Frequency Margining with VersaClock 7

This document describes the common methods of changing output frequency for VersaClock 7 (VC7) devices (RC21005, RC21008, RC21012, RC26008, RC26012, RC31005, RC31008, and RC31012). Changing an output frequency can be implemented by changing the VCO frequency of the APLL; the output divider; or both. If input frequency change or VCO frequency change, then re-calibration is required.

Contents

1. VersaClock 7 Block Diagram

The following figure shows a typical block diagram as displayed in a VersaClock 7 datasheet.

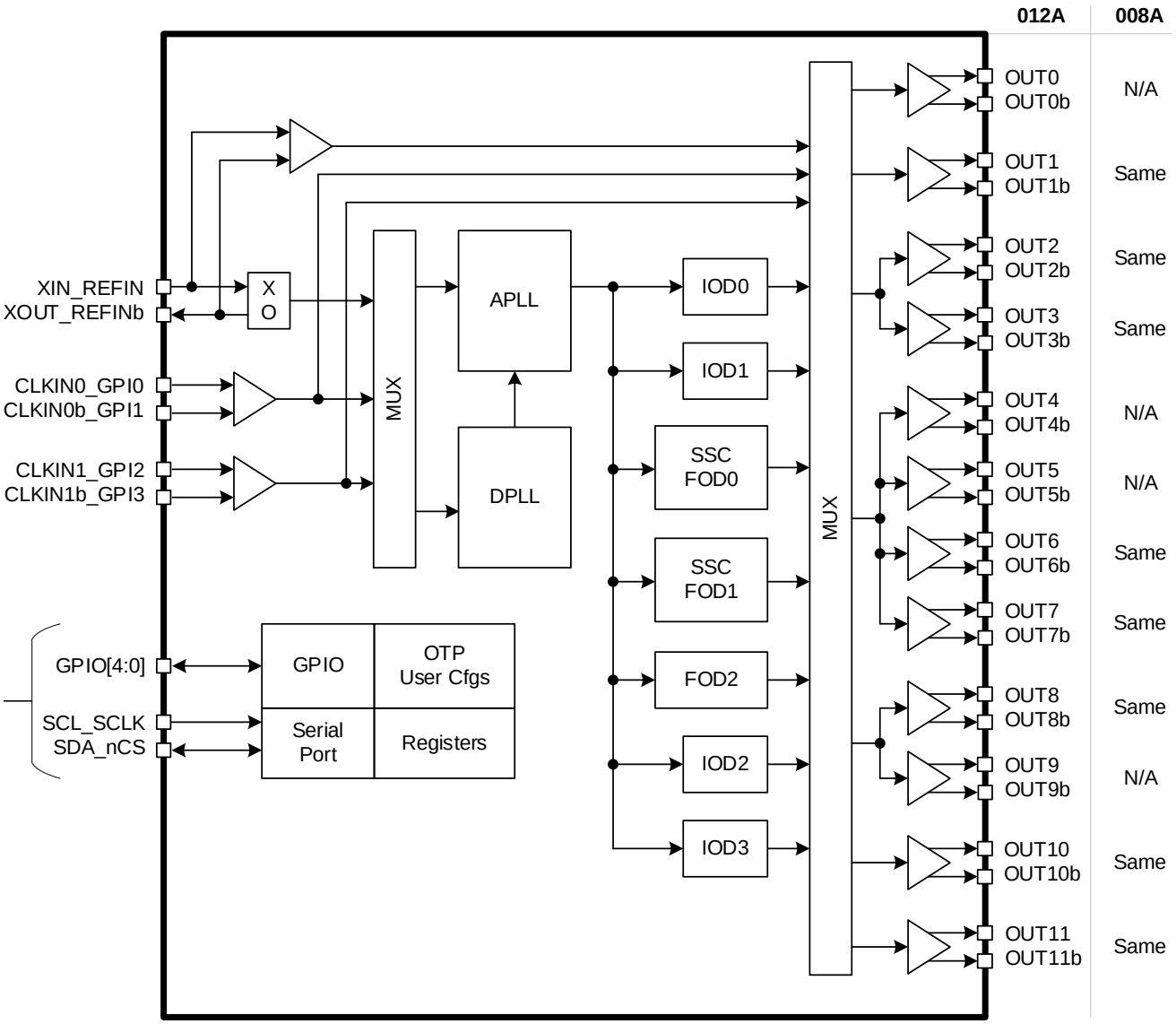

Figure 1. VersaClock 7 Block Diagram

The APLL VCO frequency is from a crystal or CLKIN multiplied by the Doubler and Feedback Divider. The high APLL frequency of around 10GHz (range of 9.5GHz to 10.7GHz) is then divided down to the outputs with output dividers.

- OUTn = VCO Frequency / Output Divider-n
- VCO Frequency = Input Frequency * Doubler * Feedback Divider
- Feedback divider = apll_fb_div_int + (apll_fb_div_frac / 2^{27})

Note: If the doubler is enabled, the Doubler = 2. If the doubler is disabled, the Doubler = 1.

2. I2C Addressing

VersaClock 7 I2C addressing is 0x09, and has a 912-byte register memory space. It also separates as 4 pages and each page is 256 bytes. So the usual 1 byte for the register address in I2C is not sufficient. There are two different methods to access all registers:

1. Paging 0, 1, 2 each have 256 registers and remaining registers in Page 3. the lower 8 bits of the register offset address come from the Offset Address byte and the upper 8 bits come from the page register. The page register can be accessed at any time using an offset byte value of 0xFD. Write to 0x00 for page 0, 0x01 for page 1, 0x02 for page 2, and 0x03 for page 3.
   - Write 0xFD 0x00 //Change page number to 0.
   - Write 0xFD 0x01 //Change page number to 1.
   - Write 0xFD 0x02 //Change page number to 2.
   - Write 0xFD 0x03 //Change page number to 3.

   Example: To program register value 64 (0x40) to page 1, 0x02 offset address:
   - Write 0xFD 0x01 //Change page number to 1.
   - Write 0x02 0x40 //Write register value 64 to 0x102 offset address
2. 2-Byte Addressing, the full 16-bit register address can be obtained from the Offset Address bytes, so the page register does not need to be set up. The MSB offset address will be the page number and the LSB address is the register address.

   Example: To program register value 64 (0x40) to 0x102 address:
   - Write 0x102 0x40 // Write register value 64 to 0x102 offset address

3. Registers

The following is a list of important registers for changing frequencies:

- IOD0, Integer Divider - 25 bits total:
  - Bits [7:0] in register 0x1C0[7:0]
  - Bits [15:8] in register 0x1C1[7:0]
  - Bits [23:16] in register 0x1C2[7:0]
  - Bit [24] in register 0x1C3[0]
- IOD1, Integer Divider - 25 bits total:
  - Bits [7:0] in register 0x1C8[7:0]
  - Bits [15:8] in register 0x1C9[7:0]
  - Bits [23:16] in register 0x1CA[7:0]
  - Bit [24] in register 0x1CB[0]
- IOD2, Integer Divider - 25 bits total:
  - Bits [7:0] in register 0x1D0[7:0]
  - Bits [15:8] in register 0x1D1[7:0]
  - Bits [23:16] in register 0x1D2[7:0]
  - Bit [24] in register 0x1D3[0]
- IOD3, Integer Divider - 25 bits total:
  - Bits [7:0] in register 0x1D8[7:0]
  - Bits [15:8] in register 0x1D9[7:0]
  - Bits [23:16] in register 0x1DA[7:0]
  - Bit [24] in register 0x1DB[0]

- FOD0, Fractional Divider - 60 Bits total:
  - FOD0 1st Integer - 9 Bits:
    - Bit [8] in register 0x1E1[0]
    - Bits [7:0] in register 0x1E0[7:0]
  - FOD0 2nd Integer - 17 Bits:
    - Bits [16:15] in register 0x1E3[1:0]
    - Bits [14:7] in register 0x1E2[7:0]
    - Bits [6:0] in register 0x1E1[7:1]
- FOD0 Fractional - 34 Bits:
  - Bits [33:30] in register 0x1E7[3:0]
  - Bits [29:22] in register 0x1E6[7:0]
  - Bits [21:14] in register 0x1E5[7:0]
  - Bits [13:6] in register 0x1E4[7:0]
  - Bits [5:0] in register 0x1E3[7:2]
- FOD1, Fractional Divider - 60 Bits total
- FOD1 1st Integer - 9 Bits:
  - Bit [8] in register 0x1F1[0]
  - Bits [7:0] in register 0x1F0[7:0]
- FOD1 2nd Integer - 17 Bits:
  - Bits [16:15] in register 0x1F3[1:0]
  - Bits [14:7] in register 0x1F2[7:0]
  - Bits [6:0] in register 0x1F1[7:1]
- FOD1 Fractional - 34 Bits:
  - Bits [33:30] in register 0x1F7[3:0]
  - Bits [29:22] in register 0x1F6[7:0]
  - Bits [21:14] in register 0x1F5[7:0]
  - Bits [13:6] in register 0x1F4[7:0]
  - Bits [5:0] in register 0x1F3[7:2]
- FOD2, Fractional Divider - 60 Bits total:
  - FOD2 1st Integer - 9 Bits:
    - Bit [8] in register 0x201[0]
    - Bits [7:0] in register 0x200[7:0]
  - FOD2 2nd Integer - 17 Bits:
    - Bits [16:15] in register 0x203[1:0]
    - Bits [14:7] in register 0x202[7:0]
    - Bits [6:0] in register 0x201[7:1]
- FOD2 Fractional - 34 Bits:
  - Bits [33:30] in register 0x207[3:0]
  - Bits [29:22] in register 0x206[7:0]
  - Bits [21:14] in register 0x205[7:0]
  - Bits [13:6] in register 0x204[7:0]
  - Bits [5:0] in register 0x203[7:2]

- APLL Feedback Divider - 37 Bits:
  - Integer - 10 Bits:
    - Bits[9:8] in register 0x125[1:0]
    - Bits[7:0] in register 0x124[7:0]
  - Fractional - 27 Bits:
    - Bits [26:24] in register 0x123[2:0]
    - Bits [23:16] in register 0x122[7:0]
    - Bits [15:8] in register 0x121[7:0]
    - Bits [7:0] in register 0x120[7:0]

4. Changing Frequencies with Synthesizer/DCO Mode

In synthesizer/DCO mode, the APLL locks to one of the following: crystal input XIN_REFIN/XOUT_REFINb (refin internally), CLKIN0/CLKIN0b (clkin0 internally), or CLKIN1/CLKIN1b (clkin2 internally), and generates output clocks as programmed. The output clock frequency can be tuned in granularity of approximately 1ppb and range of up to ±243ppm via serial interface commands. The APLL reference clock selection is a static setting and dynamically switching between references is not supported.

4.1 Changing an Output Frequency with the Output Divider

The easiest way to change an output frequency is to change the output divider for that output. Each output can be controlled individually because each output has its own output divider. Frequency change is glitchless and easy. With VCO frequency in the range of 9.5GHz to 10.7GHz, VC7 devices support an output frequency range of 1KHz to 650MHz by choosing a proper output divider value. If the desired output frequency is in an integer relationship with VCO frequency, an IOD is used; otherwise, an FOD is used. There are four IOD (IOD0-3) and three FODs (FOD0-2).

Note: If input frequency changes or VCO frequency changes, then re-calibration is required. All changes made through I2C will be temporary. When power is cycled, the original configuration is loaded from OTP memory.

4.1.1 Changing Integer Output Dividers (IOD)

Integer Output Dividers (IOD) are simple to use in your design. Each IOD is 25 bits with valid values between 14 and 33,554,431 (= 2^{25} - 1). A value smaller than 14 causes the output frequency to increase above its maximum 650MHz and the biggest value makes an output frequency below the minimum 1KHz. For example, for a 78.125MHz crystal for 156.25MHz outputs, where VCO is 10GHz, the output dividers will be 0x40(64) for 156.25MHz. When changing an IOD from 0x40(64) to 0x3F(63), the output frequency increases to 10G/63 = 158.7302MHz. 156.25MHz X (1 + 1.58725%) = 158.73MHz. This 1.58725% is the smallest step size and is the resolution for changing a 156.25MHz output using the integer output divider. As a result, the disadvantage of using an output divider to change an output frequency is that the resolution is relatively large and changes with the output frequency. For example, when the output frequency is 312.5MHz with an output divider value of 32(0x20), the resolution or step size is 3.2258%.

The following is the 1-Byte Offset Mode Programming Steps - Output divider IOD.

Example 1: Programming the registers in this example will change the output to 100MHz. VCO is 10GHz, to make OUT0 = 100MHz by change IOD0 to 100 (VCO frequency = 10GHz)

- Output Frequency = VCO Frequency / Output Divider
- 100MHz = 10GHz / 100

Programming these registers will change the output to 100MHz. VCO is 10GHz:

- Write 0xFD 0x01 (write 0x01 to 0xFD), to change the register address page to 1.
- Write 0xC0 0x64 (write 0x64 to page1, 0xC0), to change OUT0 =100MHz by changing IOD0 to 100 decimal.

Output sync up below are optional, only when the application is required:

- Write 0xFD 0x00, to change the register address page to 0.
- Write 0x11 0x30, to reset output divider.
- Write 0x11 0x32, to sync up output divider.
- Write 0x11 0x30, reset output divider.

The following is an example of the response time of output divider sync up.

The sync up output divider step is with glitch, and the response time of output divider sync up is around 4.69us; this is optional only when an application is required.

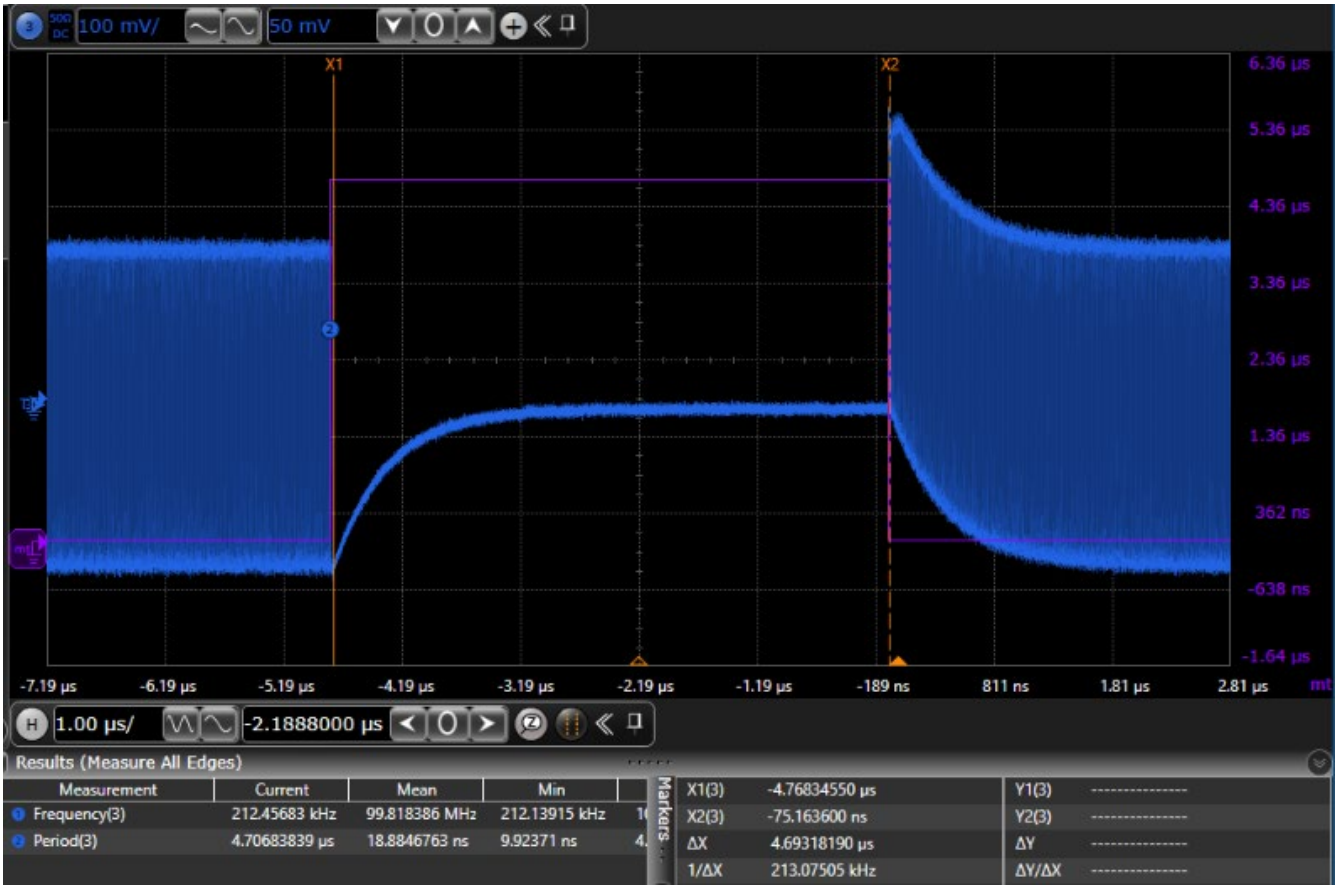

Figure 2. Response Timing of Output Sync Up

Example 2: Assuming the APLL VCO is 10.000GHz.

- Write 0xFD 0x01 to change the register address page to 1.
- Write 0xC0 0x3F to change OUT0 = 156.25MHz to 158.73MHz (+1.5873%) by change IOD0 to 63.
- Write 0xC0 0x41 to change OUT0 = 156.25MHz to 153.85MHz (-1.5386%) by change IOD0 to 65.
- Write 0xC8 0x3F to change OUT1 = 156.25MHz to 158.73MHz (+1.58725%) by change IOD1 to 63.

As you can see, this method is also useful to quickly change an output frequency to anything that can be divided down from 10GHz with an integer value.

4.1.2 Changing Fractional Output Divider

The Fraction divider consists of a 9-bit 1st integer, a 17-bit 2nd integer and a 34-bit fractional portion. The 1st stage integer covers values up to 511 and is always used. The 1st stage divides the VCO clock down to a range of 33MHz to 650MHz, giving a minimum divide ratio of 9.5GHz / 650Mhz = 14.61 and maximum divide ratio of 10.7GHz / 33 MHz = 324.25. The 2nd stage integer can cover values up to 131,071 and is used for output

frequencies below 33MHz. Fractional output divider ratio 2nd integer portion half integer divide ration of the second stage. The actual divide ratio is (fod_2nd_int*2). A setting of 1 is invalid: the minimum divide ratio is 4.

The formula for the total divider value is:

- FOUT > 33MHz: FOD = 1st Integer + Fraction
- FOUT < 33MHz: FOD = (1st Integer + Fraction) × 2 × 2nd Integer

For output frequencies above 33MHz, where the 2nd Integer is not used, the 2nd Integer value is set to 0. This causes the 2nd Integer and its additional ×2 to be bypassed.

Example 1: To make OUT0 = 156.25MHz by programming FOD0 = 64 (VCO frequency = 10GHZ)

- Output Frequency = VCO Frequency / Output Divider
- 156.25MHz = 10GHz / 64

Stage 2 (FOD0 2nd integer) is 0 to bypass since 156.25MHz > 33MHz.

- Write 0xFD 0x01 to change the register address page to 1.
- Write 0xE0 0x40 0x00 0x00 0x00 (this is a “block write” of multiple registers. 0x1E0 is the start address. 0x40 is written to page1 0xE0, 0x00 to page 1 0xE1, 0x00 to page 1 0xE2, and 0x00 to page 1 0xE3.)

Programming these registers will change the output to 156.25MHz.

Example 2: In order to make Out0 = 30MHz < 33MHz, by programming FOD0, VCO frequency = 10GHz:

- Output divider = 10GHz / 30MHz = 333. 333333333333333333
- FOD = (1st Integer + Fraction) × 2 × 2nd Integer
  = (83 + 0.333333333333333333) x 2 x 2 = 333.333333333333333333
- 1st integer = 83 = 53 hex

Factional register setting is 0.333333333 x (2^{34} - 1) = 5726623061. And the number is then rounded up to 1 55 55 55 55 hex .

The register map for FOD0 is as follows.

Table 1. Register Map for FOD0

Address	D7	D6	D5	D4	D3	D2	D1	D0
0x1e0	fod_1st_int[7]	fod_1st_int[6]	fod_1st_int[5]	fod_1st_int[4]	fod_1st_int[3]	fod_1st_int[2]	fod_1st_int[1]	fod_1st_int[0]
0x1e1	fod_2nd_int[6]	fod_2nd_int[5]	fod_2nd_int[4]	fod_2nd_int[3]	fod_2nd_int[2]	fod_2nd_int[1]	fod_2nd_int[0]	fod_1st_int[8]
0x1e2	fod_2nd_int[14]	fod_2nd_int[13]	fod_2nd_int[12]	fod_2nd_int[11]	fod_2nd_int[10]	fod_2nd_int[9]	fod_2nd_int[8]	fod_2nd_int[7]
0x1e3	fod_frac[5]	fod_frac[4]	fod_frac[3]	fod_frac[2]	fod_frac[1]	fod_frac[0]	fod_2nd_int[16]	fod_2nd_int[15]
0x1e4	fod_frac[13]	fod_frac[12]	fod_frac[11]	fod_frac[10]	fod_frac[9]	fod_frac[8]	fod_frac[7]	fod_frac[6]
0x1e5	fod_frac[21]	fod_frac[20]	fod_frac[19]	fod_frac[18]	fod_frac[17]	fod_frac[16]	fod_frac[15]	fod_frac[14]
0x1e6	fod_frac[29]	fod_frac[28]	fod_frac[27]	fod_frac[26]	fod_frac[25]	fod_frac[24]	fod_frac[23]	fod_frac[22]
0x1e7	fod_pd	fod_dis	fod_rst	fod_acc_reset	fod_frac[33]	fod_frac[32]	fod_frac[31]	fod_frac[30]

- Write 0xFD 0x01 to change the register address page to 1.
- Write 0xE0 0x53 0x04 0x00 0x54 0x55 0x55 0x55 0x05.
  - Write 0x53 to page1 0xE0, for 1st integer divider 83.
  - Write 0x04 to page 1 0xE1, for 2nd integer divider 2.

- Write 0x00 to page 1 0xE2, write 0x54 to page 1 0xE3, write 0x55 to page 1 0xE4, write 0x55 to page 1 0xE5, write 0x55 to page 1 0xE6, write 0x05 to page 1 0xE7, for fractional divider 572662306.

Programming these registers will change the output to 30MHz.

4.2 Changing an Output Frequency with the APLL Feedback Divider

The APLL Feedback Divider is a Fractional Divider with 27 bits precision for the fraction.

For the best phase noise performance, Renesas recommends configuring the APLL with an integer value for the feedback divider. When trying to make small, precise frequency changes, it is easier to work with fractions for the APLL feedback divider. For example, the VC7 with a 78.125MHz crystal for 156.25MHz or 312.5MHz outputs, where the VCO is 10GHz. The output dividers will be 64 for 156.25MHz and 32 for 312.5MHz. Assuming the doubler between crystal and phase detector is enabled, the phase detector frequency will be 156.25MHz and the feedback divider is 64.

The following describes the 1-Byte Offset Mode programming steps - APLL feedback divider.

Example: All outputs are 156.25MHz, the VCO is 10GHz, and the crystal is 78.125MHz. The objective is to increase each output with exactly +1% to 157.8125MHz.

The original APLL feedback divider is 10,000 / 78.125 / 2 = 64.

We want to change that number to 1.01 × 64 = 64.64 (+1%).

We only need to change the fraction to 0.64 to make all output frequencies go up 1%.

The fraction register setting must be $0.64 \times 2^{27} = 85,899,345.92$.

The number is then rounded up to 85,899,346 = 05 1E B8 52 hex.

The errored factional divider value is $85,899,346 / 2^{27} = 0.6400000006$.

The rounding causes a tiny error of (0.6400000006 - 0.64)/64.64 = 0.009ppb.

- Write 0xFD 0x01, to change the register address page to 1.
- Write 0x20 0x52 0xB8 0x1E 0x05, to change the fraction value to 0.64.
- Write 0x28 0x44 0x0D, to reconfigures the charge pump current.
- Write 0x2A 0x73 and write 0x2B 0x13, to reconfigure loop filter resistors and capacitors.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0xA0, APLL re-calibrate. Change register 0x013C from 20 to A0 to re-calibrate.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.

All outputs should now toggle at 157.8125MHz.

From here to all outputs at -1% or 154.6875MHz requires the feedback divider to be reprogrammed to 0.99 × 64 = 63.36 (-1%). Now we also need to update the integer value of the feedback divider.

The integer register setting will be 63 or 3F#.

The Fraction register setting will be $0.36 \times 2^{27} = 48,318,382.08$. We round that down to 48,318,382. The hex value will be 02 E1 47 AE.

Hint: To convert decimal to hex values, use Microsoft Excel® and type DEC2HEX (48318382,8). The “8” is for 8 hex characters.

The integer registers are right behind the fraction registers so this update can be done with one block write:

- Write 0xFD 0x01 to change the register address page to 1.
- Write 0x20 0xAE 0x47 0xE1 0x02 0x3F 0x00 to change the fraction value to 0.36 and the integer value to 63.
- Write 0x28 0x44 0x0D, to reconfigures the charge pump current.
- Write 0x2A 0x73 and Write 0x2B 0x13, to reconfigure loop filter resistors and capacitors.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.
- Write 0x3C 0xA0, APLL re-calibrate. Change register 0x013C from 0x20 to 0xA0 to re-calibrate.
- Write 0x3C 0x20, dummy write, to reset the calibration bit.

All outputs should now toggle at 154.6875MHz.

4.3 Comparison Between FOD Factional and APLL Feedback Divider Faction

Table 2. Compares the Behavior Between FOD Factional and APLL Feedback Divider Faction

	FOD Factional Part	APLL Feedback Divider Factional Part
Resolution without XTAL trim	2^{-34}, 58 ppt	2^{-27}, 7.45 ppb
Resolution with XTAL trim (2^{-8})	2^{-42}, 0.227 ppt	2^{-35}, 29.1 ppt
Glitch or glitch less	Glitch less	Glitch
Affected output	Selected output	All outputs

5. Changing Frequencies with Jitter Attenuator Mode

In jitter attenuator mode, the APLL locks to the crystal inputs refin. The APLL is nested inside the DPLL as a DCO, which frequency is controlled by DPLL FCW (frequency control word). The DPLL locks to one of CLKIN0/CLKIN0b/CLKIN1/CLKIN1b (internally clkin0/1/2/3) and performs jitter attenuation with a programmable bandwidth from 0.1Hz to 18kHz. All output clocks are tracking to DPLL reference clocks when DPLL is in normal lock mode. The input clocks are monitored and the DPLL may automatically switch to a different qualified input clock if the selected one becomes disqualified.

The RC310 series also can be configured as Jitter Attenuators. What this means is that the DPLL is controlling the APLL to keep the APLL synchronized with the reference clock input. When making a change to the APLL feedback divider, the DPLL must be reprogrammed as well to be synchronizing for the new APLL frequency. For this version of the document, we will not include the required changes to the DPLL settings when changing the APLL feedback divider. Renesas also recommends limiting output frequency changes with a Jitter Attenuator to changes of the integer output dividers.

- The RC210 series of Synthesizers → Full control of both APLL feedback divider and output dividers.
- The RC310 series of Jitter Attenuators → Only control output dividers.
- The RC310 series can also be configured as Synthesizers → both the APLL feedback divider and output dividers.

6. Revision History

Revision	Date	Description
1.00	May 4, 2023	Initial release.

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

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.