# RC310xx

This document describes the functional description, register organization, and byte addresses of the RC310xx. Detailed register definitions can be located by following the links in the document.

## Contents

# 1. Functional Description

The RC310xx is a small-form factor, fully integrated, low-power, high-performance frequency synthesizer with jitter attenuation and network synchronization capabilities. The device is optimized to deliver excellent phase noise as required for driving Ethernet PHYs/switch, ASICs, or FPGAs. The RC310xx supports JEDEC JESD204B/C for converter synchronization, JEDEC JESD204B/C converter synchronization, IEEE1588, and SyncE for network-based synchronization.

The following sections provide an overview of the RC310xx.

## 1.1 Power-Up, Configuration, and Serial Interfaces

The RC310xx can be powered up and configured in three ways:

- From internal non-volatile memory using OTP user configurations (UserCfgs)
- From its slave serial interface
- From an external I2C EEPROM

The power-up sequence loads one of up to 27 *internal* UserCfgs from OTP. This configuration can be selected via external GPIO pins or by programming a field to select the default configuration to load. This is useful when external GPIO are not used to select the UserCfg at power-up. After the device is powered up, the slave serial interface can be used to select preprogrammed UserCfgs or load entirely new UserCfgs stored outside the device.

The RC310xx supports three slave serial interfaces: I2C, SPI, SMBUS, and one serial master interface (I2C).

These interfaces share the same pins so only one is available at a time. The I2C master interface that is used to load UserCfgs from an external I2C EEPROM is only active after loading an OTP UserCfg that indicates a further load from external I2C EEPROM. An external master (I2C, SPI, SMBUS) can be used to access internal registers. If the slave serial port is configured as SPI then loading from external EEPROM is not available.

## 1.2 Input Clocks

The RC310xx supports one crystal/reference input that is used as a reference to the analog PLL (APLL), and up to two differential or four single-ended clock inputs that is used as a reference to the digital PLL (DPLL) and support hitless reference switching.

### 1.2.1 Crystal/Reference Input

The crystal input supports crystal frequencies of 8MHz to 80MHz. It has programmable internal load capacitors to support crystals with CL = 6pF to 12pF. Internal crystal variants of RC310xx support a trim value in OTP that can be set during ATE to compensate for initial frequency offset of the internal crystal. The crystal input supports being overdriven with a clipped sine-wave TCXO with 0.8VPP signal.

If an external square wave clock is used to drive the XIN_REFIN pin, the default is amplitude limit is 1.2V. If sel_ib_xo is set to 0 and xo_ib_cmos_sel is set to 1 amplitude limitations are determined by the VDDX supply rail.

The crystal input can be overdriven with differential or single-ended inputs with proper external terminations. The supported frequency range is same as reference clock inputs:

- 1kHz to 650MHz in differential mode
- 1kHz to 200MHz in single-ended mode

An available LOS monitor detects the loss of signal on crystal input.

### 1.2.2 Clock Inputs

There are two differential clock inputs that support LVDS, HCSL, or single-ended CMOS logic levels without external terminations. LVPECL or CML clock inputs can be supported with external terminations and/or AC coupling. Internal terminations are available for both HCSL and LVDS logic levels. Additionally, HCSL input terminations support both 100ohm and 85ohm operating environments.

If the cmos_sel register bit is set to single-ended type, the differential inputs turn into two single-ended inputs. CLKIN0 drives clkin0 internally, CLKIN0b drives clkin1 internally. CLKIN1 drives clkin2 internally, and CLKIN1b drives clkin3 internally. If set to differential type, CLKIN0/CLKIN0b pair drives clkin0, while CLKIN1/CLKIN1b pair drives clkin2. Internal biasing is available for AC-coupled applications. The two clock inputs can be left floating when unused. An available LOS monitor detects the loss of signal on crystal input. The LOS status is stored in register bits and can also be steered to a GPIO pin. Frequency monitoring is also available on the clock inputs.

### 1.2.3 Clock Input Monitors

There are two types of reference clock monitors. The APLL input is monitored for Loss of Signal (LOS). While the DPLL clock inputs (CLKIN0, CLKIN0B, CLKIN1, CLKIN1B) each have LOS, activity, and frequency monitoring.

- The LOS monitor detects missing edges over a window of several reference clock periods. For the best accuracy, it is recommended to program the window to be equal to at least 8 times that of the measuring clock period.
- The frequency monitor can be configured to measure the reference over a nominal 5ms time window in order to achieve ~1ppm granularity.
- The frequency monitor can be configured to measure the reference over a nominal 0.4s time window in order to achieve ~12ppb granularity.

## 1.3 APLL

The APLL is fractional LC-VCO based PLL with an operating range from 9.5GHz to 10.7GHz. Any of the available input clocks can be selected to drive the APLL, and the input clock can be frequency doubled for increased performance. The APLL is temperature compensated for the utmost frequency stability. For synchronous, deterministic requirements, the APLL also supports ZDB mode where CLKIN0 is used for the feedback input.

### 1.3.1 APLL Lock Detector

The APLL lock detector indicates whether the APLL is locked to a functioning crystal or reference input by monitoring the phase errors. Lock status can be sent on to a GPIO pin and register apll_lock_sts. The falling edge of apll_lock_sts sets apll_lol event sticky bit. The apll_lol event also increments a 7-bit loss of lock counter that starts from power on. The counter values represents the total number of loss of lock since power on and can be read back from register apll_lol_cnt.

## 1.4 DPLL

To operate as a network synchronizer or jitter attenuator, the DPLL and APLL are nested and form a fractional-N DPLL architecture. The System APLL locks to an input clock from a crystal or a crystal oscillator and generates an output clock of approximately 10GHz. The APLL uses a fractional feedback divider with 26-b numerator and fixed 26-b denominator to generate its feedback clock. The fractional feedback divide ratio is dynamically controlled by the DPLL. The DPLL also uses the APLL's VCO clock to generate the fractional divided DPLL feedback clock. The DPLL fractional feedback divider, which is comprised of 48-b numerator and 48-b denominator, is static during normal operation. The DPLL can also be optionally disabled to operate the RC31008/31012A in synthesizer/DCO mode.

### 1.4.1 DPLL Loop Filter

The all-digital DPLL loop filter is a Proportional - Integral (PI) type filter with a bandwidth that is programmable from approximately 0.06Hz to 14kHz.

## 1.5 DPLL Reference Selection

The DPLL can lock to either of the two differential or the four single-ended input clocks. The reference selection can be either automatic or manual and when enabled, hitless switching results in negligible (< 100ps) output clock initial phase hit during reference switching or the DPLL exiting from holdover.

### 1.5.1 Manual Reference Selection

In manual mode, the selection is set either by register or by pin which is set by dpll_ref_sel_mode

- By register, the selection is set by dpll_ref_sel
- By pin, the selection is set by two GPI or GPIO pins assigned as DPLL_CLK_SEL[1:0] using gpi_func or gpio_func.
  - 00 = clkin0
  - 01 = clkin1
  - 10 = clkin2
  - 11 = clkin3

### 1.5.2 Automatic Reference Selection

In automatic mode, the selection is based on clock quality statuses and priorities. The quality statuses are from clock monitors. The priorities can be re-programmed in register ref_priority. If two clock inputs are programmed to the same priority, the one with lower index number takes precedence, for example, clkin0 takes precedence over clkin1.

The automatic reference selection can be either revertive or non-revertive, which is set in register dpll_revertive_en. In revertive mode, the reference clock that is qualified and of the highest priority is always selected. If a reference clock of higher priority than the currently selected one becomes qualified, the DPLL will switch to that reference clock; if a reference clock of equal or lower priority than the currently selected one becomes qualified, the DPLL will keep the current reference clock. In non-revertive mode, if there is a higher priority reference clock coming back (from disqualified to qualified), the current selected reference clock remains selected unless it gets disqualified.

### 1.5.3 Hitless Reference Switching

If hitless switching is enabled by setting register dpll_hitless_en to 1, the output clock initial phase hit will be minimized (< 100ps) during reference switching or the DPLL exiting from holdover, while the input clock and output clock may no longer be aligned.

If hitless switching is disabled, the output clock phase change slope is determined by DPLL loop characteristics and phase slope limit settings in register phase_slope_limit.

Minimal initial phase hit of < 100ps can only be met during reference switching when the reference clocks are of same fractional frequency offset. If they are of different fractional frequency offset (up to 200ppm), the output clock phase will track to the new reference clock. Although the initial phase hit can be minimized by setting dpll_hitless_en to 1, the total amount of output phase change and the change slope depends on the fractional frequency offset difference, the loop characteristics, and phase slope limit settings.

When dpll_hitless_en is set to 1, the phase_offset register is ignored by the DPLL and only the internally stored hitless offset affects the input-output phase offset. When dpll_hitless_en is set to 0, the hitless phase offset stored in the hitless handler is reset and the phase_offset register is used again. Hitless switching minimizes the output phase movement at the expense of a defined input – output phase offset while the use of the phase offset register enables a defined input-output phase offset at the expense of output phase movement during reference switching.

## 1.6 DPLL Operating Modes

The DPLL can operate in six different states: Free-run, Acquire, Normal, Holdover, Hitless-switch, and Write-frequency. The state transitions can be either manual or automatic, and are set in register dpll_mode. In manual mode, the states of Free-run, Holdover, and Write-frequency can be forced in the register dpll_mode.

### 1.6.1 Free-run State

During power-on reset or VCO calibration or in synthesizer mode, the DPLL is in the Free-run state. In this state, no reference clock is used and the output clocks are tracking the APLL reference clock.

### 1.6.2 Acquire State

When there is at least one qualified reference, the DPLL will track the selected qualified reference at the acquisition bandwidth and damping factor settings. If the reference clock is disqualified and no other qualified reference clock is available, the DPLL transitions to either the Free-run state or the Holdover state depending on the value of los_to_freerun. When lock-detector detects a lock, DPLL transitions to the normal state.

### 1.6.3 Normal State

In the Normal state, the DPLL is tracking the selected reference clock with the normal locking bandwidth and damping factor settings.

If the selected reference clock is disqualified, the state machine goes to either the Holdover or the Free-run state according to los_to_freerun. At a reference switch, the state machine goes via the Holdover state to the Hitless-switch state or the Acquire state.

### 1.6.4 Holdover State

In the Holdover state, the DPLL output frequency will be held at the instantaneous value or a a value that is low pass filtered and/or restored from the holdover history registers. This can be selected through the CSR register HOLDOVER_CNFG - Holdover Configuration. The initial holdover accuracy is less than 50ppb.

### 1.6.5 Hitless Switch State

At a Hitless reference switch or a hitless transition from the holdover state, the DPLL's TDC will measure the phase offset between the (newly) selected reference clock and the feedback clock, both of which are averaged. This offset is stored in an internal phase offset register. As a result, the output clocks will experience a minimal phase transient due to the reference switch or coming out of holdover. After the hitless switch procedure is finished, the state machine transitions to the Acquire state unless the reference clock fails.

### 1.6.6 Write-frequency State

In the Write-frequency mode the DPLL is not tracking any reference clock. The DPLL output frequency offset is directly controlled by the write_freq value.

### 1.6.7 Manual Mode

The DPLL operation can be forced to the Free-run, Holdover, and Write-frequency states by configuring register dpll_mode.

## 1.7 DPLL Lock Detector

The DPLL lock detector declares lock when the phase from the phase detector remains within a programmable range that is set in register dpll_lock_thresh for a programmable time interval that is set in register dpll_lock_timer. This indicates that the DPLL is locked to the reference clock input. Once the phase output from the phase detector has been below the lock threshold for half of the programmed lock interval, the internal lock signal is asserted and the normal loop filter bandwidth and damping is applied to the DPLL's loop filter instead of the acquire filter settings. The lock signal can be indicated on the GPIO pins and dpll_lock_sts register is asserted when the phase from the phase detector has been within the lock threshold for the full lock interval. Lock status can be sent on to pin and register dpll_lock_sts.

If the phase output from the phase detector exceeds the lock threshold for more than half the lock interval time, the lock status bit is de-asserted. The loss of lock event will increment a 7-bit loss of lock counter that starts from power on. The counter values represents the total number of loss of lock since power on and can be read back from register dpll_lol_cnt.

## 1.8 Output Dividers

The RC310xx provides four integer and three fractional output dividers.

### 1.8.1 Integer Output Dividers

All four IODs are identical and use a 25-bit divider to provide output frequencies from 1kHz to 650MHz derived from the VCO clock. Changing IOD values results in an immediate change to the new frequency. Glitch-less squelch and release of the IOD clock is supported with an iod_squelch bit. When enabled, this mimics a gapped clock behavior when an IOD frequency is changed.

#### 1.8.1.1 SYSREF Generation

The RC310xx supports pulse mode SYSREF generation within each IOD. The number of pulses is programmable and SYSREF can be triggered using register programming to register bit sysref_trig, or an assigned GPI or GPIO pin that can act as SYSREF_IN.

Any output that selects a SYSREF IOD must also have out_oe_mode set to 1 (asynchronous OE mode) since the IOD clock is not free running. Partial SYSREF (generating SYSREF pulses on a subset of the outputs configured for SYSREF) can be accomplished by setting out_dis to 1 on those outputs that should not send pulses. To guarantee glitchless operation, the out_dis value must not change while driving SYSREF pulse(s). The phase of each IOD in the group can be independently adjusted if skew is intended.

### 1.8.2 Fractional Output Dividers

There are three fractional output dividers (FOD). Each FOD can divide down the VCO clock to provide frequencies of 1kHz to 650MHz. Each FOD is implemented in two stages. The first stage is an 8-bit fractional divider with Digital Control Delay (DCD) correction followed by a divide-by-2. The DCD FOD allows a divide down of the VCO clock from 30MHz to 657MHz. The FOD's second stage divider is a 17-bit integer divider with minimum divide ratio of 4. This allows output frequencies lower than 30MHz. For output frequencies above 30MHz, this second-stage divider may be bypassed.

#### 1.8.2.1 Spread-Spectrum Clocking (SSC)

FOD0 and FOD1 support spread-spectrum clocking.

If spreading is enabled by setting ssc_en to 1, the spread-spectrum engine generates a triangular frequency modulation on to FOD's divider ratio. The modulation amplitude is programmable in ssc_ampl register fields. The modulation can be programmed to either down spread or center spread in register ssc_mode. The peak-to-peak amplitude is two times of ssc_ampl for center spreading, and one ssc_ampl for down spreading. The supported modulation frequency is from 30kHz to 63kHz. It can be set by programming register ssc_step based on the equations provided in the register description.

When turning off spread, it stops when the current spreading cycle's modulation returns to zero.

If FOD0 and FOD1 SSC are programmed to the same modulation frequency, the register bit ssc_share can be set to 1 to ensure that SSC for FOD0 and FOD1 are in phase. The modulation amplitude and mode (down or center spread) can be set differently. The spread engine of FOD0 will act as the master for the spread engine of FOD1 with respect to synchronization. The zero crossing of the spread triangles is where the synchronization occurs. If the center spread is used on FOD0, then the zero crossing of the upwards frequency ramp of the triangle is the synchronization point. When ssc_share is set to 1, then FOD1 ssc_en must be set to 1 before FOD0 ssc_en is set to 1 since FOD1 SSC will start when FOD0 ssc_en is set to 1. This restriction does not apply when loading the device configuration from OTP/EEPROM on startup, but does apply if dynamically changing these settings later through a dynamic configuration load from the OTP/EEPROM, or by writing registers from the serial interface.

The minimum output frequency that can be spread is 33MHz. A spreading output clock meets the PCIe Gen1 to Gen6 standard at 100MHz.

#### 1.8.2.2 Sync and Phase Adjustment

Each FOD can adjust its output clock phase with a step size of 1/4 VCO period up to about ±20ns. The amount of phase adjustment is programmed in register fod_phase. The adjustment can be of either positive or negative directions. The phase adjustment can be applied immediately if the fod_ph_adj_now bit is set to 1, and/or it will be applied after each time the divider is synchronized if fod_ph_adj_post_sync is set to 1.

IOD phase adjustment is same as, or maybe mimics, FOD phase adjustment but with a step size of one VCO period. The amount of phase adjustment is programmed in register iod_phase. The phase adjustment can be applied immediately if the iod_ph_adj_now bit is set to 1, and/or it will be applied after each time the divider is synchronized if iod_ph_adj_post_sync is set to 1.

There are two sync groups, group0 and group1. An FOD or IOD can be assigned to either group or none by fod_sync_group or iod_sync_group, respectively.The DPLL feedback divider can be assigned to either group or none by dpll_sync_group. The dividers in the same group can be re-synchronized together after any one of them is re-programmed by writing 1 to od_grp0_sync or od_grp1_sync. To disable all clock outputs sourced from the dividers belonging to sync group 0 or 1 prior to re-programming until re-synchronization completes, first write 1 to clr_grp0_oe or clr_grp1_oe. If no divider exists in neither group then it will not re-synchronize.

Upon power-up, after OTP has been loaded and VCO calibration completes and APLL gets locked, a sync pulse is generated automatically to all dividers including FODs, IODs, and DPLL feedback divider, that are assigned to sync group 0 or 1.

A sync pulse can also be initiated by writing 1 to register divider_sync to synchronize all dividers assigned to group 0 and 1, or od_grp0_sync and od_grp1_sync can be used to synchronize only the dividers assigned to each group.

Writing the apll_reinit bit causes the power-up sequence to restart from the VCO calibration step, which will synchronize the dividers after the APLL locks.

#### 1.8.2.3 Digitally Controlled Oscillator (DCO) Mode

In DCO mode, a frequency control word is passed directly from an external processor or FPGA to the DPLL with a step size of $1/2^{40}$ or 0.91 parts per trillion (ppt) and a full-range of ±244 parts per million (ppm) from the nominal DPLL output frequency. The frequency control word (FCW) is written to a 29-b wide write_freq register in two's-complement and then applied to the DPLL feedback divider. The reference clock inputs are unused in this mode. The FCW (positive or negative integer) can be calculated from Fractional Frequency Offset (FFO, in ppm) as follows and then converted to a 29-b two's-complement value.

$$FCW = \left(1 - \frac{1}{1 + \frac{FFO}{10^6}}\right) x 2^{40}$$

#### 1.8.2.4 Numerically Controlled Oscillator (NCO) Mode

In NCO mode, each FOD can adjust its output clock frequency with a step size of $1/2^{34}$ or 58.21ppt, and is based on incrementing the numerator while holding the 34-b denominator at a fixed value. This frequency change at the output clock is gradual and without glitches. The APLL can be in either clock synthesizer/DCO or in jitter attenuator mode.

## 1.9 Clock Outputs

The RC310xx supports up to 12 differential or 24 single-ended clock outputs, or any combination of differential and single-ended clock outputs. Each differential clock output can be programmed as two single-ended clock outputs.

### 1.9.1 Output Types

Differential outputs can be set to 85ohm HCSL, 100ohm HCSL, or LVDS. The HCSL outputs types are low-power push-pull HCSL (LPHCSL) with integrated terminations. They do not require external terminations to drive standard HCSL inputs, such as those found in PCIe applications. HCSL outputs have programmable output swing and HCSL outputs also have two slew rate settings (2V/ns to 4V/ns and 3V/ns to 5V/ns). LVDS outputs require only a 100ohm resistor between the true and complement inputs of the clock input being driven. Both LVDS and HCSL provide output swing levels that are compatible with LVPECL and CML with external AC coupling.

If set to single-ended mode, the output pair can drive both pins. If both pins are enabled, they can be in phase or inverted phase. The single-ended outputs support CMOS swings of 1.8V, 2.5V, or 3.3V as determined by their VDDO voltage.

### 1.9.2 Output Banks

The RC310xx maps the internal and external frequency sources to output banks that can be programmed in register output_bank_src, according to the following table. There are up to 12 clock outputs arranged in seven output banks. Each bank sits on its own VDDO (each VDDO also supplies an IOD or FOD as listed below).

**Table 1. Output Bank Source Mapping**

<table>
<tr><th rowspan="2">output_bank_src</th><th>Bank 0</th><th>Bank 1</th><th>Bank 2</th><th>Bank 3</th><th>Bank 4</th><th>Bank 5</th><th>Bank 6</th></tr>
<tr><th>OUT0</th><th>OUT1</th><th>OUT[2:3]</th><th>OUT[4:7]</th><th>OUT[8:9]</th><th>OUT10</th><th>OUT11</th></tr>
<tr><td>0x0</td><td colspan="2">IOD0</td><td colspan="2">N/A</td><td colspan="3">CLKIN1</td></tr>
<tr><td>0x1</td><td colspan="3">IOD1</td><td colspan="2">N/A</td><td>XIN_REFIN</td><td>N/A</td></tr>
<tr><td>0x2</td><td colspan="4">N/A</td><td colspan="3">IOD2</td></tr>
<tr><td>0x3</td><td colspan="5">N/A</td><td colspan="2">IOD3</td></tr>
<tr><td>0x4</td><td colspan="4">FOD0</td><td colspan="3">N/A</td></tr>
<tr><td>0x5</td><td colspan="7">FOD1</td></tr>
<tr><td>0x6</td><td colspan="3">N/A</td><td colspan="4">FOD2</td></tr>
<tr><td>0x7</td><td colspan="4">N/A</td><td colspan="3">CLKIN0</td></tr>
</table>

## 1.10 Output Enable Control

During the power-up sequence, the clock output drivers are tri-stated until the power supplies have stabilized, then both OUTx and OUTxb are held low. After the OTP configuration load completes, the clock output drivers can be enabled or held disabled until the APLL and/or DPLL lock according to the setting of out_startup. This behavior can be overridden by setting out_dis_group to 0x7.

After power-up, the clock output driver is then enabled, either by setting the corresponding out_dis register bit or by the designated OE pin, if assigned to an OE group. The output driver is enabled when both the register bit and the OE pin are active. If configured in CMOS mode, OUT*x* and OUT*x*b can be enabled or disabled individually through out_prog3 and out_prog2.

There are five OE groups, each output driver can optionally be assigned to a OE group in register out_dis_group. A GPI or GPIO pin can be assigned as OE pin to a OE group in register gpi_func or gpio_func or can also be assigned as a global OE (GOE) pin with the goe register bit.

# 2. Serial Interfaces

The RC310xx can be configured in three ways:

- From internal non-volatile memory using OTP user configurations
- From its slave serial interface and program by a master device
- From an external I2C EEPROM. The device will become master at power-up and load data from I2C EEPROM.

The RC310xx supports three slave serial interfaces: I2C, SPI, and SMBUS. The interface type is selected by the ssi_enable (0x0026[1:0]) register field. If OTP is not programmed, the device powers up in I2C 1-byte offset mode. These serial interfaces share the same pins so only one is available at a time. The I2C master interface that is used to load User Configurations from an external I2C EEPROM is only active after loading an OTP Configuration with EEPROM function enable. An external master (I2C, SPI, SMBUS) can be used to access internal registers after power-up and the configuration load process is completed.

## 2.1 1-Byte and 2-Byte Offset Mode

The RC310xx has a 912-byte register memory space. It also separates as four pages, and each page is 256 bytes.

The RC310xx supports 1-byte and 2-byte offset mode for all three slave serial interfaces, I2C, SPI, and SMBUS. The user can choose to operate as 1-byte or 2-byte offset mode, and can be configured through register ssi_addr_size (0x0026[2]) register field which defaults as 1-byte offset mode. These offsets are used in conjunction with the page register to access registers internal to the device. Because the I2C protocol already includes a read/write bit with the Dev Addr, all bits of the 1-byte or 2-byte offset field can be used to address internal registers.

- The 1-byte offset mode – It also called page mode where part of the address offset is provided in each transaction and another part comes from an internal page register in each serial port. For an I2C 1-byte offset mode operation example, see Figure 1.
- The 2-Byte offset mode – Use two byte as the serial port providing the full offset address within each burst. For an I2C 2-byte offset mode operation example, see Figure 2.

# 3. I2C Slave

The I2C slave protocol complies with the *I2C Specification*, version UM10204 Rev.6, 4 April 2014. The SCL_SCLK and SDA_nCS pins are 3.3V tolerant. The Dev Addr shown in the Figure 1 represents the I2C bus address that the device will respond to. This 7-bit value in the i2c_addr register field defaults to 0x09 if not programmed via the OTP load.

## 3.1 1-Byte Offset Mode

In 1-byte mode, the lower 8 bits of the register offset address originate from the Offset Addr 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 0 for page0, 1 for page1, 2 for page2, and 3 for page3.

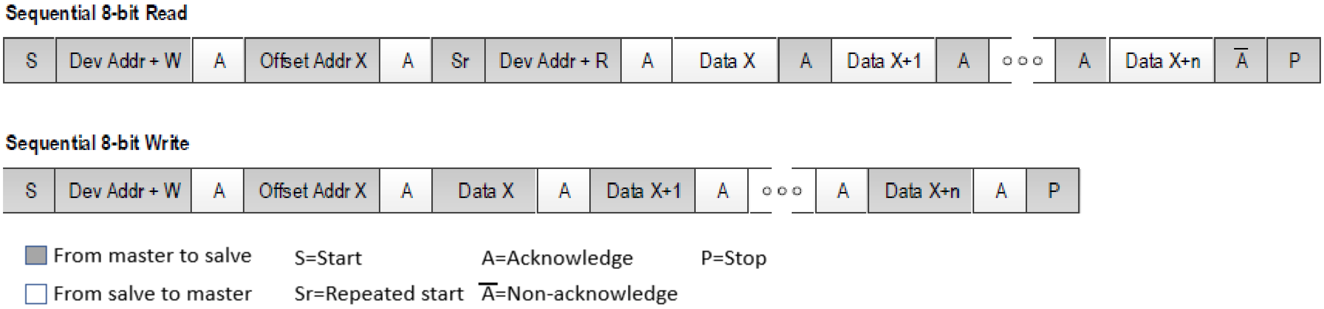


**Figure 1. I2C 1-Byte Offset Mode Slave Sequencing**

## 3.2 2-Byte Offset Mode

In 2-byte mode, the full 16-bit register address can be obtained from the Offset Addr bytes, so the page register does not need to be set up. The MSB offset address is the page number and the LSB address is the register address.

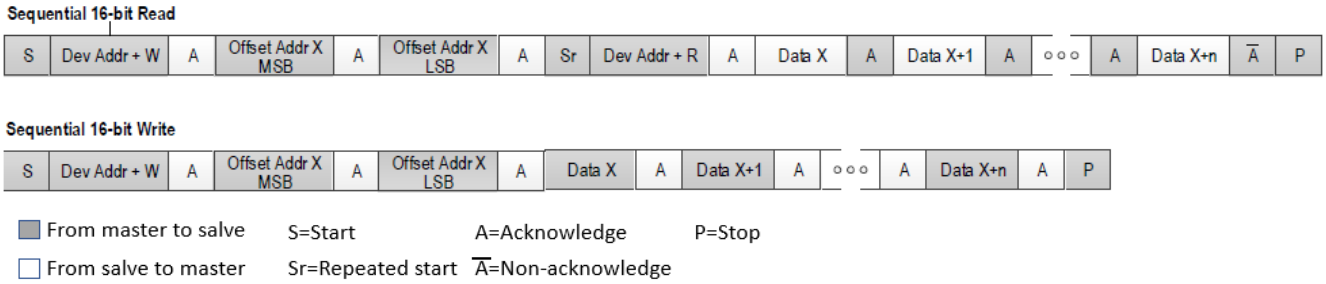


**Figure 2. I2C 2-Byte Offset Mode Slave Sequencing**

*Note*: All serial port configuration will take effect after the write cycle is completed.

# 4. SPI Slave

This device supports 4-wire or 3-wire SPI operation as a selectable protocol on the serial port. The 3-wire or 4-wire mode is selected by the spi_3wire register bit. In 4-wire mode, there are separate data in and data out signals (SDI and SDO, respectively). In 3-wire mode, the SDIO signal is used as a single, bidirectional data signal.

Figure 3 shows the sequencing of address and data on the serial port in both 3-wire and 4-wire SPI mode. 4-wire SPI mode is the default. The R/W bit is high for read cycles and low for write cycles.

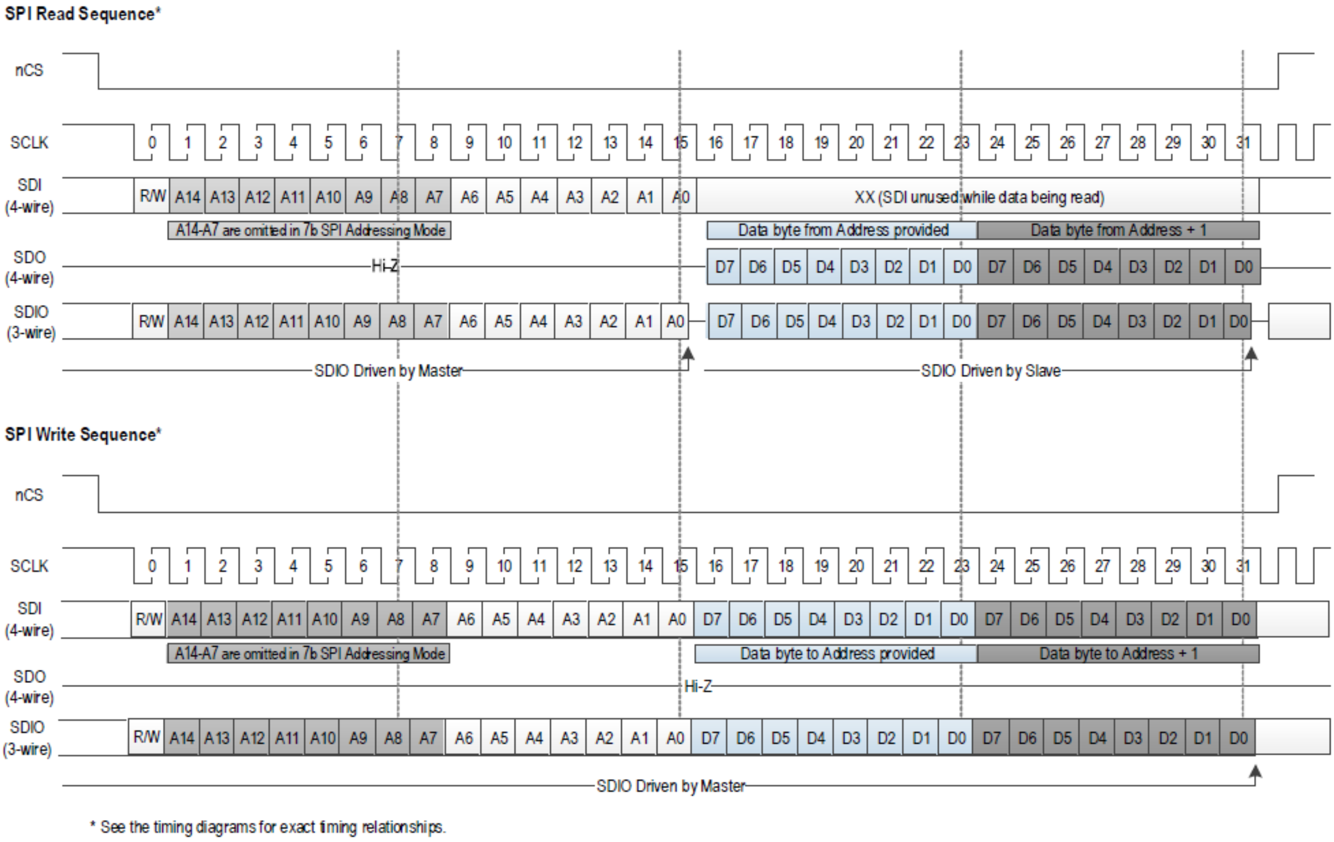


**Figure 3. SPI Sequencing Diagram**

SPI operation can be configured for the following settings through register fields: 1-byte or 2-byte offset addressing (ssi_addr_size).

In 1-byte operation, the 16-bit register address is formed by using the 7 bits of address supplied in the SPI access and taking the upper 9 bits from the page register. The page register is accessed using an offset address of 0xFD. For an 1-byte offset operation example, see Example of I2C 1-Byte Offset Mode Programming.

In 2-byte operation, the 16-bit register address is formed by using the 15 bits of address supplied in the SPI access and taking the upper 1-bit from the page register. For an 2-byte offset operation example, see also Example of I2C 2-Byte Offset Mode Programming.

- Data sampling on falling or rising edge of SCLK (spi_clk_sel)
- Output (read) data positioning relative to active SCLK edge (spi_del_out)

*Note*: SPI burst mode operation is required to ensure data integrity of multi-byte registers. When accessing a multi-byte register, all data bytes must be written or read in a single SPI burst access. Bursts can be of greater length if desired but must not extend beyond the end of the register page. An internal address pointer is incremented automatically as each data byte is written or read.

The SPI interface operating at 10MHz supports a DCO update rate of approximately 200k updates per second.

# 5. SMBus Slave

This device supports a standard SMBus v2.0 and v3.1 compliant interface operating at 100kHz, 400kHz, and 1MHz. The SCL_SCLK and SDA_nCS pins are 3.3V tolerant.

The SMBus interface supports block write and byte write modes. SMBus only supports 7-bit sub-addresses and the ssi_addr_size setting is ignored. A writeable page register selects the upper address bits.

The SMBus slave protocol of the RC310xx complies with the SMBus v2.0 and v3.1 standard.

In the following description, SCL refers to the SCL_SCLK pin and SDA refers to the SDA_nCS pin. Figure 4 shows the sequence of states on the SMBus SDA signal for the supported modes of operation.

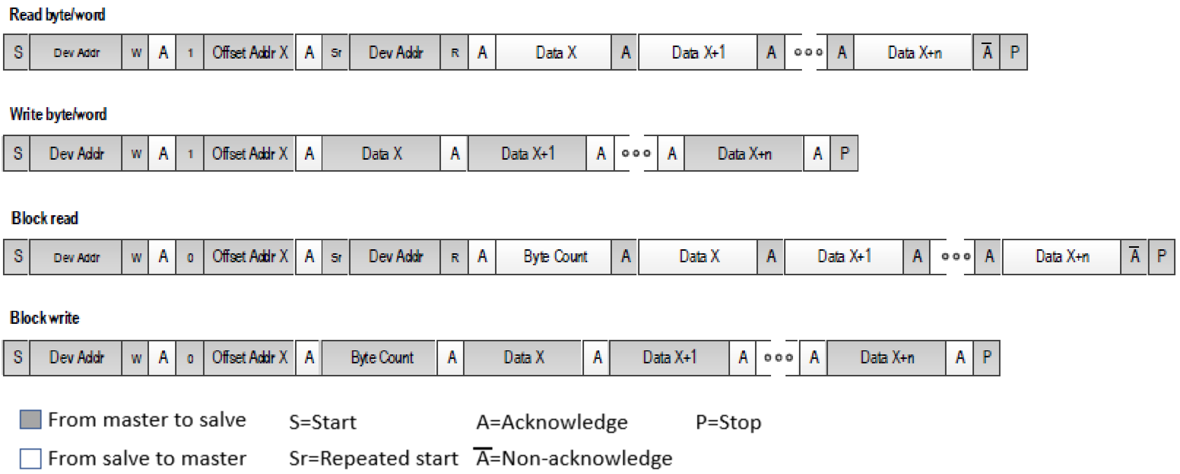


**Figure 4. I2C Slave Timing Diagram**

The Dev Addr shown in the figure represents the SMBus address that the device will respond to. This 7-bit value in the i2c_addr register field defaults to 0x09 if not programmed via the OTP load.

In SMBus, the MSB of the offset address is used to determine whether Block mode is selected, and means the address space is only 128 locations. The internal 16-bit register address is formed by using the 7 bits of address supplied in the SMBus access and taking the upper 9 bits from the page register. The page register is accessed using an offset address of 0xFD.

Read byte/word and Write byte/word are essentially the same as the reads and writes in I2C mode. The main difference is in that when this mode is selected, the MSB of the offset address to read is set to 1. Block read and Block write modes are selected when the MSB of the offset address is set to 0. In block mode, the byte count is always the first value of the data phase. The byte count is stored in the byte count register field and defaults to 0x8.

*Note*: Same as for I2C, external to the scope of the SMBus-specific logic, burst mode operation is required to ensure data integrity of multi-byte registers. When accessing a multi-byte register, all data bytes must be written or read in a single SMBus burst access. Bursts can be of greater length if desired, but must not extend the end of the register page (Offset Addr 0x7F). An internal address pointer is incremented automatically as each data byte is written or read.

## 5.1 SMBus 1-Byte/2-Byte Offset Mode Programming Examples

In 1-byte operation, same as I2C 1-byte offset mode operation. For an 1-byte offset operation example, see Example of I2C 1-Byte Offset Mode Programming.

In 2-byte operation, same as I2C 2-byte offset mode operation. For an 2-byte offset operation example, see also Example of I2C 2-Byte Offset Mode Programming.

# 6. Example of Programming Process

The following two examples show how to use the rbs setting file register value and the customer's own I2C master to program the RC310xx:

- The I2C 1-byte offset mode programming with page number for more than 256 register offset addresses.
- The I2C 2-byte offset mode programming will use the most significant offset bye for the page number. It also applies to SMBus and SPI.

There are four steps to program the RC310xx:

1. Program all register values to the device.
2. Complete a VCO calibration for the new setting.
   *Note*: This step applies only if the VCO frequency will change.
3. Complete an output divider sync up.
4. Change synchronous serial bus setting. The setting is effective after the write cycle is completed. If no change on synchronous serial bus change, then skip this step.

## 6.1 Getting Register Value from rbs Setting File

The register values of a configuration can be exported from the RICBox GUI when the configuration file is loaded, or it can be retrieved from an rbs file by running a Python script on the configuration file. The register map file is a text file that can be integrated into the user's software (i.e., a head file or a data array), and it will list all of the register values at the beginning of the file (see the following example).

| Offset Number | Binary Value | Hex Value | Offset Number |
|---|---|---|---|
| 00 | 00110011 | 33 | 00 |
| 01 | 00010000 | 10 | 01 |
| 02 | 00000000 | 00 | 02 |
| 03 | 00000000 | 00 | 03 |
| 04 | 00100010 | 22 | 04 |
| 05 | 00000001 | 01 | 05 |
| 06 | 10101001 | A9 | 06 |
| 07 | 00000000 | 00 | 07 |
| 08 | 00101010 | 2A | 08 |

## 6.2 Example of I2C 1-Byte Offset Mode Programming

**//Step 1 – Program register value**

```
write 0x9 fd 00   // Change page number to page 0.
write 0x9 02 data(0x0002 to 0x0019)
write 0x9 28 data(0x0028 to 0x00F7) // Skip register 0x20 to 0x27, the synchronous
serial bus setting at the end.
write 0x9 fd 01   //Change page number to 1:
write 0x9 00 data(0x0100 to 0x01FA)
write 0x9 fd 02   //Change page number to 2:
write 0x9 00 data(0x0200 to 0x01FB)
write 0x9 fd 03  //Change page number to 3:
write 0x9 00 data(0x0300 to 0x031F)
write 0x9 fd 01  //Change page number to 1
```

**//Step 2 – VCO re-calibration**

```
write 0x9 3c 20   // Dummy write, delay
write 0x9 3c A0   // Apll re-calibrate
write 0x9 3c 20   // Dummy write, wait for the re-calibrate.
                  // If the master has wait function, then it can be replay by wait
                  command. (Wait about 400us)
write 0x9 3c 20   // Dummy write
write 0x9 3c 20   // Dummy write
write 0x9 3c 20   // Dummy write
write 0x9 3c 20  // Dummy write
write 0x9 3c 20  // Dummy write
write 0x9 3c 20  // Dummy write
write 0x9 3c 20  // Dummy write
write 0x9 3c 20  // Dummy write
write 0x9 3c 20  // Dummy write
write 0x9 3c 20  // Dummy write
write 0x9 3c 20  // Dummy write
```

**//Step 3 – Output divider sync up**

```
write 0x9 fd 00  //Change page number to 0
write 0x9 11 30
write 0x9 11 32  // output divider sync up
write 0x9 11 30
```

**//Step 4 – Synchronous serial bus setting**

```
write 0x9 00 20 data(0x0020 to 0x0027)   //change synchronous serial bus setting. It
will be affected after the write cycle completed.
```

## 6.3 Example of I2C 2-Byte Offset Mode Programming

**//Step 1 Program register value**

```
write 0x9 00 02 data(0x0002 to 0x0019)  // burst write    //Skip register 0x0020 to
0x0027.
write 0x9 00 28 data(0x0028 to 0x00F7)  // burst write
write 0x9 01 00 data(0x0100 to 0x01FA)  // burst write
write 0x9 02 00 data(0x0200 to 0x01FB)  // burst write
write 0x9 03 00 data(0x0300 to 0x031F)  // burst write
```

**//Step 2 VCO re-calibration**

```
write 0x9 01 3c 20     // Dummy write, delay
write 0x9 01 3c 20
write 0x9 01 3c A0   // Apll re-calibrate. Change register 0x013C from 20 to A0 to
re-calibrate
write 0x9 01 3c 20    // Dummy write, wait for the re-calibrate.
write 0x9 01 3c 20   // Dummy write
write 0x9 01 3c 20   // Dummy write
write 0x9 01 3c 20   // Dummy write
write 0x9 01 3c 20  // Dummy write
write 0x9 01 3c 20  // Dummy write
write 0x9 01 3c 20  // Dummy write
```

```
write 0x9 01 3c 20  // Dummy write
write 0x9 01 3c 20  // Dummy write
write 0x9 01 3c 20  // Dummy write
write 0x9 01 3c 20  // Dummy write
write 0x9 01 3c 20  // Dummy write
```

**//Step 3 Output divider sync up**

```
write 0x9 00 11 30
write 0x9 00 11 32  // divider sync up
write 0x9 00 11 30
```

**//Step 4 Synchronous serial bus setting**

```
write 0x9 00 20 data(0x0020 to 0x0027)   //change synchronous serial bus setting. It will be affected after the write cycle completed.
```

# 7. Example of Changing Output Frequency

Changing an output frequency does not require all registers to be programmed. Changing an output frequency can be implemented by changing either or both of the following:

- VCO frequency of the APLL
- Output divider

Output dividers include integer output dividers (IODs) and fractional output dividers (FODs). Modifying only the output dividers will result in a glitchless frequency change when the last byte of output divider value is updated by an I2C block write event. The frequency change is instantaneous; one clock cycle is still the old frequency and the next clock cycle is the new frequency. Re-calibration is not required; however, if the VCO frequency is modified, it will trigger a VCO calibration. The resulting output frequency will not be glitchless.

APLL VCO frequency is determined by input clock frequency and the 27-bit Feedback divider. The VCO frequency range is between 9.5GHz and 10.7GHz. With VCO frequency in the range of 9.5GHz to 10.7GHz, the RC310xx supports 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. FOD provides a frequency resolution up to 1ppt (1 per trillion). There are four IODs (IOD0-3) and three FODs (FOD0-2).

*Note*: If the input frequency or VCO frequency is changed, 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.

## 7.1 Output Frequency Calculation

The following formulas provide a method to calculate the desired output frequency. With output frequency determined, we then select an output divider to get the VCO frequency, which must be within 9.5GHz to 10.7GHz.

- Output Frequency = VCO Frequency / Output Divider
- VCO Frequency = Input Frequency * Doubler * (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.

## 7.2 Feedback Divider

The VCO feedback divider consists of a 10-bit integer and a 27-bit fractional portion. The offset addresses of the feedback divider register are as follows:

Integer:

- FB_INT_DIV[9:8] = 0x125[1:0]
- FB_INT_DIV[7:0] = 0x124[7:0]

Fractional:

- FB_FRAC_DIV[26:24] = 0x123[2:0]
- FB_FRAC_DIV[23:16] = 0x122[7:0]
- FB_FRAC_DIV[15:8] = 0x121[7:0]
- FB_FRAC_DIV[7:0] = 0x120[7:0]

**Example**: Programming a VCO frequency of 10GHz with 60MHz crystal.

With a crystal frequency of 60MHz, in order to make a VCO frequency of 10GHz:

- FB_DIV = 10GHz / (60MHz x 2) =83.3333333333333333333
- FB_INT_DIV = 83
- FB_FRAC_DIV = 0.6666666666666667 x 2^27 = 44,739,243 = ‘0h02 AA AA AB

Program the following registers will make VCO frequency = 10GHz:

- 0x125[1:0] = 0x0
- 0x124[7:0] = 0x53
- 0x123[2:0] = 0x02
- 0x122[7:0] = 0xAA
- 0x121[7:0] = 0xAA
- 0x120[7:0] = 0xAB

## 7.3 Integer Output Divider

Each Integer Output Divider (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.

IOD0 registers:

- IOD0[24] = Register 0x1C3[0]
- IOD0[23:16] = Register 0x1C2[7:0]
- IOD0[15:8] = Register 0x1C1[7:0]
- IOD0[7:0] = Register 0x1C0[7:0]

Register 0x1C3 only uses bit [0] as IOD0’s MSB. All four registers must be written for the IOD value to actually change an output frequency. Register 0x1C3 has remaining bits that have other functions. Remember to keep other bits intact when modifying 0x1C3[0]. Also, read registers 0x1C0~3 to confirm the current value and to know the remaining bits in register 0xC3.

In addition, the other three IODs are taking up the register ranges listed below:

- IOD1[24:0] are in registers 0x1C8, 0x1C9, 0x1CA and 0x1CB.
- IOD2[24:0] are in registers 0x1D0, 0x1D1, 0x1D2 and 0x1D3.
- IOD3[24:0] are in registers 0x1D8, 0x1D9, 0x1DA and 0x1DB.

**Example**: To make OUT0 = 100MHz by change IOD0 to 100 (VCO frequency = 10GHz)

Output Frequency = VCO Frequency / Output Divider

100MHz = 10GHz / 100

Programming the following registers will change the output to 100MHz. Assuming the VCO is 10GHz.

- Register 0x1C0[7:0] value 0x64
- Register 0x1C1[7:0] value 0x00
- Register 0x1C2[7:0] value 0x00
- Register 0x1C3[0] value 0x0

## 7.4 Fractional Output Dividers

The fractional output divider consists of a 9-bit 1st integer, a 17-bit 2nd integer, and a 34-bit fractional portion. Together, it can provide a frequency resolution better than 1ppt. The 1st integer covers values up to 511 and is always used. The 2nd integer can cover values up to 131,071 and is used for output frequencies below 33MHz. The formula for the total divider value is as follows:

- $F_{OUT}$ > 33MHz: FOD = 1st Integer + Fraction
- $F_{OUT}$ < 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 1. This causes the 2nd Integer and its additional ×2 to be bypassed. Both integer and fractional registers are as follows.

1st Integer:

- FOD0 1st Integer [8] = Register 0x1E1[0]
- FOD0 1st Integer [7:0] = Register 0x1E0[7:0]

2nd Integer:

- FOD0 2nd Integer [16:15] = Register 0x1E3[1:0]
- FOD0 2nd Integer [14:7] = Register 0x1E2[7:0]
- FOD0 2nd Integer [6:0] = Register 0x1E1[7:1]

Fractional:

- FOD0 Fractional [33:30] = Register 0x1E7[3:0]
- FOD0 Fractional [29:22] = Register 0x1E6[7:0]
- FOD0 Fractional [21:14] = Register 0x1E5[7:0]
- FOD0 Fractional [13:6] = Register 0x1E4[7:0]
- FOD0 Fractional [5:0] = Register 0x1E3[7:2]

In addition, FOD1 values are carried by registers 0x1F0 to 0x1F7, and FOD2 is carried by registers in 0x200 to 0x207. For more information, see Register Offsets.

**Example**: Setting OUT0 = 156.25MHz by programming FOD0 = 64 (VCO frequency = 10GHZ)

Output Frequency = VCO Frequency / Output Divider

156.25MHz = 10GHz / 64

Program the following registers will change the output to 156.25MHz

- FOD0 Integer Register programming:
  - Register 0x1E0[7:0] value 0x40
  - Register 0x1E1[7:0] value 0x00

- Register 0x1E2[7:0] value 0x00
- Register 0x1E3[1:0] value 0x00
- FOD0 Fractional register programming:
  - Register 0x1E3[7:2] value 0x00
  - Register 0x1E4[6:0] value 0x00
  - Register 0x1E5[6:0] value 0x00
  - Register 0x1E6[6:0] value 0x00
  - Register 0x1E7[3:0] value 0x00

# 8. Example of Device Programming with the Driver

For PC or Linux users, they can also leverage the RC310xx Python driver to program the device. The driver will take care of the programming process.

The requirement: Install RICBox and the VC7 plug-in file, or install Python 3.9 and the VC7 driver.

## 8.1 Example of Python Code

Save the following code to “VC7_Config_Builder.py”

```
import rbcore
from r_drv_vc7.device_abstraction.config_builder import *
from r_drv_vc7.device_abstraction.versaclock7 import Versaclock7
from rbcore.io.ftdi import FtdiI2cSettings
## Create VC7 device. (VC7_Family, Version)
vc7 = Versaclock7( Versaclock7.create_device(r'RC21012B', ‘B’))
## Connect FTDI device, I2C address = 0x9
vc7.connect(FtdiI2cSettings(index='0', address=0x09))
## Program VC7
vc7.program_settings_file(r'RC21012B065_config0.rbs')
```

## 8.2 Example of DOS Batch File

The RICBox VC7 plug-in file comes with the Python 3.9 environment. The following example shows the RICBox environment path. Save the following command to batch file so that it can execute the Python code.

```
Echo off
Echo Start!
C:\Users\%username%\AppData\Roaming\RICBox\venvs\VersaClock7-x64\Scripts\python
VC7_Config_Builder.py
pause
```

# 9. Register Organization

## 9.1 Overview

Register sizes are denoted as:

- byte: 8-bit
- hword: 16-bit
- word: 32-bit
- dword: 64-bit

Register types are defined in the following table.

**Table 2. Register Type Definition**

| Type Indicator | Definition |
|---|---|
| R/W1C | Read/Write 1 to Clear |
| R/W1S | Read/Write 1 to Set |
| RO | Read-only |
| RW | Read/Write |
| WO | Write-only |

## 9.2 Register Block Offsets

**Table 3. Register Block Offsets**

| Block Offset | Block Name | Address Table | Registers |
|---|---|---|---|
| 0x000 | GLOBAL | Global Addresses | Global Registers |
| 0x020 | SSI | SSI Addresses | SSI Registers |
| 0x02C | XO | Crystal Addresses | XO Register |
| 0x030 += 0x004 | CLKIN[0:1] | Clock Input Addresses | CLKIN Register |
| 0x038 += 0x004 | REF[0:3] | Clock Reference Addresses | REF Registers |
| 0x050 += 0x010 | LOSMON[0:4] | LOS Monitor Addresses | LOSMON Registers |
| 0xA0 += 0x020 | FREQMON[0:3] | Frequency Monitor Addresses | FREQMON Registers |
| 0x120 | APLL | APLL Addresses | APLL Registers |
| 0x160 | TDCAPLL | TDC APLL Addresses | TDC APLL Registers |
| 0x180 | DPLL | DPLL Addresses | DPLL Registers |
| 0x1C0 += 0x008 | IOD[0:3] | IOD Addresses | IOD Registers |
| 0x1E0 += 0x010 | FOD[0:2] | FOD Addresses | FOD Registers |
| 0x240 += 0x004 | OUT[0:11] | Clock Output Driver Addresses | OUT Registers |
| 0x280 += 0x004 | BANK[0:6] | Output Bank Addresses | BANK Register |
| 0x29C += 0x004 | GPI[0:3] | GPI Addresses | GPI Registers |
| 0x2AC += 0x004 | GPIO[0:4] | GPIO Addresses | GPIO Registers |
| 0x300 += 0x004 | SSC[0:1] | SSC Addresses | SSC Register |
| 0x308 | INT | Interrupt Addresses | INT Registers |

## 9.3 Register Offsets

### 9.3.1 Global Address Map

**Table 4. Global Addresses**

| Block Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | hword | VENDOR_ID | VENDOR_ID - Vendor ID |
| 0x02 | hword | DEVICE_ID | DEVICE_ID - Device ID |
| 0x04 | hword | DEVICE_REV | DEVICE_REV - Device Revision |
| 0x06 | hword | DEVICE_PGM | DEVICE_PGM - Device Programming |
| 0x08 | word | DEVICE_CNFG | DEVICE_CNFG - Device Configuration |
| 0x0E | byte | PWR_CTL | PWR_CTL - Power Control |
| 0x10 | byte | REG_LOCK | REG_LOCK - Configuration Register Lock |
| 0x11 | byte | INIT_SYNC | INIT_SYNC - Initialization and Synchronization Register |
| 0x12 | hword | SW_RESET | SW_RESET - Software Reset Register |
| 0x16 | hword | MISC_CNFG | MISC_CNFG - Miscellaneous Configuration |
| 0x1C | hword | STARTUP_STS | STARTUP_STS - Start-up Status |
| 0x1E | hword | DEVICE_STS | DEVICE_STS - Device Status |

### 9.3.2 SSI Address Map

**Table 5. SSI Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | byte | SPI_CNFG | SPI_CNFG - SPI Configuration |
| 0x01 | byte | I2C_FLTR_CNFG | I2C_FLTR_CNFG - I2C Filter Configuration |
| 0x02 | byte | I2C_TIMING_CNFG | I2C_TIMING_CNFG - I2C Timing Configuration |
| 0x03 | byte | I2C_ADDR_CNFG | I2C_ADDR_CNFG - I2C Address Configuration |
| 0x04 | byte | BYTE_CNT | BYTE_CNT - Byte Count |
| 0x05 | byte | SMB_CTL | SMB_CTL - SMBus Control |
| 0x06 | byte | SSI_GLOBAL_CNFG | SSI_GLOBAL_CNFG - SSI Global Configuration |
| 0x07 | byte | SSI_STS | SSI_STS - Serial Port Status |

### 9.3.3 Crystal Address Map

**Table 6. Crystal Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | word | XO_CNFG | XO_CNFG - Crystal Configuration |

### 9.3.4 Clock Input Address Map

**Table 7. Clock Input Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | hword | CLKIN_CNFG | CLKIN_CNFG - Clock Input Pad Configuration |

### 9.3.5 Clock Reference Address Map

**Table 8. Clock Reference Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | word | PREDIV_CNFG | PREDIV_CNFG - Reference Clock Input Pre-divider Configuration |

### 9.3.6 LOS Monitor Address Map

**Table 9. LOS Monitor Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | hword | LOSMON_WINDOW | LOSMON_WINDOW - LOS Monitor Window Configuration |
| 0x02 | hword | LOSMON_NOMINAL | LOSMON_NOMINAL - LOS Monitor Nominal Number Configuration |
| 0x04 | word | LOSMON_THRESH | LOSMON_THRESH - LOS Monitor Threshold Configuration |
| 0x08 | byte | LOSMON_QUAL | LOSMON_QUAL - LOS Monitor Qualify Counter Configuration |
| 0x09 | byte | LOSMON_STS | LOSMON_STS - LOS Monitor Status |
| 0x0A | byte | LOSMON_EVENT | LOSMON_EVENT - LOS Monitor Event Status |
| 0x0B | byte | LOSMON_CNT | LOSMON_CNT - LOS Monitor Count |

### 9.3.7 Frequency Monitor Address Map

**Table 10. Frequency Monitor Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | word | FREQMON_WINDOW | FREQMON_WINDOW - Frequency Monitor Window Configuration |
| 0x04 | word | FREQMON_NOMINAL | FREQMON_NOMINAL - Frequency Monitor Nominal Number Configuration |
| 0x08 | dword | FREQMON_THRESH | FREQMON_THRESH - Frequency Monitor Threshold Configuration |
| 0x10 | byte | FREQMON_STS | FREQMON_STS - Frequency Monitor Status |
| 0x11 | byte | FREQMON_EVENT | FREQMON_EVENT - Frequency Monitor Event Status |
| 0x14 | word | FREQMON_OFFSET | FREQMON_OFFSET - Frequency Monitor Frequency Offset Status |

### 9.3.8 TDC APLL Address Map

**Table 11. TDC APLL Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | hword | TDC_APLL_CNFG | TDC_APLL_CNFG - TDC APLL Configuration |
| 0x02 | hword | TDC_FB_DIV_FRAC | TDC_FB_DIV_FRAC - TDC APLL Feedback Divider Fraction Configuration |
| 0x04 | byte | TDC_FB_DIV_INT | TDC_FB_DIV_INT - TDC APLL Feedback Divider Integer Configuration |
| 0x05 | byte | TDC_FB_SDM_CNFG | TDC_FB_SDM_CNFG - TDC APLL Feedback SDM Configuration |
| 0x06 | byte | TDC_REF_DIV_CNFG | TDC_REF_DIV_CNFG - TDC APLL Reference Divider Configuration |
| 0x0A | hword | TDC_FILTER_STS | TDC_FILTER_STS - TDC APLL Filter Status |

### 9.3.9 APLL Address Map

**Table 12. APLL Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | word | APLL_FB_DIV_FRAC | APLL_FB_DIV_FRAC - APLL Feedback Divider Fraction Configuration |
| 0x04 | hword | APLL_FB_DIV_INT | APLL_FB_DIV_INT - APLL Feedback Divider Integer Configuration |
| 0x06 | byte | APLL_FB_SDM_CNFG | APLL_FB_SDM_CNFG - APLL Feedback SDM Configuration |
| 0x07 | byte | APLL_CNFG | APLL_CNFG - APLL Configuration |
| 0x08 | hword | CP_CNFG | CP_CNFG - APLL Charge Pump Configuration |
| 0x0A | byte | LPF_CNFG | LPF_CNFG - APLL Loop Filter Configuration |
| 0x0B | byte | LPF_3RD_CNFG | LPF_3RD_CNFG - APLL Loop Filter 3rd Pole Configuration |
| 0x15 | byte | APLL_REF_FB_CNFG | APLL_REF_FB_CNFG - APLL Ref and Fb Clock Configuration |
| 0x18 | word | BANK_MUX_CLK_EN | BANK_MUX_CLK_EN - Bank Mux Clock Enable |
| 0x1F | byte | APLL_STS | APLL_STS - APLL Status |
| 0x20 | byte | APLL_EVENT | APLL_EVENT - APLL Event Status |
| 0x21 | byte | APLL_LOL_CNT | APLL_LOL_CNT - APLL Loss-of-Lock Counter |
| 0x28 | hword | ANA_SPARE_CNFG | ANA_SPARE_CNFG - Analog Spare Configuration |
| 0x2A | hword | ANA_SPARE_STS | ANA_SPARE_STS - Analog Spare Status |

### 9.3.10 DPLL Address Map

**Table 13. DPLL Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | byte | DPLL_REF_FB_CNFG | DPLL_REF_FB_CNFG - DPLL Ref and Fb Clock Configuration |
| 0x01 | byte | DPLL_MODE | DPLL_MODE - DPLL Mode Configuration |
| 0x02 | byte | DPLL_DECIMATOR | DPLL_DECIMATOR - DPLL Decimator Configuration |
| 0x03 | byte | DPLL_XTAL_OFFSET | DPLL_XTAL_OFFSET - DPLL Crystal Trim Offset Configuration |
| 0x04 | hword | HOLDOVER_CNFG | HOLDOVER_CNFG - Holdover Configuration |
| 0x06 | hword | DPLL_BANDWIDTH | DPLL_BANDWIDTH - DPLL Bandwidth Configuration |
| 0x08 | hword | DPLL_DAMPING | DPLL_DAMPING - DPLL Damping Configuration |
| 0x0A | hword | DPLL_FB_CORR | DPLL_FB_CORR - DPLL Feedback Correction Configuration |
| 0x0C | word | DPLL_PHASE_SLOPE_LIMIT | DPLL_PHASE_SLOPE_LIMIT - DPLL Phase Slope Limit Configuration |
| 0x10 | word | DPLL_PHASE_OFFSET | DPLL_PHASE_OFFSET - DPLL Phase Offset Configuration |
| 0x14 | word | DPLL_WRITE_FREQ | DPLL_WRITE_FREQ - DPLL Write Frequency Configuration |
| 0x18 | dword | DPLL_FB_DIV_NUM | DPLL_FB_DIV_NUM - DPLL Feedback Divider Numerator Configuration |
| 0x20 | dword | DPLL_FB_DIV_DEN | DPLL_FB_DIV_DEN - DPLL Feedback Divider Denominator Configuration |
| 0x28 | word | DPLL_FB_DIV_INT | DPLL_FB_DIV_INT - DPLL Feedback Divider Integer Configuration |
| 0x2C | word | DPLL_LOCK | DPLL_LOCK - DPLL Lock Configuration |
| 0x31 | byte | DPLL_STS | DPLL_STS - DPLL Status |
| 0x32 | byte | DPLL_EVENT | DPLL_EVENT - DPLL Event Status |
| 0x33 | byte | DPLL_LOL_CNT | DPLL_LOL_CNT - DPLL Loss-of-Lock Counter |
| 0x34 | word | DPLL_FILTER_STS | DPLL_FILTER_STS - DPLL Filter Status |
| 0x38 | word | DPLL_PHASE_STS | DPLL_PHASE_STS - DPLL Phase Status |

### 9.3.11 IOD Address Map

**Table 14. IOD Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | word | IOD_INT_CNFG | IOD_INT_CNFG - IOD Integer Ratio Configuration |
| 0x04 | hword | IOD_PHASE_CNFG | IOD_PHASE_CNFG - IOD Phase Configuration |
| 0x06 | hword | SYSREF_CNFG | SYSREF_CNFG - SYSREF Configuration |

### 9.3.12 FOD Address Map

**Table 15. FOD Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | dword | FOD_INT_CNFG | FOD_INT_CNFG - FOD Integer Configuration |
| 0x08 | hword | FOD_PHASE_CNFG | FOD_PHASE_CNFG - FOD Phase Configuration |

### 9.3.13 SSC Address Map

**Table 16. SSC Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | word | SSC_CNFG | SSC_CNFG - Spectrum Spreading Configuration |

### 9.3.14 Output Bank Address Map

**Table 17. Output Bank Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | byte | OUT_BANK_CNFG | OUT_BANK_CNFG - Output Bank Configuration |

### 9.3.15 Clock Output Driver Address Map

**Table 18. Clock Output Driver Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | byte | ODRV_EN | ODRV_EN - Output Driver Enable |
| 0x02 | hword | ODRV_CNFG | ODRV_CNFG - Output Driver Configuration |

### 9.3.16 GPI Address Map

**Table 19. GPI Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | hword | GPI_CNFG | GPI_CNFG - GPI Configuration |
| 0x02 | byte | GPI_STS | GPI_STS - GPI Status |

### 9.3.17 GPIO Address Map

**Table 20. GPIO Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | hword | GPIO_CNFG | GPIO_CNFG - GPIO Configuration |
| 0x02 | byte | GPIO_STS | GPIO_STS - GPIO Status |

### 9.3.18 Interrupt Address Map

**Table 21. Interrupt Addresses**

| Offset | Size | Register Name | Register Description |
|---|---|---|---|
| 0x00 | word | SCRATCH0 | SCRATCH0 - Software Scratch Register 0 |
| 0x04 | word | INT_EN | INT_EN - Interrupt Enable |
| 0x08 | word | INT_STS | INT_STS - Interrupt Status |

# 10. Physical Register Addresses

## 10.1 Global Block

Table 22. Global Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x0 | GLOBAL | VENDOR_ID[0] | RO | 0x33 |
| 0x1 | GLOBAL | VENDOR_ID[1] | RO | 0x10 |
| 0x2 | GLOBAL | DEVICE_ID[0] | RW | 0x00 |
| 0x3 | GLOBAL | DEVICE_ID[1] | RW | 0x00 |
| 0x4 | GLOBAL | DEVICE_REV[0] | RO | 0x22 |
| 0x5 | GLOBAL | DEVICE_REV[1] | RO | 0x01 |
| 0x6 | GLOBAL | DEVICE_PGM[0] | RW | 0x00 |
| 0x7 | GLOBAL | DEVICE_PGM[1] | RW | 0x00 |
| 0x8 | GLOBAL | DEVICE_CNFG[0] | RW | 0x2a |
| 0x9 | GLOBAL | DEVICE_CNFG[1] | RW | 0xf0 |
| 0xa | GLOBAL | DEVICE_CNFG[2] | RW | 0x12 |
| 0xb | GLOBAL | DEVICE_CNFG[3] | RW | 0x00 |
| 0xe | GLOBAL | PWR_CTL | RW | 0x01 |
| 0x10 | GLOBAL | REG_LOCK | RW | 0x00 |
| 0x11 | GLOBAL | INIT_SYNC | R/W1S | 0x30 |
| 0x12 | GLOBAL | SW_RESET[0] | RW | 0x00 |
| 0x13 | GLOBAL | SW_RESET[1] | RW | 0x00 |
| 0x16 | GLOBAL | MISC_CNFG[0] | RW | 0x05 |
| 0x17 | GLOBAL | MISC_CNFG[1] | RW | 0x80 |
| 0x1c | GLOBAL | STARTUP_STS[0] | RO | 0x00 |
| 0x1d | GLOBAL | STARTUP_STS[1] | RO | 0x00 |
| 0x1e | GLOBAL | DEVICE_STS[0] | RO | 0x00 |
| 0x1f | GLOBAL | DEVICE_STS[1] | RO | 0x00 |

## 10.2 SSI Block

Table 23. SSI Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x20 | SSI | SPI_CNFG | RW | 0x00 |
| 0x21 | SSI | I2C_FLTR_CNFG | RW | 0x01 |
| 0x22 | SSI | I2C_TIMING_CNFG | RW | 0x22 |
| 0x23 | SSI | I2C_ADDR_CNFG | RW | 0x09 |
| 0x24 | SSI | BYTE_CNT | RW | 0x08 |
| 0x25 | SSI | SMB_CTL | RW | 0x79 |

Table 23. SSI Block Physical Register Addresses (Cont.)

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x26 | SSI | SSI_GLOBAL_CNFG | RW | 0x01 |
| 0x27 | SSI | SSI_STS | R/W1C | 0x00 |

## 10.3 XO Block

Table 24. XO Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x2c | XO | XO_CNFG[0] | RW | 0x45 |
| 0x2d | XO | XO_CNFG[1] | RW | 0x28 |
| 0x2e | XO | XO_CNFG[2] | RW | 0x68 |
| 0x2f | XO | XO_CNFG[3] | RW | 0x20 |

## 10.4 CLKIN Block

Table 25. CLKIN Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x30 | CLKIN[0] | CLKIN_CNFG[0] | RW | 0x01 |
| 0x31 | CLKIN[0] | CLKIN_CNFG[1] | RW | 0x80 |
| 0x34 | CLKIN[1] | CLKIN_CNFG[0] | RW | 0x01 |
| 0x35 | CLKIN[1] | CLKIN_CNFG[1] | RW | 0x80 |

## 10.5 REF Block

Table 26. REF Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x38 | REF[0] | PREDIV_CNFG[0] | RW | 0x00 |
| 0x39 | REF[0] | PREDIV_CNFG[1] | RW | 0x00 |
| 0x3a | REF[0] | PREDIV_CNFG[2] | RW | 0x50 |
| 0x3b | REF[0] | PREDIV_CNFG[3] | RW | 0x00 |
| 0x3c | REF[1] | PREDIV_CNFG[0] | RW | 0x00 |
| 0x3d | REF[1] | PREDIV_CNFG[1] | RW | 0x00 |
| 0x3e | REF[1] | PREDIV_CNFG[2] | RW | 0x50 |
| 0x3f | REF[1] | PREDIV_CNFG[3] | RW | 0x00 |
| 0x40 | REF[2] | PREDIV_CNFG[0] | RW | 0x00 |
| 0x41 | REF[2] | PREDIV_CNFG[1] | RW | 0x00 |
| 0x42 | REF[2] | PREDIV_CNFG[2] | RW | 0x50 |
| 0x43 | REF[2] | PREDIV_CNFG[3] | RW | 0x00 |

**Table 26. REF Block Physical Register Addresses (Cont.)**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x44 | REF[3] | PREDIV_CNFG[0] | RW | 0x00 |
| 0x45 | REF[3] | PREDIV_CNFG[1] | RW | 0x00 |
| 0x46 | REF[3] | PREDIV_CNFG[2] | RW | 0x50 |
| 0x47 | REF[3] | PREDIV_CNFG[3] | RW | 0x00 |

## 10.6 LOSMON Block

**Table 27. LOSMON Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x50 | LOSMON[0] | LOSMON_WINDOW[0] | RW | 0x00 |
| 0x51 | LOSMON[0] | LOSMON_WINDOW[1] | RW | 0x00 |
| 0x52 | LOSMON[0] | LOSMON_NOMINAL[0] | RW | 0x00 |
| 0x53 | LOSMON[0] | LOSMON_NOMINAL[1] | RW | 0x00 |
| 0x54 | LOSMON[0] | LOSMON_THRESH[0] | RW | 0x00 |
| 0x55 | LOSMON[0] | LOSMON_THRESH[1] | RW | 0x00 |
| 0x56 | LOSMON[0] | LOSMON_THRESH[2] | RW | 0x00 |
| 0x57 | LOSMON[0] | LOSMON_THRESH[3] | RW | 0x00 |
| 0x58 | LOSMON[0] | LOSMON_QUAL | RW | 0x44 |
| 0x59 | LOSMON[0] | LOSMON_STS | RO | 0x03 |
| 0x5a | LOSMON[0] | LOSMON_EVENT | R/W1C | 0x01 |
| 0x5b | LOSMON[0] | LOSMON_CNT | RW | 0x00 |
| 0x60 | LOSMON[1] | LOSMON_WINDOW[0] | RW | 0x00 |
| 0x61 | LOSMON[1] | LOSMON_WINDOW[1] | RW | 0x00 |
| 0x62 | LOSMON[1] | LOSMON_NOMINAL[0] | RW | 0x00 |
| 0x63 | LOSMON[1] | LOSMON_NOMINAL[1] | RW | 0x00 |
| 0x64 | LOSMON[1] | LOSMON_THRESH[0] | RW | 0x00 |
| 0x65 | LOSMON[1] | LOSMON_THRESH[1] | RW | 0x00 |
| 0x66 | LOSMON[1] | LOSMON_THRESH[2] | RW | 0x00 |
| 0x67 | LOSMON[1] | LOSMON_THRESH[3] | RW | 0x00 |
| 0x68 | LOSMON[1] | LOSMON_QUAL | RW | 0x44 |
| 0x69 | LOSMON[1] | LOSMON_STS | RO | 0x03 |
| 0x6a | LOSMON[1] | LOSMON_EVENT | R/W1C | 0x01 |
| 0x6b | LOSMON[1] | LOSMON_CNT | RW | 0x00 |
| 0x70 | LOSMON[2] | LOSMON_WINDOW[0] | RW | 0x00 |
| 0x71 | LOSMON[2] | LOSMON_WINDOW[1] | RW | 0x00 |
| 0x72 | LOSMON[2] | LOSMON_NOMINAL[0] | RW | 0x00 |
| 0x73 | LOSMON[2] | LOSMON_NOMINAL[1] | RW | 0x00 |

Table 27. LOSMON Block Physical Register Addresses (Cont.)

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x74 | LOSMON[2] | LOSMON_THRESH[0] | RW | 0x00 |
| 0x75 | LOSMON[2] | LOSMON_THRESH[1] | RW | 0x00 |
| 0x76 | LOSMON[2] | LOSMON_THRESH[2] | RW | 0x00 |
| 0x77 | LOSMON[2] | LOSMON_THRESH[3] | RW | 0x00 |
| 0x78 | LOSMON[2] | LOSMON_QUAL | RW | 0x44 |
| 0x79 | LOSMON[2] | LOSMON_STS | RO | 0x03 |
| 0x7a | LOSMON[2] | LOSMON_EVENT | R/W1C | 0x01 |
| 0x7b | LOSMON[2] | LOSMON_CNT | RW | 0x00 |
| 0x80 | LOSMON[3] | LOSMON_WINDOW[0] | RW | 0x00 |
| 0x81 | LOSMON[3] | LOSMON_WINDOW[1] | RW | 0x00 |
| 0x82 | LOSMON[3] | LOSMON_NOMINAL[0] | RW | 0x00 |
| 0x83 | LOSMON[3] | LOSMON_NOMINAL[1] | RW | 0x00 |
| 0x84 | LOSMON[3] | LOSMON_THRESH[0] | RW | 0x00 |
| 0x85 | LOSMON[3] | LOSMON_THRESH[1] | RW | 0x00 |
| 0x86 | LOSMON[3] | LOSMON_THRESH[2] | RW | 0x00 |
| 0x87 | LOSMON[3] | LOSMON_THRESH[3] | RW | 0x00 |
| 0x88 | LOSMON[3] | LOSMON_QUAL | RW | 0x44 |
| 0x89 | LOSMON[3] | LOSMON_STS | RO | 0x03 |
| 0x8a | LOSMON[3] | LOSMON_EVENT | R/W1C | 0x01 |
| 0x8b | LOSMON[3] | LOSMON_CNT | RW | 0x00 |
| 0x90 | LOSMON[4] | LOSMON_WINDOW[0] | RW | 0x00 |
| 0x91 | LOSMON[4] | LOSMON_WINDOW[1] | RW | 0x00 |
| 0x92 | LOSMON[4] | LOSMON_NOMINAL[0] | RW | 0x00 |
| 0x93 | LOSMON[4] | LOSMON_NOMINAL[1] | RW | 0x00 |
| 0x94 | LOSMON[4] | LOSMON_THRESH[0] | RW | 0x00 |
| 0x95 | LOSMON[4] | LOSMON_THRESH[1] | RW | 0x00 |
| 0x96 | LOSMON[4] | LOSMON_THRESH[2] | RW | 0x00 |
| 0x97 | LOSMON[4] | LOSMON_THRESH[3] | RW | 0x00 |
| 0x98 | LOSMON[4] | LOSMON_QUAL | RW | 0x44 |
| 0x99 | LOSMON[4] | LOSMON_STS | RO | 0x03 |
| 0x9a | LOSMON[4] | LOSMON_EVENT | R/W1C | 0x01 |
| 0x9b | LOSMON[4] | LOSMON_CNT | RW | 0x00 |

## 10.7 FREQMON Block

**Table 28. FREQMON Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0xa0 | FREQMON[0] | FREQMON_WINDOW[0] | RW | 0x00 |
| 0xa1 | FREQMON[0] | FREQMON_WINDOW[1] | RW | 0x00 |
| 0xa2 | FREQMON[0] | FREQMON_WINDOW[2] | RW | 0x00 |
| 0xa3 | FREQMON[0] | FREQMON_WINDOW[3] | RW | 0x00 |
| 0xa4 | FREQMON[0] | FREQMON_NOMINAL[0] | RW | 0x00 |
| 0xa5 | FREQMON[0] | FREQMON_NOMINAL[1] | RW | 0x00 |
| 0xa6 | FREQMON[0] | FREQMON_NOMINAL[2] | RW | 0x00 |
| 0xa7 | FREQMON[0] | FREQMON_NOMINAL[3] | RW | 0x00 |
| 0xa8 | FREQMON[0] | FREQMON_THRESH[0] | RW | 0x00 |
| 0xa9 | FREQMON[0] | FREQMON_THRESH[1] | RW | 0x00 |
| 0xaa | FREQMON[0] | FREQMON_THRESH[2] | RW | 0x00 |
| 0xab | FREQMON[0] | FREQMON_THRESH[3] | RW | 0x00 |
| 0xac | FREQMON[0] | FREQMON_THRESH[4] | RW | 0x00 |
| 0xad | FREQMON[0] | FREQMON_THRESH[5] | RW | 0x00 |
| 0xae | FREQMON[0] | FREQMON_THRESH[6] | RW | 0x00 |
| 0xaf | FREQMON[0] | FREQMON_THRESH[7] | RW | 0x00 |
| 0xb0 | FREQMON[0] | FREQMON_STS | RO | 0x01 |
| 0xb1 | FREQMON[0] | FREQMON_EVENT | R/W1C | 0x01 |
| 0xb4 | FREQMON[0] | FREQMON_OFFSET[0] | RO | 0xff |
| 0xb5 | FREQMON[0] | FREQMON_OFFSET[1] | RO | 0xff |
| 0xb6 | FREQMON[0] | FREQMON_OFFSET[2] | RO | 0xff |
| 0xb7 | FREQMON[0] | FREQMON_OFFSET[3] | RO | 0x07 |
| 0xc0 | FREQMON[1] | FREQMON_WINDOW[0] | RW | 0x00 |
| 0xc1 | FREQMON[1] | FREQMON_WINDOW[1] | RW | 0x00 |
| 0xc2 | FREQMON[1] | FREQMON_WINDOW[2] | RW | 0x00 |
| 0xc3 | FREQMON[1] | FREQMON_WINDOW[3] | RW | 0x00 |
| 0xc4 | FREQMON[1] | FREQMON_NOMINAL[0] | RW | 0x00 |
| 0xc5 | FREQMON[1] | FREQMON_NOMINAL[1] | RW | 0x00 |
| 0xc6 | FREQMON[1] | FREQMON_NOMINAL[2] | RW | 0x00 |
| 0xc7 | FREQMON[1] | FREQMON_NOMINAL[3] | RW | 0x00 |
| 0xc8 | FREQMON[1] | FREQMON_THRESH[0] | RW | 0x00 |
| 0xc9 | FREQMON[1] | FREQMON_THRESH[1] | RW | 0x00 |
| 0xca | FREQMON[1] | FREQMON_THRESH[2] | RW | 0x00 |
| 0xcb | FREQMON[1] | FREQMON_THRESH[3] | RW | 0x00 |
| 0xcc | FREQMON[1] | FREQMON_THRESH[4] | RW | 0x00 |
| 0xcd | FREQMON[1] | FREQMON_THRESH[5] | RW | 0x00 |

Table 28. FREQMON Block Physical Register Addresses (Cont.)

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0xce | FREQMON[1] | FREQMON_THRESH[6] | RW | 0x00 |
| 0xcf | FREQMON[1] | FREQMON_THRESH[7] | RW | 0x00 |
| 0xd0 | FREQMON[1] | FREQMON_STS | RO | 0x01 |
| 0xd1 | FREQMON[1] | FREQMON_EVENT | R/W1C | 0x01 |
| 0xd4 | FREQMON[1] | FREQMON_OFFSET[0] | RO | 0xff |
| 0xd5 | FREQMON[1] | FREQMON_OFFSET[1] | RO | 0xff |
| 0xd6 | FREQMON[1] | FREQMON_OFFSET[2] | RO | 0xff |
| 0xd7 | FREQMON[1] | FREQMON_OFFSET[3] | RO | 0x07 |
| 0xe0 | FREQMON[2] | FREQMON_WINDOW[0] | RW | 0x00 |
| 0xe1 | FREQMON[2] | FREQMON_WINDOW[1] | RW | 0x00 |
| 0xe2 | FREQMON[2] | FREQMON_WINDOW[2] | RW | 0x00 |
| 0xe3 | FREQMON[2] | FREQMON_WINDOW[3] | RW | 0x00 |
| 0xe4 | FREQMON[2] | FREQMON_NOMINAL[0] | RW | 0x00 |
| 0xe5 | FREQMON[2] | FREQMON_NOMINAL[1] | RW | 0x00 |
| 0xe6 | FREQMON[2] | FREQMON_NOMINAL[2] | RW | 0x00 |
| 0xe7 | FREQMON[2] | FREQMON_NOMINAL[3] | RW | 0x00 |
| 0xe8 | FREQMON[2] | FREQMON_THRESH[0] | RW | 0x00 |
| 0xe9 | FREQMON[2] | FREQMON_THRESH[1] | RW | 0x00 |
| 0xea | FREQMON[2] | FREQMON_THRESH[2] | RW | 0x00 |
| 0xeb | FREQMON[2] | FREQMON_THRESH[3] | RW | 0x00 |
| 0xec | FREQMON[2] | FREQMON_THRESH[4] | RW | 0x00 |
| 0xed | FREQMON[2] | FREQMON_THRESH[5] | RW | 0x00 |
| 0xee | FREQMON[2] | FREQMON_THRESH[6] | RW | 0x00 |
| 0xef | FREQMON[2] | FREQMON_THRESH[7] | RW | 0x00 |
| 0xf0 | FREQMON[2] | FREQMON_STS | RO | 0x01 |
| 0xf1 | FREQMON[2] | FREQMON_EVENT | R/W1C | 0x01 |
| 0xf4 | FREQMON[2] | FREQMON_OFFSET[0] | RO | 0xff |
| 0xf5 | FREQMON[2] | FREQMON_OFFSET[1] | RO | 0xff |
| 0xf6 | FREQMON[2] | FREQMON_OFFSET[2] | RO | 0xff |
| 0xf7 | FREQMON[2] | FREQMON_OFFSET[3] | RO | 0x07 |
| 0x100 | FREQMON[3] | FREQMON_WINDOW[0] | RW | 0x00 |
| 0x101 | FREQMON[3] | FREQMON_WINDOW[1] | RW | 0x00 |
| 0x102 | FREQMON[3] | FREQMON_WINDOW[2] | RW | 0x00 |
| 0x103 | FREQMON[3] | FREQMON_WINDOW[3] | RW | 0x00 |
| 0x104 | FREQMON[3] | FREQMON_NOMINAL[0] | RW | 0x00 |
| 0x105 | FREQMON[3] | FREQMON_NOMINAL[1] | RW | 0x00 |
| 0x106 | FREQMON[3] | FREQMON_NOMINAL[2] | RW | 0x00 |

Table 28. FREQMON Block Physical Register Addresses (Cont.)

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x107 | FREQMON[3] | FREQMON_NOMINAL[3] | RW | 0x00 |
| 0x108 | FREQMON[3] | FREQMON_THRESH[0] | RW | 0x00 |
| 0x109 | FREQMON[3] | FREQMON_THRESH[1] | RW | 0x00 |
| 0x10a | FREQMON[3] | FREQMON_THRESH[2] | RW | 0x00 |
| 0x10b | FREQMON[3] | FREQMON_THRESH[3] | RW | 0x00 |
| 0x10c | FREQMON[3] | FREQMON_THRESH[4] | RW | 0x00 |
| 0x10d | FREQMON[3] | FREQMON_THRESH[5] | RW | 0x00 |
| 0x10e | FREQMON[3] | FREQMON_THRESH[6] | RW | 0x00 |
| 0x10f | FREQMON[3] | FREQMON_THRESH[7] | RW | 0x00 |
| 0x110 | FREQMON[3] | FREQMON_STS | RO | 0x01 |
| 0x111 | FREQMON[3] | FREQMON_EVENT | R/W1C | 0x01 |
| 0x114 | FREQMON[3] | FREQMON_OFFSET[0] | RO | 0xff |
| 0x115 | FREQMON[3] | FREQMON_OFFSET[1] | RO | 0xff |
| 0x116 | FREQMON[3] | FREQMON_OFFSET[2] | RO | 0xff |
| 0x117 | FREQMON[3] | FREQMON_OFFSET[3] | RO | 0x07 |

## 10.8 APLL Block

Table 29. APLL Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x120 | APLL | APLL_FB_DIV_FRAC[0] | RW | 0x00 |
| 0x121 | APLL | APLL_FB_DIV_FRAC[1] | RW | 0x00 |
| 0x122 | APLL | APLL_FB_DIV_FRAC[2] | RW | 0x00 |
| 0x123 | APLL | APLL_FB_DIV_FRAC[3] | RW | 0x00 |
| 0x124 | APLL | APLL_FB_DIV_INT[0] | RW | 0x69 |
| 0x125 | APLL | APLL_FB_DIV_INT[1] | RW | 0x00 |
| 0x126 | APLL | APLL_FB_SDM_CNFG | RW | 0x03 |
| 0x127 | APLL | APLL_CNFG | RW | 0x01 |
| 0x128 | APLL | CP_CNFG[0] | RW | 0x33 |
| 0x129 | APLL | CP_CNFG[1] | RW | 0x03 |
| 0x12a | APLL | LPF_CNFG | RW | 0x74 |
| 0x12b | APLL | LPF_3RD_CNFG | RW | 0x44 |
| 0x135 | APLL | APLL_REF_FB_CNFG | RW | 0x00 |
| 0x13f | APLL | APLL_STS | RO | 0x00 |
| 0x140 | APLL | APLL_EVENT | R/W1C | 0x00 |
| 0x141 | APLL | APLL_LOL_CNT | RW | 0x00 |

## 10.9 TDCAPLL Block

**Table 30. TDCAPLL Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x160 | TDCAPLL | TDC_APLL_CNFG[0] | RW | 0x4a |
| 0x161 | TDCAPLL | TDC_APLL_CNFG[1] | RW | 0x1e |
| 0x162 | TDCAPLL | TDC_FB_DIV_FRAC[0] | RW | 0x00 |
| 0x163 | TDCAPLL | TDC_FB_DIV_FRAC[1] | RW | 0x28 |
| 0x164 | TDCAPLL | TDC_FB_DIV_INT | RW | 0x24 |
| 0x165 | TDCAPLL | TDC_FB_SDM_CNFG | RW | 0x81 |
| 0x166 | TDCAPLL | TDC_REF_DIV_CNFG | RW | 0x01 |
| 0x16a | TDCAPLL | TDC_FILTER_STS[0] | RO | 0x00 |
| 0x16b | TDCAPLL | TDC_FILTER_STS[1] | RO | 0x00 |

## 10.10 DPLL Block

**Table 31. DPLL Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x180 | DPLL | DPLL_REF_FB_CNFG | RW | 0x04 |
| 0x181 | DPLL | DPLL_MODE | RW | 0x81 |
| 0x182 | DPLL | DPLL_DECIMATOR | RW | 0x36 |
| 0x183 | DPLL | DPLL_XTAL_OFFSET | RW | 0x00 |
| 0x184 | DPLL | HOLDOVER_CNFG[0] | RW | 0x00 |
| 0x185 | DPLL | HOLDOVER_CNFG[1] | RW | 0x38 |
| 0x186 | DPLL | DPLL_BANDWIDTH[0] | RW | 0x58 |
| 0x187 | DPLL | DPLL_BANDWIDTH[1] | RW | 0x70 |
| 0x188 | DPLL | DPLL_DAMPING[0] | RW | 0x00 |
| 0x189 | DPLL | DPLL_DAMPING[1] | RW | 0x29 |
| 0x18a | DPLL | DPLL_FB_CORR[0] | RW | 0x00 |
| 0x18b | DPLL | DPLL_FB_CORR[1] | RW | 0x00 |
| 0x18c | DPLL | DPLL_PHASE_SLOPE_LIMIT[0] | RW | 0xff |
| 0x18d | DPLL | DPLL_PHASE_SLOPE_LIMIT[1] | RW | 0xff |
| 0x18e | DPLL | DPLL_PHASE_SLOPE_LIMIT[2] | RW | 0xff |
| 0x18f | DPLL | DPLL_PHASE_SLOPE_LIMIT[3] | RW | 0x1f |
| 0x190 | DPLL | DPLL_PHASE_OFFSET[0] | RW | 0x00 |
| 0x191 | DPLL | DPLL_PHASE_OFFSET[1] | RW | 0x00 |
| 0x192 | DPLL | DPLL_PHASE_OFFSET[2] | RW | 0x00 |
| 0x193 | DPLL | DPLL_PHASE_OFFSET[3] | RW | 0x00 |
| 0x194 | DPLL | DPLL_WRITE_FREQ[0] | RW | 0x00 |
| 0x195 | DPLL | DPLL_WRITE_FREQ[1] | RW | 0x00 |

**Table 31. DPLL Block Physical Register Addresses (Cont.)**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x196 | DPLL | DPLL_WRITE_FREQ[2] | RW | 0x00 |
| 0x197 | DPLL | DPLL_WRITE_FREQ[3] | RW | 0x00 |
| 0x198 | DPLL | DPLL_FB_DIV_NUM[0] | RW | 0x00 |
| 0x199 | DPLL | DPLL_FB_DIV_NUM[1] | RW | 0x00 |
| 0x19a | DPLL | DPLL_FB_DIV_NUM[2] | RW | 0x00 |
| 0x19b | DPLL | DPLL_FB_DIV_NUM[3] | RW | 0x00 |
| 0x19c | DPLL | DPLL_FB_DIV_NUM[4] | RW | 0x00 |
| 0x19d | DPLL | DPLL_FB_DIV_NUM[5] | RW | 0x00 |
| 0x1a0 | DPLL | DPLL_FB_DIV_DEN[0] | RW | 0x00 |
| 0x1a1 | DPLL | DPLL_FB_DIV_DEN[1] | RW | 0x00 |
| 0x1a2 | DPLL | DPLL_FB_DIV_DEN[2] | RW | 0x80 |
| 0x1a3 | DPLL | DPLL_FB_DIV_DEN[3] | RW | 0x00 |
| 0x1a4 | DPLL | DPLL_FB_DIV_DEN[4] | RW | 0x00 |
| 0x1a5 | DPLL | DPLL_FB_DIV_DEN[5] | RW | 0x00 |
| 0x1a8 | DPLL | DPLL_FB_DIV_INT[0] | RW | 0x90 |
| 0x1a9 | DPLL | DPLL_FB_DIV_INT[1] | RW | 0x01 |
| 0x1aa | DPLL | DPLL_FB_DIV_INT[2] | RW | 0x00 |
| 0x1ac | DPLL | DPLL_LOCK[0] | RW | 0x55 |
| 0x1ad | DPLL | DPLL_LOCK[1] | RW | 0x01 |
| 0x1ae | DPLL | DPLL_LOCK[2] | RW | 0xff |
| 0x1af | DPLL | DPLL_LOCK[3] | RW | 0x00 |
| 0x1b1 | DPLL | DPLL_STS | RO | 0x00 |
| 0x1b2 | DPLL | DPLL_EVENT | R/W1C | 0x00 |
| 0x1b3 | DPLL | DPLL_LOL_CNT | RW | 0x00 |
| 0x1b4 | DPLL | DPLL_FILTER_STS[0] | RO | 0x00 |
| 0x1b5 | DPLL | DPLL_FILTER_STS[1] | RO | 0x00 |
| 0x1b6 | DPLL | DPLL_FILTER_STS[2] | RO | 0x00 |
| 0x1b7 | DPLL | DPLL_FILTER_STS[3] | RO | 0x00 |
| 0x1b8 | DPLL | DPLL_PHASE_STS[0] | RO | 0x00 |
| 0x1b9 | DPLL | DPLL_PHASE_STS[1] | RO | 0x00 |
| 0x1ba | DPLL | DPLL_PHASE_STS[2] | RO | 0x00 |
| 0x1bb | DPLL | DPLL_PHASE_STS[3] | RO | 0x00 |

## 10.11 IOD Block

**Table 32. IOD Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x1c0 | IOD[0] | IOD_INT_CNFG[0] | RW | 0x64 |
| 0x1c1 | IOD[0] | IOD_INT_CNFG[1] | RW | 0x00 |
| 0x1c2 | IOD[0] | IOD_INT_CNFG[2] | RW | 0x00 |
| 0x1c3 | IOD[0] | IOD_INT_CNFG[3] | RW | 0x00 |
| 0x1c4 | IOD[0] | IOD_PHASE_CNFG[0] | RW | 0x00 |
| 0x1c5 | IOD[0] | IOD_PHASE_CNFG[1] | R/W1S | 0x00 |
| 0x1c6 | IOD[0] | SYSREF_CNFG[0] | RW | 0x00 |
| 0x1c7 | IOD[0] | SYSREF_CNFG[1] | RW | 0x00 |
| 0x1c8 | IOD[1] | IOD_INT_CNFG[0] | RW | 0x64 |
| 0x1c9 | IOD[1] | IOD_INT_CNFG[1] | RW | 0x00 |
| 0x1ca | IOD[1] | IOD_INT_CNFG[2] | RW | 0x00 |
| 0x1cb | IOD[1] | IOD_INT_CNFG[3] | RW | 0x00 |
| 0x1cc | IOD[1] | IOD_PHASE_CNFG[0] | RW | 0x00 |
| 0x1cd | IOD[1] | IOD_PHASE_CNFG[1] | R/W1S | 0x00 |
| 0x1ce | IOD[1] | SYSREF_CNFG[0] | RW | 0x00 |
| 0x1cf | IOD[1] | SYSREF_CNFG[1] | RW | 0x00 |
| 0x1d0 | IOD[2] | IOD_INT_CNFG[0] | RW | 0x64 |
| 0x1d1 | IOD[2] | IOD_INT_CNFG[1] | RW | 0x00 |
| 0x1d2 | IOD[2] | IOD_INT_CNFG[2] | RW | 0x00 |
| 0x1d3 | IOD[2] | IOD_INT_CNFG[3] | RW | 0x00 |
| 0x1d4 | IOD[2] | IOD_PHASE_CNFG[0] | RW | 0x00 |
| 0x1d5 | IOD[2] | IOD_PHASE_CNFG[1] | R/W1S | 0x00 |
| 0x1d6 | IOD[2] | SYSREF_CNFG[0] | RW | 0x00 |
| 0x1d7 | IOD[2] | SYSREF_CNFG[1] | RW | 0x00 |
| 0x1d8 | IOD[3] | IOD_INT_CNFG[0] | RW | 0x64 |
| 0x1d9 | IOD[3] | IOD_INT_CNFG[1] | RW | 0x00 |
| 0x1da | IOD[3] | IOD_INT_CNFG[2] | RW | 0x00 |
| 0x1db | IOD[3] | IOD_INT_CNFG[3] | RW | 0x00 |
| 0x1dc | IOD[3] | IOD_PHASE_CNFG[0] | RW | 0x00 |
| 0x1dd | IOD[3] | IOD_PHASE_CNFG[1] | R/W1S | 0x00 |
| 0x1de | IOD[3] | SYSREF_CNFG[0] | RW | 0x00 |
| 0x1df | IOD[3] | SYSREF_CNFG[1] | RW | 0x00 |

## 10.12 FOD Block

Table 33. FOD Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x1e0 | FOD[0] | FOD_INT_CNFG[0] | RW | 0x64 |
| 0x1e1 | FOD[0] | FOD_INT_CNFG[1] | RW | 0x00 |
| 0x1e2 | FOD[0] | FOD_INT_CNFG[2] | RW | 0x00 |
| 0x1e3 | FOD[0] | FOD_INT_CNFG[3] | RW | 0x00 |
| 0x1e4 | FOD[0] | FOD_INT_CNFG[4] | RW | 0x00 |
| 0x1e5 | FOD[0] | FOD_INT_CNFG[5] | RW | 0x00 |
| 0x1e6 | FOD[0] | FOD_INT_CNFG[6] | RW | 0x00 |
| 0x1e7 | FOD[0] | FOD_INT_CNFG[7] | RW | 0x00 |
| 0x1e8 | FOD[0] | FOD_PHASE_CNFG[0] | RW | 0xf0 |
| 0x1e9 | FOD[0] | FOD_PHASE_CNFG[1] | R/W1S | 0x43 |
| 0x1f0 | FOD[1] | FOD_INT_CNFG[0] | RW | 0x64 |
| 0x1f1 | FOD[1] | FOD_INT_CNFG[1] | RW | 0x00 |
| 0x1f2 | FOD[1] | FOD_INT_CNFG[2] | RW | 0x00 |
| 0x1f3 | FOD[1] | FOD_INT_CNFG[3] | RW | 0x00 |
| 0x1f4 | FOD[1] | FOD_INT_CNFG[4] | RW | 0x00 |
| 0x1f5 | FOD[1] | FOD_INT_CNFG[5] | RW | 0x00 |
| 0x1f6 | FOD[1] | FOD_INT_CNFG[6] | RW | 0x00 |
| 0x1f7 | FOD[1] | FOD_INT_CNFG[7] | RW | 0x00 |
| 0x1f8 | FOD[1] | FOD_PHASE_CNFG[0] | RW | 0xf0 |
| 0x1f9 | FOD[1] | FOD_PHASE_CNFG[1] | R/W1S | 0x43 |
| 0x200 | FOD[2] | FOD_INT_CNFG[0] | RW | 0x64 |
| 0x201 | FOD[2] | FOD_INT_CNFG[1] | RW | 0x00 |
| 0x202 | FOD[2] | FOD_INT_CNFG[2] | RW | 0x00 |
| 0x203 | FOD[2] | FOD_INT_CNFG[3] | RW | 0x00 |
| 0x204 | FOD[2] | FOD_INT_CNFG[4] | RW | 0x00 |
| 0x205 | FOD[2] | FOD_INT_CNFG[5] | RW | 0x00 |
| 0x206 | FOD[2] | FOD_INT_CNFG[6] | RW | 0x00 |
| 0x207 | FOD[2] | FOD_INT_CNFG[7] | RW | 0x00 |
| 0x208 | FOD[2] | FOD_PHASE_CNFG[0] | RW | 0xf0 |
| 0x209 | FOD[2] | FOD_PHASE_CNFG[1] | R/W1S | 0x43 |

## 10.13 OUT Block

**Table 34. OUT Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x240 | OUT[0] | ODRV_EN | RW | 0x06 |
| 0x242 | OUT[0] | ODRV_CNFG[0] | RW | 0x3c |
| 0x243 | OUT[0] | ODRV_CNFG[1] | RW | 0x00 |
| 0x244 | OUT[1] | ODRV_EN | RW | 0x06 |
| 0x246 | OUT[1] | ODRV_CNFG[0] | RW | 0x3c |
| 0x247 | OUT[1] | ODRV_CNFG[1] | RW | 0x00 |
| 0x248 | OUT[2] | ODRV_EN | RW | 0x06 |
| 0x24a | OUT[2] | ODRV_CNFG[0] | RW | 0x3c |
| 0x24b | OUT[2] | ODRV_CNFG[1] | RW | 0x00 |
| 0x24c | OUT[3] | ODRV_EN | RW | 0x06 |
| 0x24e | OUT[3] | ODRV_CNFG[0] | RW | 0x3c |
| 0x24f | OUT[3] | ODRV_CNFG[1] | RW | 0x00 |
| 0x250 | OUT[4] | ODRV_EN | RW | 0x06 |
| 0x252 | OUT[4] | ODRV_CNFG[0] | RW | 0x3c |
| 0x253 | OUT[4] | ODRV_CNFG[1] | RW | 0x00 |
| 0x254 | OUT[5] | ODRV_EN | RW | 0x06 |
| 0x256 | OUT[5] | ODRV_CNFG[0] | RW | 0x3c |
| 0x257 | OUT[5] | ODRV_CNFG[1] | RW | 0x00 |
| 0x258 | OUT[6] | ODRV_EN | RW | 0x06 |
| 0x25a | OUT[6] | ODRV_CNFG[0] | RW | 0x3c |
| 0x25b | OUT[6] | ODRV_CNFG[1] | RW | 0x00 |
| 0x25c | OUT[7] | ODRV_EN | RW | 0x06 |
| 0x25e | OUT[7] | ODRV_CNFG[0] | RW | 0x3c |
| 0x25f | OUT[7] | ODRV_CNFG[1] | RW | 0x00 |
| 0x260 | OUT[8] | ODRV_EN | RW | 0x06 |
| 0x262 | OUT[8] | ODRV_CNFG[0] | RW | 0x3c |
| 0x263 | OUT[8] | ODRV_CNFG[1] | RW | 0x00 |
| 0x264 | OUT[9] | ODRV_EN | RW | 0x06 |
| 0x266 | OUT[9] | ODRV_CNFG[0] | RW | 0x3c |
| 0x267 | OUT[9] | ODRV_CNFG[1] | RW | 0x00 |
| 0x268 | OUT[10] | ODRV_EN | RW | 0x06 |
| 0x26a | OUT[10] | ODRV_CNFG[0] | RW | 0x3c |
| 0x26b | OUT[10] | ODRV_CNFG[1] | RW | 0x00 |
| 0x26c | OUT[11] | ODRV_EN | RW | 0x06 |
| 0x26e | OUT[11] | ODRV_CNFG[0] | RW | 0x3c |
| 0x26f | OUT[11] | ODRV_CNFG[1] | RW | 0x00 |

## 10.14 BANK Block

**Table 35. BANK Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x280 | BANK[0] | OUT_BANK_CNFG | RW | 0x05 |
| 0x284 | BANK[1] | OUT_BANK_CNFG | RW | 0x05 |
| 0x288 | BANK[2] | OUT_BANK_CNFG | RW | 0x05 |
| 0x28c | BANK[3] | OUT_BANK_CNFG | RW | 0x05 |
| 0x290 | BANK[4] | OUT_BANK_CNFG | RW | 0x05 |
| 0x294 | BANK[5] | OUT_BANK_CNFG | RW | 0x05 |
| 0x298 | BANK[6] | OUT_BANK_CNFG | RW | 0x05 |

## 10.15 GPI Block

**Table 36. GPI Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x29c | GPI[0] | GPI_CNFG[0] | RW | 0x7f |
| 0x29d | GPI[0] | GPI_CNFG[1] | RW | 0x00 |
| 0x29e | GPI[0] | GPI_STS | RO | 0x00 |
| 0x2a0 | GPI[1] | GPI_CNFG[0] | RW | 0x7f |
| 0x2a1 | GPI[1] | GPI_CNFG[1] | RW | 0x00 |
| 0x2a2 | GPI[1] | GPI_STS | RO | 0x00 |
| 0x2a4 | GPI[2] | GPI_CNFG[0] | RW | 0x7f |
| 0x2a5 | GPI[2] | GPI_CNFG[1] | RW | 0x00 |
| 0x2a6 | GPI[2] | GPI_STS | RO | 0x00 |
| 0x2a8 | GPI[3] | GPI_CNFG[0] | RW | 0x7f |
| 0x2a9 | GPI[3] | GPI_CNFG[1] | RW | 0x00 |
| 0x2aa | GPI[3] | GPI_STS | RO | 0x00 |

## 10.16 GPIO Block

**Table 37. GPIO Block Physical Register Addresses**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x2ac | GPIO[0] | GPIO_CNFG[0] | RW | 0x90 |
| 0x2ad | GPIO[0] | GPIO_CNFG[1] | RW | 0x04 |
| 0x2ae | GPIO[0] | GPIO_STS | RO | 0x00 |
| 0x2b0 | GPIO[1] | GPIO_CNFG[0] | RW | 0x90 |
| 0x2b1 | GPIO[1] | GPIO_CNFG[1] | RW | 0x04 |
| 0x2b2 | GPIO[1] | GPIO_STS | RO | 0x00 |
| 0x2b4 | GPIO[2] | GPIO_CNFG[0] | RW | 0x90 |

Table 37. GPIO Block Physical Register Addresses (Cont.)

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x2b5 | GPIO[2] | GPIO_CNFG[1] | RW | 0x04 |
| 0x2b6 | GPIO[2] | GPIO_STS | RO | 0x00 |
| 0x2b8 | GPIO[3] | GPIO_CNFG[0] | RW | 0x90 |
| 0x2b9 | GPIO[3] | GPIO_CNFG[1] | RW | 0x04 |
| 0x2ba | GPIO[3] | GPIO_STS | RO | 0x00 |
| 0x2bc | GPIO[4] | GPIO_CNFG[0] | RW | 0x90 |
| 0x2bd | GPIO[4] | GPIO_CNFG[1] | RW | 0x04 |
| 0x2be | GPIO[4] | GPIO_STS | RO | 0x00 |

## 10.17 SSC Block

Table 38. SSC Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x300 | SSC[0] | SSC_CNFG[0] | RW | 0x8c |
| 0x301 | SSC[0] | SSC_CNFG[1] | RW | 0x2b |
| 0x302 | SSC[0] | SSC_CNFG[2] | RW | 0x51 |
| 0x303 | SSC[0] | SSC_CNFG[3] | RW | 0x00 |
| 0x304 | SSC[1] | SSC_CNFG[0] | RW | 0x8c |
| 0x305 | SSC[1] | SSC_CNFG[1] | RW | 0x2b |
| 0x306 | SSC[1] | SSC_CNFG[2] | RW | 0x51 |
| 0x307 | SSC[1] | SSC_CNFG[3] | RW | 0x00 |

## 10.18 INT Block

Table 39. INT Block Physical Register Addresses

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x308 | INT | SCRATCH0[0] | RW | 0x00 |
| 0x309 | INT | SCRATCH0[1] | RW | 0x00 |
| 0x30a | INT | SCRATCH0[2] | RW | 0x00 |
| 0x30b | INT | SCRATCH0[3] | RW | 0x00 |
| 0x30c | INT | INT_EN[0] | RW | 0x00 |
| 0x30d | INT | INT_EN[1] | RW | 0x00 |
| 0x30e | INT | INT_EN[2] | RW | 0x00 |
| 0x30f | INT | INT_EN[3] | RW | 0x00 |
| 0x310 | INT | INT_STS[0] | RO | 0x00 |
| 0x311 | INT | INT_STS[1] | RO | 0x00 |

**Table 39. INT Block Physical Register Addresses (Cont.)**

| Address | Block Name | Register | Type | Default |
|---|---|---|---|---|
| 0x312 | INT | INT_STS[2] | RO | 0x00 |
| 0x313 | INT | INT_STS[3] | RO | 0x00 |

# 11. Register Descriptions

## 11.1 Global Registers

**Table 40. VENDOR_ID - Vendor ID**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:12 | dev_id_type | RO | 0x1 | Device ID Block Type.<br>A value of 0x1 indicates that this register is followed by a 16-bit Device ID register and an 16-bit Device Revision register, and a 16-bit Device Programming register. |
| 11 | reserved | RO | 0x0 | Reserved. |
| 10:0 | vendor_id | RO | 0x33 | Vendor ID.<br>Renesas/IDT JEDEC ID. |

**Table 41. DEVICE_ID - Device ID**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:0 | device_id | RW | 0x0 | Device ID.<br>For default value refer to Product Identification. This field is writeable so it can be configured from OTP. |

**Table 42. DEVICE_REV - Device Revision**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:0 | device_revision | RO | 0x0443 | Device Revision.<br>Decode as follows:<br>▪ 0x0111 = First silicon; not released.<br>▪ 0x0333 = Rev A<br>▪ 0x0443 = Rev B |

**Table 43. DEVICE_PGM - Device Programming**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:0 | dash_code | RW | 0x0 | Dash Code.<br>Decimal value assigned by IDT to identify the user configuration loaded in OTP at the factory. This field is write-able and is configured from the OTP common configuration programmed at the factory.<br>▪ 0x0 = No user configuration has been programmed at the factory |

**Table 44. DEVICE_CNFG - Device Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | reserved | RO | 0x0 | Reserved. |
| 30:24 | reserved | RW | 0x0 | Reserved. |
| 30 | float_vddo6 | RW | 0x0 | Float VDDO6 Supply.<br>▪ Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider IOD3 and output driver 11 in the VDDO6 domain.0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDO6 pin floats |
| 29 | float_vddo5 | RW | 0x0 | Float VDDO5 Supply.<br>Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider IOD2 and output driver 10 in the VDDO5 domain.<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDO5 pin floats |
| 28 | float_vddo4 | RW | 0x0 | Float VDDO4 Supply.<br>Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider FOD2 and output drivers 8/9 in the VDDO4 domain.<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDO4 pin floats |
| 27 | float_vddo3 | RW | 0x0 | Float VDDO3 Supply.<br>Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider FOD1 and output drivers 4/5/6/7in the VDDO3 domain.<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDO3 pin floats |
| 26 | float_vddo2 | RW | 0x0 | Float VDDO2 Supply.<br>Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider FOD0 and output drivers 2/3 in the VDDO2 domain.<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDO2 pin floats |
| 25 | float_vddo1 | RW | 0x0 | Float VDDO1 Supply.<br>Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider IOD1 and output driver 1 in the VDDO1 domain.<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDO1 pin floats |
| 24 | float_vddo0 | RW | 0x0 | Float VDDO0 Supply.<br>Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider IOD0 and output driver 0 in the VDDO0 domain.<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDO0 pin floats |
| 23 | reserved | RW | 0x0 | Reserved. |

**Table 44. DEVICE_CNFG - Device Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 22 | dyn_csel_deb | RW | 0x0 | Dynamic CSEL De-bounce Interval.<br>The input value across GPI/GPIO pins configured as Dynamic CSEL must be stable for this number of system clock cycles before it takes effect.<br>▪ 0x0 = 8 system clock cycles (133ns)<br>▪ 0x1 = 256 system clock cycles (4.2us) |
| 21:20 | dpll_clk_sel_deb | RW | 0x1 | DPLL_CLK_SEL De-bounce Interval.<br>The input value across GPI/GPIO pins configured as DPLL_CLK_SEL[0] and DPLL_CLK_SEL[1] must be stable for this number of system clock cycles before it takes effect.<br>▪ 0x0 = 2 system clock cycles (33ns)<br>▪ 0x1 = 4 system clock cycles (67ns)<br>▪ 0x2 = 6 system clock cycles (100ns)<br>▪ 0x3 = 8 system clock cycles (133ns) |
| 19:16 | sync_dis_wait | RW | 0x2 | Divider Synchronization Output Clock Disable Wait Time.<br>During the divider synchronization procedure, after de-asserting the output enable to the affected output drivers, the control logic waits for this period of time before stopping the clocks and synchronizing the dividers. If enabled, this must be set longer than the period of the slowest output clock, and longer than the SSC modulation interval.<br>▪ 0x0 = 1us<br>▪ 0x1 = 2us<br>▪ 0x2 = 4us<br>▪ 0x3 = 8us<br>▪ 0x4 = 16us<br>▪ 0x5 = 32us<br>▪ 0x6 = 64us<br>▪ 0x7 = 128us<br>▪ 0x8 = 256us<br>▪ 0x9 = 512us<br>▪ 0xA = 1024us<br>▪ 0xB = 2048us<br>▪ 0xC = 4096us |
| 15:12 | pwrdn_sel | RW | 0xF | PWRGD/PWRDN# Pin Select.<br>Selects the pin used as PWRGD/PWRDN#, or disables the functionality. If a pin is selected, the device waits for that pin to be high before loading the OTP/EEPROM UserCfg. That pin may also be assigned as a 2-level Dynamic CSEL pin with *gpio_func* and gpio_type, and polarity inversion must be disabled (*gpio_pol*. This setting must be programmed by the OTP/EEPROM common configuration in order to take effect during the power-up sequence.<br>▪ 0x0 = GPIO0<br>▪ 0x1 = GPIO1<br>▪ 0x2 = GPIO2<br>▪ 0x3 = GPIO3<br>▪ 0x4 = GPIO4<br>▪ 0x5 = GPI0<br>▪ 0x6 = GPI1<br>▪ 0x7 = GPI2<br>▪ 0x8 = GPI3<br>▪ 0xF: No pin selected, associated functionality disabled |

**Table 44. DEVICE_CNFG - Device Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 11:6 | reserved | RW | 0x0 | Reserved. |
| 11 | float_vddr3 | RW | 0x0 | Float VDDR3 Supply.<br>Controls Iref and Ibias to the Input Buffer LDO and IB2 in the VDDR3 domain (bonded to VDDR in the current 48 and 40 pin package variants).<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDR3 pin floats |
| 10 | float_vddr2 | RW | 0x0 | Float VDDR2 Supply.<br>Controls Iref and Ibias to the Input Buffer LDO and IB1 in the VDDR2 domain (bonded to VDDR in the current 48 and 40 pin package variants).<br>▪ 0x0 = Reference currents enabled<br>▪ 0x1 = Reference currents disabled, must be set when the VDDR2 pin floats |
| 9:6 | reserved | RW | 0x0 | Reserved. |
| 5:4 | static_csel2 | RW | 0x2 | Static Configuration Select Bit 2.<br>Selects the GPIO pin or fixed value used for static configuration select bit index 2. This setting must be programmed by the OTP/EEPROM common configuration in order to take effect during the power-up sequence.<br>▪ 0x0 = No pin selected, treated as Low (tri-level)<br>▪ 0x1 = No pin selected, treated as Mid (tri-level)<br>▪ 0x2 = GPIO2 (tri-level)<br>▪ 0x3 = No pin selected, treated as High (tri-level) |
| 3:2 | static_csel1 | RW | 0x2 | Static Configuration Select Bit 1.<br>Selects the GPIO pin or fixed value used for static configuration select bit index 1. This setting must be programmed by the OTP/EEPROM common configuration in order to take effect during the power-up sequence.<br>▪ 0x0 = No pin selected, treated as Low (tri-level)<br>▪ 0x1 = No pin selected, treated as Mid (tri-level)<br>▪ 0x2 = GPIO1 (tri-level)<br>▪ 0x3 = No pin selected, treated as High (tri-level) |
| 1:0 | static_csel0 | RW | 0x2 | Static Configuration Select Bit 0.<br>Selects the GPIO pin or fixed value used for static configuration select bit index 0. This setting must be programmed by the OTP/EEPROM common configuration in order to take effect during the power-up sequence.<br>▪ 0x0 = No pin selected, treated as Low (tri-level)<br>▪ 0x1 = No pin selected, treated as Mid (tri-level)<br>▪ 0x2 = GPIO0 (tri-level)<br>▪ 0x3 = No pin selected, treated as High (tri-level) |

**Table 45. PWR_CTL - Power Control**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:1 | reserved | RO | 0x0 | Reserved. |
| 0 | pd_restoreb | RW | 0x1 | Power Down Restore (Active Low).<br>Controls the device behavior when PWRGD/PWRDN# falls to enter power down.If pd_restoreb is set to 0, it will be reset back to 1 by the reset sequence.<br>▪ 0x0 = Full power-on-reset. The falling edge of the PWRGD/PWRDN# pin resets the device, including configuration registers, to the initial power-on-reset state. The startup sequence is executed. It will wait for the rising edge of the PWRGD/PWRDN# pin to re-latch static CSEL inputs and load the corresponding user configuration, as on initial power-up.<br>▪ 0x1 = Quick power down and power up sequence. The falling and subsequent rising edge of the PWRGD/PWRDN# pin trigger dynamic configuration loads which program the device to enter the powered down and powered up states, respectively. The startup sequence is not executed, and static CSEL inputs are not re-latched. |

**Table 46. REG_LOCK - Configuration Register Lock**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:0 | reg_lock_key | RW | 0x0 | Configuration Register Lock Key.<br>Writing this field with 0xCB sets the config_reg_ro bit to 1. Writing this field with 0x34 clears the config_reg_ro bit to 0. |

**Table 47. INIT_SYNC - Initialization and Synchronization Register**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | clr_grp1_oe | RW1S | 0x0 | Clear Reset Sequencer Sync Group 1 Output Enable.<br>Writing this bit to 1 clears the sync group 1 output enable from the reset sequencer. The reset sequencer will set it back to 1 after the next group 1 divider synchronization completes. This affects all outputs in banks that select a divider assigned to sync group 1. Self-cleared immediately. Must not be set to 1 at the same time as any of divider_sync, od_grp0_sync and od_grp1_sync are set to 1. |
| 6 | clr_grp0_oe | RW1S | 0x0 | Clear Reset Sequencer Sync Group 0 Output Enable.<br>Writing this bit to 1 clears the sync group 0 output enable from the reset sequencer. The reset sequencer will set it back to 1 after the next group 0 divider synchronization completes. This affects all outputs in banks that select a divider assigned to sync group 0. Self-cleared immediately. Must not be set to 1 at the same time as any of divider_sync, od_grp0_sync and od_grp1_sync are set to 1. |
| 5 | id_global_setb | RW | 0x1 | Input Dividers Common Set.<br>When cleared, all input dividers are held in set mode (bit is active low). This allows set and release of all dividers at roughly the same time. |
| 4 | goe | RW | 0x1 | Output Global OE.<br>This bit allows manual CSR control of the global output OE. |

**Table 47. INIT_SYNC - Initialization and Synchronization Register**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3 | od_grp1_sync | RW1S | 0x0 | Divider Group 1 Sync Trigger.<br>Write 1 to trigger synchronization of DPLL and output dividers in group 1 as defined in dpll_sync_group, iod_sync_group and fod_sync_group. The affected output clocks will be squelched for approximately the sync_dis_wait duration. Self-cleared immediately. |
| 2 | od_grp0_sync | RW1S | 0x0 | Divider Group 0 Sync Trigger.<br>Write 1 to trigger synchronization of DPLL and output dividers in group 0 as defined in dpll_sync_group, iod_sync_group and fod_sync_group. The affected output clocks will be squelched for approximately the sync_dis_wait duration. Self-cleared immediately. |
| 1 | divider_sync | RW1S | 0x0 | Divider Synchronization.<br>Write 1 to trigger synchronization of DPLL and output dividers in groups 0 and 1 as defined in dpll_sync_group, iod_sync_group and fod_sync_group.The affected output clocks will be squelched for approximately the sync_dis_wait duration. Self-cleared immediately. |
| 0 | apll_reinit | RW1S | 0x0 | APLL Reinitialization.<br>Writing this bit to 1 re-starts the startup sequence from the VCO calibration step, including divider synchronization. Self-cleared immediately. |

**Table 48. SW_RESET - Software Reset Register**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:12 | reserved | RO | 0x0 | Reserved. |
| 11 | out_rst | RW | 0x0 | Output Driver Reset.<br>This bit resets and disables all output drivers. Reset assertion acts asynchronously and can cause output glitches or runt pulses. Reset assertion acts synchronously. |
| 10 | config_rst | RW | 0x0 | Configuration Reset.<br>Writing this bit to 1 resets the digital logic including the registers and re-starts the startup sequence at the bias calibration, if the device is not already executing the startup sequence (during the startup sequence, writes to this bit are masked). This bit can be set to 1 by the first write in a static OTP/EEPROM User Configuration in order to revert the CSRs to their default values. The static CSEL pins are not latched. If written from a User Configuration, that same User Configuration is loaded during the startup sequence. If written from the serial port, the User Configuration to load during the startup sequence is determined by the originally latched CSEL pins, as on device power-up. If written from the serial port, the reset is triggered after the serial port write transaction completes. Self-cleared by the reset. |
| 9 | dig_sw_rst | RW | 0x0 | Digital Logic Software Reset.<br>Writing this bit to 1 resets all digital logic including the registers and re-starts the startup sequence at the latching of the static CSEL pins. The reset is triggered after the serial port write transaction completes. Self-cleared by the reset. |
| 8 | otp_sw_rst | RW | 0x0 | Configuration Loader Software Reset.<br>The configuration loader logic is held in reset while this bit is set to 1. This bit must not be set through the OTP/EEPROM configuration, otherwise the part will become unresponsive. |

Table 48. SW_RESET - Software Reset Register (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | oc_ssc_sw_rst | RW | 0x0 | Over-Clock and Spread-Spectrum Engine Software Reset.<br>The over-clocking engine and both spread-spectrum blocks are held in reset while this bit is set to 1. |
| 6 | tdc_apll_dig_sw_rst | RW | 0x0 | TDC APLL Digital Logic Reset.<br>The TDC APLL digital logic is held in reset while this bit is set to 1.<br>While re-programming the TDC APLL CSRs after device start-up, tdc_apll_dig_sw_rst must be set to 1. |
| 5 | dpll_sw_rst | RW | 0x0 | DPLL Software Reset.<br>The DPLL is held in reset while this bit is set to 1. |
| 4 | clkmon4_sw_rst | RW | 0x0 | CLKMON4 Software Reset.<br>The Clock Monitor 4 is held in reset while this bit is set to 1.<br>While re-programming the refin LOS monitor CSRs after device start-up, clkmon4_sw_rst must be set to 1. |
| 3 | clkmon3_sw_rst | RW | 0x0 | CLKMON3 Software Reset.<br>The Clock Monitor 3 is held in reset while this bit is set to 1.<br>While re-programming the clkin3 LOS or Frequency monitor CSRs after device start-up, clkmon3_sw_rst must be set to 1. |
| 2 | clkmon2_sw_rst | RW | 0x0 | CLKMON2 Software Reset.<br>The Clock Monitor 2 is held in reset while this bit is set to 1.<br>While re-programming the clkin2 LOS or Frequency monitor CSRs after device start-up, clkmon2_sw_rst must be set to 1. |
| 1 | clkmon1_sw_rst | RW | 0x0 | CLKMON1 Software Reset.<br>The Clock Monitor 1 is held in reset while this bit is set to 1.<br>While re-programming the clkin1 LOS or Frequency monitor CSRs after device start-up, clkmon1_sw_rst must be set to 1. |
| 0 | clkmon0_sw_rst | RW | 0x0 | CLKMON0 Software Reset.<br>The Clock Monitor 0 is held in reset while this bit is set to 1.<br>While re-programming the clkin0 LOS or Frequency monitor CSRs after device start-up, clkmon0_sw_rst must be set to 1. |

Table 49. CLOCK_GATE - Clock Gate Register

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:7 | reserved | RO | 0x0 | Reserved. |
| 6 | dpll_cg | RW | 0x0 | DPLL Clock Gate.<br>The DPLL system clock domain is clock gated while this bit is set to 1. |
| 5 | clkmon4_cg | RW | 0x0 | CLKMON4 Clock Gate.<br>The Clock Monitor 4 is clock gated while this bit is set to 1. |
| 4 | clkmon3_cg | RW | 0x0 | CLKMON3 Clock Gate.<br>The Clock Monitor 3 is clock gated while this bit is set to 1. |
| 3 | clkmon2_cg | RW | 0x0 | CLKMON2 Clock Gate.<br>The Clock Monitor 2 is clock gated while this bit is set to 1. |
| 2 | clkmon1_cg | RW | 0x0 | CLKMON1 Clock Gate.<br>The Clock Monitor 1 is clock gated while this bit is set to 1. |

**Table 49. CLOCK_GATE - Clock Gate Register (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 1 | clkmon0_cg | RW | 0x0 | CLKMON0 Clock Gate.<br>The Clock Monitor 0 is clock gated while this bit is set to 1. |
| 0 | otp_cg | RW | 0x0 | OTP Logic Clock Gate.<br>The OTP interface logic is clock gated while this bit is set to 1. The user must set this bit to 0 to access the OTP. This bit must not be set through the OTP config, otherwise the part will become unresponsive. |

**Table 50. MISC_CNFG - Miscellaneous Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:11 | Reserved | RW | 0x20 | Reserved. |
| 10 | losmon_clksel | RW | 0x0 | LOS Monitor Measuring Clock Select.<br>Selects the measuring clock for the LOS monitor blocks.<br>▪ 0x0 = quad system clock (nominally 240 MHz)<br>▪ 0x1 = TDC ring oscillator frequency divided by 2 (nominally 432MHz). |
| 9:8 | fanout_buf_mode1 | RW | 0x0 | CLKIN1 Fan-out Buffer Mode.<br>Configures the device to operate in fan-out buffer mode from CLKIN1. Independent from the CLKIN0 fanout_buf_mode setting.<br>▪ 0x0 = CLKIN1 Fan-out buffer mode disabled or Manual fan-out buffer mode, output banks select their clock sources according to output_bank_src. fanout_clkmode1 reads back as 0.<br>▪ 0x1 = Automatic fan-out buffer mode. In banks where bank_fanout_mode is set to 2 (not available on Bank 6), output bank clock source selection is based on CLKIN1 LOS and is latched at the rising edge of PERST1#. The selected mode can be read in fanout_clkmode1.<br>▪ 0x3 = Manual fan-out buffer mode, CLKIN1 fans out to output clocks in banks where bank_fanout_mode is set to 2. fanout_clkmode1 reads back as 1. |
| 7:6 | out_startup | RW | 0x0 | Output Disable on Startup until PLL locks.<br>Controls whether the clock output drivers are disabled until the APLL or DPLL locks during the startup sequence. When output_startup is not set to 0x2, individual output clocks may be configured to ignore APLL or DPLL lock status by setting their out_dis_group to 0x7.<br>▪ 0x0 = Clock output drivers are disabled until APLL lock asserts<br>▪ 0x1 = Clock output drivers are disabled until DPLL lock asserts<br>▪ 0x2 = Clock output drivers are not disabled by APLL or DPLL lock status<br>▪ 0x3 = Reserved |
| 5 | ssc_share | RW | 0x0 | Spectrum Spreading Phase Share.<br>When two SSC engines have the same modulation frequency, setting this bit to 1 will align the phase of FOD1 SSC to FOD0 SSC. For restrictions on enabling the SSCs when ssc_share is set to 1, see ssc_en. |

**Table 50. MISC_CNFG - Miscellaneous Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4:3 | fanout_buf_mode | RW | 0x0 | CLKIN0 Fan-out Buffer Mode.<br>Configures the device to operate in fan-out buffer mode from CLKIN0. Independent from the CLKIN1 fanout_buf_mode1 setting.<br>▪ 0x0 = CLKIN0 Fan-out buffer mode disabled or Manual fan-out buffer mode, output banks select their clock sources according to output_bank_src. fanout_clkmode reads back as 0.<br>▪ 0x1 = Automatic fan-out buffer mode. In banks where bank_fanout_mode is set to 1, output bank clock source selection is based on CLKIN0 LOS and is latched at the rising edge of PERST#. The selected mode can be read in fanout_clkmode.<br>▪ 0x3 = Manual fan-out buffer mode, CLKIN0 fans out to output clocks in banks where bank_fanout_mode is set to 1. fanout_clkmode reads back as 1. |
| 2:0 | reserved | RW | 0x5 | Reserved |

**Table 51. STARTUP_STS - Start-up Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:6 | reserved | RO | 0x0 | Reserved. |
| 5:4 | gpio2_latched | RO | 0x0 | GPIO2 Value Latched at Startup.<br>Value of GPIO2 latched at startup (when PWRGD/PWRDN# first becomes high, if pwrdn_sel selects a PWRGD/PWRDN# pin).<br>▪ 0x0 = Tri-level low<br>▪ 0x1 = Tri-level mid<br>▪ 0x2 = Unused<br>▪ 0x3 = Tri-level high |
| 3:2 | gpio1_latched | RO | 0x0 | GPIO1 Value Latched at Startup.<br>Value of GPIO1 latched at startup (when PWRGD/PWRDN# first becomes high, if pwrdn_sel selects a PWRGD/PWRDN# pin).<br>▪ 0x0 = Tri-level low<br>▪ 0x1 = Tri-level mid<br>▪ 0x2 = Unused<br>▪ 0x3 = Tri-level high |
| 1:0 | gpio0_latched | RO | 0x0 | GPIO0 Value Latched at Startup.<br>Value of GPIO0 latched at startup (when PWRGD/PWRDN# first becomes high, if pwrdn_sel selects a PWRGD/PWRDN# pin).<br>▪ 0x0 = Tri-level low<br>▪ 0x1 = Tri-level mid<br>▪ 0x2 = Unused<br>▪ 0x3 = Tri-level high |

**Table 52. DEVICE_STS - Device Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | fanout_clkmode1 | RO | 0x0 | CLKIN1 Fanout Clock Mode.<br>Set to 1 when CLKIN1 fanout buffer mode is enabled (fanout_buf_mode1 is not set to 0x0) and CLKIN1 is passed to the output clocks. |
| 14 | fanout_clkmode | RO | 0x0 | CLKIN0 Fanout Clock Mode.<br>Set to 1 when CLKIN0 fanout buffer mode is enabled (fanout_buf_mode is not set to 0x0) and CLKIN0 is passed to the output clocks. |
| 13:10 | reserved | RO | 0x0 | Reserved. |
| 9 | device_ready | RO | 0x0 | Device Ready.<br>Set to 1 when the configuration load (OTP and/or EEPROM) completes during the startup sequence. Cleared during a dynamic configuration load. |
| 8 | config_reg_ro | RO | 0x0 | Configuration Register Read-only Status.<br>When this bit is 1, writes to configuration registers are ignored. Writes to bits of type RW1C, reg_lock_key and certain RW and RW1S bits are not ignored. Use the reg_lock_key field to set/clear this bit. |
| 7 | otp_detect_se | RO | 0x0 | OTP Loader Detected Single-Ended Mode.<br>When high, indicates that the OTP loader detected that the OTP image is configured for single-ended mode. |
| 6 | eeprom_config_valid | RO | 0x0 | Valid EEPROM User Configuration Loaded.<br>Indicates that the user configuration in config_loaded was successfully loaded from EEPROM. Only valid when device_ready is 1. |
| 5 | otp_config_valid | RO | 0x0 | Valid OTP User Configuration Loaded.<br>Indicates that the user configuration in config_loaded was successfully loaded from OTP. Only valid when device_ready is 1. |
| 4:0 | config_loaded | RO | 0x0 | User Configuration Loaded.<br>Indicates the user configuration loaded from OTP/EEPROM on start-up or a dynamic configuration load. Note that on startup, the common configuration is always loaded prior to the user configuration. Only valid when device_ready is 1. |

## 11.2 SSI Registers

**Table 53. SPI_CNFG - SPI Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3 | spi_del_out | RW | 0x0 | SDO Delay.<br>▪ 0x0 = SDO is driven on opposite SCLK edge than the sampling edge<br>▪ 0x1 = SDO is delayed one half cycle of SCLK |
| 2 | reserved | RO | 0x0 | Reserved. |

**Table 53. SPI_CNFG - SPI Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 1 | spi_clk_sel | RW | 0x0 | SDI Sampling Edge Selection.<br>▪ 0x0 = SDI is sampled on rising SCLK edge<br>▪ 0x1 = SDI is sampled on falling SCLK edge |
| 0 | spi_3wire | RW | 0x0 | Select SPI 3 or 4-wire Mode.<br>▪ 0x0 = Normal 4-wire SPI. Data is received on the GPI/GPIO pin assigned as SDI, and transmitted on the GPIO pin assigned as SDO.<br>▪ 0x1 = 3-wire SPI. Data is received and transmitted on the GPIO pin assigned as SDIO. |

**Table 54. I2C_FLTR_CNFG - I2C Filter Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | i2c_spike_fltr | RW | 0x1 | I2C/SMBus Digital Spike Filter Duration.<br>Controls the duration of the digital spike filters on the SCL and SDA inputs, specified in number of system clock cycles (16.7ns). 0 disables filtering. |

**Table 55. I2C_TIMING_CNFG - I2C Timing Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | i2c_sda_high_hold | RW | 0x2 | I2C/SMBus Transmit One Bit Delay.<br>Delays transmission of 1 value by this number of 133ns periods (8 system clock cycles). |
| 3:0 | i2c_sda_low_hold | RW | 0x2 | I2C/SMBus Transmit Zero Bit Delay.<br>Delays transmission of 0 value by this number of 133ns periods (8 system clock cycles). |

**Table 56. I2C_ADDR_CNFG - I2C Address Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | reserved | RO | 0x0 | Reserved. |
| 6:0 | i2c_addr | RW | 0x09 | I2C Device Address.<br>Sets I2C or SMBus device address that the SSI will acknowledge and accept accesses on. |

**Table 57. BYTE_CNT - Byte Count**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:6 | reserved | RO | 0x0 | Reserved. |
| 5:0 | bc | RW | 0x08 | Byte Count.<br>Defines how many bytes are in an SMBus block transaction. This register is modified by the SMBus interface logic when a block transaction is processed. |

**Table 58. SMB_CTL - SMBus Control**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | reserved | RO | 0x0 | Reserved. |
| 6 | smb_d_to_en | RW | 0x1 | SMBus Data Timeout Enable.<br>Enables the SMBus data timeout check. When this bit is set to 1, the clock timeout must also be enabled (smb_to_cnt is not 0). When a data timeout is detected, the smb_d_to bit is set to 1. |
| 5 | smb_to_en | RW | 0x1 | Reserved. |
| 4:0 | smb_to_cnt | RW | 0x19 | SMBus Timeout Duration.<br>Sets the clock and data timeout duration in milliseconds. When this field is set to 0, the clock timeout is disabled and smb_d_to_en must also be set to 0 to disable the data timeout. When a clock timeout is detected, the smb_to bit is set to 1. |

**Table 59. SSI_GLOBAL_CNFG - SSI Global Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:5 | reserved | RO | 0x0 | Reserved. |
| 4:3 | sda_sdi_drv | RW | 0x0 | I2C/SMBus Drive Strength.<br>Selects the output driver slew rate of the SDA_nCS and SCL_SCLK pins when the serial slave interface is configured for I2C/SMBus mode (higher settings means higher drive strength). This setting does not affect the internal timing. As an I2C/SMBus slave, the external I2C/SMBus master must provide the appropriate SCL frequency and other timing requirements according to the selected speed.<br>▪ 0x0 = 1.8V Standard mode (100 kHz) or 2.5V/3.3V standard (100kHz) and Fast mode (400kHz)<br>▪ 0x1 = 1.8V Fast mode (400 kHz)<br>▪ 0x2 = Reserved<br>▪ 0x3 = 1.8V/2.5V/3.3V Fast mode plus (1MHz) |
| 2 | ssi_addr_size | RW | 0x0 | SSI Address Size.<br>Sets the number of bytes expected when providing a CSR address. Upper address bits are taken from the SSI's page register to create a full 32-bit CSR address.<br>▪ 0x0 = 1-byte address<br>▪ 0x1 = 2-byte address |
| 1:0 | ssi_enable | RW | 0x1 | SSI Mode.<br>▪ 0x0 = SSI is disabled<br>▪ 0x1 = SSI is in I2C mode<br>▪ 0x2 = SSI is in SPI mode<br>▪ 0x3 = SSI is in SMBus mode |

**Table 60. SSI_STS - Serial Port Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:2 | reserved | RO | 0x0 | Reserved. |
| 1 | smb_d_to | RW1C | 0x0 | SMBus Data Timeout.<br>Set when the SMBus interface detects a data timeout. |
| 0 | smb_to | RW1C | 0x0 | SMBus Clock Timeout.<br>Set when the SMBus interface detects a clock timeout. |

## 11.3 XO Register

**Table 61. XO_CNFG - Crystal Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:30 | reserved | RO | 0x0 | Reserved. |
| 29 | xo_ib_h_div_setb | RW | 0x1 | XO / Input Buffer High-Frequency Divider Set.<br>When cleared, the XO/IB high frequency divider is held in set mode (bit is active low). Unless this divider is bypassed (xo_ib_h_div = 0), clearing this bit will halt the refin_div clock. |
| 28:24 | xo_ib_h_div | RW | 0x0 | XO / Input Buffer High-Frequency Divide Ratio.<br>Divide by 2 to 31. Bypass if set to 0. 1 is reserved. |
| 23:22 | en_gain | RW | 0x1 | XO Gain Boosting Control.<br>Selects the number of gain boosting amplifiers enabled during startup.<br>▪ 0x0 = Gain boosting amplifiers are disabled<br>▪ 0x1 = One parallel amplifier is enabled<br>▪ 0x2 = Two parallel amplifiers are enabled<br>▪ 0x3 = All three parallel amplifiers are enabled |
| 21:16 | en_cap_x2 | RW | 0x28 | XO Capacitance at X2 Terminal.<br>Controls the internal tuning capacitance applied at the XOUT_REFINb terminal integrated crystal. The capacitance rises monotonically in steps of 0.42pF from 0pF to 26.8pF as the control setting increases from 0x00 to the maximum of 0x3F. This must be set to 0 when sel_ib_xo is set to 0. |
| 15 | reserved | RO | 0x0 | Reserved. |
| 14 | xo_buff_dis | RW | 0x0 | XO Buffer Disable.<br>Forces the XO buffer to the core logic to be disabled. This setting is intended for debug purposes only.<br>▪ 0x0 = XO buffer enable is controlled by the hardware. It is enabled during the startup sequence and remains enabled until the device is power cycled.<br>▪ 0x1 = XO buffer is disabled |
| 13:8 | en_cap_x1 | RW | 0x28 | XO Capacitance at X1 Terminal.<br>Controls the internal tuning capacitance applied at the XIN_REFIN terminal integrated crystal. The capacitance rises monotonically in steps of 0.42pF from 0pF to 26.8pF as the control setting increases from 0x00 to the maximum of 0x3F. This must be set to 0 when sel_ib_xo is set to 0. |
| 7 | reserved | RO | 0x0 | Reserved. |
| 6:4 | xo_res | RW | 0x4 | XO Resistor Configuration.<br>Series resistance array control for limiting driving power level. |
| 3 | xo_ib_en_dc_bias | RW | 0x0 | Input Buffer Internal DC Bias Enable.<br>When the input buffer clock input signal is AC-coupled external to the device, the internal DC bias voltage must be enabled.<br>▪ 0x0 = Internal DC bias is disabled (input signal is DC-coupled)<br>▪ 0x1 = Internal DC bias is enabled (input signal is AC-coupled) |
| 2 | sel_ib_xo | RW | 0x1 | XO / Input Buffer Select.<br>Selects the mode of the XO / Input Buffer.<br>▪ 0x0 = Input buffer<br>▪ 0x1 = XO |

**Table 61. XO_CNFG - Crystal Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 1 | xo_ib_cmos_sel | RW | 0x0 | Input Buffer CMOS / Differential Select.<br>Configures the input buffer for single-ended CMOS or differential input signal.<br>▪ 0x0 = Differential input is selected<br>▪ 0x1 = CMOS input is selected |
| 0 | xo_ib_p_n_diff_sel | RW | 0x1 | Input Buffer PMOS / NMOS Select.<br>Configures the input buffer according to the common mode voltage of the provided input signal.<br>▪ 0x0 = PMOS input pair is enabled (low common mode voltage)<br>▪ 0x1 = NMOS input pair is enabled (higher common mode voltage) |

## 11.4 CLKIN Register

**Table 62. CLKIN_CNFG - Clock Input Pad Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | h_div_setb | RW | 0x1 | Reference Clock High-Frequency Divider Set.<br>When cleared, the corresponding reference clock high-frequency divider is held in set mode (bit is active low). Unless this divider is bypassed (h_div = 0), clearing this bit will halt the clkin0_div/clkin2_div clock. |
| 14:10 | h_div | RW | 0x0 | Reference Clock High-Frequency Divide Ratio.<br>Divide by 2 to 31. Bypass if set to 0. 1 is reserved. |
| 9 | trim_term | RW | 0x0 | Reference Clock Input Pad Termination Trim.<br>Selects the HCSL and LVDS termination resistance to ground / across the inputs for modes to:<br>▪ 0x0 = 50Ω / 100Ω<br>▪ 0x1 = 42.5Ω / 85Ω |
| 8 | en_lvpecl | RW | 0x0 | Reference Clock Input Pad LVPECL Termination Enable.<br>Enables compatible termination when the reference clock input signal is LVPECL.<br>▪ 0x0 = LVPECL input termination is disabled<br>▪ 0x1 = LVPECL input termination is enabled |
| 7 | en_lvds | RW | 0x0 | Reference Clock Input Pad LVDS Termination Enable.<br>Enables compatible termination when the reference clock input signal is LVDS.<br>▪ 0x0 = LVDS input termination is disabled<br>▪ 0x1 = LVDS input termination is enabled |
| 6 | en_hcsl | RW | 0x0 | Reference Clock Input Pad HCSL Termination Enable.<br>Enables compatible termination when the reference clock input signal is HCSL.<br>▪ 0x0 = HCSL input termination is disabled<br>▪ 0x1 = HCSL input termination is enabled |
| 5 | en_cml | RW | 0x0 | Reference Clock Input Pad CML Termination Enable.<br>Enables compatible termination when the reference clock input signal is CML.<br>▪ 0x0 = CML input termination is disabled<br>▪ 0x1 = CML input termination is enabled |

**Table 62. CLKIN_CNFG - Clock Input Pad Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4 | en_dc_bias | RW | 0x0 | Reference Clock Input Pad Internal DC Bias Enable.<br>When the reference clock input signal is AC-coupled external to the device, the internal DC bias voltage must be enabled.<br>▪ 0x0 = Internal DC bias is disabled (input signal is DC-coupled)<br>▪ 0x1 = Internal DC bias is enabled (input signal is AC-coupled) |
| 3 | en_inbuff | RW | 0x0 | Reference Clock Input Pad Enable.<br>The reference clock input pad must be enabled to allow the clock to be used by the device, and should be left disabled if unused. To fully power down the input pad, en_inbuff, en_dc_bias, en_cml, en_hcsl, en_lvds, and en_lvpecl all must be set to 0.<br>▪ 0x0 = Input pad is disabled<br>▪ 0x1 = Input pad is enabled |
| 2:1 | cmos_sel | RW | 0x0 | Reference Clock Input Pad CMOS / Differential Select.<br>Configures the reference clock input pad for two single-ended CMOS or differential input signal.<br>▪ 0x0 = Differential input is selected<br>▪ 0x1 = Single CMOS input is selected on positive pin<br>▪ 0x2 = Single CMOS input is selected on negative pin<br>▪ 0x3 = Dual CMOS inputs are selected |
| 0 | p_n_diff_sel | RW | 0x1 | Reference Clock Input Pad PMOS / NMOS Select.<br>Configures the reference clock input pad according to the common mode voltage of the provided input signal.<br>▪ 0x0 = PMOS input pair is enabled (low common mode voltage)<br>▪ 0x1 = NMOS input pair is enabled (higher common mode voltage) |

## 11.5 REF Registers

**Table 63. PREDIV_CNFG - Reference Clock Input Pre-divider Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:26 | reserved | RO | 0x0 | Reserved. |
| 25:24 | ref_priority | RW | 0x0 | Reference Clock Priority.<br>Sets the clock's priority for DPLL reference switching. If multiple clocks are set to the same priority level, they are prioritized from the lowest numbered (clkin0) to the highest numbered (clkin3).<br>▪ 0x0 = First priority<br>▪ 0x1 = Second priority<br>▪ 0x2 = Third priority<br>▪ 0x3 = Fourth priority |
| 23 | reserved | RO | 0x0 | Reserved. |
| 22 | id_setb | RW | 0x1 | Input Divider Set.<br>When cleared, the corresponding input divider is held in set mode (bit is active low). Unless this divider is bypassed (id_byp_en = 1), clearing this bit will halt the corresponding reference clock. |
| 21 | reserved | RO | 0x0 | Reserved. |

**Table 63. PREDIV_CNFG - Reference Clock Input Pre-divider Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 20 | id_byp_en | RW | 0x1 | Reference Clock Pre-divider Bypass.<br>Allows the input divider to be bypassed and the reference clock input is passed directly to the DPLL. Bypass must be disabled if the reference clock frequency is greater than 33MHz.<br>▪ 0x0 = Divided reference clock is selected, divide ratio is id_ratio<br>▪ 0x1 = Reference clock is selected, effective divide ratio is 1 |
| 19:0 | id_ratio | RW | 0x0 | Reference Clock Pre-divider Ratio.<br>The reference clock frequency divided by this value must be no more than 33 MHz, and must be equal to the DPLL feedback clock frequency. The minimum divide value is 2. To divide by 1 (when the input reference clock frequency is no more than 33 MHz), bypass the divider by setting id_byp_en to 1.<br>To support hitless switching between reference clocks, they must have the same nominal frequency after the pre-divider. |

## 11.6 LOSMON Registers

Note that before reprogramming a Loss of Signal Monitor, the corresponding clkmon0_sw_rst, clkmon1_sw_rst, clkmon2_sw_rst, clkmon3_sw_rst, or clkmon4_sw_rst bit should be set. Once programming is done, it should then be cleared.

**Table 64. LOSMON_WINDOW - LOS Monitor Window Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:12 | reserved | RO | 0x0 | Reserved. |
| 11:8 | los_cnt_thresh | RW | 0x0 | Loss-of-Signal Counter Threshold.<br>While the Loss-of-Signal counter (los_cnt) exceeds this threshold, the los_lmt_evt bit is set. |
| 7 | reserved | RO | 0x0 | Reserved. |
| 6 | ref_disable | RW | 0x0 | Reference Clock Selection Disable.<br>Controls whether this reference clock can be selected as the DPLL reference clock. Not applicable for LOSMON4 (XO/IB monitor).<br>▪ 0x0 = Reference clock can be selected, subject to qualification by the Loss-of-Signal and Frequency monitors, and prioritization according to ref_priority<br>▪ 0x1 = Reference clock cannot be selected, ref_invalid set to 1 |
| 5 | los_fail_mask | RW | 0x0 | LOS Monitor Failure Mask.<br>Masks the LOS monitor status contribution to ref_invalid.<br>▪ 0x0 = los_sts contributes to ref_invalid<br>▪ 0x1 = los_sts does not contribute to ref_invalid |
| 4:0 | los_div_ratio | RW | 0x0 | LOS Monitor Divide Ratio.<br>This divide ratio must be set such that the monitored clock nominal frequency divided by los_div_ratio is less than 1/8 of the measuring clock frequency (see losmon_clksel) to achieve 25% accuracy. One period of the divided clock is the monitoring window duration. A value of 0 or 1 means divide by 1. |

Table 65. LOSMON_NOMINAL - LOS Monitor Nominal Number Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:0 | los_nom_num | RW | 0x0 | LOS Monitor Nominal Cycle Count.<br>Sets the expected number of measuring clock periods (see losmon_clksel) within one monitor window. Set to 0x0 to disable the LOS monitor. Disabling the monitor will cause the los_sts bit to be asserted, therefore the los_fail_mask bit should also be set when this field is written to 0x0. |

Table 66. LOSMON_THRESH - LOS Monitor Threshold Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | reserved | RO | 0x0 | Reserved. |
| 30:16 | los_acc_margin | RW | 0x0 | LOS Monitor Accept Threshold.<br>An accepted clock monitoring window occurs when the final monitor counter value is within los_nom_num ± los_acc_margin. |
| 15 | reserved | RO | 0x0 | Reserved. |
| 14:0 | los_rej_margin | RW | 0x0 | LOS Monitor Reject Threshold.<br>A rejected clock monitoring window occurs when the final monitor counter value is outside of los_nom_num ± los_rej_margin. |

Table 67. LOSMON_QUAL - LOS Monitor Qualify Counter Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | los_good_times | RW | 0x4 | LOS Monitor Qualification Count.<br>If this number of consecutive accepted clock LOS monitoring windows occur without a rejected window, then the clock is qualified and los_sts is set to 0. A value of 0 is reserved. |
| 3:0 | los_fail_times | RW | 0x4 | LOS Monitor Disqualification Count.<br>If this number of rejected clock LOS monitoring windows occur without qualifying the clock, then the clock is disqualified and los_sts is set to 1. A value of 0 is reserved. |

Table 68. LOSMON_STS - LOS Monitor Status

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:2 | reserved | RO | 0x0 | Reserved. |
| 1 | ref_invalid | RO | 0x1 | Reference Clock Invalid Status.<br>Indicates whether this reference clock is currently considered to be invalid. This occurs if the clock is disqualified by one or more of the Loss-of-Signal and Frequency monitors, or ref_disable is set to 1.<br>▪ 0x0 = Clock is valid<br>▪ 0x1 = Clock is invalid |
| 0 | los_sts | RO | 0x1 | Loss-of-Signal Status.<br>Current value of the LOS status from the clock monitor:<br>▪ 0x0 = Clock meets the monitoring criteria<br>▪ 0x1 = Loss-of-signal detected |

**Table 69. LOSMON_EVENT - LOS Monitor Event Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:2 | reserved | RO | 0x0 | Reserved. |
| 1 | los_lmt_evt | RW1C | 0x0 | Loss-of-Signal Counter Threshold Exceeded Status.<br>Set while the Loss-of-Signal counter (los_cnt) exceeds the threshold set in los_cnt_thresh. This bit cannot be cleared by software while the condition persists.<br>▪ 0x0 = Loss-of-signal counter has not exceeded the threshold since the last time the bit was cleared<br>▪ 0x1 = Loss-of-signal counter exceeded the threshold since the last time the bit was cleared |
| 0 | los_evt | RW1C | 0x1 | Loss-of-Signal Event Status.<br>Set while the clock monitor asserts LOS. This bit cannot be cleared by software while the LOS condition persists. This bit is set when the block comes out of reset and needs to be cleared after proper programming.<br>▪ 0x0 = Loss-of-signal not detected since the last time the bit was cleared<br>▪ 0x1 = Loss-of-signal detected since the last time the bit was cleared |

**Table 70. LOSMON_CNT - LOS Monitor Count**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | los_cnt | RW | 0x0 | Loss-of-Signal Failure Counter.<br>This counter increments each time the clock monitor asserts LOS and saturates at 0xF. It is cleared by writing 0x0 to it, and can be preset by writing the desired value. Preset can be used either as a debug tool or to cause a threshold alarm to happen sooner.<br>This register can only be written if the block is not clock gated (clkmon0_cg:clkmon4_cg) or held in reset (clkmon0_sw_rst:clkmon4_sw_rst). |

## 11.7 FREQMON Registers

Note that before reprogramming a Frequency Monitor, the corresponding clkmon0_sw_rst, clkmon1_sw_rst, clkmon2_sw_rst, or clkmon3_sw_rst bit should be set. Once programming is done, it should then be cleared.

**Table 71. FREQMON_WINDOW - Frequency Monitor Window Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | freq_fail_mask | RW | 0x0 | Frequency Monitor Failure Mask.<br>Masks the frequency monitor status contribution to ref_invalid.<br>▪ 0x0 = freq_sts contributes to ref_invalid<br>▪ 0x1 = freq_sts does not contribute to ref_invalid |
| 30:28 | reserved | RO | 0x0 | Reserved. |
| 27:0 | freq_div_ratio | RW | 0x0 | Frequency Monitor Divide Ratio.<br>This divide ratio must be set such that the monitored clock nominal frequency divided by freq_div_ratio is as close as possible to 2.5Hz, creating a 0.4s monitoring window. A value of 0 or 1 means divide by 1. |

**Table 72. FREQMON_NOMINAL - Frequency Monitor Nominal Number Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:27 | reserved | RO | 0x0 | Reserved. |
| 26:0 | freq_nom_num | RW | 0x0 | Frequency Monitor Nominal Cycle Count.<br>Sets the expected number of clock periods of the TDC ring oscillator frequency divided by 4 (nominally 216MHz) within one monitor window. Set to 0x0 to disable the frequency monitor. Disabling the monitor will cause the freq_sts bit to be asserted, therefore the freq_fail_mask bit should also be set when this field is written to 0x0. |

**Table 73. FREQMON_THRESH - Frequency Monitor Threshold Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 63:59 | reserved | RO | 0x0 | Reserved. |
| 58:32 | freq_acc_margin | RW | 0x0 | Frequency Monitor Accept Threshold.<br>An accepted clock monitoring window occurs when the final monitor counter value is within freq_nom_num ± freq_acc_margin. One accepted window qualifies the clock and freq_sts is set to 0. |
| 31:27 | reserved | RO | 0x0 | Reserved. |
| 26:0 | freq_rej_margin | RW | 0x0 | Frequency Monitor Reject Threshold.<br>A rejected clock monitoring window occurs when the final monitor counter value is outside of freq_nom_num ± freq_rej_margin. One rejected window disqualifies the clock and freq_sts is set to 1. |

**Table 74. FREQMON_STS - Frequency Monitor Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:1 | reserved | RO | 0x0 | Reserved. |
| 0 | freq_sts | RO | 0x1 | Frequency Monitor Status.<br>Current value of the qualification status from the frequency monitor:<br>▪ 0x0 = Clock meets the monitoring criteria, clock qualified<br>▪ 0x1 = Failure detected, clock disqualified |

**Table 75. FREQMON_EVENT - Frequency Monitor Event Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:1 | reserved | RO | 0x0 | Reserved. |
| 0 | freq_evt | RW1C | 0x1 | Frequency Monitor Event Status.<br>Set while the frequency monitor disqualifies the clock. This bit cannot be cleared by software while the disqualified condition persists. This bit is set when the block comes out of reset and needs to be cleared after proper programming.<br>▪ 0x0 = Frequency monitor has not disqualified the clock since the last time the bit was cleared<br>▪ 0x1 = Frequency monitor has disqualified the clock since the last time the bit was cleared |

**Table 76. FREQMON_OFFSET - Frequency Monitor Frequency Offset Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:28 | reserved | RO | 0x0 | Reserved. |
| 27:0 | freq_offset_sts | RO | 0x7FFFFFF | Frequency Count Status.<br>Cycle count measured by the frequency monitor, updated at the end of each monitoring window. The count saturates.to 0x7FF_FFFF when it reaches freq_nom_num*2, which typically indicates that the monitored clock is not toggling, or is less than half the expected nominal frequency. It may be converted to ppm as follows:<br>ppm offset = 1e6 * (freq_nom_num - freq_offset_sts)/freq_offset_sts |

## 11.8 TDC APLL Registers

Note that before reprogramming the TDC APLL registers, the tdc_apll_dig_sw_rst bit should be set. Once programming is done, it should then be cleared.

**Table 77. TDC_APLL_CNFG - TDC APLL Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | tdc_en | RW | 0x0 | TDC Enable.<br>Controls whether the TDC is enabled. Must be enabled n Jitter Attenuator mode, when enabling a reference clock LOS monitor when losmon_clksel is set to 1, and when enabling a reference clock frequency monitor. |
| 14 | reserved | RO | 0x0 | Reserved. |
| 13:9 | tdc_kp_coef | RW | 0xF | TDC APLL Proportional Filter Coefficient (Kp).<br>The filter output is (up/down) * $2^{Kp}$ + integrator. |
| 8:4 | tdc_ki_coef | RW | 0x4 | TDC APLL Integral Filter Coefficient (Ki).<br>The integrator is (up/down) * $2^{Ki}$ + integrator. |
| 3:2 | tdc_fast_lock | RW | 0x2 | TDC APLL Fast Lock Mode.<br>Controls the TDC APLL lock speed by increasing the effective Ki values while the input up/down is railed.<br>▪ 0x0 = Normal lock. Ki as programmed.<br>▪ 0x1 = Ki increased by 2<br>▪ 0x2 = Ki increased by 4<br>▪ 0x3 = Ki increased by 6 |
| 1:0 | tdc_dac_sdm_order | RW | 0x2 | TDC APLL DAC SDM (Sigma Delta Modulator) Order.<br>Selects the order of the SDM controlling the DAC for the TDC APLL.<br>▪ 0x0 = No accumulation<br>▪ 0x1 = 1st order<br>▪ 0x2 = 2nd order<br>▪ 0x3 = Reserved |

**Table 78. TDC_FB_DIV_FRAC - TDC APLL Feedback Divider Fraction Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:0 | tdc_fb_div_frac | RW | 0x2800 | TDC APLL Feedback Divider Fraction.<br>Fraction of the TDC APLL feedback divider. The fraction is calculated as follows: tdc_fb_div_frac/$2^{16}$. |

**Table 79. TDC_FB_DIV_INT - TDC APLL Feedback Divider Integer Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:0 | tdc_fb_div_int | RW | 0x24 | TDC APLL Feedback Divider Integer.<br>Integer portion of the TDC APLL feedback divider. |

**Table 80. TDC_FB_SDM_CNFG - TDC APLL Feedback SDM Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | tdc_fb_sdm_en | RW | 0x1 | TDC APLL Feedback SDM Enable.<br>Enables the SDM controlling the TDC APLL feedback divider.<br>▪ 0x0 = SDM disabled, constant integer division by tdc_fb_div_int<br>▪ 0x1 = SDM enabled, MMD mode |
| 6:2 | reserved | RO | 0x0 | Reserved. |
| 1:0 | tdc_fb_sdm_order | RW | 0x1 | TDC APLL Feedback SDM Order.<br>Selects the order of the SDM controlling the feedback divider for the TDC APLL.<br>▪ 0x0 = Integer<br>▪ 0x1 = 1st order<br>▪ 0x2 = 2nd order<br>▪ 0x3 = 3rd order |

**Table 81. TDC_REF_DIV_CNFG - TDC APLL Reference Divider Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:3 | reserved | RO | 0x0 | Reserved. |
| 2:0 | tdc_ref_div_cnfg | RW | 0x1 | TDC Reference Divider Control.<br>Controls the divide ratio of the TDC reference (either XO input or reference clock input, selected by apll_ref_sel). This field should be programmed such that the reference to the TDC APLL is between 10MHz and 30MHz.<br>▪ 0x0 = Bypass divider<br>▪ 0x1 = Divide by 2<br>▪ 0x2 = Divide by 4<br>▪ 0x3 = Divide by 8<br>▪ 0x4 = Divide by 16 |

**Table 82. TDC_FILTER_STS - TDC APLL Filter Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:0 | tdc_filter_status | RO | 0x0 | TDC Loop Filter Status.<br>Provides the TDC loop filter output value. It is a signed 16-bit number. |

## 11.9 APLL Registers

**Table 83. APLL_FB_DIV_FRAC - APLL Feedback Divider Fraction Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:27 | reserved | RO | 0x0 | Reserved. |
| 26:0 | apll_fb_div_frac | RW | 0x0 | APLL Feedback Divider Fraction.<br>APLL feedback divider numerator value. The denominator is a fixed value of 2^27. |

**Table 84. APLL_FB_DIV_INT - APLL Feedback Divider Integer Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:10 | reserved | RO | 0x0 | Reserved. |
| 9:0 | apll_fb_div_int | RW | 0x69 | APLL Feedback Divider Integer.<br>APLL feedback divider integer value. |

**Table 85. APLL_FB_SDM_CNFG - APLL Feedback SDM Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:6 | reserved | RO | 0x0 | Reserved. |
| 5 | apll_fb_dither_en | RW | 0x0 | APLL Feedback SDM Dither Enable.<br>Dither enable for the SDM controlling the APLL feedback divider. After device startup, should only be changed while filter_update_dis is set to 1.<br>▪ 0x0 = Dither disabled<br>▪ 0x1 = Dither enabled |
| 4 | apll_fb_dither_ns | RW | 0x0 | APLL Feedback SDM Dither Noise Shaping.<br>Dither noise shaping enable for the SDM controlling the APLL feedback divider. After device startup, should only be changed while filter_update_dis is set to 1.<br>▪ 0x0 = Dither not shaped<br>▪ 0x1 = Dither shaped |
| 3:2 | apll_fb_dither_gain | RW | 0x0 | APLL Feedback SDM Dither Gain.<br>Gain control for the SDM controlling the APLL feedback divider. After device startup, should only be changed while filter_update_dis is set to 1.<br>▪ 0x0 = LSB<br>▪ 0x1 = 2*LSB<br>▪ 0x2 = 4*LSB<br>▪ 0x3 = 8*LSB |
| 1:0 | apll_fb_sdm_order | RW | 0x3 | APLL Feedback SDM Order.<br>Selects the order of the SDM controlling the feedback divider for the APLL. When the feedback divide ratio is fractional, or to allow xtal_trim or write_freq to operate correctly, the order must be set greater than 0. After device startup, should only be changed while filter_update_dis is set to 1.<br>▪ 0x0 = Integer<br>▪ 0x1 = 1st order<br>▪ 0x2 = 2nd order<br>▪ 0x3 = 3rd order |

**Table 86. APLL_CNFG - APLL Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:2 | reserved | RO | 0x0 | Reserved. |
| 1 | doubler_sel | RW | 0x0 | Frequency Doubler Select.<br>Selects the frequency doubler to use when en_doubler is set to 1.<br>▪ 0x0 = PLL frequency doubler<br>▪ 0x1 = PFD frequency doubler |
| 0 | en_doubler | RW | 0x1 | Frequency Doubler Enable.<br>Enables the frequency doubler selected by the doubler_sel bit.<br>▪ 0x0 = Disable<br>▪ 0x1 = Enable |

**Table 87. CP_CNFG - APLL Charge Pump Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:14 | reserved | RO | 0x0 | Reserved. |
| 13 | cp_offset_en | RW | 0x0 | Charge Pump Offset Current Enable.<br>Enables the charge pump offset current. The magnitude of the current is selected by cnf_cp_offset.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 12 | cp_offset_boost | RW | 0x0 | Charge Pump Offset Current Boost.<br>Increases the magnitude of the offset current.<br>▪ 0x0 = Charge pump offset range from 0uA to 145uA<br>▪ 0x1 = Charge pump offset range from 0uA to 236uA |
| 11:8 | cnf_cp_offset | RW | 0x3 | Charge Pump Offset Current Setting.<br>Controls the charge pump offset current when enabled by cp_offset_en. When cp_offset_boost is set to 0 / 1, the charge pump current is:<br>▪ 0x0 = 0uA / 0uA<br>▪ 0x1 = 10.3uA / 99.7uA<br>▪ 0x2 = 20.5uA / 118.7uA<br>▪ 0x3 = 30.6uA / 128.1uA<br>▪ 0x4 = 40.7uA / 137.5uA<br>▪ 0x5 = 50.5uA / 146.7uA<br>▪ 0x6 = 60.3uA / 155.9uA<br>▪ 0x7 = 70.0uA / 165.1uA<br>▪ 0x8 = 79.7uA / 174.2uA<br>▪ 0x9 = 89.2uA / 183.2uA<br>▪ 0xA = 98.7uA / 192.2uA<br>▪ 0xB = 108.1uA / 201.2uA<br>▪ 0xC = 117.5uA / 210.0uA<br>▪ 0xD = 126.7uA / 218.9uA<br>▪ 0xE = 135.9uA / 227.7uA<br>▪ 0xF = 145.0uA / 236.0uA |
| 7 | reserved | RO | 0x0 | Reserved. |

**Table 87. CP_CNFG - APLL Charge Pump Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 6:4 | cnf_cp_up | RW | 0x3 | Charge Pump Up Current Setting.<br>▪ 0x0 = 250uA<br>▪ 0x1 = 500uA<br>▪ 0x2 = 750uA<br>▪ 0x3 = 1mA<br>▪ 0x4 = 1.25mA<br>▪ 0x5 = 1.5mA<br>▪ 0x6 = 1.75mA<br>▪ 0x7 = 2mA |
| 3 | reserved | RO | 0x0 | Reserved. |
| 2:0 | cnf_cp_dn | RW | 0x3 | Charge Pump Down Current Setting.<br>▪ 0x0 = 250uA<br>▪ 0x1 = 500uA<br>▪ 0x2 = 750uA<br>▪ 0x3 = 1mA<br>▪ 0x4 = 1.25mA<br>▪ 0x5 = 1.5mA<br>▪ 0x6 = 1.75mA<br>▪ 0x7 = 2mA |

**Table 88. LPF_CNFG - APLL Loop Filter Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | apll_vco_filter_byp | RW | 0x0 | VCO Current Source Filter Bypass.<br>▪ 0x0 = Filter active<br>▪ 0x1 = Filter bypassed |
| 6:4 | cnf_lpf_cp | RW | 0x7 | Loop Filter Pole Capacitor Setting.<br>▪ 0x0 = 11pF<br>▪ 0x1 = 14.7pF<br>▪ 0x2 = 18.4pF<br>▪ 0x3 = 22.1pF<br>▪ 0x4 = 25.8pF<br>▪ 0x5 = 29.5pF<br>▪ 0x6 = 33.2pF<br>▪ 0x7 = 36.9pF |

Table 88. LPF_CNFG - APLL Loop Filter Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3:0 | cnf_lpf_res | RW | 0x4 | Loop Filter Resistor Setting.<br>▪ 0x0 = 0Ω<br>▪ 0x1 = 400Ω<br>▪ 0x2 = 800Ω<br>▪ 0x3 = 1.2kΩ<br>▪ 0x4 = 1.6kΩ<br>▪ 0x5 = 2kΩ<br>▪ 0x6 = 2.4kΩ<br>▪ 0x7 = 2.8kΩ<br>▪ 0x8 = 3.2kΩ<br>▪ 0x9 = 3.6kΩ<br>▪ 0xA = 4kΩ<br>▪ 0xB = 4.4kΩ<br>▪ 0xC = 4.8kΩ<br>▪ 0xD = 5.2kΩ<br>▪ 0xE = 5.6kΩ<br>▪ 0xF = 6kΩ |

Table 89. LPF_3RD_CNFG - APLL Loop Filter 3rd Pole Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | byp_p3 | RW | 0x0 | Bypass 3rd Pole.<br>This bit can only be set to 1 when operating with an integer feedback divider.<br>▪ 0x0 = 3rd pole active<br>▪ 0x1 = 3rd pole bypassed |
| 6:4 | cnf_lpf_R3 | RW | 0x4 | Loop Filter 3rd Pole Resistor Setting.<br>▪ 0x0 = 0Ω<br>▪ 0x1 = 800Ω<br>▪ 0x2 = 1.6kΩ<br>▪ 0x3 = 2.4kΩ<br>▪ 0x4 = 3.2kΩ<br>▪ 0x5 = 4kΩ<br>▪ 0x6 = 4.8kΩ<br>▪ 0x7 = 5.6kΩ |
| 3 | reserved | RO | 0x0 | Reserved. |
| 2:0 | cnf_lpf_C3 | RW | 0x4 | Loop Filter 3rd Pole Capacitor Setting.<br>▪ 0x0 = 0pF<br>▪ 0x1 = 1pF<br>▪ 0x2 = 2pF<br>▪ 0x3 = 3pF<br>▪ 0x4 = 4pF<br>▪ 0x5 = 5pF<br>▪ 0x6 = 6pF<br>▪ 0x7 = 7pF |

**Table 90. APLL_REF_FB_CNFG - APLL Ref and Fb Clock Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | apll_lol_cnt_thresh | RW | 0x0 | APLL Loss-of-Lock Counter Threshold.<br>While the APLL Loss-of-Lock counter (apll_lol_cnt) exceeds this threshold, the apll_lol_lmt bit is set. |
| 3 | apll_fb_div_dis | RW | 0x0 | APLL Feedback Divider Disable.<br>Disables the APLL feedback divider. May be set to 1 to reduce power consumption when an external feedback clock is selected with apll_fb_sel.<br>▪ 0x0 = APLL feedback divider enabled<br>▪ 0x1 = APLL feedback divider disabled |
| 2 | apll_fb_sel | RW | 0x0 | APLL Feedback Clock Selection.<br>Selects the APLL feedback clock source. When an external source is selected, the APLL feedback divider may be disabled by setting apll_fb_div_dis to 1. If the external clock is sourced from an output clock from the device, and out_startup is not set to 0x2, then that output clock's out_dis_group must be set to 0x7.<br>▪ 0x0 = Internal from APLL feedback MMD<br>▪ 0x1 = External from CLKIN0 – ZDB mode |
| 1:0 | apll_ref_sel | RW | 0x0 | APLL Reference Clock Selection.<br>▪ 0x0 = XIN_REFIN (refin_div)<br>▪ 0x1 = CLKIN0 (clkin0_div)<br>▪ 0x2 = CLKIN1 (clkin2_div)<br>▪ 0x3 = VSS |

**Table 91. BANK_MUX_CLK_EN - Bank Mux Clock Enable**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | en_refin_omux | RW | 0x1 | XO/IB Clock to Bank 5 Output Mux Enable.<br>Enables fanout of the XO/IB clock to Bank 5 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 30 | en_clkin2_bnk6 | RW | 0x1 | Reference Clock 2 to Bank 6 Output Mux Enable.<br>Enables fanout of reference clock 2 (CLKIN1) to Bank 6 output mux. When bank 6 is in automatic fanout buffer mode and bank_fanout_mode is set to 2, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 29 | en_clkin2_bnk5 | RW | 0x1 | Reference Clock 2 to Bank 5 Output Mux Enable.<br>Enables fanout of reference clock 2 (CLKIN1) to Bank 5 output mux. When bank 5 is in automatic fanout buffer mode and bank_fanout_mode is set to 2, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |

Table 91. BANK_MUX_CLK_EN - Bank Mux Clock Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 28 | en_clkin2_bnk4 | RW | 0x1 | Reference Clock 2 to Bank 4 Output Mux Enable.<br>Enables fanout of reference clock 2 (CLKIN1) to Bank 4 output mux. When bank 4 is in automatic fanout buffer mode and bank_fanout_mode is set to 2, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 27 | en_clkin0_bnk6 | RW | 0x1 | Reference Clock 0 to Bank 6 Output Mux Enable.<br>Enables fanout of reference clock 0 (CLKIN0) to Bank 6 output mux. When bank 6 is in automatic fanout buffer mode and bank_fanout_mode is set to 1, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 26 | en_clkin0_bnk5 | RW | 0x1 | Reference Clock 0 to Bank 5 Output Mux Enable.<br>Enables fanout of reference clock 0 (CLKIN0) to Bank 5 output mux. When bank 5 is in automatic fanout buffer mode and bank_fanout_mode is set to 1, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 25 | en_clkin0_bnk4 | RW | 0x1 | Reference Clock 0 to Bank 4 Output Mux Enable.<br>Enables fanout of reference clock 0 (CLKIN0) to Bank 4 output mux. When bank 4 is in automatic fanout buffer mode and bank_fanout_mode is set to 1, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 24 | en_fod2_bnk6 | RW | 0x1 | FOD 2 to Bank 6 Output Mux Enable.<br>Enables fanout of FOD 2 output clock to Bank 6 output mux. When bank 6 is in automatic fanout buffer mode and output_bank_src is set to 6, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 23 | en_fod2_bnk5 | RW | 0x1 | FOD 2 to Bank 5 Output Mux Enable.<br>Enables fanout of FOD 2 output clock to Bank 5 output mux. When bank 4 is in automatic fanout buffer mode and output_bank_src is set to 6, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 22 | en_fod2_bnk4 | RW | 0x1 | FOD 2 to Bank 4 Output Mux Enable.<br>Enables fanout of FOD 2 output clock to Bank 4 output mux. When bank 4 is in automatic fanout buffer mode and output_bank_src is set to 6, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |

Table 91. BANK_MUX_CLK_EN - Bank Mux Clock Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 21 | en_fod2_bnk3 | RW | 0x1 | FOD 2 to Bank 3 Output Mux Enable.<br>Enables fanout of FOD 2 output clock to Bank 3 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 20 | en_fod1_bnk6 | RW | 0x1 | FOD 1 to Bank 6 Output Mux Enable.<br>Enables fanout of FOD 1 output clock to Bank 6 output mux. When bank 6 is in automatic fanout buffer mode and output_bank_src is set to 5, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 19 | en_fod1_bnk5 | RW | 0x1 | FOD 1 to Bank 5 Output Mux Enable.<br>Enables fanout of FOD 1 output clock to Bank 5 output mux. When bank 5 is in automatic fanout buffer mode and output_bank_src is set to 5, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 18 | en_fod1_bnk4 | RW | 0x1 | FOD 1 to Bank 4 Output Mux Enable.<br>Enables fanout of FOD 1 output clock to Bank 4 output mux. When bank 4 is in automatic fanout buffer mode and output_bank_src is set to 5, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 17 | en_fod1_bnk3 | RW | 0x1 | FOD 1 to Bank 3 Output Mux Enable.<br>Enables fanout of FOD 1 output clock to Bank 3 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 16 | en_fod1_bnk2 | RW | 0x1 | FOD 1 to Bank 2 Output Mux Enable.<br>Enables fanout of FOD 1 output clock to Bank 2 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 15 | en_fod1_bnk1 | RW | 0x1 | FOD 1 to Bank 1 Output Mux Enable.<br>Enables fanout of FOD 1 output clock to Bank 1 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 14 | en_fod1_bnk0 | RW | 0x1 | FOD 1 to Bank 0 Output Mux Enable.<br>Enables fanout of FOD 1 output clock to Bank 0 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 13 | en_fod0_bnk3 | RW | 0x1 | FOD 0 to Bank 3 Output Mux Enable.<br>Enables fanout of FOD 0 output clock to Bank 3 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |

Table 91. BANK_MUX_CLK_EN - Bank Mux Clock Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 12 | en_fod0_bnk2 | RW | 0x1 | FOD 0 to Bank 2 Output Mux Enable.<br>Enables fanout of FOD 0 output clock to Bank 2 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 11 | en_fod0_bnk1 | RW | 0x1 | FOD 0 to Bank 1 Output Mux Enable.<br>Enables fanout of FOD 0 output clock to Bank 1 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 10 | en_fod0_bnk0 | RW | 0x1 | FOD 0 to Bank 0 Output Mux Enable.<br>Enables fanout of FOD 0 output clock to Bank 0 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 9 | en_iod3_bnk6 | RW | 0x1 | IOD 3 to Bank 6 Output Mux Enable.<br>Enables fanout of IOD 3 output clock to Bank 6 output mux. When bank 6 is in automatic fanout buffer mode and output_bank_src is set to 3, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 8 | en_iod3_bnk5 | RW | 0x1 | IOD 3 to Bank 5 Output Mux Enable.<br>Enables fanout of IOD 3 output clock to Bank 5 output mux. When bank 5 is in automatic fanout buffer mode and output_bank_src is set to 3, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 7 | en_iod2_bnk6 | RW | 0x1 | IOD 2 to Bank 6 Output Mux Enable.<br>Enables fanout of IOD 2 output clock to Bank 6 output mux. When bank 6 is in automatic fanout buffer mode and output_bank_src is set to 2, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 6 | en_iod2_bnk5 | RW | 0x1 | IOD 2 to Bank 5 Output Mux Enable.<br>Enables fanout of IOD 2 output clock to Bank 5 output mux. When bank 5 is in automatic fanout buffer mode and output_bank_src is set to 2, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 5 | en_iod2_bnk4 | RW | 0x1 | IOD 2 to Bank 4 Output Mux Enable.<br>Enables fanout of IOD 2 output clock to Bank 4 output mux. When bank 4 is in automatic fanout buffer mode and output_bank_src is set to 2, this bit is automatically controlled according to the output mux selection and cannot be written.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |

**Table 91. BANK_MUX_CLK_EN - Bank Mux Clock Enable (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4 | en_iod1_bnk2 | RW | 0x1 | IOD 1 to Bank 2 Output Mux Enable.<br>Enables fanout of IOD 1 output clock to Bank 2 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 3 | en_iod1_bnk1 | RW | 0x1 | IOD 1 to Bank 1 Output Mux Enable.<br>Enables fanout of IOD 1 output clock to Bank 1 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 2 | en_iod1_bnk0 | RW | 0x1 | IOD 1 to Bank 0 Output Mux Enable.<br>Enables fanout of IOD 1 output clock to Bank 0 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 1 | en_iod0_bnk1 | RW | 0x1 | IOD 0 to Bank 1 Output Mux Enable.<br>Enables fanout of IOD 0 output clock to Bank 1 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 0 | en_iod0_bnk0 | RW | 0x1 | IOD 0 to Bank 0 Output Mux Enable.<br>Enables fanout of IOD 0 output clock to Bank 0 output mux.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |

**Table 92. APLL_STS - APLL Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:1 | reserved | RO | 0x0 | Reserved. |
| 0 | apll_lock_sts | RO | 0x0 | APLL Lock Status.<br>Set to 1 when the APLL is locked to its reference.<br>▪ 0x0 = Unlocked<br>▪ 0x1 = Locked |

**Table 93. APLL_EVENT - APLL Event Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3 | apll_rail_high_evt | RW1C | 0x0 | APLL Rail High Event.<br>Set to 1 when the APLL lock detects a rail high status. Once asserted, this bit will remain asserted until cleared by a write of '1' to this bit position. |
| 2 | apll_rail_low_evt | RW1C | 0x0 | APLL Rail Low Event.<br>Set to 1 when the APLL lock detects a rail low status. Once asserted, this bit will remain asserted until cleared by a write of '1' to this bit position. |

**Table 93. APLL_EVENT - APLL Event Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 1 | apll_lol_lmt | RW1C | 0x0 | APLL Loss-of-Lock Counter Threshold Exceeded Status.<br>Set while the APLL Loss-of-Lock counter (apll_lol_cnt) exceeds the threshold set in apll_lol_cnt_thresh. This bit cannot be cleared by software while the condition persists.<br>▪ 0x0 = Loss-of-lock counter has not exceeded the threshold since the last time the bit was cleared<br>▪ 0x1 = Loss-of-lock counter exceeded the threshold since the last time the bit was cleared |
| 0 | apll_lol | RW1C | 0x0 | APLL Loss-of-lock Event.<br>Set to 1 when the APLL lock status transitions from locked to unlocked. |

**Table 94. APLL_LOL_CNT - APLL Loss-of-Lock Counter**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | apll_lol_cnt | RW | 0x0 | APLL Loss-of-Lock Counter.<br>This counter increments each time the APLL lock status de-asserts, and saturates at 0xF. It is cleared by writing it to 0x0, and may be preset by writing the desired value. |

**Table 95. ANA_SPARE_CNFG - Analog Spare Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | regulator_dis | RW | 0x0 | Regulator Disable.<br>This field can disable the following regulators: LD_XO_IB, LDO_IB1, LDO_IB2, LDO_DIG, LDO_CP, LDO_VCO, DIG_LDO_DIV, DIG_LDO_FOD.<br>▪ 0x0 = Regulators listed above are enabled.<br>▪ 0x1 = Regulators listed above are disabled. |
| 14 | ana_spare | RW | 0x0 | Reserved |
| 13 | sense_mode_sel | RW | 0x0 | One-shot VDDO Sensing.<br>Provides one-shot VDDO sensing for system clock gating function controlled by sysclk_gate_start and sysclk_gate_bypass. Should be left at 0x0. |
| 12 | sysclk_gate_bypass | RW | 0x0 | System Clock Gating Bypass.<br>Setting this bit to 1 in the last byte of the COMMON OTP will bypass the above function in sysclk_gate_start or sysclk_gate_start may be set to 1. |
| 11 | sysclk_gate_start | RW | 0x0 | System Clock Start.<br>Setting this bit to 1 in the last byte of the COMMON OTP config will allow the part to wait to load the USER OTP config until all VDDO that are not programmed as float_vddo* in DEVICE_CNFG are present. |

**Table 95. ANA_SPARE_CNFG - Analog Spare Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 10 | BNK6_pwrsel | RW | 0x0 | Bank6 Regulator Select.<br>This bit controls whether to enable or bypass the 1.8V regulator in the output driver.<br>▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator.<br>▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 9 | BNK5_pwrsel | RW | 0x0 | Bank5 Regulator Select.<br>This bit controls whether to enable or bypass the 1.8V regulator in the output driver.<br>▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator.<br>▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 8 | BNK4_pwrsel | RW | 0x0 | Bank4 Regulator Select.<br>This bit controls whether to enable or bypass the 1.8V regulator in the output driver.<br>▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator.<br>▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 7 | BNK3_pwrsel | RW | 0x0 | Bank3 Regulator Select.<br>This bit controls whether to enable or bypass the 1.8V regulator in the output driver.<br>▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator.<br>▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 6 | BNK2_pwrsel | RW | 0x0 | Bank2 Regulator Select.<br>This bit controls whether to enable or bypass the 1.8V regulator in the output driver.<br>▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator.<br>▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 5 | BNK1_pwrsel | RW | 0x0 | Bank1 Regulator Select.<br>This bit controls whether to enable or bypass the 1.8V regulator in the output driver.<br>▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator.<br>▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 4 | BNK0_pwrsel | RW | 0x0 | Bank0 Regulator Select.<br>This bit controls whether to enable or bypass the 1.8V regulator in the output driver.<br>▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator.<br>▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 3:0 | bg_trim | RW | 0x0 | Bandgap Trim. PTAT current trimming for bandgap Vbg voltage. |

**Table 96. ANA_SPARE_STS - Analog Spare Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:0 | ana_spare_sts | RO | 0x00 | Reserved |

## 11.10 DPLL Registers

Table 97. DPLL_REF_FB_CNFG - DPLL Ref and Fb Clock Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:6 | dpll_ref_sel_mode | RW | 0x0 | DPLL Reference Clock Selection Mode<br>▪ 0x0 = Manual mode, reference selection is based on the setting of dpll_ref_sel<br>▪ 0x1 = Manual mode, reference selection is based on GPI/GPIO pins<br>▪ 0x2 = Auto mode<br>▪ 0x3 = Auto mode |
| 5:4 | dpll_ref_sel | RW | 0x0 | DPLL Manual Reference Clock Selection<br>▪ 0x0 = CLKIN0 single-ended or CLKIN0/CLKIN0b differential (clkin0)<br>▪ 0x1 = CLKIN0b single-ended (clkin1)<br>▪ 0x2 = CLKIN1 single-ended or CLKIN1/CLKIN1b differential (clkin2)<br>▪ 0x3 = CLKIN1b single-ended (clkin3) |
| 3:2 | dpll_fb_sel | RW | 0x1 | DPLL Feedback Clock Selection<br>Selects the DPLL feedback clock source. When an external source is selected, the DPLL feedback divider may be disabled by setting dpll_fb_div_dis to 1. If the external clock is sourced from an output clock from the device, and out_startup is not set to 0x2, then that output clock's out_dis_group must be set to 0x7.<br>0x0 = External from CLKIN1 single-ended or CLKIN1/CLKIN1b differential (clkin2) --- Zero Delay Buffer (ZDB) mode<br>0x1 = Internal from DPLL feedback MMD |
| 1 | dpll_revertive_en | RW | 0x0 | DPLL Revertive Reference Switch<br>▪ 0x0 = Non-revertive<br>▪ 0x1 = Revertive |
| 0 | dpll_hitless_en | RW | 0x0 | DPLL Hitless Reference Switch<br>▪ 0x0 = Hitless disabled<br>▪ 0x1 = Hitless enabled |

Table 98. DPLL_MODE - DPLL Mode Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | bw_damp_sw | RW | 0x1 | Automatic Bandwidth/Damping Switching.<br>Enables the DPLL to switch to the Locking Loop Filter bandwidth and damping settings when the DPLL is in the Acquire state while locking. Refer to dpll_lock_timer.<br>▪ 0x0 = Always use Normal Operation settings.<br>▪ 0x1 = Use Locking settings when the DPLL is in the Acquire state. |
| 6 | los_to_freerun | RW | 0x0 | Reference Loss-of-Signal to Freerun.<br>Controls whether the DPLL enters Freerun or Holdover mode when the current reference clock is invalid.<br>▪ 0x0 = Holdover.<br>▪ 0x1 = Freerun. |
| 5:4 | dpll_sync_group | RW | 0x0 | DPLL Feedback Divider Sync Group.<br>Sets the sync group that the DPLL feedback divider belongs to.<br>▪ 0x0 = Group0<br>▪ 0x1 = Group1<br>▪ 0x2 = None<br>▪ 0x3 = None |

**Table 98. DPLL_MODE - DPLL Mode Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3 | filter_update_dis | RW | 0x0 | DPLL Filter Update Disable.<br>This bit must be set to 1 before reconfiguring any of the APLL feedback SDM register settings after startup, and then must be cleared after the reconfiguration finishes.These settings are: apll_fb_dither_en, apll_fb_dither_ns, apll_fb_dither_gain, and apll_fb_sdm_order. |
| 2 | dpll_en | RW | 0x0 | DPLL Enable.<br>Controls whether the DPLL is enabled.<br>▪ 0x0 = Synthesizer/DCO mode.<br>▪ 0x1 = Jitter Attenuator mode. DPLL is enabled. |
| 1:0 | dpll_mode | RW | 0x1 | DPLL Mode Selection.<br>Selects DPLL mode:<br>▪ 0x0 = Forces DPLL into Freerun state.<br>▪ 0x1 = Places the DPLL in Normal (automatic) mode. This is the normal setting for Jitter Attenuator mode.<br>▪ 0x2 = Forces DPLL into Holdover state.<br>▪ 0x3 = Places DPLL in Write Frequency mode. This is the normal setting in DCO mode. |

**Table 99. DPLL_DECIMATOR - DPLL Decimator Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | reserved | RO | 0x0 | Reserved. |
| 6:4 | dec_hitless_bw_shift | RW | 0x3 | Hitless Switch Decimator Bandwidth.<br>Shift to set the decimator bandwidth during a hitless reference switch or holdover-normal switch for measuring the phase offset. If *dpll_hitless_en* is set to zero, this field is ignored. After device startup, should only be changed while tdc_en is set to 0. |
| 3:0 | dec_bw_shift | RW | 0x6 | Decimator Bandwidth.<br>Shift to set the decimator bandwidth. 0 puts the decimator in feed-through (infinite bandwidth). After device startup, should only be changed while tdc_en is set to 0. |

**Table 100. DPLL_XTAL_OFFSET - DPLL Crystal Trim Offset Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:0 | xtal_trim | RW | 0x0 | Crystal Trim Offset.<br>Crystal fractional frequency offset compensation. This is an 8-bit 2's complement value. Resolution = $2^{-20} \approx 1$ ppm, Range = $\pm 2^{-13} \approx \pm 122$ ppm.<br>apll_fb_sdm_order must be set to a value greater than 0 for xtal_trim to operate correctly. |

**Table 101. HOLDOVER_CNFG - Holdover Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:11 | holdover_bw_shift | RW | 0x7 | Holdover Filter Bandwidth Shift.<br>Coarse control of the holdover bandwidth. A value of zero disables the holdover filter (infinite bandwidth). The valid range is 0 to 20. Values larger than 20 are limited internally to 20. |
| 10:8 | holdover_bw_mult | RW | 0x0 | Holdover Filter Bandwidth Multiplier.<br>Fine control of the holdover filter bandwidth. A value of zero disables the holdover filter (infinite bandwidth), which is also the default setting. |
| 7:4 | holdover_history | RW | 0x0 | Holdover History.<br>▪ 0x0 = Instantaneous (no history)<br>▪ 0x1 = 1 second<br>▪ 0x2 = 2 seconds<br>▪ 0x3 = 3 seconds<br>▪ 0x4 = 4 seconds<br>▪ 0x5 = 5 seconds<br>▪ 0x6 = 6 seconds<br>▪ 0x7 = 7 seconds<br>▪ 0x8 = 8 seconds<br>▪ 0x9 = 9 seconds<br>▪ 0xA = 10 seconds |
| 3:0 | dpll_lol_cnt_thresh | RW | 0x0 | DPLL Loss-of-Lock Counter Threshold.<br>While the DPLL Loss-of-Lock counter (*dpll_lol_cnt*) exceeds this threshold, the *dpll_lol_lmt* bit is set. |

**Table 102. DPLL_BANDWIDTH - DPLL Bandwidth Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:11 | acquire_bw_shift | RW | 0xE | Acquire Loop Filter Bandwidth Shift.<br>Coarse control of the DPLL loop filter bandwidth in the Acquire state while locking to the input clock. Default bandwidth = 1023Hz. |
| 10:8 | acquire_bw_mult | RW | 0x0 | Acquire Loop Filter Bandwidth Multiplier.<br>Fine control of the DPLL loop filter bandwidth in the Acquire state while locking to the input clock. Default bandwidth = 1023Hz |
| 7:3 | normal_bw_shift | RW | 0xB | Normal Operation Loop Filter Bandwidth Shift.<br>Coarse control of the DPLL loop filter bandwidth in the Normal state when locked to the input clock. Default bandwidth = 127Hz. |
| 2:0 | normal_bw_mult | RW | 0x0 | Normal Operation Loop Filter Bandwidth Multiplier.<br>Fine control of the DPLL loop filter bandwidth in the Normal state when locked to the input clock. Default bandwidth = 127Hz. |

**Table 103. DPLL_DAMPING - DPLL Damping Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:14 | reserved | RO | 0x0 | Reserved. |
| 13:11 | acquire_damping_shift | RW | 0x5 | Acquire Loop Filter Damping Shift.<br>Coarse control of the DPLL loop filter damping in the Acquire state while locking to the input clock. Default damping causes 1.1 dB peaking in the frequency domain jitter transfer function. |

**Table 103. DPLL_DAMPING - DPLL Damping Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 10:8 | acquire_damping_mult | RW | 0x1 | Acquire Loop Filter Damping Multiplier.<br>Fine control of the DPLL loop filter damping in the Acquire state while locking to the input clock. Default damping causes 1.1 dB peaking in the frequency domain jitter transfer function. |
| 7:6 | reserved | RO | 0x0 | Reserved. |
| 5:3 | normal_damping_shift | RW | 0x0 | Normal Operation Loop Filter Damping Shift.<br>Coarse control of the DPLL loop filter damping in the Normal state when locked to the input clock. Default damping causes 0.1 dB peaking in the frequency domain jitter transfer function. |
| 2:0 | normal_damping_mult | RW | 0x0 | Normal Operation Loop Filter Damping Multiplier.<br>Fine control of the DPLL loop filter damping in the Normal state when locked to the input clock. Default damping causes 0.1 dB peaking in the frequency domain jitter transfer function. |

**Table 104. DPLL_PHASE_SLOPE_LIMIT - DPLL Phase Slope Limit Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:29 | reserved | RO | 0x0 | Reserved. |
| 28:0 | phase_slope_limit | RW | 0x1FFFFFFF | Phase Slope Limit.<br>Control of the phase slope limit of the output clocks. This represents the maximum instant relative frequency change of the output clock. This is an unsigned unitless number although it is often expressed as μs/s or ns/s. It is recommended to program a value that is approx. 10% smaller than the required limit to leave some room for the integrator to adjust to frequency offsets.<br>The resolution of 1 LSB is $2^{-35}$ = 2.91e-11 = 29.1 ps/s. |

**Table 105. DPLL_PHASE_OFFSET - DPLL Phase Offset Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:30 | reserved | RO | 0x0 | Reserved. |
| 29:0 | phase_offset | RW | 0x0 | Phase Offset.<br>Manually sets the phase offset between the reference and feedback clocks. This is a 30-bit 2's complement value. The resolution is the TDC resolution / 8 (≈ 2.3 ps) and the range is ≈ ±1.26 ms. This allows all outputs to be adjusted in terms of their phase relationship to the input. All outputs move together using this precision setting.<br>This field is not used when hitless switching is enabled.<br>This register is atomic. When the most significant byte (bits [29:24]) is written, the new value is applied to the APLL. |

Table 106. DPLL_WRITE_FREQ - DPLL Write Frequency Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:29 | reserved | RO | 0x0 | Reserved. |
| 28:0 | write_freq | RW | 0x0 | Write Frequency.<br>Frequency control word for synthesizer/DCO mode. This is a 29-bit 2's complement value. The units are $2^{-40}$ * 1e6 [ppm]. The maximum setting is ±243ppm (±267,386,880).<br>apll_fb_sdm_order must be set to a value greater than 0 for write_freq to operate correctly.<br>An update to this multi-byte register will only take effect when the most significant byte (bits [28:24]) are written, the new value is applied to the DPLL. |

Table 107. DPLL_FB_DIV_NUM - DPLL Feedback Divider Numerator Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 63:48 | reserved | RO | 0x0 | Reserved. |
| 47:0 | fb_div_num | RW | 0x0 | Feedback Divider Numerator.<br>DPLL feedback divide numerator value. Refer to fb_div_int for details.<br>*Note*: This register field is part of an atomic group consisting of fb_div_num, fb_div_den and fb_div_int. When the most significant byte (bits [47:40]) of fb_div_num or fb_div_den, or the most significant byte (bits 20:16]) of fb_div_int is written, the value of all these fields are applied to the DPLL. |

Table 108. DPLL_FB_DIV_DEN - DPLL Feedback Divider Denominator Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 63:48 | reserved | RO | 0x0 | Reserved. |
| 47:0 | fb_div_den | RW | 0x800000 | Feedback Divider Denominator.<br>DPLL feedback divide denominator value. Refer to *fb_div_int* for details.<br>*Note*: The MSB (bit 47) of fb_div_den must be a 1. For an arbitrary fraction *M/N*, this may be accomplished by left shifting the denominator value *N* until the MSB becomes 1, and then left shifting the numerator value *M* by the same number of bits to obtain the *fb_div_num* value.<br>This register field is part of an atomic group consisting of *fb_div_num*, fb_div_den and fb_div_int. When the most significant byte (bits [47:40]) of *fb_div_num* or *fb_div_den,* or the most significant byte (bits 20:16]) of fb_div_int is written, the value of all these fields are applied to the DPLL. |

**Table 109. DPLL_FB_DIV_INT - DPLL Feedback Divider Integer Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:21 | reserved | RO | 0x0 | Reserved. |
| 20:0 | fb_div_int | RW | 0x190 | DPLL Feedback Clock Divider Integer.<br>DPLL feedback divide integer value. The DPLL feedback clock frequency must be no more than 33 MHz, and must be equal to the frequency of the reference clock divided by *id_ratio*, or equal to the reference clock when the input divider is bypassed by *id_byp_en*.<br>This register field is part of an atomic group consisting of *fb_div_num*, fb_div_den and fb_div_int. When the most significant byte (bits [47:40]) of *fb_div_num* or *fb_div_den,* or the most significant byte (bits 20:16]) of fb_div_int is written, the value of all these fields are applied to the DPLL. |

**Table 110. DPLL_FB_CORR - DPLL Feedback Correction Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | dpll_fb_div_dis | RW | 0x0 | DPLL Feedback Divider Disable.<br>Disables the DPLL feedback divider. May be set to 1 to reduce power consumption when an external feedback clock is selected with dpll_fb_sel. The feedback divider is automatically disabled when dpll_en is set to 0.<br>▪ 0x0 = DPLL feedback divider enabled if dpll_en is set to 1<br>▪ 0x1 = DPLL feedback divider disabled |
| 14:10 | reserved | RO | 0x0 | Reserved. |
| 9 | fine_rev | RW | 0x0 | TDC Fine Timestamp Bit Reversal.<br>Selects the bit ordering of the fine timestamp signals from the TDC analog to digital. This setting is intended for debug purposes only.<br>▪ 0x0 = Analog 30:0 maps to digital 30:0<br>▪ 0x1 = Analog 0:30 maps to digital 30:0 |
| 8:7 | pec_delay | RW | 0x0 | PEC Delay.<br>Phase error correction delay. Intended for debug purposes only. After device startup, should only be changed while tdc_en is set to 0.<br>▪ 0x0 = sdm error no delay<br>▪ 0x1 = sdm error delay one cycle<br>▪ 0x2 = Same as 0x0<br>▪ 0x3 = Same as 0x0 |
| 6:0 | pec_corr_mult | RW | 0x0 | Feedback Correction Multiplier.<br>Multiplier to get the FB SDM remainder bits on the same resolution as the TDC phase detector bits.(resolution ≈18.7 ps if the TDC APLL runs at 864MHz). After device startup, should only be changed while tdc_en is set to 0. |

**Table 111. DPLL_LOCK - DPLL Lock Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:16 | dpll_lock_timer | RW | 0x00FF | DPLL Lock Timer.<br>Specifies the time interval during which the absolute value of the phase detector error must remain below the DPLL lock threshold (dpll_lock_thresh) in order to declare lock. The DPLL switches from the Acquire state to the Normal state when the threshold has been met for half of this time interval. If enabled by bw_damp_sw, the loop filter bandwidth and damping settings revert at this time from the Acquire settings to the Normal settings. When the threshold has been met again for half of this time interval, the DPLL declares lock. The minimum value is 2. The units are ms. |
| 15:0 | dpll_lock_thresh | RW | 0x0155 | DPLL Lock Threshold.<br>Specifies the threshold that the absolute value of the phase detector error must remain below during the DPLL lock timer (dpll_lock_timer) in order to declare lock. The units are 8 * TDC resolution. |

**Table 112. DPLL_STS - DPLL Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | reserved | RO | 0x0 | Reserved. |
| 6:4 | dpll_state_sts | RO | 0x0 | DPLL Frequency Switch Mode (FSM) State.<br>Decode as follows:<br>▪ 0x0 = Freerun<br>▪ 0x1 = Normal / locked<br>▪ 0x2 = Holdover<br>▪ 0x3 = Write frequency<br>▪ 0x4 = Acquire<br>▪ 0x5 = Hitless switch |
| 3 | reserved | RO | 0x0 | Reserved. |
| 2:1 | dpll_ref_sel_sts | RO | 0x0 | DPLL Reference Clock Selection Status.<br>Indicates the reference clock selected by the DPLL.<br>▪ 0x0 = clkin0<br>▪ 0x1 = clkin1<br>▪ 0x2 = clkin2<br>▪ 0x3 = clkin3 |
| 0 | dpll_lock_sts | RO | 0x0 | DPLL Lock Status.<br>Indicates the DPLL lock status:<br>▪ 0x0 = Unlocked<br>▪ 0x1 = Locked |

**Table 113. DPLL_EVENT - DPLL Event Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3 | dpll_state_ch | RW1C | 0x0 | DPLL State Change Event.<br>Set to 1 when the DPLL state machine changes state. Once asserted, this bit will remain asserted until cleared by a write of 1 to this bit position. |

**Table 113. DPLL_EVENT - DPLL Event Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 2 | dpll_holdover | RW1C | 0x0 | DPLL Holdover Event.<br>Set to 1 when the DPLL state machine enters the holdover state. Once asserted, this bit will remain asserted until cleared by a write of 1 to this bit position. |
| 1 | dpll_lol_lmt | RW1C | 0x0 | DPLL Loss-of-Lock Counter Threshold Exceeded Status.<br>Set while the DPLL Loss-of-Lock counter (dpll_lol_cnt) exceeds the threshold set in *dpll_lol_cnt_thresh*. This bit cannot be cleared by software while the condition persists.<br>▪ 0x0 = Loss-of-lock counter has not exceeded the threshold since the last time the bit was cleared<br>▪ 0x1 = Loss-of-lock counter exceeded the threshold since the last time the bit was cleared |
| 0 | dpll_lol | RW1C | 0x0 | DPLL Loss-of-lock Event.<br>Set to 1 when the DPLL lock status transitions from locked to unlocked. Once asserted, this bit will remain asserted until cleared by a write of 1 to this bit position. |

**Table 114. DPLL_LOL_CNT - DPLL Loss-of-Lock Counter**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | dpll_lol_cnt | RW | 0x0 | DPLL Loss-of-Lock Counter.<br>This counter increments each time the DPLL lock status de-asserts, and saturates at 0xF. It is cleared by writing it to 0x0, and may be preset by writing the desired value.<br>This register can only be written if the block is not clock gated (dpll_cg) or held in reset (dpll_sw_rst). |

**Table 115. DPLL_FILTER_STS - DPLL Filter Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:29 | reserved | RO | 0x0 | Reserved. |
| 28:0 | filter_status | RO | 0x0 | DPLL Filter Status.<br>Provides the integrator value from the filter. |

**Table 116. DPLL_PHASE_STS - DPLL Phase Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:30 | reserved | RO | 0x0 | Reserved. |
| 29:0 | phase_status | RO | 0x0 | DPLL Phase Status.<br>Provides the phase data from the decimator. This is a 32-bit 2's complement value. The units are the TDC APLL VCO period divided by 62*8 (2.333ps for the nominal 864MHz). |

## 11.11 IOD Registers

**Table 117. IOD_INT_CNFG - IOD Integer Ratio Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | iod_pd | RW | 0x0 | Integer Output Divider Power Down.<br>Powers down the integer output divider by turning off the regulator. If this bit is set to 1, iod_rst must also be set to 1. When clearing this bit, iod_rst must remain set and then it can be cleared afterwards.<br>▪ 0x0 = Divider is powered up<br>▪ 0x1 = Divider is powered down |
| 30 | iod_dis | RW | 0x0 | Integer Output Divider Disable.<br>Disables the integer output divider. If this bit is set to 1, iod_rst must also be set to 1. When clearing this bit, iod_rst must remain set and then it can be cleared afterwards.<br>▪ 0x0 = Divider enabled<br>▪ 0x1 = Divider disabled |
| 29 | iod_rst | RW | 0x0 | Integer Output Divider Reset.<br>Resets the integer output divider.<br>▪ 0x0 = Divider reset de-asserted<br>▪ 0x1 = Divider reset asserted |
| 28 | iod_squelch | RW | 0x0 | Integer Output Divider Squelch.<br>Synchronously squelches and releases the IOD output clock.<br>▪ 0x0 = IOD output is not squelched<br>▪ 0x1 = IOD output is squelched |
| 27:25 | reserved | RO | 0x0 | Reserved. |
| 24:0 | iod_int | RW | 0x64 | Integer Output Divider Ratio<br>Integer output divider ratio. The minimum value is 14.<br>This register is atomic. When the most significant byte (bit [24]) is written, the new value is applied to the IOD. |

**Table 118. IOD_PHASE_CNFG - IOD Phase Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | iod_ph_adj_now | RW1S | 0x0 | Integer Output Divider Phase Adjustment Now.<br>When this bit is written from 0 to 1, the phase adjustment in iod_phase is applied to the divider. This bit self-clears when the adjust completes. |
| 14 | iod_ph_adj_post_sync | RW | 0x0 | Integer Output Divider Phase Adjustment After Synchronization.<br>When this bit is set to 1, the phase adjustment in iod_phase is applied to the divider whenever the divider is synchronized. |
| 13:11 | reserved | RO | 0x0 | Reserved. |

**Table 118. IOD_PHASE_CNFG - IOD Phase Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 10:9 | iod_sync_group | RW | 0x0 | Integer Output Divider Sync Group.<br>Sets the sync group that this divider belongs to<br>▪ 0x0 = Group0<br>▪ 0x1 = Group1<br>▪ 0x2 = None<br>▪ 0x3 = None |
| 8:0 | iod_phase | RW | 0x0 | Integer Output Divider Phase Configuration.<br>Signed 2's complementary value sets the phase, a positive value means lag from 0 phase, a negative value means lead from 0 phase, in steps of one VCO period.<br>The available range is ± 0~255 steps (approximately ± 0~20ns). In SYSREF mode (sysref_mode is set to 1), the range is also limited to ± (iod_int/2 - 5) steps.<br>This register is atomic. When the most significant byte (bit [8]) is written, the new value is applied to the IOD according to iod_ph_adj_now and iod_ph_adj_post_sync. |

**Table 119. SYSREF_CNFG - SYSREF Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:14 | reserved | RO | 0x0 | Reserved. |
| 13:12 | sysref_sync_src_sel | RW | 0x0 | SYSREF Synchronization Source Select.<br>Selects the source SYSREF instance used to trigger this SYSREF instance. If multiple SYSREF instances are intended to be aligned (part of a SYSREF group), they all must select the same source instance. If a SYSREF instance is independent, this field should be set to its own instance number. The default setting is for SYSREF instance zero to trigger all SYSREF instances.<br>To be used as a synchronization source, a SYSREF instance must:<br>▪ Be enabled and not held in reset (iod_pd, iod_dis and iod_rst are set to 0).<br>▪ Select itself if selected by other SYSREF instances |
| 11 | reserved | RO | 0x0 | Reserved. |
| 10 | sysref_src | RW | 0x0 | SYSREF Trigger Source.<br>Selects the source for the SYSREF trigger.<br>▪ 0x0 = Trigger from SYSREF_IN<br>▪ 0x1 = Trigger from register sysref_trig |
| 9 | sysref_trig | RW | 0x0 | SYSREF Trigger Signal.<br>Software SYSREF trigger. Writing this bit from 0 to 1 triggers SYSREF if sysref_src selects it. |

**Table 119. SYSREF_CNFG - SYSREF Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 8 | sysref_mode | RW | 0x0 | SYSREF Modes.<br>This bit defines whether the IOD operates normally (outputting a clock continuously), or whether it operates in SYSREF mode (outputting a burst of clocks after receiving a trigger). This bit must be set to 0 to output the clock from the IOD even if SYSREF functionality is not required.<br>▪ 0x0 = Normal mode, continuous output clock<br>▪ 0x1 = SYSREF mode, pulsed output clock (see sysref_pulse_cnt) |
| 7:0 | sysref_pulse_cnt | RW | 0x0 | SYSREF Pulse Counter Configuration.<br>Sets the number of sysref clocks to output after receiving a trigger. The number of clocks is one more than the value of this field, so the possible number of clocks is 1 to 256. |

## 11.12 FOD Registers

**Table 120. FOD_INT_CNFG - FOD Integer Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 63 | fod_pd | RW | 0x0 | Fractional Output Divider Power Down.<br>Powers down the fractional output divider by turning off the regulator. If this bit is set to 1, fod_rst must also be set to 1. When clearing this bit, fod_rst must remain set and then it can be cleared afterwards.<br>▪ 0x0 = Divider is powered up<br>▪ 0x1 = Divider is powered down |
| 62 | fod_dis | RW | 0x0 | Fractional Output Divider Disable.<br>Disables the fractional output divider. If this bit is set to 1, fod_rst must also be set to 1. When clearing this bit, fod_rst must remain set and then it can be cleared afterwards.<br>▪ 0x0 = Divider enabled<br>▪ 0x1 = Divider disabled |
| 61 | fod_rst | RW | 0x0 | Fractional Output Divider Reset.<br>Resets the fractional output divider.<br>▪ 0x0 = Divider reset de-asserted<br>▪ 0x1 = Divider reset asserted |
| 60 | fod_acc_reset | RW | 0x0 | Fractional Output Divider Accumulator Reset.<br>Resets the FOD SDM accumulator.<br>▪ 0x0 = Accumulator reset de-asserted<br>▪ 0x1 = Accumulator reset asserted |
| 59:26 | fod_frac | RW | 0x0 | Fractional Output Divider Ratio Fraction Portion.<br>Denominator is fixed to 2^34.<br>This register field is part of an atomic group consisting of fod_1st_int and fod_frac. When the most significant byte (bits [33:30]) of fod_frac is written, the value of all these fields are applied to the FOD.<br>*Note*: When an FOD has spread-spectrum fod_frac[5:0] *must* be set to 0x0. |

**Table 120. FOD_INT_CNFG - FOD Integer Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 25:9 | fod_2nd_int | RW | 0x0 | Fractional Output Divider Ratio 2nd Integer Portion.<br>Half integer divide ratio of second stage. The actual divide ratio is (fod_2nd_int * 2). A setting of 1 is invalid: the minimum divide ratio is 4. Set to 0 to bypass the second stage. |
| 8:0 | fod_1st_int | RW | 0x64 | Fractional Output Divider Ratio 1st Integer Portion.<br>Integer divide ratio of first stage (MMD).<br>The first 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 / 33MHz = 324.25<br>If the first stage frequency is less than 70MHz, the fod_slow_freq_en bit must be set to 1.<br>This register field is part of an atomic group consisting of fod_1st_int and fod_frac. When the most significant byte (bits [33:30]) of fod_frac is written, the value of all these fields are applied to the FOD.<br>*Note*: Refer to fod_frac for details about serial bus writes to this register field and automatic updates performed by the over-clocking engine. |

**Table 121. FOD_PHASE_CNFG - FOD Phase Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | fod_ph_adj_now | RW1S | 0x0 | Fraction Output Divider Phase Adjustment Now.<br>When this bit is written from 0 to 1, the phase adjustment in fod_phase is applied to the divider. This bit self-clears when the adjust completes. |
| 14 | fod_ph_adj_post_sync | RW | 0x1 | Fraction Output Divider Phase Adjustment After Synchronization.<br>When this bit is set to 1, the phase adjustment in fod_phase is applied to the divider whenever the divider is synchronized. |
| 13 | fod_slow_freq_en | RW | 0x0 | FOD Slow Frequency Enable.<br>Must be set to 1 when the MMD (first stage) frequency is under 70MHz. |
| 12 | reserved | RO | 0x0 | Reserved. |
| 11:10 | fod_sync_group | RW | 0x0 | Fraction Output Divider Sync Group.<br>Sets the sync group that this divider belongs to:<br>▪ 0x0 = Group0<br>▪ 0x1 = Group1<br>▪ 0x2 = None<br>▪ 0x3 = None |
| 9:0 | fod_phase | RW | 0x3F0 | Fraction Output Divider Phase Configuration.<br>Signed 2's complementary value sets the phase. A positive value means lag from 0 phase, and a negative value means lead from 0 phase, in steps of 1/4 VCO period.<br>The default value of -16 (decimal), or -4.0 VCO periods, approximately aligns the FOD output clock with the IOD output clock, when the FOD is configured with an integer divide ratio.<br>This register is atomic. When the most significant byte (bits [9:8]) is written, the new value is applied to the FOD according to fod_ph_adj_now and fod_ph_adj_post_sync. |

## 11.13 SSC Register

**Table 122. SSC_CNFG - Spectrum Spreading Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | ssc_en | RW | 0x0 | Spread-Spectrum Clocking Enable.<br>Enable spread spectrum. The spread configuration is determined by the other register fields in this register.<br>▪ 0x0 = SSC disabled<br>▪ 0x1 = SSC enabled<br>If the FOD0 and FOD1 SSC modulation frequencies are the same, the FOD1 SSC phase can be aligned to FOD0 SSC by setting ssc_share to 1.<br>*Note*: When ssc_share is set to 1, then FOD1 ssc_en must be set to 1 before FOD0 ssc_en is set to 1 since FOD1 SSC will start when FOD0 ssc_en is set to 1. This restriction does not apply when loading the device configuration from OTP/EEPROM on startup, but does apply if dynamically changing these settings later through a dynamic configuration load from the OTP/EEPROM, or by writing registers from the serial interface. |
| 30 | ssc_mode | RW | 0x0 | Spectrum Spreading Mode.<br>▪ 0x0 = Down spreading<br>▪ 0x1 = Center spreading |
| 29:24 | reserved | RO | 0x0 | Reserved. |
| 23:16 | ssc_ampl | RW | 0x51 | Spectrum Spreading Amplitude.<br>Sets the positive and negative spreading amplitude. For down spread, ssc_ampl is only used for the negative limit and the positive limit is internally set to 0. For center spread, the peak-to-peak spread amplitude is twice the specified amplitude (for a 1% peak-to-peak center spread, define ssc_ampl as 0.5%).<br>ssc_ampl = spread_percentage / 100 * 2^14<br>For example, for 1% spread, set ssc_ampl to 0.01 * 2^14 = 163 decimal, or 0xA3.<br>The default value corresponds to a 0.5% down spread at 31.5kHz. |
| 15:0 | ssc_step | RW | 0x2B8C | Spectrum Spreading Step Size.<br>Set ramp step size to get the target modulation rate.<br>For down spread:<br>▪ ssc_step = ssc_ampl * 2^16 * ssc_freq / 15MHz<br>For center spread:<br>▪ ssc_step = 2 * ssc_ampl * 2^16 * ssc_freq / 15MHz<br>where:<br>ssc_freq is the target modulation rate from 30kHz to 63kHz<br>15MHz is the system clock divided by 4, assuming the system clock is 60MHz<br>*Example 1*. For a 32kHz 1% down spread:<br>ssc_ampl = 163<br>ssc_step = 163 * 2^16 * 32kHz / 15MHz = 0x5905<br>*Example 2*. For a 32kHz ± 0.5% center spread:<br>ssc_ampl = 81<br>ssc_step = 2 * 81 * 2^16 * 32kHz / 15MHz = 0x5879<br>The default value corresponds to a 0.5% down spread at 31.5kHz. |

## 11.14 BANK Register

**Table 123. OUT_BANK_CNFG - Output Bank Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:6 | reserved | RO | 0x0 | Reserved. |
| 5 | bank_pd | RW | 0x0 | Output Bank Power Down.<br>Powers down the output bank by turning off the regulator. When a bank is powered down, all output driver(s) in that bank should also be powered down by setting their out_pd bit(s) to 1.<br>▪ 0x0 = Bank is powered up<br>▪ 0x1 = Bank is powered down |
| 4:3 | bank_fanout_mode | RW | 0x0 | Output Bank Fan-out Buffer Mode.<br>Configures the bank selection for fanout buffer mode. Only available on banks 4, 5, and 6.<br>The device must be configured for fanout buffer mode by fanout_buf_mode (CLKIN0) / fanout_buf_mode1 (CLKIN1) for this setting to take effect.<br>▪ 0x0 = Normal operation mode. The output bank source clock is selected by output_bank_src.<br>▪ 0x1 = CLKIN0 Fan-out buffer mode. When fanout_clkmode is 1, the output bank selects CLKIN0. When fanout_clkmode is 0, the output bank source clock is selected by output_bank_src.<br>▪ 0x2 = CLKIN1 Fan-out buffer mode. When fanout_clkmode1 is 1, the output bank selects CLKIN1. When fanout_clkmode1 is 0, the output bank source clock is selected by output_bank_src. |
| 2:0 | output_bank_src | RW | 0x5 | Output Bank Source.<br>Sets the clock source of each output bank, in conjunction with bank_fanout_mode. Some configurations can be reserved based on Output Bank Source Mapping table. The bits in BANK_MUX_CLK_EN - Bank Mux Clock Enable must be set appropriately to enable only the selected source for each bank.<br>▪ 0x0 = IOD0 for Banks 0, 1 / clkin2_div (CLKIN1) for Banks 4, 5, 6<br>▪ 0x1 = IOD1 for Banks 0, 1, 2 / refin_div (XIN_REF) for Bank 5<br>▪ 0x2 = IOD2 for Banks 4, 5, 6<br>▪ 0x3 = IOD3 for Banks 5, 6<br>▪ 0x4 = FOD0 for Banks 0, 1, 2, 3<br>▪ 0x5 = FOD1<br>▪ 0x6 = FOD2 for Banks 3, 4, 5, 6<br>▪ 0x7 = clkin0_div (CLKIN0) for Banks 4, 5, 6 |

## 11.15 OUT Registers

Table 124. ODRV_EN - Output Driver Enable

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | out_pd | RW | 0x0 | Output Driver Power Down.<br>Powers down the output clock driver.<br>▪ 0x0 = Output driver is powered up<br>▪ 0x1 = Output driver is powered down |
| 6 | out_oe_mode | RW | 0x0 | Output Driver OE Mode.<br>Controls whether the output enable acts synchronously or asynchronously with respect to the output divider clock. Must be set to asynchronous mode when outputting SYSREF.<br>▪ 0x0 = OE is synchronized to the divider clock. Enabling and disabling the output clock is glitchless. OE transitions take effect after 1 divider clock cycle.<br>▪ 0x1 = OE is asynchronous to the divider clock. OE transitions while the divider clock is toggling can result in glitches/runt pulses. |
| 5:3 | out_dis_group | RW | 0x0 | Output Driver OE Group Select and Global Output Enable Exclusion.<br>Sets which OE group this driver is in, and if not assigned to a group, can also exclude global output enables from applying to the clock.<br>▪ 0x0 = Group0<br>▪ 0x1 = Group1<br>▪ 0x2 = Group2<br>▪ 0x3 = Group3<br>▪ 0x4 = Group4<br>▪ 0x5 = None<br>▪ 0x7 = None, and exclude global output enables<br>When set to 0x7, allows the output clock to be enabled regardless of:<br>▪ APLL or DPLL lock status (equivalent to out_startup set to 2)<br>▪ goe register bit<br>▪ GOE pin, if one is assigned with gpi_func/gpio_func<br>The output clock can still be disabled by the out_dis register bit.<br>This setting is intended to allow an output clock to run freely following the configuration load so that it can be used as an external APLL or DPLL feedback clock. In that application, if the output clock's divider is resynchronized after the startup sequence completes, the output clock will stop running for less than 1us and then the APLL or DPLL will re-align the feedback clock to the reference clock. |
| 2:1 | out_dis_state | RW | 0x3 | OUT Driver Disabled State.<br>Controls the state of OUTx / OUTxb when the output driver is disabled.<br>▪ 0x0 = Held High / Low<br>▪ 0x1 = Held Low / High<br>▪ 0x2 = Held Hi-Z / Hi-Z<br>▪ 0x3 = Held Low / Low (except LVDS mode is held High / Low) |
| 0 | out_dis | RW | 0x0 | OUTx and/or OUTxb Driver Disable.<br>Forces both OUTx and OUTxb Drivers to be disabled if in differential mode or forces OUT Driver to be disabled if in CMOS mode. For more information, see Output Enable Control.<br>▪ 0x0 = OUTx and/or OUTxb Driver is enabled if not disabled by other means<br>▪ 0x1 = OUTx and/or OUTxb Driver is disabled |

Table 125. ODRV_CNFG - Output Driver Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | out_prog7 | RW | 0x0 | Output Driver Programmability Bit 7.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>reserved / flip output polarity / flip output polarity |
| 14 | out_prog6 | RW | 0x0 | Output Driver Programmability Bit 6.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>reserved / reserved / reserved |
| 13 | out_prog5 | RW | 0x0 | Output Driver Programmability Bit 5.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>reserved / cross point lower / cross point tune |
| 12 | out_prog4 | RW | 0x0 | Output Driver Programmability Bit 4.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>reserved / cross point increase for double termination / cross point tune |
| 11 | out_prog3 | RW | 0x0 | Output Driver Programmability Bit 3.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>tristate OUT*x* / driver impedance -5% / cross point tune |
| 10 | out_prog2 | RW | 0x0 | Output Driver Programmability Bit 2.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>tristate OUT*x*b / driver impedance +5% / cross point tune |
| 9 | out_prog1 | RW | 0x0 | Output Driver Programmability Bit 1.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>flip OUT*x* polarity / amplitude -10% / amplitude -10% |
| 8 | out_prog0 | RW | 0x0 | Output Driver Programmability Bit 0.<br>When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode:<br>flip OUT*x*b polarity / amplitude +5% / amplitude +5% |
| 7 | reserved | RO | 0x0 | Reserved. |
| 6 | out_lpamp | RW | 0x0 | Output Driver LPHCSL Amplitude Control.<br>Controls the amplitude of the output driver when LPHCSL mode is selected.<br>▪ 0x0 = 800mV<br>▪ 0x1 = 900mV |
| 5 | out_lpsr | RW | 0x1 | Output Driver LPHCSL Slew Rate Control.<br>Controls the slew rate of the output driver when LPHCSL mode is selected. Based on 5" transmission line simulation condition. Slew rates are measured from -150mV to +150mV from crossing point.<br>▪ 0x0 = Slow, 2-4 V/ns<br>▪ 0x1 = Fast, > 4 V/ns |

**Table 125. ODRV_CNFG - Output Driver Configuration (Cont.)**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4 | out_lpimp | RW | 0x1 | Output Driver LPHCSL Impedance Control.<br>Controls the output impedance of the output driver when LPHCSL mode is selected.<br>▪ 0x0 = 85Ω<br>▪ 0x1 = 100Ω |
| 3:2 | out_cmdrv | RW | 0x3 | Output Driver CMOS Slew Rate Control.<br>Controls the slew rate of the output driver (in V/ns) when CMOS mode is selected, according to the supply voltage level of 3.3V / 2.5V / 1.8V:<br>▪ 0x0 = 4.2 / 2.7 / 1.8<br>▪ 0x1 = 2.2 / 1.5 / 1.8<br>▪ 0x2 = 2.2 / 1.5 / 1.8<br>▪ 0x3 = 3.4 / 2.0 / 1.9 |
| 1:0 | out_mode | RW | 0x0 | Output Driver Type.<br>Selects the output driver type.<br>▪ 0x0 = LPHCSL<br>▪ 0x1 = LVDS<br>▪ 0x2 = LVDS<br>▪ 0x3 = CMOS |

## 11.16 GPI Registers

**Table 126. GPI_CNFG - GPI Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:10 | reserved | RO | 0x0 | Reserved. |
| 9 | gpi_pol | RW | 0x0 | GPI Polarity<br>Sets the pin polarity. This bit is ignored if gpi_func configures the pin as a reference clock input.<br>▪ 0x0 = Normal sense. Pin functions denoted with a # are active low, others are active high.<br>▪ 0x1 = Inverted sense. Pin functions denoted with a # are active high, others are active low. |
| 8 | gpi_pullup | RW | 0x0 | GPI Pull-up Enable<br>Sets the internal pull-up mode. This bit is ignored and the internal pull-down is enabled if gpi_func configures the pin as a reference clock input.<br>▪ 0x0 = Pull-up disabled<br>▪ 0x1 = Pull-up enabled |

**Table 126. GPI_CNFG - GPI Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | gpi_pulldn | RW | 0x0 | GPI Pull-down Enable.<br>Sets the internal pull-up or pull-down modes. This bit is ignored and the internal pull-down is enabled if gpi_func configures the pin as a reference clock input.<br>▪ 0x0 = Pull-down disabled<br>▪ 0x1 = Pull-down enabled |
| 6:0 | gpi_func | RW | 0x7F | GPI Functions.<br>Sets the general purpose input function.<br>▪ 0x0 = OE[0], input, enable output drivers in OE group 0<br>▪ 0x1 = OE[1], input, enable output drivers in OE group 1<br>▪ 0x2 = OE[2], input, enable output drivers in OE group 2<br>▪ 0x3 = OE[3], input, enable output drivers in OE group 3<br>▪ 0x4 = OE[4], input, enable output drivers in OE group 4<br>▪ 0x5 = PERST#, input, latches CLKIN0 fanout buffer mode clock selection on active edge<br>▪ 0x6 = GOE, input, enable all output drivers<br>▪ 0x7 = DPLL_CLK_SEL[0], input<br>▪ 0x8 = DPLL_CLK_SEL[1], input<br>▪ 0x9 = PERST1#, input, latches XO/IB fanout buffer mode clock selection on active edge<br>▪ 0xA = Reserved<br>▪ 0xE = SDI, input (SPI 4-wire mode)<br>▪ 0xF = SYSREF_IN<br>▪ 0x10 = GPI, input, input status allowed read back via SSI<br>▪ 0x7F = GPI function disabled, pin used as reference clock input |

**Table 127. GPI_STS - GPI Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | gpi_sts | RO | 0x0 | GPI Status.<br>Indicates the status of the GPI0/1/2/3 pins without latching and without applying optional polarity inversion (gpi_pol). If a pin is configured to be a reference clock input (see gpi_func), the status reads back as 0.<br>Bit [3] = GPI3 status<br>Bit [2] = GPI2 status<br>Bit [1] = GPI1 status<br>Bit [0] = GPI0 status |

## 11.17 GPIO Registers

**Table 128. GPIO_CNFG - GPIO Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15 | reserved | RO | 0x0 | Reserved. |
| 14:13 | gpo_drv | RW | 0x0 | GPO Drive Strength.<br>Applies to the pad when configured as an output (gpio_type is 0x0 or 0x2). Drive strength increases as this setting increases. |
| 12 | gpo_ctrl | RW | 0x0 | GPO Output Control Signal Value.<br>Sets the value to drive the GPO pin when configured as a general purpose output. |
| 11:10 | gpio_type | RW | 0x1 | GPIO Type.<br>Sets the direction and type following reset.<br>▪ 0x0 = Output (driven high/low), or bidirectional if configured as SDIO by gpio_func<br>▪ 0x1 = Input (2-level)<br>▪ 0x2 = Output (open-drain)<br>▪ 0x3 = Input (tri-level). Only valid for GPIO0/1/2 when set to GPI mode or Dynamic CSEL; reserved for GPIO3/4. |
| 9 | gpio_pol | RW | 0x0 | GPIO Polarity.<br>Sets the pin polarity. This bit is ignored if gpio_func configures the pin as a tri-level Dynamic CSEL, GPO, or test clock output.<br>▪ 0x0 = Normal sense. Pin functions denoted with a # are active low, others are active high.<br>▪ 0x1 = Inverted sense. Pin functions denoted with a # are active high, others are active low. |
| 8 | gpio_pullup | RW | 0x0 | GPIO Pull-up Enable.<br>Sets the internal pull-up mode.<br>▪ 0x0 = Pull-up disabled<br>▪ 0x1 = Pull-up enabled |
| 7 | gpio_pulldn | RW | 0x1 | GPIO Pull-down Enable.<br>Sets the internal pull-down mode.<br>▪ 0x0 = Pull-down disabled<br>▪ 0x1 = Pull-down enabled |

Table 128. GPIO_CNFG - GPIO Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 6:0 | gpio_func | RW | 0x10 | GPIO Functions.<br>Sets the general purpose input/output function. Refer to pwrdn_sel for PWRGD/PWRDN# assignment.<br>0x0 = OE[0], input, enable output drivers in OE group 0<br>0x1 = OE[1], input, enable output drivers in OE group 1<br>0x2 = OE[2], input, enable output drivers in OE group 2<br>0x3 = OE[3], input, enable output drivers in OE group 3<br>0x4 = OE[4], input, enable output drivers in OE group 4<br>0x5 = PERST#, input, latches CLKIN0 fanout buffer mode clock selection on active edge<br>0x6 = GOE, input, enable all output drivers<br>0x7 = DPLL_CLK_SEL[0], input<br>0x8 = DPLL_CLK_SEL[1], input<br>0x9 = PERST1#, input, latches CLKIN1 fanout buffer mode clock selection on active edge<br>0xE = SDI, input (SPI 4-wire mode)<br>0xF = SYSREF_IN, input<br>0x10 = GPI, input, input status allowed read back via SSI<br>0x11 = Dynamic CSEL, input, dynamic configuration control. Only valid for GPIO0/1/2. May be tri-level or 2-level.<br>0x1C = INT, output<br>0x1D = GPO, output, to control external functions such as LEDs. The output value is set in gpo_ctrl.<br>0x1E = SDO, output (SPI 4-wire mode) or SDIO, bidirectional (SPI-3-wire mode; gpio_type must be set to 0, and is internally controlled as 0 or 1 by the SPI logic).<br>0x20 = clkin0 los_sts, output<br>0x21 = clkin1 los_sts, output<br>0x22 = clkin2 los_sts, output<br>0x23 = clkin3 los_sts, output<br>0x24 = refin los_sts, output<br>0x25 = apll_lock_sts, output<br>0x26 = dpll_lock_sts, output<br>0x28 = clkin0 los_evt, output<br>0x29 = clkin1 los_evt, output<br>0x2A = clkin2 los_evt, output<br>0x2B = clkin3 los_evt, output<br>0x2C = refin los_evt, output<br>0x2D = apll_lol, output<br>0x2E = dpll_lol, output<br>0x30 = clkin0 los_lmt_evt, output<br>0x31 = clkin1 los_lmt_evt, output<br>0x32 = clkin2 los_lmt_evt, output<br>0x33 = clkin3 los_lmt_evt, output<br>0x34 = refin los_lmt_evt, output<br>0x35 = apll_lol_lmt, output<br>0x36 = dpll_lol_lmt, output<br>0x37 = CLKMODE (fanout_clkmode), output<br>0x38 = device_ready, output<br>0x39 = CLKMODE1 (fanout_buf_mode1), output |

**Table 129. GPIO_STS - GPIO Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:0 | gpio_sts | RO | 0x0 | GPIO Status.<br>Indicates the status of the GPIO0/1/2/3/4 pins without latching and without applying optional polarity inversion (gpio_pol).<br>For GPIO0/1/2, the possible encodings are:<br>▪ 0x0 = Tri-level low, 2-level low<br>▪ 0x1 = Tri-level mid, 2-level unused<br>▪ 0x2 = Unused<br>▪ 0x3 = Tri-level high, 2-level high<br>Bit [7] = GPIO4 status<br>Bit [6] = GPIO3 status<br>Bits [5:4] = GPIO2 status<br>Bits [3:2] = GPIO1 status<br>Bits [1:0] = GPIO0 status |

## 11.18 EEPROM Registers

**Table 130. EEPROM_CNFG - EEPROM Load Configuration**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 63:56 | reserved | RO | 0x0 | Reserved. |
| 55:44 | eeprom_fall | RW | 0x96 | EEPROM Falling Edge Time.<br>Cycle number (counting down from eeprom_cycle) at which the SCL falling edge occurs. The default value is for a 60MHz clock and must be scaled according to the actual system clock frequency. |
| 43:32 | eeprom_rise | RW | 0x1C2 | EEPROM Rising Edge Time.<br>Cycle number (counting down from eeprom_cycle) at which the SCL rising edge occurs. The default value is for a 60MHz clock and must be scaled according to the actual system clock frequency. |
| 31:20 | eeprom_cycle | RW | 0x258 | EEPROM Cycle Time.<br>Number of system clock cycles in one SCL period when running at 100kHz. The default value is for a 60MHz clock and must be scaled according to the actual system clock frequency. |
| 19:15 | reserved | RO | 0x0 | Reserved. |
| 14:13 | eeprom_i2c_drv | RW | 0x0 | I2C/SMBus Drive Strength.<br>Selects the output driver slew rate of the SDA_nCS and SCL_SCLK pins when the internal I2C master is active. (higher settings means higher drive strength). This setting does not affect the internal timing, refer to eeprom_i2c_speed.<br>0x0 = 1.8V Standard mode (100 kHz) or 2.5V/3.3V standard (100kHz) and Fast mode (400kHz)<br>▪ 0x1 = 1.8V Fast mode (400 kHz)<br>▪ 0x2 = Reserved<br>▪ 0x3 = 1.8V/2.5V/3.3V Fast mode plus (1 MHz) |

Table 130. EEPROM_CNFG - EEPROM Load Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 12 | eeprom_ext_addr | RW | 0x0 | EEPROM Extended Address Enable.<br>Allows extended 10-bit addressing with a 1-byte I2C address, if supported by the EEPROM.<br>▪ 0x0 = The address is sent outside of the device address.<br>▪ 0x1 = Address bits 10:8 are sent in bits 2:0 of the device address, and address bits 7:0 are sent in the 1-byte address. eeprom_addr_size must be set to 0. |
| 11 | eeprom_addr_size | RW | 0x1 | EEPROM Address Size.<br>Number of address bytes sent to the EEPROM during a read.<br>0x0 = 1-byte address<br>0x1 = 2-byte address |
| 10:7 | eeprom_length | RW | 0x4 | EEPROM Size.<br>Selects the number of bytes in the EEPROM for storing configurations.<br>▪ 0x0 = 128B<br>▪ 0x1 = 256B<br>▪ 0x2 = 512B<br>▪ 0x3 = 1KB<br>▪ 0x4 = 2KB<br>▪ 0x5 = 4KB<br>▪ 0x6 = 8KB<br>▪ 0x7 = 16KB<br>▪ 0x8 = 32KB<br>▪ 0x9 = 64KB |
| 6 | eeprom_load_en | RW | 0x0 | EEPROM Load Enable.<br>Enables loading of the common and/or user configurations from an external EEPROM device. The device loads configurations in the following order: OTP common, EEPROM common (if eeprom_load_en is set to 1), OTP user, and EEPROM user (if eeprom_load_en is set to 1). This bit may be programmed in the OTP common and/or user configurations to control whether the device attempts to load the common and/or user configurations from EEPROM.<br>*Warning*: If the SDA pin is held low or floating when the $I^2C$ master attempts to read the EEPROM, the $I^2C$ master will wait indefinitely until SDA becomes high before beginning the read request.<br>▪ 0x0 = Disabled<br>▪ 0x1 = Enabled |
| 5:4 | eeprom_i2c_speed | RW | 0x0 | EEPROM I2C Speed.<br>Selects the I2C master speed for EEPROM load. When the speed is 400kHz or 1MHz, eeprom_fall, eeprom_rise and eeprom_cycle are internally divided by 4 or 10 respectively to achieve the faster timing. The pad drive strength (eeprom_i2c_drv) should also be set according to the speed.<br>▪ 0x0 = 100kHz<br>▪ 0x1 = 400kHz<br>▪ 0x2 = 1MHz<br>▪ 0x3 = Reserved |
| 3:0 | eeprom_retry_count | RW | 0x4 | EEPROM Load Retry Count.<br>Number of times to attempt to load EEPROM. If eeprom_bad or load failed (CRC error) after this number of attempts, the eeprom_load_fail status bit is set. |

Table 131. EEPROM_ADDR - EEPROM Address Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | reserved | RO | 0x0 | Reserved. |
| 6:0 | eeprom_addr | RW | 0x50 | EEPROM Device Address.<br>Sets the I2C device address of the EEPROM to load. |

Table 132. EEPROM_EVENT - EEPROM Load Event Status

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3 | eeprom_bad | RW1C | 0x0 | EEPROM Not Detected.<br>When high, indicates the EEPROM did not acknowledge a read access during device startup or a dynamic configuration load. In this case, eeprom_load_fail is not set. If EEPROM load is disabled (eeprom_load_en is set to 0), then VC7 will not attempt to read the EERPOM and this bit cannot be set.<br>Cleared by writing it to 1. |
| 2 | eeprom_config_empty | RW1C | 0x0 | EEPROM Load of Empty Configuration.<br>When high, indicates the EEPROM load attempted to load a configuration that did not select any blocks, during device startup or a dynamic configuration load.<br>Cleared by writing it to 1. |
| 1 | eeprom_load_fail | RW1C | 0x0 | EEPROM Load Failure.<br>When high, indicates the EEPROM load failed during device startup or a dynamic configuration load. This bit is not set if the EEPROM does not respond (eeprom_bad is set instead).<br>Cleared by writing it to 1. |
| 0 | eeprom_crc_err | RW1C | 0x0 | EEPROM Load CRC Error.<br>When high, indicates the EEPROM load encountered one or more CRC errors during device startup or a dynamic configuration load.<br>Cleared by writing it to 1. |

Table 133. EEPROM_ERR_CNT - EEPROM CRC Error Count

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | eeprom_crc_err_cnt | RW | 0x0 | EEPROM CRC Error Counter.<br>This counter increments each time the loader detects a CRC error while reading the EEPROM, and saturates at 0xF. It is cleared by writing it to 0x0, and may be preset by writing the desired value. Preset may be used as a debug tool.<br>This register can only be written if the block is not clock gated (otp_cg) or held in reset (otp_sw_rst). |

**Table 134. OTP_VPPLMT - OTP VPP Monitor Limit**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 15:14 | reserved | RO | 0x0 | Reserved. |
| 13:0 | vpplmt | RW | 0x3D0 | OTP VPP Monitor Limit.<br>Number of VPP monitor clock cycles before the VPP monitor issues an error (vpp_error). Counted in multiples of 1024 us. The default value is approximately 1 second. |

**Table 135. OTP_EVENT - OTP Event Status**

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | manual_rdy | RW1C | 0x0 | Manual Ready Indicator.<br>When high, indicates a manual request (including Program Assist) completed. Cleared by writing it to 1. |
| 6 | status_latched | RW1C | 0x0 | OTP Latched Status.<br>Latched value of the OTP STATUS pin. Can be cleared by writing 1 to it. If STATUS is still high when clearing is attempted, this bit will immediately be set to 1 again. |
| 5 | vpp_error | RW1C | 0x0 | OTP VPP Error.<br>This error bit signals that the VPP monitor has detected that the OTP internal charge pump was enabled for longer than the time defined in the vpplmt field.<br>This bit gets cleared by writing one to it. It cannot be cleared unless the internal condition has gone away (i.e., VPP_MON has been de-asserted). |
| 4 | reserved | RO | 0x0 | Reserved. |
| 3 | pgm_assist_fail | RW1C | 0x0 | Program Assist Failure.<br>When high, indicates that the Program Assist sequence failed to program one or more bits in the OTP word. |
| 2 | otp_config_empty | RW1C | 0x0 | OTP Load of Empty Configuration.<br>When high, indicates the OTP load attempted to load a configuration that did not select any blocks. Cleared by writing it to 1. |
| 1 | otp_load_fail | RW1C | 0x0 | OTP Load Failure.<br>When high, indicates the OTP load failed during device startup or a dynamic configuration load. Cleared by writing it to 1. |
| 0 | otp_crc_err | RW1C | 0x0 | OTP Load CRC Error.<br>When high, indicates the OTP load encountered one or more CRC errors during device startup or a dynamic configuration load. Cleared by writing it to 1. |

## 11.19 INT Registers

Table 136. SCRATCH0 - Software Scratch Register 0

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31:0 | scratch0 | RW | 0x0 | Scratch Register and Power-down Controls.<br>For arbitrary software use.<br>In VC7A+, bits[3:0] have the following function:<br>▪ scratch0[3]: System clock divider disable override; When high, this bit overrides the normal enable pin of the system clock divider and forces it to be low (disabled). This bit should only be set when the digital core is running on the RC oscillator clock.<br>▪ scratch0[2]: APLL feedback divider disable override; When high, this bit overrides the normal enable pin of the APLL feedback divider and forces it to be low (disabled). scratch0[1]: System clock divider reset; When high, this bit resets the system clock divider. This bit should only be set when the digital core is running on the RC oscillator clock.<br>▪ scratch0[0]: System clock select; When high, this bit cleanly switches the digital core system clock to the RC oscillator clock. This bit should only be set once the RC oscillator has been enabled (by programming the por_osc_sel field to 0x3) once it is cleanly toggling. To switch the digital core back to the VCO clock, this bit must be written back to zero, but only once the VCO has been properly enabled (see vco_dis) and after the system clock divider has been reset (by writing to one and then zero bit[1] of this register). |

Table 137. INT_EN - Interrupt Enable

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | device_int_en | RW | 0x0 | Device Interrupt Enable.<br>Overall device interrupt enable. When this field is set to 1, the device interrupt is asserted while device_int_sts is 1. |
| 30 | reserved | RO | 0x0 | Reserved. |
| 29 | load_fail_int_en | RW | 0x0 | Configuration Loader Failure Interrupt Enable.<br>When this field is set to 1, the load_fail_int_sts bit contributes to the device interrupt |
| 28 | load_err_int_en | RW | 0x0 | Configuration Loader Error Interrupt Enable.<br>When this field is set to 1, the load_err_int_sts bit contributes to the device interrupt |
| 27 | otp_manual_rdy_int_en | RW | 0x0 | OTP Manual Request Ready Interrupt Enable.<br>When this field is set to 1, the otp_manual_rdy_int_sts bit contributes to the device interrupt |
| 26 | reserved | RO | 0x0 | Reserved. |
| 25 | los4_lmt_int_en | RW | 0x0 | REFIN Monitor LOS Threshold Exceeded Interrupt Enable.<br>When this field is set to 1, the los4_lmt_int_sts bit contributes to the device interrupt |
| 24 | los3_lmt_int_en | RW | 0x0 | CLKIN3 Monitor LOS Threshold Exceeded Interrupt Enable.<br>When this field is set to 1, the los3_lmt_int_sts bit contributes to the device interrupt |
| 23 | los2_lmt_int_en | RW | 0x0 | CLKIN2 Monitor LOS Threshold Exceeded Interrupt Enable.<br>When this field is set to 1, the los2_lmt_int_sts bit contributes to the device interrupt |

Table 137. INT_EN - Interrupt Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 22 | los1_lmt_int_en | RW | 0x0 | CLKIN1 Monitor LOS Threshold Exceeded Interrupt Enable.<br>When this field is set to 1, the los1_lmt_int_sts bit contributes to the device interrupt |
| 21 | los0_lmt_int_en | RW | 0x0 | CLKIN0 Monitor LOS Threshold Exceeded Interrupt Enable.<br>When this field is set to 1, the los0_lmt_int_sts bit contributes to the device interrupt |
| 20 | dpll_lol_lmt_int_en | RW | 0x0 | DPLL Loss-of-lock Threshold Exceeded Interrupt Enable.<br>When this field is set to 1, the dpll_lol_lmt_int_sts bit contributes to the device interrupt |
| 19 | apll_lol_lmt_int_en | RW | 0x0 | APLL Loss-of-lock Threshold Exceeded Interrupt Enable.<br>When this field is set to 1, the apll_lol_lmt_int_sts bit contributes to the device interrupt |
| 18 | reserved | RO | 0x0 | Reserved. |
| 17 | freq3_int_en | RW | 0x0 | CLKIN3 Frequency Monitor Interrupt Enable.<br>When this field is set to 1, the freq3_int_sts bit contributes to the device interrupt |
| 16 | freq2_int_en | RW | 0x0 | CLKIN2 Frequency Monitor Interrupt Enable.<br>When this field is set to 1, the freq2_int_sts bit contributes to the device interrupt |
| 15 | freq1_int_en | RW | 0x0 | CLKIN1 Frequency Monitor Interrupt Enable.<br>When this field is set to 1, the freq1_int_sts bit contributes to the device interrupt |
| 14 | freq0_int_en | RW | 0x0 | CLKIN0 Frequency Monitor Interrupt Enable.<br>When this field is set to 1, the freq0_int_sts bit contributes to the device interrupt. |
| 13:9 | reserved | RO | 0x0 | Reserved. |
| 8 | los4_int_en | RW | 0x0 | REFIN Monitor Loss-of-Signal interrupt Enable.<br>When this field is set to 1, the los4_int_sts bit contributes to the device interrupt |
| 7 | los3_int_en | RW | 0x0 | CLKIN3 Monitor Loss-of-Signal Interrupt Enable.<br>When this field is set to 1, the los3_int_sts bit contributes to the device interrupt |
| 6 | los2_int_en | RW | 0x0 | CLKIN2 Monitor Loss-of-Signal Interrupt Enable.<br>When this field is set to 1, the los2_int_sts bit contributes to the device interrupt |
| 5 | los1_int_en | RW | 0x0 | CLKIN1 Monitor Loss-of-Signal Interrupt Enable.<br>When this field is set to 1, the los1_int_sts bit contributes to the device interrupt |
| 4 | los0_int_en | RW | 0x0 | CLKIN0 Monitor Loss-of-Signal Interrupt Enable.<br>When this field is set to 1, the los0_int_sts bit contributes to the device interrupt. |
| 3 | dpll_state_ch_int_en | RW | 0x0 | DPLL State Change Interrupt Enable.<br>When this field is set to 1, the dpll_state_ch_int_sts bit contributes to the device interrupt. |
| 2 | dpll_holdover_int_en | RW | 0x0 | DPLL Holdover Interrupt Enable.<br>When this field is set to 1, the dpll_holdover_int_sts bit contributes to the device interrupt. |

Table 137. INT_EN - Interrupt Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 1 | dpll_lol_int_en | RW | 0x0 | DPLL Loss-of-Lock Interrupt Enable.<br>When this field is set to 1, the dpll_lol_int_sts bit contributes to the device interrupt. |
| 0 | apll_lol_int_en | RW | 0x0 | APLL Loss-of-Lock Interrupt Enable.<br>When this field is set to 1, the apll_lol_int_sts bit contributes to the device interrupt. |

Table 138. INT_STS - Interrupt Status

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | device_int_sts | RO | 0x0 | Device Interrupt Status.<br>Overall device interrupt status. This bit is the OR of all the other interrupt status bits in this register after masking by their respective interrupt enable bits in the INT_EN - Interrupt Enable register. This bit is masked by device_int_en. The resulting signal can be output on the assigned GPIO pin. |
| 30 | reserved | RO | 0x0 | Reserved. |
| 29 | load_fail_int_sts | RO | 0x0 | Configuration Loader Failure Interrupt Status.<br>The logical OR of the otp_load_fail and eeprom_load_fail event bits |
| 28 | load_err_int_sts | RO | 0x0 | Configuration Loader Error Interrupt Status.<br>The logical OR of the otp_crc_err and eeprom_crc_err event bits |
| 27 | otp_manual_rdy_int_sts | RO | 0x0 | OTP Manual Request Ready Interrupt Status.<br>Mirrors the OTP manual_rdy event bit |
| 26 | reserved | RO | 0x0 | Reserved. |
| 25 | los4_lmt_int_sts | RO | 0x0 | REFIN Monitor LOS Threshold Exceeded Interrupt Status.<br>Mirrors the REFIN los_lmt_evt event bit |
| 24 | los3_lmt_int_sts | RO | 0x0 | CLKIN3 Monitor LOS Threshold Exceeded Interrupt Status.<br>Mirrors the CLKIN3 los_lmt_evt event bit |
| 23 | los2_lmt_int_sts | RO | 0x0 | CLKIN2 Monitor LOS Threshold Exceeded Interrupt Status.<br>Mirrors the CLKIN2 los_lmt_evt event bit |
| 22 | los1_lmt_int_sts | RO | 0x0 | CLKIN1 Monitor LOS Threshold Exceeded Interrupt Status.<br>Mirrors the CLKIN1 los_lmt_evt event bit |
| 21 | los0_lmt_int_sts | RO | 0x0 | CLKIN0 Monitor LOS Threshold Exceeded Interrupt Status.<br>Mirrors the CLKIN0 los_lmt_evt event bit |
| 20 | dpll_lol_lmt_int_sts | RO | 0x0 | DPLL Loss-of-lock Threshold Exceeded Interrupt Status.<br>Mirrors the dpll_lol_lmt event bit |
| 19 | apll_lol_lmt_int_sts | RO | 0x0 | APLL Loss-of-lock Threshold Exceeded Interrupt Status.<br>Mirrors the apll_lol_lmt event bit |
| 18 | reserved | RO | 0x0 | Reserved. |
| 17 | freq3_int_sts | RO | 0x0 | CLKIN3 Frequency Monitor Interrupt Status.<br>Mirrors the CLKIN3 freq_evt event bit |
| 16 | freq2_int_sts | RO | 0x0 | CLKIN2 Frequency Monitor Interrupt Status.<br>Mirrors the CLKIN2 freq_evt event bit |
| 15 | freq1_int_sts | RO | 0x0 | CLKIN1 Frequency Monitor Interrupt Status.<br>Mirrors the CLKIN1 freq_evt event bit |

Table 138. INT_STS - Interrupt Status (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|---|---|---|---|---|
| 14 | freq0_int_sts | RO | 0x0 | CLKIN0 Frequency Monitor Interrupt Status.<br>Mirrors the CLKIN0 freq_evt event bit |
| 13:9 | reserved | RO | 0x0 | Reserved. |
| 8 | los4_int_sts | RO | 0x0 | REFIN Monitor Loss-of-Signal Interrupt Status.<br>Mirrors the REFIN los_evt event bit |
| 7 | los3_int_sts | RO | 0x0 | CLKIN3 Monitor Loss-of-Signal Interrupt Status.<br>Mirrors the CLKIN3 los_evt event bit |
| 6 | los2_int_sts | RO | 0x0 | CLKIN2 Monitor Loss-of-Signal Interrupt Status.<br>Mirrors the CLKIN2 los_evt event bit |
| 5 | los1_int_sts | RO | 0x0 | CLKIN1 Monitor Loss-of-Signal Interrupt Status.<br>Mirrors the CLKIN1 los_evt event bit |
| 4 | los0_int_sts | RO | 0x0 | CLKIN0 Monitor Loss-of-Signal Interrupt Status.<br>Mirrors the CLKIN0 los_evt event bit |
| 3 | dpll_state_ch_int_sts | RO | 0x0 | DPLL State Change Interrupt Status.<br>Mirrors the dpll_state_ch event bit |
| 2 | dpll_holdover_int_sts | RO | 0x0 | DPLL Holdover Interrupt Status.<br>Mirrors the dpll_holdover event bit |
| 1 | dpll_lol_int_sts | RO | 0x0 | DPLL Loss-of-lock Interrupt Status.<br>Mirrors the dpll_lol event bit |
| 0 | apll_lol_int_sts | RO | 0x0 | APLL Loss-of-lock Interrupt Status.<br>Mirrors the apll_lol event bit |

# 12. Revision History

| Revision | Date | Description |
|---|---|---|
| 1.01 | Aug 14, 2023 | ▪ Updated part number references to RC310xx<br>▪ Updated device_revision in Table 42 (DEVICE_REV)<br>▪ Updated the description of en_cap_x2 and en_cap_x1 in Table 61 (XO_CNFG)<br>▪ Added Table 95 (ANA_SPARE_CNFG) and Table 96 (ANA_SPARE_STS) |
| 1.00 | Oct 26, 2022 | 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/.