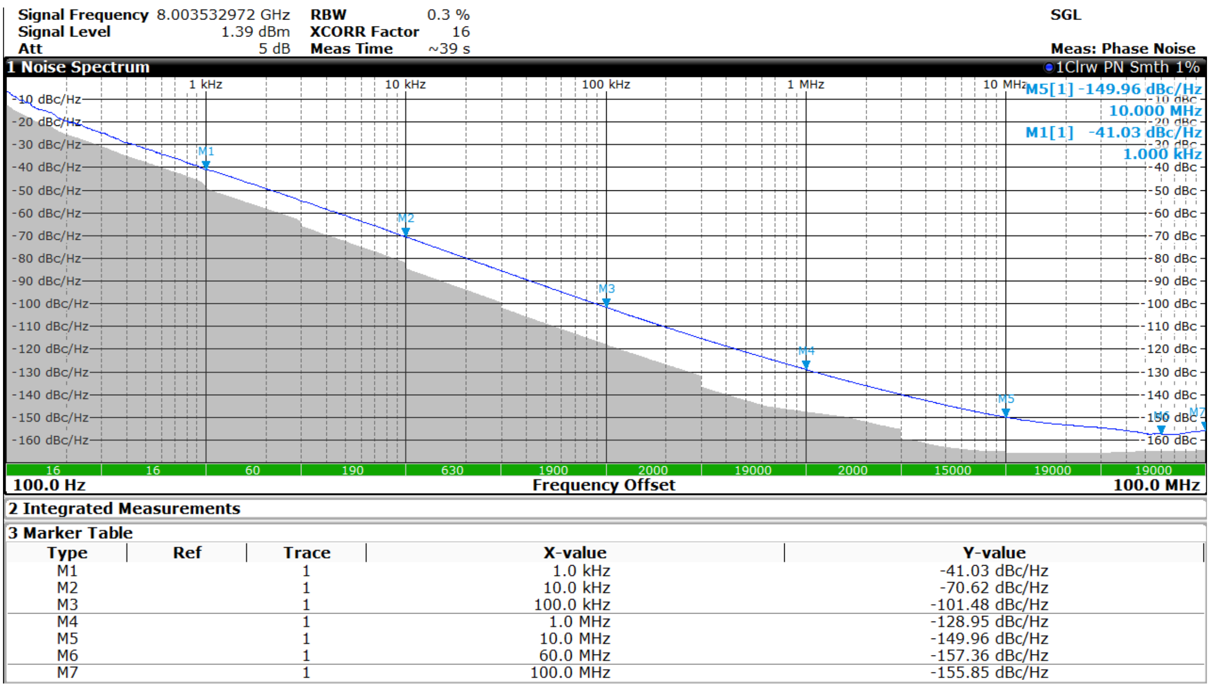
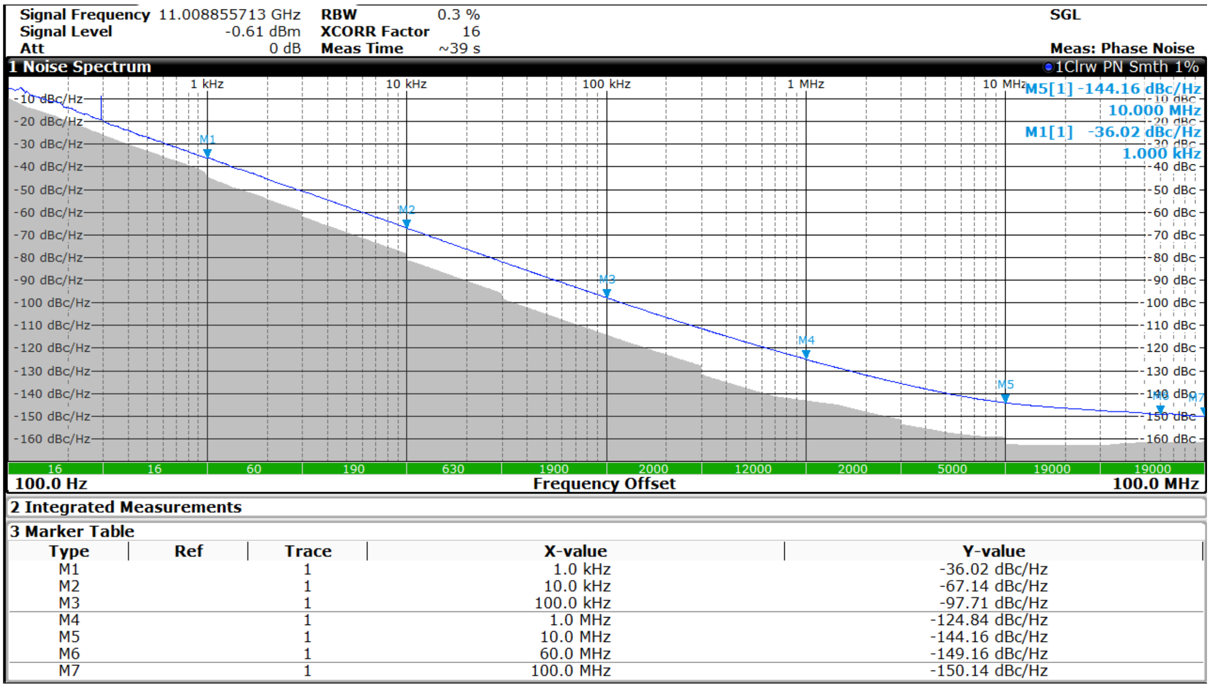
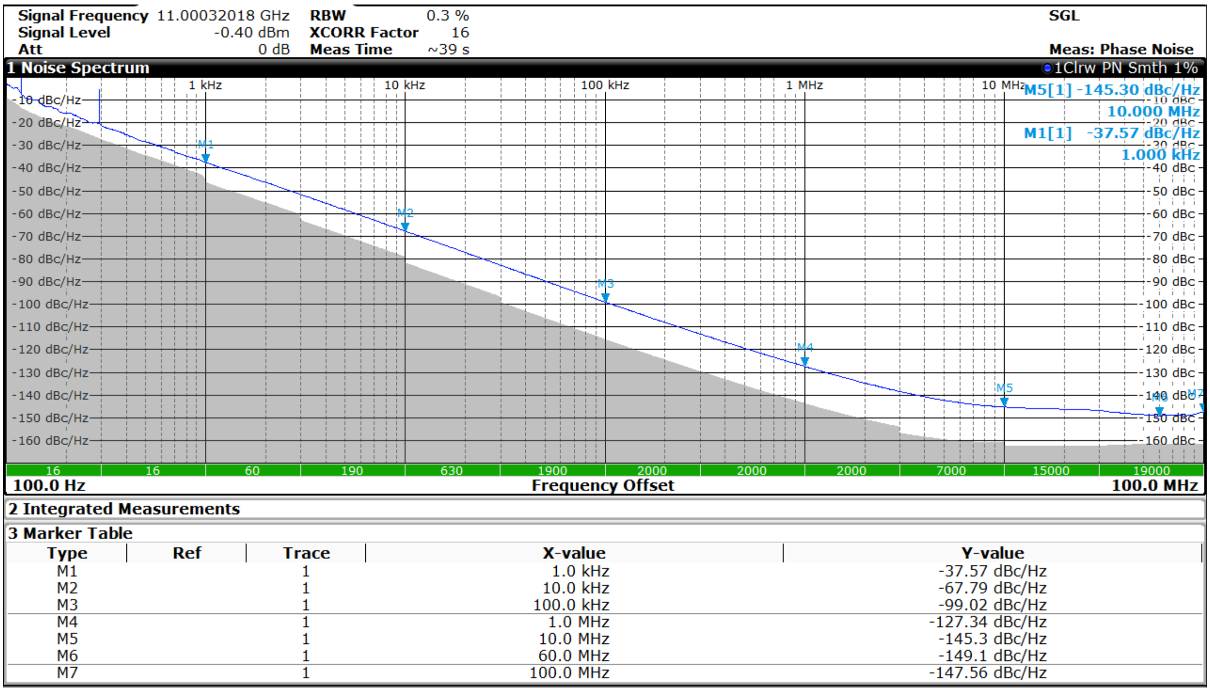


Figure 10. f_{VCO} = 8GHz at 90°C



Type	Ref	Trace	X-value	Y-value
M1		1	1.0 kHz	-36.02 dBc/Hz
M2		1	10.0 kHz	-67.14 dBc/Hz
M3		1	100.0 kHz	-97.71 dBc/Hz
M4		1	1.0 MHz	-124.84 dBc/Hz
M5		1	10.0 MHz	-144.16 dBc/Hz
M6		1	60.0 MHz	-149.16 dBc/Hz
M7		1	100.0 MHz	-150.14 dBc/Hz

Figure 11. f_{VCO} = 11GHz at 25°C



Type	Ref	Trace	X-value	Y-value
M1		1	1.0 kHz	-37.57 dBc/Hz
M2		1	10.0 kHz	-67.79 dBc/Hz
M3		1	100.0 kHz	-99.02 dBc/Hz
M4		1	1.0 MHz	-127.34 dBc/Hz
M5		1	10.0 MHz	-145.3 dBc/Hz
M6		1	60.0 MHz	-149.1 dBc/Hz
M7		1	100.0 MHz	-147.56 dBc/Hz

Figure 12. f_{VCO} = 11GHz at -40°C

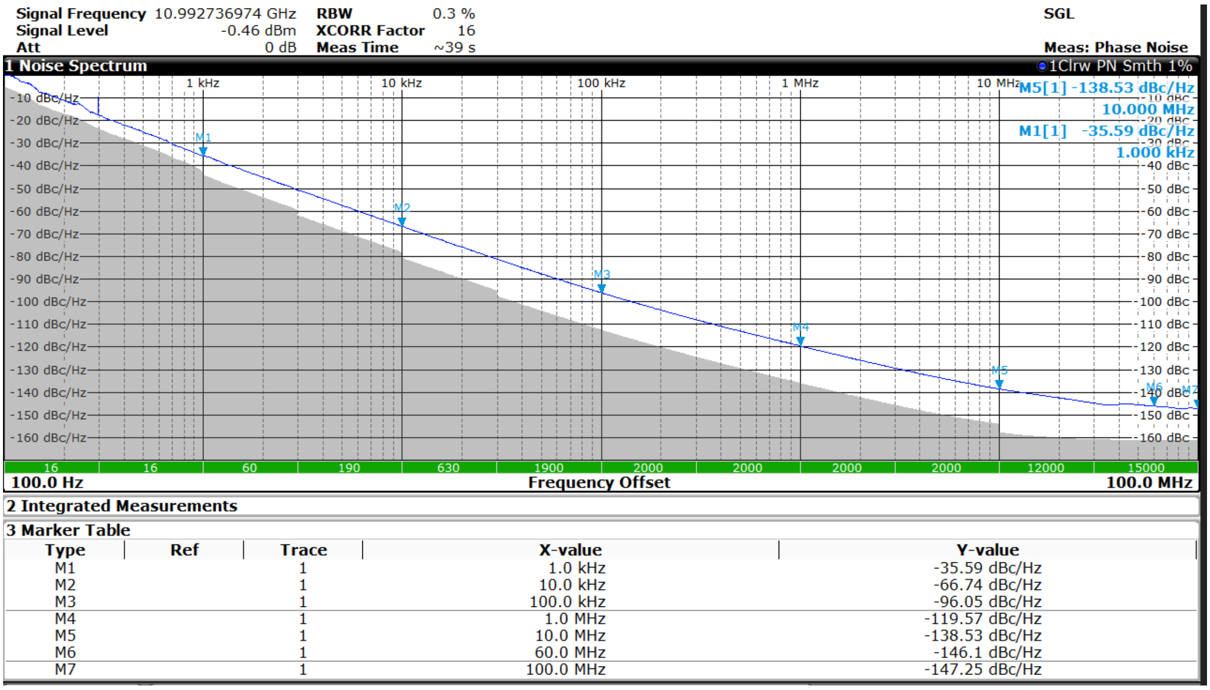


Figure 13. f_{VCO} = 11GHz at 90°C

10. Appendix B – Python Code

```
from fractions import Fraction
f_vco = raw_input ('Enter VCO frequency and Press Enter to continue ')
f_pfd = raw_input ('Enter PFD frequency and Press Enter to continue ')
## Calculate fractional feedback divider ratio
inputReal = (float(f_vco)/float(f_pfd))
## Convert decimal feedback divider to improper fraction
improper_fraction = Fraction(inputReal).limit_denominator(2**32)
## Get fractional part
frac_part = inputReal % 1.0
## Get denominator
denom = Fraction(inputReal).limit_denominator(2**32).denominator
## Get numerator
numer = frac_part*denom
## Find multiplcation factor to obtain equivalent fraction
## with numerator/denominator closest to 2^^32
a = (2**32 - 1)/numer
b = (2**32 - 1)/denom
multiplication_factor = min(a,b)
## Calculate Nfrac, Nmod
Nfrac = int(numer * multiplication_factor)
Nmod = int(denom * multiplication_factor)
print ('Nfrac value to be programmed to Nfrac[31:0] registers is '), Nfrac
print ('Nmod value to be programmed to Nmod[31:0] registers is '), Nmod
```

11. Revision History

Revision	Date	Description
1.0	Sept.23.20	Initial release.

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/.