

RC210xx

This document describes the functional description, register organization, and byte addresses of the RC210xx. Detailed register definitions can be located by following the links in the document.

Contents

| | |
|--|-----------|
| 1. Functional Description | 3 |
| 1.1 Power-Up, Configuration, and Serial Interfaces | 3 |
| 1.2 Input Clocks | 3 |
| 1.2.1 Crystal/Reference Input | 3 |
| 1.2.2 Clock Inputs | 3 |
| 1.3 APLL | 4 |
| 1.3.1 APLL Lock Detector | 4 |
| 1.4 Output Dividers | 4 |
| 1.4.1 Integer Output Dividers | 4 |
| 1.4.2 Fractional Output Dividers | 4 |
| 1.5 Clock Outputs | 5 |
| 1.5.1 Output Types | 5 |
| 1.5.2 Output Banks | 6 |
| 1.6 Output Enable Control | 6 |
| 2. Serial Interfaces | 7 |
| 2.1 1-Byte and 2-Byte Offset Mode | 7 |
| 3. I2C Slave | 8 |
| 3.1 1-Byte Offset Mode | 8 |
| 3.2 2-Byte Offset Mode | 8 |
| 4. SPI Slave | 9 |
| 5. SMBus Slave | 10 |
| 5.1 SMBus 1-Byte/2-Byte Offset Mode Programming Examples | 10 |
| 6. Example of Programming Process | 11 |
| 6.1 Getting Register Value from rbs Setting File | 11 |
| 6.2 Example of I2C 1-Byte Offset Mode Programming | 11 |
| 6.3 Example of I2C 2-Byte Offset Mode Programming | 12 |
| 7. Example of Changing Output Frequency | 13 |
| 7.1 Output Frequency Calculation | 13 |
| 7.2 Feedback Divider | 14 |
| 7.3 Integer Output Divider | 14 |
| 7.4 Fractional Output Dividers | 15 |
| 8. Example of Device Programming with the Driver | 16 |
| 8.1 Example of Python Code | 16 |
| 8.2 Example of DOS Batch File | 16 |
| 9. Register Organization | 17 |
| 9.1 Overview | 17 |
| 9.2 Register Block Offsets | 17 |
| 9.3 Register Offsets | 18 |

| | | |
|------------|------------------------------------|-----------|
| 9.3.1 | Global Address Map | 18 |
| 9.3.2 | SSI Address Map | 18 |
| 9.3.3 | Crystal Address Map | 18 |
| 9.3.4 | Clock Input Address Map | 19 |
| 9.3.5 | LOS Monitor Address Map | 19 |
| 9.3.6 | APLL Address Map | 19 |
| 9.3.7 | IOD Address Map | 20 |
| 9.3.8 | FOD Address Map | 20 |
| 9.3.9 | SSC Address Map | 20 |
| 9.3.10 | Output Bank Address Map | 20 |
| 9.3.11 | Clock Output Driver Address Map | 20 |
| 9.3.12 | GPI Address Map | 20 |
| 9.3.13 | GPIO Address Map | 21 |
| 9.3.14 | Interrupt Address Map | 21 |
| 10. | Physical Register Addresses | 21 |
| 10.1 | Global Block | 21 |
| 10.2 | SSI Block | 22 |
| 10.3 | XO Block | 22 |
| 10.4 | CLKIN Block | 22 |
| 10.5 | LOSMON Block | 23 |
| 10.6 | APLL Block | 24 |
| 10.7 | IOD Block | 25 |
| 10.8 | FOD Block | 26 |
| 10.9 | OUT Block | 27 |
| 10.10 | BANK Block | 28 |
| 10.11 | GPI Block | 28 |
| 10.12 | GPIO Block | 28 |
| 10.13 | SSC Block | 29 |
| 10.14 | INT Block | 29 |
| 11. | Register Descriptions | 30 |
| 11.1 | Global Registers | 30 |
| 11.2 | SSI Registers | 39 |
| 11.3 | XO Register | 42 |
| 11.4 | CLKIN Register | 43 |
| 11.5 | LOSMON Registers | 44 |
| 11.6 | FREQMON Registers | 47 |
| 11.7 | TDC APLL Registers | 48 |
| 11.8 | APLL Registers | 50 |
| 11.9 | IOD Registers | 61 |
| 11.10 | FOD Registers | 63 |
| 11.11 | SSC Register | 65 |
| 11.12 | BANK Register | 66 |
| 11.13 | OUT Registers | 67 |
| 11.14 | GPI Registers | 69 |
| 11.15 | GPIO Registers | 71 |
| 11.16 | EEPROM Registers | 73 |
| 11.17 | INT Registers | 77 |
| 12. | Revision History | 80 |

1. Functional Description

The RC210xx is a small-form factor, fully integrated, low-power, high-performance frequency synthesizer providing excellent PCIe and Ethernet phase jitter, while covering a wide range of output frequencies up to 650MHz. It can simultaneously provide low phase jitter non-spreading clocks for Ethernet and storage applications, while providing spread-spectrum PCIe Gen6 clocks.

The following sections provide an overview of the RC210xx.

1.1 Power-Up, Configuration, and Serial Interfaces

The RC210xx 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 RC210xx 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 RC210xx supports one crystal/reference input and up to two differential or four single-ended clock inputs.

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 RC210xx 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 `clk3` internally. If set to differential type, `CLKIN0/CLKIN0b` pair drives `clk0`, while `CLKIN1/CLKIN1b` pair drives `clk2`. 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.

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 Output Dividers

The RC210xx provides four integer and three fractional output dividers.

1.4.1 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.

1.4.2 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.

1.4.2.1 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.

1.4.2.2 Sync and Phase Adjustment

Each FOD can adjust its output clock phase with a step size of 1/4 VCO period up to about ± 20 ns. 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 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, 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 `apl_reinit` bit causes the power-up sequence to restart from the VCO calibration step, which will synchronize the dividers after the APLL locks.

1.4.2.3 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.

1.5 Clock Outputs

The RC210xx 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.

1.5.1 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.

1.5.2 Output Banks

The RC210xx 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

| output_bank_src | Bank 0 | Bank 1 | Bank 2 | Bank 3 | Bank 4 | Bank 5 | Bank 6 |
|-----------------|--------|--------|----------|----------|----------|-----------|--------|
| | OUT0 | OUT1 | OUT[2:3] | OUT[4:7] | OUT[8:9] | OUT10 | OUT11 |
| 0x0 | IOD0 | | N/A | | CLKIN1 | | |
| 0x1 | IOD1 | | | N/A | | XIN_REFIN | N/A |
| 0x2 | N/A | | | | IOD2 | | |
| 0x3 | N/A | | | | | IOD3 | |
| 0x4 | FOD0 | | | | N/A | | |
| 0x5 | FOD1 | | | | | | |
| 0x6 | N/A | | | FOD2 | | | |
| 0x7 | N/A | | | | CLKIN0 | | |

1.6 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 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, OUTx and OUTxb 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 RC210xx 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 RC210xx 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 RC210xx has a 912-byte register memory space. It also separates as four pages, and each page is 256 bytes.

The RC210xx 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.

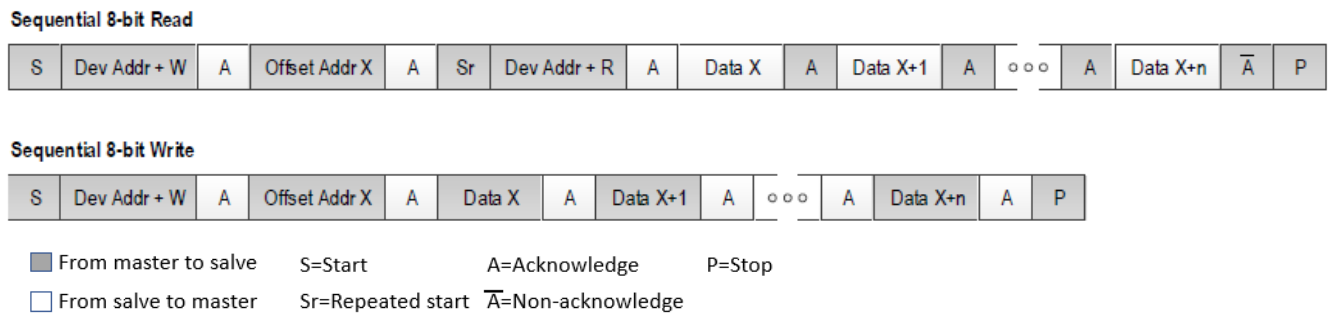


Figure 1. I2C 1-Byte Offset Mode Slave Sequencing

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.

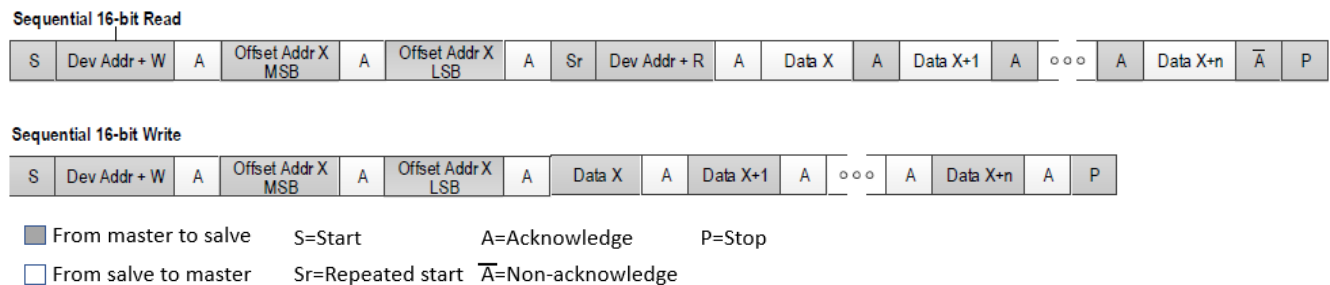


Figure 2. I2C 2-Byte Offset Mode Slave Sequencing

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.

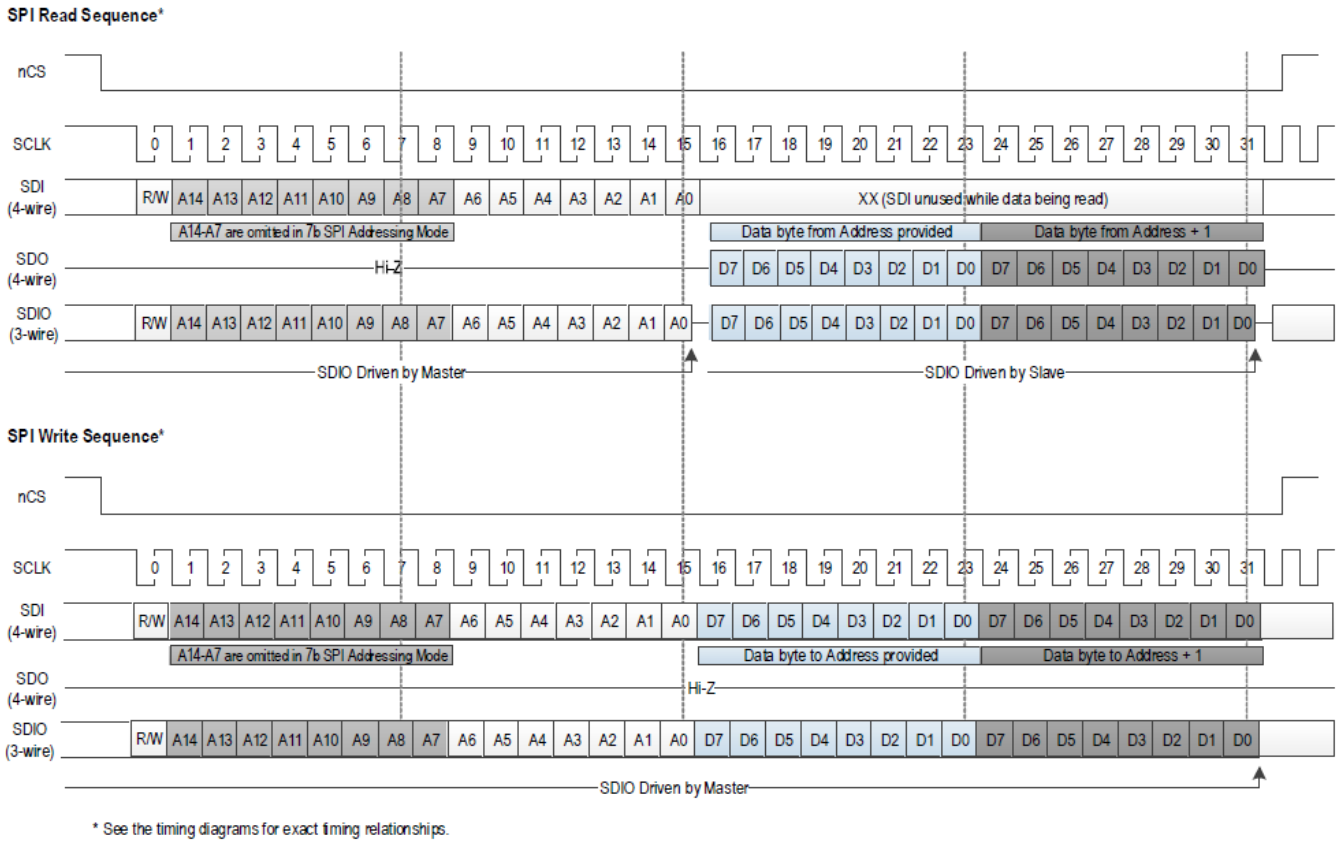


Figure 3. SPI Sequencing Diagram

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 RC210xx 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.

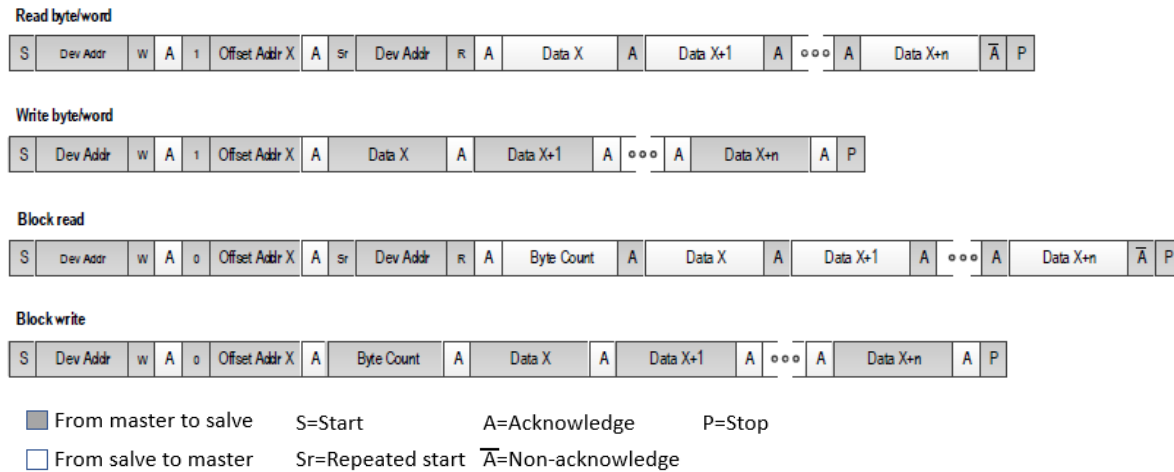


Figure 4. I2C Slave Timing Diagram

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 RC210xx:

- 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 byte for the page number. It also applies to SMBus and SPI.

There are four steps to program the RC210xx:

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 // Apl1 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
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 // Apl1 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 RC210xx 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) = 83.333333333333333333333333
- FB_INT_DIV = 83
- FB_FRAC_DIV = 0.6666666666666667 x 2²⁷ = 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 (= 2²⁵ - 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

$$100\text{MHz} = 10\text{GHz} / 100$$

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:

- $F_{\text{OUT}} > 33\text{MHz}$: $\text{FOD} = 1^{\text{st}} \text{ Integer} + \text{Fraction}$
- $F_{\text{OUT}} < 33\text{MHz}$: $\text{FOD} = (1^{\text{st}} \text{ Integer} + \text{Fraction}) \times 2 \times 2^{\text{nd}} \text{ Integer}$

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 $\times 2$ 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

$$156.25\text{MHz} = 10\text{GHz} / 64$$

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 RC210xx 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 |
| 0x050 += 0x010 | LOSMON[0:4] | LOS Monitor Addresses | LOSMON Registers |
| 0x120 | APLL | APLL Addresses | APLL 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 LOS Monitor Address Map

Table 8. 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 |

9.3.6 APLL Address Map

Table 9. 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 |

9.3.7 IOD Address Map

Table 10. 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 |

9.3.8 FOD Address Map

Table 11. 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 |

9.3.9 SSC Address Map

Table 12. SSC Addresses

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

9.3.10 Output Bank Address Map

Table 13. Output Bank Addresses

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

9.3.11 Clock Output Driver Address Map

Table 14. 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 |

9.3.12 GPI Address Map

Table 15. GPI Addresses

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

9.3.13 GPIO Address Map

Table 16. GPIO Addresses

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

9.3.14 Interrupt Address Map

Table 17. 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 18. 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 |

Table 18. Global Block Physical Register Addresses (Cont.)

| Address | Block Name | Register | Type | Default |
|---------|------------|----------------|------|---------|
| 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 19. 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 |
| 0x26 | SSI | SSI_GLOBAL_CNFG | RW | 0x01 |
| 0x27 | SSI | SSI_STS | R/W1C | 0x00 |

10.3 XO Block

Table 20. 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 21. 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 LOSMON Block

Table 22. 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 |
| 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 |

Table 22. LOSMON Block Physical Register Addresses (Cont.)

| Address | Block Name | Register | Type | Default |
|---------|------------|-------------------|-------|---------|
| 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 |

10.6 APLL Block

Table 23. 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 |

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

| Address | Block Name | Register | Type | Default |
|---------|------------|------------------|-------|---------|
| 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 |

10.7 IOD Block

Table 24. 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 |
| 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 |
| 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 |
| 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 |

10.8 FOD Block

Table 25. 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 |

10.9 OUT Block

Table 26. 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 |

10.10 BANK Block

Table 27. 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 |

10.11 GPI Block

Table 28. 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 |

10.12 GPIO Block

Table 29. 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 29. 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 |

10.13 SSC Block

Table 30. 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 |

10.14 INT Block

Table 31. 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 31. 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 32. VENDOR_ID - Vendor ID

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-------------|------|---------------|---|
| 15:12 | dev_id_type | RO | 0x1 | Device ID Block Type. 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. Renesas/IDT JEDEC ID. |

Table 33. DEVICE_ID - Device ID

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

Table 34. DEVICE_REV - Device Revision

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-----------------|------|---------------|--|
| 15:0 | device_revision | RO | 0x0443 | Device Revision. Decode as follows: <ul style="list-style-type: none"> ▪ 0x0111 = First silicon; not released. ▪ 0x0333 = Rev A ▪ 0x0443 = Rev B |

Table 35. DEVICE_PGM - Device Programming

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 15:0 | dash_code | RW | 0x0 | Dash Code. 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. <ul style="list-style-type: none"> ▪ 0x0 = No user configuration has been programmed at the factory |

Table 36. 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. <ul style="list-style-type: none"> ▪ 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 ▪ 0x1 = Reference currents disabled, must be set when the VDDO6 pin floats |
| 29 | float_vddo5 | RW | 0x0 | Float VDDO5 Supply. Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider IOD2 and output driver 10 in the VDDO5 domain. <ul style="list-style-type: none"> ▪ 0x0 = Reference currents enabled ▪ 0x1 = Reference currents disabled, must be set when the VDDO5 pin floats |
| 28 | float_vddo4 | RW | 0x0 | Float VDDO4 Supply. Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider FOD2 and output drivers 8/9 in the VDDO4 domain. <ul style="list-style-type: none"> ▪ 0x0 = Reference currents enabled ▪ 0x1 = Reference currents disabled, must be set when the VDDO4 pin floats |
| 27 | float_vddo3 | RW | 0x0 | Float VDDO3 Supply. Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider FOD1 and output drivers 4/5/6/7 in the VDDO3 domain. <ul style="list-style-type: none"> ▪ 0x0 = Reference currents enabled ▪ 0x1 = Reference currents disabled, must be set when the VDDO3 pin floats |
| 26 | float_vddo2 | RW | 0x0 | Float VDDO2 Supply. Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider FOD0 and output drivers 2/3 in the VDDO2 domain. <ul style="list-style-type: none"> ▪ 0x0 = Reference currents enabled ▪ 0x1 = Reference currents disabled, must be set when the VDDO2 pin floats |
| 25 | float_vddo1 | RW | 0x0 | Float VDDO1 Supply. Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider IOD1 and output driver 1 in the VDDO1 domain. <ul style="list-style-type: none"> ▪ 0x0 = Reference currents enabled ▪ 0x1 = Reference currents disabled, must be set when the VDDO1 pin floats |
| 24 | float_vddo0 | RW | 0x0 | Float VDDO0 Supply. Controls Iref and Ibias to the output divider LDO, output bank LDO, output divider IOD0 and output driver 0 in the VDDO0 domain. <ul style="list-style-type: none"> ▪ 0x0 = Reference currents enabled ▪ 0x1 = Reference currents disabled, must be set when the VDDO0 pin floats |
| 23 | reserved | RW | 0x0 | Reserved. |

Table 36. DEVICE_CNFG - Device Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|---------------|------|---------------|--|
| 22 | dyn_csel_deb | RW | 0x0 | Dynamic CSEL De-bounce Interval. 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. <ul style="list-style-type: none"> 0x0 = 8 system clock cycles (133ns) 0x1 = 256 system clock cycles (4.2us) |
| 21:20 | reserved | RW | 0x1 | reserved. |
| 19:16 | sync_dis_wait | RW | 0x2 | Divider Synchronization Output Clock Disable Wait Time. 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. <ul style="list-style-type: none"> 0x0 = 1us 0x1 = 2us 0x2 = 4us 0x3 = 8us 0x4 = 16us 0x5 = 32us 0x6 = 64us 0x7 = 128us 0x8 = 256us 0x9 = 512us 0xA = 1024us 0xB = 2048us 0xC = 4096us |
| 15:12 | pwrnd_sel | RW | 0xF | PWRGD/PWRDN# Pin Select. 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 <i>gpio_func</i> and <i>gpio_type</i> , and polarity inversion must be disabled (<i>gpio_pol</i>). This setting must be programmed by the OTP/EEPROM common configuration in order to take effect during the power-up sequence. <ul style="list-style-type: none"> 0x0 = GPIO0 0x1 = GPIO1 0x2 = GPIO2 0x3 = GPIO3 0x4 = GPIO4 0x5 = GPI0 0x6 = GPI1 0x7 = GPI2 0x8 = GPI3 0xF: No pin selected, associated functionality disabled |
| 11:6 | reserved | RW | 0x0 | Reserved. |

Table 36. DEVICE_CNFG - Device Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------|------|---------------|--|
| 11 | float_vddr3 | RW | 0x0 | Float VDDR3 Supply. 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). <ul style="list-style-type: none"> 0x0 = Reference currents enabled 0x1 = Reference currents disabled, must be set when the VDDR3 pin floats |
| 10 | float_vddr2 | RW | 0x0 | Float VDDR2 Supply. 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). <ul style="list-style-type: none"> 0x0 = Reference currents enabled 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. 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. <ul style="list-style-type: none"> 0x0 = No pin selected, treated as Low (tri-level) 0x1 = No pin selected, treated as Mid (tri-level) 0x2 = GPIO2 (tri-level) 0x3 = No pin selected, treated as High (tri-level) |
| 3:2 | static_csel1 | RW | 0x2 | Static Configuration Select Bit 1. 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. <ul style="list-style-type: none"> 0x0 = No pin selected, treated as Low (tri-level) 0x1 = No pin selected, treated as Mid (tri-level) 0x2 = GPIO1 (tri-level) 0x3 = No pin selected, treated as High (tri-level) |
| 1:0 | static_csel0 | RW | 0x2 | Static Configuration Select Bit 0. 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. <ul style="list-style-type: none"> 0x0 = No pin selected, treated as Low (tri-level) 0x1 = No pin selected, treated as Mid (tri-level) 0x2 = GPIO0 (tri-level) 0x3 = No pin selected, treated as High (tri-level) |

Table 37. PWR_CTL - Power Control

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-------------|------|---------------|--|
| 7:1 | reserved | RO | 0x0 | Reserved. |
| 0 | pd_restoreb | RW | 0x1 | <p>Power Down Restore (Active Low). 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.</p> <ul style="list-style-type: none"> 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. 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 38. REG_LOCK - Configuration Register Lock

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------|------|---------------|---|
| 7:0 | reg_lock_key | RW | 0x0 | <p>Configuration Register Lock Key. 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.</p> |

Table 39. INIT_SYNC - Initialization and Synchronization Register

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|----------------|------|---------------|--|
| 7 | clr_grp1_oe | RW1S | 0x0 | <p>Clear Reset Sequencer Sync Group 1 Output Enable. 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.</p> |
| 6 | clr_grp0_oe | RW1S | 0x0 | <p>Clear Reset Sequencer Sync Group 0 Output Enable. 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.</p> |
| 5 | id_global_setb | RW | 0x1 | <p>Input Dividers Common Set. 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.</p> |
| 4 | goe | RW | 0x1 | <p>Output Global OE. This bit allows manual CSR control of the global output OE.</p> |

Table 39. 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. Write 1 to trigger synchronization of output dividers in group 1 as defined in 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. Write 1 to trigger synchronization of output dividers in group 0 as defined in 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. Write 1 to trigger synchronization of output dividers in groups 0 and 1 as defined in 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. Writing this bit to 1 re-starts the startup sequence from the VCO calibration step, including divider synchronization. Self-cleared immediately. |

Table 40. 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. 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. 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. 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. 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 40. 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. 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. The TDC APLL digital logic is held in reset while this bit is set to 1. While re-programming the TDC APLL CSRs after device start-up, tdc_apll_dig_sw_rst must be set to 1. |
| 5 | reserved | RW | 0x0 | Reserved |
| 4 | clkmon4_sw_rst | RW | 0x0 | CLKMON4 Software Reset. The Clock Monitor 4 is held in reset while this bit is set to 1. 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. The Clock Monitor 3 is held in reset while this bit is set to 1. 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. The Clock Monitor 2 is held in reset while this bit is set to 1. 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. The Clock Monitor 1 is held in reset while this bit is set to 1. 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. The Clock Monitor 0 is held in reset while this bit is set to 1. While re-programming the clkin0 LOS or Frequency monitor CSRs after device start-up, clkmon0_sw_rst must be set to 1. |

Table 41. CLOCK_GATE - Clock Gate Register

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

Table 41. CLOCK_GATE - Clock Gate Register (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|--|
| 1 | clkmon0_cg | RW | 0x0 | CLKMON0 Clock Gate. The Clock Monitor 0 is clock gated while this bit is set to 1. |
| 0 | otp_cg | RW | 0x0 | OTP Logic Clock Gate. 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 42. MISC_CNFG - Miscellaneous Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------------|------|---------------|---|
| 15:11 | Reserved | RW | 0x20 | Reserved. |
| 10 | losmon_cksel | RW | 0x0 | LOS Monitor Measuring Clock Select. Selects the measuring clock for the LOS monitor blocks. <ul style="list-style-type: none"> 0x0 = quad system clock (nominally 240 MHz) 0x1 = TDC ring oscillator frequency divided by 2 (nominally 432MHz). |
| 9:8 | fanout_buf_mode1 | RW | 0x0 | CLKIN1 Fan-out Buffer Mode. Configures the device to operate in fan-out buffer mode from CLKIN1. Independent from the CLKIN0 fanout_buf_mode setting. <ul style="list-style-type: none"> 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. 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. 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. Controls whether the clock output drivers are disabled until the APLL locks during the startup sequence. When output_startup is not set to 0x2, individual output clocks may be configured to ignore APLL lock status by setting their out_dis_group to 0x7. <ul style="list-style-type: none"> 0x0 = Clock output drivers are disabled until APLL lock asserts 0x1 = Reserved 0x2 = Clock output drivers are not disabled by APLL lock status 0x3 = Reserved |
| 5 | ssc_share | RW | 0x0 | Spectrum Spreading Phase Share. 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 42. MISC_CNFG - Miscellaneous Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-----------------|------|---------------|---|
| 4:3 | fanout_buf_mode | RW | 0x0 | <p>CLKIN0 Fan-out Buffer Mode.</p> <p>Configures the device to operate in fan-out buffer mode from CLKIN0. Independent from the CLKIN1 fanout_buf_mode1 setting.</p> <ul style="list-style-type: none"> ▪ 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. ▪ 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. ▪ 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 43. 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 | <p>GPIO2 Value Latched at Startup.</p> <p>Value of GPIO2 latched at startup (when PWRGD/PWRDN# first becomes high, if pwrnd_sel selects a PWRGD/PWRDN# pin).</p> <ul style="list-style-type: none"> ▪ 0x0 = Tri-level low ▪ 0x1 = Tri-level mid ▪ 0x2 = Unused ▪ 0x3 = Tri-level high |
| 3:2 | gpio1_latched | RO | 0x0 | <p>GPIO1 Value Latched at Startup.</p> <p>Value of GPIO1 latched at startup (when PWRGD/PWRDN# first becomes high, if pwrnd_sel selects a PWRGD/PWRDN# pin).</p> <ul style="list-style-type: none"> ▪ 0x0 = Tri-level low ▪ 0x1 = Tri-level mid ▪ 0x2 = Unused ▪ 0x3 = Tri-level high |
| 1:0 | gpio0_latched | RO | 0x0 | <p>GPIO0 Value Latched at Startup.</p> <p>Value of GPIO0 latched at startup (when PWRGD/PWRDN# first becomes high, if pwrnd_sel selects a PWRGD/PWRDN# pin).</p> <ul style="list-style-type: none"> ▪ 0x0 = Tri-level low ▪ 0x1 = Tri-level mid ▪ 0x2 = Unused ▪ 0x3 = Tri-level high |

Table 44. DEVICE_STS - Device Status

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|----------------------|------|---------------|--|
| 15 | fanout_clkmode1 | RO | 0x0 | CLKIN1 Fanout Clock Mode. 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. 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. 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. 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. When high, indicates that the OTP loader detected that the OTP image is configured for single-ended mode. |
| 6 | eeeprom_config_valid | RO | 0x0 | Valid EEPROM User Configuration Loaded. 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. 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. 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 45. 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. <ul style="list-style-type: none"> ▪ 0x0 = SDO is driven on opposite SCLK edge than the sampling edge ▪ 0x1 = SDO is delayed one half cycle of SCLK |
| 2 | reserved | RO | 0x0 | Reserved. |

Table 45. SPI_CNFG - SPI Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-------------|------|---------------|---|
| 1 | spi_clk_sel | RW | 0x0 | SDI Sampling Edge Selection. <ul style="list-style-type: none"> 0x0 = SDI is sampled on rising SCLK edge 0x1 = SDI is sampled on falling SCLK edge |
| 0 | spi_3wire | RW | 0x0 | Select SPI 3 or 4-wire Mode. <ul style="list-style-type: none"> 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. 0x1 = 3-wire SPI. Data is received and transmitted on the GPIO pin assigned as SDIO. |

Table 46. I2C_FLTR_CNFG - I2C Filter Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|---------------|------|---------------|--|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | i2c_spike_ftr | RW | 0x1 | I2C/SMBus Digital Spike Filter Duration. 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 47. 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. 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. Delays transmission of 0 value by this number of 133ns periods (8 system clock cycles). |

Table 48. 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. Sets I2C or SMBus device address that the SSI will acknowledge and accept accesses on. |

Table 49. 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. 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 50. 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. Enables the SMBus data timeout check. When this bit is set to 1, the clock timeout must also be enabled (<code>smb_to_cnt</code> is not 0). When a data timeout is detected, the <code>smb_d_to</code> bit is set to 1. |
| 5 | smb_to_en | RW | 0x1 | Reserved. |
| 4:0 | smb_to_cnt | RW | 0x19 | SMBus Timeout Duration. Sets the clock and data timeout duration in milliseconds. When this field is set to 0, the clock timeout is disabled and <code>smb_d_to_en</code> must also be set to 0 to disable the data timeout. When a clock timeout is detected, the <code>smb_to</code> bit is set to 1. |

Table 51. 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. 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. <ul style="list-style-type: none"> ▪ 0x0 = 1.8V Standard mode (100 kHz) or 2.5V/3.3V standard (100kHz) and Fast mode (400kHz) ▪ 0x1 = 1.8V Fast mode (400 kHz) ▪ 0x2 = Reserved ▪ 0x3 = 1.8V/2.5V/3.3V Fast mode plus (1MHz) |
| 2 | ssi_addr_size | RW | 0x0 | SSI Address Size. 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. <ul style="list-style-type: none"> ▪ 0x0 = 1-byte address ▪ 0x1 = 2-byte address |
| 1:0 | ssi_enable | RW | 0x1 | SSI Mode. <ul style="list-style-type: none"> ▪ 0x0 = SSI is disabled ▪ 0x1 = SSI is in I2C mode ▪ 0x2 = SSI is in SPI mode ▪ 0x3 = SSI is in SMBus mode |

Table 52. 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. Set when the SMBus interface detects a data timeout. |
| 0 | smb_to | RW1C | 0x0 | SMBus Clock Timeout. Set when the SMBus interface detects a clock timeout. |

11.3 XO Register

Table 53. 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. When cleared, the XO/IB high frequency divider is held in set mode (bit is active low). Unless this divider is bypassed (<code>xo_ib_h_div = 0</code>), clearing this bit will halt the <code>refin_div</code> clock. |
| 28:24 | xo_ib_h_div | RW | 0x0 | XO / Input Buffer High-Frequency Divide Ratio. Divide by 2 to 31. Bypass if set to 0. 1 is reserved. |
| 23:22 | en_gain | RW | 0x1 | XO Gain Boosting Control. Selects the number of gain boosting amplifiers enabled during startup. <ul style="list-style-type: none"> 0x0 = Gain boosting amplifiers are disabled 0x1 = One parallel amplifier is enabled 0x2 = Two parallel amplifiers are enabled 0x3 = All three parallel amplifiers are enabled |
| 21:16 | en_cap_x2 | RW | 0x28 | XO Capacitance at X2 Terminal. 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 <code>sel_ib_xo</code> is set to 0. |
| 15 | reserved | RO | 0x0 | Reserved. |
| 14 | xo_buff_dis | RW | 0x0 | XO Buffer Disable. Forces the XO buffer to the core logic to be disabled. This setting is intended for debug purposes only. <ul style="list-style-type: none"> 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. 0x1 = XO buffer is disabled |
| 13:8 | en_cap_x1 | RW | 0x28 | XO Capacitance at X1 Terminal. 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 <code>sel_ib_xo</code> is set to 0. |
| 7 | reserved | RO | 0x0 | Reserved. |
| 6:4 | xo_res | RW | 0x4 | XO Resistor Configuration. Series resistance array control for limiting driving power level. |
| 3 | xo_ib_en_dc_bias | RW | 0x0 | Input Buffer Internal DC Bias Enable. When the input buffer clock input signal is AC-coupled external to the device, the internal DC bias voltage must be enabled. <ul style="list-style-type: none"> 0x0 = Internal DC bias is disabled (input signal is DC-coupled) 0x1 = Internal DC bias is enabled (input signal is AC-coupled) |
| 2 | sel_ib_xo | RW | 0x1 | XO / Input Buffer Select. Selects the mode of the XO / Input Buffer. <ul style="list-style-type: none"> 0x0 = Input buffer 0x1 = XO |

Table 53. 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. Configures the input buffer for single-ended CMOS or differential input signal. <ul style="list-style-type: none"> 0x0 = Differential input is selected 0x1 = CMOS input is selected |
| 0 | xo_ib_p_n_diff_sel | RW | 0x1 | Input Buffer PMOS / NMOS Select. Configures the input buffer according to the common mode voltage of the provided input signal. <ul style="list-style-type: none"> 0x0 = PMOS input pair is enabled (low common mode voltage) 0x1 = NMOS input pair is enabled (higher common mode voltage) |

11.4 CLKIN Register

Table 54. 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. When cleared, the corresponding reference clock high-frequency divider is held in set mode (bit is active low). Unless this divider is bypassed (<code>h_div = 0</code>), clearing this bit will halt the <code>clkln0_div/clkln2_div</code> clock. |
| 14:10 | h_div | RW | 0x0 | Reference Clock High-Frequency Divide Ratio. Divide by 2 to 31. Bypass if set to 0. 1 is reserved. |
| 9 | trim_term | RW | 0x0 | Reference Clock Input Pad Termination Trim. Selects the HCSL and LVDS termination resistance to ground / across the inputs for modes to: <ul style="list-style-type: none"> 0x0 = 50Ω / 100Ω 0x1 = 42.5Ω / 85Ω |
| 8 | en_lvpecl | RW | 0x0 | Reference Clock Input Pad LVPECL Termination Enable. Enables compatible termination when the reference clock input signal is LVPECL. <ul style="list-style-type: none"> 0x0 = LVPECL input termination is disabled 0x1 = LVPECL input termination is enabled |
| 7 | en_lvds | RW | 0x0 | Reference Clock Input Pad LVDS Termination Enable. Enables compatible termination when the reference clock input signal is LVDS. <ul style="list-style-type: none"> 0x0 = LVDS input termination is disabled 0x1 = LVDS input termination is enabled |
| 6 | en_hcsl | RW | 0x0 | Reference Clock Input Pad HCSL Termination Enable. Enables compatible termination when the reference clock input signal is HCSL. <ul style="list-style-type: none"> 0x0 = HCSL input termination is disabled 0x1 = HCSL input termination is enabled |
| 5 | en_cml | RW | 0x0 | Reference Clock Input Pad CML Termination Enable. Enables compatible termination when the reference clock input signal is CML. <ul style="list-style-type: none"> 0x0 = CML input termination is disabled 0x1 = CML input termination is enabled |

Table 54. 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. When the reference clock input signal is AC-coupled external to the device, the internal DC bias voltage must be enabled. <ul style="list-style-type: none"> 0x0 = Internal DC bias is disabled (input signal is DC-coupled) 0x1 = Internal DC bias is enabled (input signal is AC-coupled) |
| 3 | en_inbuff | RW | 0x0 | Reference Clock Input Pad Enable. 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. <ul style="list-style-type: none"> 0x0 = Input pad is disabled 0x1 = Input pad is enabled |
| 2:1 | cmos_sel | RW | 0x0 | Reference Clock Input Pad CMOS / Differential Select. Configures the reference clock input pad for two single-ended CMOS or differential input signal. <ul style="list-style-type: none"> 0x0 = Differential input is selected 0x1 = Single CMOS input is selected on positive pin 0x2 = Single CMOS input is selected on negative pin 0x3 = Dual CMOS inputs are selected |
| 0 | p_n_diff_sel | RW | 0x1 | Reference Clock Input Pad PMOS / NMOS Select. Configures the reference clock input pad according to the common mode voltage of the provided input signal. <ul style="list-style-type: none"> 0x0 = PMOS input pair is enabled (low common mode voltage) 0x1 = NMOS input pair is enabled (higher common mode voltage) |

11.5 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 55. 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. While the Loss-of-Signal counter (los_cnt) exceeds this threshold, the los_lmt_evt bit is set. |
| 7 | reserved | RO | 0x0 | Reserved. |
| 6 | reserved | RW | 0x0 | Reserved. |
| 5 | los_fail_mask | RW | 0x0 | LOS Monitor Failure Mask. Masks the LOS monitor status contribution to ref_invalid . <ul style="list-style-type: none"> 0x0 = los_sts contributes to ref_invalid 0x1 = los_sts does not contribute to ref_invalid |

Table 55. LOSMON_WINDOW - LOS Monitor Window Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|---------------|------|---------------|--|
| 4:0 | los_div_ratio | RW | 0x0 | <p>LOS Monitor Divide Ratio.</p> <p>This divide ratio must be set such that the monitored clock nominal frequency divided by <code>los_div_ratio</code> is less than 1/8 of the measuring clock frequency (see <code>losmon_clkssel</code>) 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.</p> |

Table 56. LOSMON_NOMINAL - LOS Monitor Nominal Number Configuration

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

Table 57. 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 | <p>LOS Monitor Accept Threshold.</p> <p>An accepted clock monitoring window occurs when the final monitor counter value is within <code>los_nom_num ± los_acc_margin</code>.</p> |
| 15 | reserved | RO | 0x0 | Reserved. |
| 14:0 | los_rej_margin | RW | 0x0 | <p>LOS Monitor Reject Threshold.</p> <p>A rejected clock monitoring window occurs when the final monitor counter value is outside of <code>los_nom_num ± los_rej_margin</code>.</p> |

Table 58. LOSMON_QUAL - LOS Monitor Qualify Counter Configuration

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

Table 59. 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. 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. <ul style="list-style-type: none"> 0x0 = Clock is valid 0x1 = Clock is invalid |
| 0 | los_sts | RO | 0x1 | Loss-of-Signal Status. Current value of the LOS status from the clock monitor: <ul style="list-style-type: none"> 0x0 = Clock meets the monitoring criteria 0x1 = Loss-of-signal detected |

Table 60. 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. 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. <ul style="list-style-type: none"> 0x0 = Loss-of-signal counter has not exceeded the threshold since the last time the bit was cleared 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. 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. <ul style="list-style-type: none"> 0x0 = Loss-of-signal not detected since the last time the bit was cleared 0x1 = Loss-of-signal detected since the last time the bit was cleared |

Table 61. 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. 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. 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). |

11.6 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 62. FREQMON_WINDOW - Frequency Monitor Window Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|----------------|------|---------------|---|
| 31 | freq_fail_mask | RW | 0x0 | Frequency Monitor Failure Mask. Masks the frequency monitor status contribution to <code>ref_invalid</code> . <ul style="list-style-type: none"> 0x0 = <code>freq_sts</code> contributes to <code>ref_invalid</code> 0x1 = <code>freq_sts</code> does not contribute to <code>ref_invalid</code> |
| 30:28 | reserved | RO | 0x0 | Reserved. |
| 27:0 | freq_div_ratio | RW | 0x0 | Frequency Monitor Divide Ratio. This divide ratio must be set such that the monitored clock nominal frequency divided by <code>freq_div_ratio</code> 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 63. 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. 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 <code>freq_sts</code> bit to be asserted, therefore the <code>freq_fail_mask</code> bit should also be set when this field is written to 0x0. |

Table 64. 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. An accepted clock monitoring window occurs when the final monitor counter value is within <code>freq_nom_num ± freq_acc_margin</code> . One accepted window qualifies the clock and <code>freq_sts</code> is set to 0. |
| 31:27 | reserved | RO | 0x0 | Reserved. |
| 26:0 | freq_rej_margin | RW | 0x0 | Frequency Monitor Reject Threshold. A rejected clock monitoring window occurs when the final monitor counter value is outside of <code>freq_nom_num ± freq_rej_margin</code> . One rejected window disqualifies the clock and <code>freq_sts</code> is set to 1. |

Table 65. 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. Current value of the qualification status from the frequency monitor: <ul style="list-style-type: none"> 0x0 = Clock meets the monitoring criteria, clock qualified 0x1 = Failure detected, clock disqualified |

Table 66. 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 | <p>Frequency Monitor Event Status.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = Frequency monitor has not disqualified the clock since the last time the bit was cleared ▪ 0x1 = Frequency monitor has disqualified the clock since the last time the bit was cleared |

Table 67. 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 | 0x7FFFFFFF | <p>Frequency Count Status.</p> <p>Cycle count measured by the frequency monitor, updated at the end of each monitoring window. The count saturates to 0x7FF_FFFF when it reaches $\text{freq_nom_num} \times 2$, 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:</p> <p>ppm offset = $1e6 \times (\text{freq_nom_num} - \text{freq_offset_sts}) / \text{freq_offset_sts}$</p> |

11.7 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 68. TDC_APLL_CNFG - TDC APLL Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-------------|------|---------------|---|
| 15 | tdc_en | RW | 0x0 | <p>TDC Enable.</p> <p>Controls whether the TDC is enabled. Must be enabled when enabling a reference clock LOS monitor when <code>losmon_clkssel</code> is set to 1, and when enabling a reference clock frequency monitor.</p> |
| 14 | reserved | RO | 0x0 | Reserved. |
| 13:9 | tdc_kp_coef | RW | 0xF | <p>TDC APLL Proportional Filter Coefficient (Kp).</p> <p>The filter output is (up/down) * 2^{Kp} + integrator.</p> |
| 8:4 | tdc_ki_coef | RW | 0x4 | <p>TDC APLL Integral Filter Coefficient (Ki).</p> <p>The integrator is (up/down) * 2^{Ki} + integrator.</p> |

Table 68. TDC_APLL_CNFG - TDC APLL Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-------------------|------|---------------|--|
| 3:2 | tdc_fast_lock | RW | 0x2 | TDC APLL Fast Lock Mode. Controls the TDC APLL lock speed by increasing the effective Ki values while the input up/down is railed. <ul style="list-style-type: none"> ▪ 0x0 = Normal lock. Ki as programmed. ▪ 0x1 = Ki increased by 2 ▪ 0x2 = Ki increased by 4 ▪ 0x3 = Ki increased by 6 |
| 1:0 | tdc_dac_sdm_order | RW | 0x2 | TDC APLL DAC SDM (Sigma Delta Modulator) Order. Selects the order of the SDM controlling the DAC for the TDC APLL. <ul style="list-style-type: none"> ▪ 0x0 = No accumulation ▪ 0x1 = 1st order ▪ 0x2 = 2nd order ▪ 0x3 = Reserved |

Table 69. 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. Fraction of the TDC APLL feedback divider. The fraction is calculated as follows: $tdc_fb_div_frac/2^{16}$. |

Table 70. 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. Integer portion of the TDC APLL feedback divider. |

Table 71. 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. Enables the SDM controlling the TDC APLL feedback divider. <ul style="list-style-type: none"> ▪ 0x0 = SDM disabled, constant integer division by tdc_fb_div_int ▪ 0x1 = SDM enabled, MMD mode |
| 6:2 | reserved | RO | 0x0 | Reserved. |
| 1:0 | tdc_fb_sdm_order | RW | 0x1 | TDC APLL Feedback SDM Order. Selects the order of the SDM controlling the feedback divider for the TDC APLL. <ul style="list-style-type: none"> ▪ 0x0 = Integer ▪ 0x1 = 1st order ▪ 0x2 = 2nd order ▪ 0x3 = 3rd order |

Table 72. 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. 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. <ul style="list-style-type: none"> ▪ 0x0 = Bypass divider ▪ 0x1 = Divide by 2 ▪ 0x2 = Divide by 4 ▪ 0x3 = Divide by 8 ▪ 0x4 = Divide by 16 |

Table 73. 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. Provides the TDC loop filter output value. It is a signed 16-bit number. |

11.8 APLL Registers

Table 74. 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. APLL feedback divider numerator value. The denominator is a fixed value of 2^{27} . |

Table 75. 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. APLL feedback divider integer value. |

Table 76. 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. <ul style="list-style-type: none"> ▪ Dither enable for the SDM controlling the APLL feedback divider. 0x0 = Dither disabled ▪ 0x1 = Dither enabled |

Table 76. APLL_FB_SDM_CNFG - APLL Feedback SDM Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------------|------|---------------|---|
| 4 | apl_fb_dither_ns | RW | 0x0 | APLL Feedback SDM Dither Noise Shaping. <ul style="list-style-type: none"> ▪ Dither noise shaping enable for the SDM controlling the APLL feedback divider. 0x0 = Dither not shaped ▪ 0x1 = Dither shaped |
| 3:2 | apl_fb_dither_gain | RW | 0x0 | APLL Feedback SDM Dither Gain. <ul style="list-style-type: none"> ▪ Gain control for the SDM controlling the APLL feedback divider. 0x0 = LSB ▪ 0x1 = 2*LSB ▪ 0x2 = 4*LSB ▪ 0x3 = 8*LSB |
| 1:0 | apl_fb_sdm_order | RW | 0x3 | APLL Feedback SDM Order. <ul style="list-style-type: none"> ▪ Selects the order of the SDM controlling the feedback divider for the APLL. When the feedback divide ratio is fractional, the order must be set greater than 0. 0x0 = Integer ▪ 0x1 = 1st order ▪ 0x2 = 2nd order ▪ 0x3 = 3rd order |

Table 77. 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. Selects the frequency doubler to use when en_doubler is set to 1. <ul style="list-style-type: none"> ▪ 0x0 = PLL frequency doubler ▪ 0x1 = PFD frequency doubler |
| 0 | en_doubler | RW | 0x1 | Frequency Doubler Enable. Enables the frequency doubler selected by the doubler_sel bit. <ul style="list-style-type: none"> ▪ 0x0 = Disable ▪ 0x1 = Enable |

Table 78. 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. Enables the charge pump offset current. The magnitude of the current is selected by cnf_cp_offset . <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 12 | cp_offset_boost | RW | 0x0 | Charge Pump Offset Current Boost. Increases the magnitude of the offset current. <ul style="list-style-type: none"> ▪ 0x0 = Charge pump offset range from 0uA to 145uA ▪ 0x1 = Charge pump offset range from 0uA to 236uA |

Table 78. CP_CNFG - APLL Charge Pump Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|---------------|------|---------------|---|
| 11:8 | cnf_cp_offset | RW | 0x3 | <p>Charge Pump Offset Current Setting.</p> <p>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:</p> <ul style="list-style-type: none"> ▪ 0x0 = 0uA / 0uA ▪ 0x1 = 10.3uA / 99.7uA ▪ 0x2 = 20.5uA / 118.7uA ▪ 0x3 = 30.6uA / 128.1uA ▪ 0x4 = 40.7uA / 137.5uA ▪ 0x5 = 50.5uA / 146.7uA ▪ 0x6 = 60.3uA / 155.9uA ▪ 0x7 = 70.0uA / 165.1uA ▪ 0x8 = 79.7uA / 174.2uA ▪ 0x9 = 89.2uA / 183.2uA ▪ 0xA = 98.7uA / 192.2uA ▪ 0xB = 108.1uA / 201.2uA ▪ 0xC = 117.5uA / 210.0uA ▪ 0xD = 126.7uA / 218.9uA ▪ 0xE = 135.9uA / 227.7uA ▪ 0xF = 145.0uA / 236.0uA |
| 7 | reserved | RO | 0x0 | Reserved. |
| 6:4 | cnf_cp_up | RW | 0x3 | <p>Charge Pump Up Current Setting.</p> <ul style="list-style-type: none"> ▪ 0x0 = 250uA ▪ 0x1 = 500uA ▪ 0x2 = 750uA ▪ 0x3 = 1mA ▪ 0x4 = 1.25mA ▪ 0x5 = 1.5mA ▪ 0x6 = 1.75mA ▪ 0x7 = 2mA |
| 3 | reserved | RO | 0x0 | Reserved. |
| 2:0 | cnf_cp_dn | RW | 0x3 | <p>Charge Pump Down Current Setting.</p> <ul style="list-style-type: none"> ▪ 0x0 = 250uA ▪ 0x1 = 500uA ▪ 0x2 = 750uA ▪ 0x3 = 1mA ▪ 0x4 = 1.25mA ▪ 0x5 = 1.5mA ▪ 0x6 = 1.75mA ▪ 0x7 = 2mA |

Table 79. 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. <ul style="list-style-type: none"> ▪ 0x0 = Filter active ▪ 0x1 = Filter bypassed |
| 6:4 | cnf_lpf_cp | RW | 0x7 | Loop Filter Pole Capacitor Setting. <ul style="list-style-type: none"> ▪ 0x0 = 11pF ▪ 0x1 = 14.7pF ▪ 0x2 = 18.4pF ▪ 0x3 = 22.1pF ▪ 0x4 = 25.8pF ▪ 0x5 = 29.5pF ▪ 0x6 = 33.2pF ▪ 0x7 = 36.9pF |
| 3:0 | cnf_lpf_res | RW | 0x4 | Loop Filter Resistor Setting. <ul style="list-style-type: none"> ▪ 0x0 = 0Ω ▪ 0x1 = 400Ω ▪ 0x2 = 800Ω ▪ 0x3 = 1.2kΩ ▪ 0x4 = 1.6kΩ ▪ 0x5 = 2kΩ ▪ 0x6 = 2.4kΩ ▪ 0x7 = 2.8kΩ ▪ 0x8 = 3.2kΩ ▪ 0x9 = 3.6kΩ ▪ 0xA = 4kΩ ▪ 0xB = 4.4kΩ ▪ 0xC = 4.8kΩ ▪ 0xD = 5.2kΩ ▪ 0xE = 5.6kΩ ▪ 0xF = 6kΩ |

Table 80. 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. This bit can only be set to 1 when operating with an integer feedback divider. <ul style="list-style-type: none"> ▪ 0x0 = 3rd pole active ▪ 0x1 = 3rd pole bypassed |
| 6:4 | cnf_lpf_R3 | RW | 0x4 | Loop Filter 3rd Pole Resistor Setting. <ul style="list-style-type: none"> ▪ 0x0 = 0Ω ▪ 0x1 = 800Ω ▪ 0x2 = 1.6kΩ ▪ 0x3 = 2.4kΩ ▪ 0x4 = 3.2kΩ ▪ 0x5 = 4kΩ ▪ 0x6 = 4.8kΩ ▪ 0x7 = 5.6kΩ |

Table 80. LPF_3RD_CNFG - APLL Loop Filter 3rd Pole Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 3 | reserved | RO | 0x0 | Reserved. |
| 2:0 | cnf_lpf_C3 | RW | 0x4 | <p>Loop Filter 3rd Pole Capacitor Setting.</p> <ul style="list-style-type: none"> ▪ 0x0 = 0pF ▪ 0x1 = 1pF ▪ 0x2 = 2pF ▪ 0x3 = 3pF ▪ 0x4 = 4pF ▪ 0x5 = 5pF ▪ 0x6 = 6pF ▪ 0x7 = 7pF |

Table 81. 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 | <p>APLL Loss-of-Lock Counter Threshold.</p> <p>While the APLL Loss-of-Lock counter (apll_lol_cnt) exceeds this threshold, the apll_lol_lmt bit is set.</p> |
| 3 | apll_fb_div_dis | RW | 0x0 | <p>APLL Feedback Divider Disable.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = APLL feedback divider enabled ▪ 0x1 = APLL feedback divider disabled |
| 2 | apll_fb_sel | RW | 0x0 | <p>APLL Feedback Clock Selection.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = Internal from APLL feedback MMD ▪ 0x1 = External from CLKIN0 – ZDB mode |
| 1:0 | apll_ref_sel | RW | 0x0 | <p>APLL Reference Clock Selection.</p> <ul style="list-style-type: none"> ▪ 0x0 = XIN_REFIN (refin_div) ▪ 0x1 = CLKIN0 (clk0_div) ▪ 0x2 = CLKIN1 (clk1_div) ▪ 0x3 = VSS |

Table 82. 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. Enables fanout of the XO/IB clock to Bank 5 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 30 | en_clkin2_bnk6 | RW | 0x1 | Reference Clock 2 to Bank 6 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 29 | en_clkin2_bnk5 | RW | 0x1 | Reference Clock 2 to Bank 5 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 28 | en_clkin2_bnk4 | RW | 0x1 | Reference Clock 2 to Bank 4 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 27 | en_clkin0_bnk6 | RW | 0x1 | Reference Clock 0 to Bank 6 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 26 | en_clkin0_bnk5 | RW | 0x1 | Reference Clock 0 to Bank 5 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 25 | en_clkin0_bnk4 | RW | 0x1 | Reference Clock 0 to Bank 4 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |

Table 82. BANK_MUX_CLK_EN - Bank Mux Clock Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------|------|---------------|---|
| 24 | en_fod2_bnk6 | RW | 0x1 | FOD 2 to Bank 6 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 23 | en_fod2_bnk5 | RW | 0x1 | FOD 2 to Bank 5 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 22 | en_fod2_bnk4 | RW | 0x1 | FOD 2 to Bank 4 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 21 | en_fod2_bnk3 | RW | 0x1 | FOD 2 to Bank 3 Output Mux Enable. Enables fanout of FOD 2 output clock to Bank 3 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 20 | en_fod1_bnk6 | RW | 0x1 | FOD 1 to Bank 6 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 19 | en_fod1_bnk5 | RW | 0x1 | FOD 1 to Bank 5 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 18 | en_fod1_bnk4 | RW | 0x1 | FOD 1 to Bank 4 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 17 | en_fod1_bnk3 | RW | 0x1 | FOD 1 to Bank 3 Output Mux Enable. Enables fanout of FOD 1 output clock to Bank 3 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |

Table 82. BANK_MUX_CLK_EN - Bank Mux Clock Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------|------|---------------|---|
| 16 | en_fod1_bnk2 | RW | 0x1 | FOD 1 to Bank 2 Output Mux Enable. Enables fanout of FOD 1 output clock to Bank 2 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 15 | en_fod1_bnk1 | RW | 0x1 | FOD 1 to Bank 1 Output Mux Enable. Enables fanout of FOD 1 output clock to Bank 1 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 14 | en_fod1_bnk0 | RW | 0x1 | FOD 1 to Bank 0 Output Mux Enable. Enables fanout of FOD 1 output clock to Bank 0 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 13 | en_fod0_bnk3 | RW | 0x1 | FOD 0 to Bank 3 Output Mux Enable. Enables fanout of FOD 0 output clock to Bank 3 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 12 | en_fod0_bnk2 | RW | 0x1 | FOD 0 to Bank 2 Output Mux Enable. Enables fanout of FOD 0 output clock to Bank 2 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 11 | en_fod0_bnk1 | RW | 0x1 | FOD 0 to Bank 1 Output Mux Enable. Enables fanout of FOD 0 output clock to Bank 1 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 10 | en_fod0_bnk0 | RW | 0x1 | FOD 0 to Bank 0 Output Mux Enable. Enables fanout of FOD 0 output clock to Bank 0 output mux. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 9 | en_iod3_bnk6 | RW | 0x1 | IOD 3 to Bank 6 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 8 | en_iod3_bnk5 | RW | 0x1 | IOD 3 to Bank 5 Output Mux Enable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |

Table 82. BANK_MUX_CLK_EN - Bank Mux Clock Enable (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------|------|---------------|---|
| 7 | en_iod2_bnk6 | RW | 0x1 | <p>IOD 2 to Bank 6 Output Mux Enable.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 6 | en_iod2_bnk5 | RW | 0x1 | <p>IOD 2 to Bank 5 Output Mux Enable.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 5 | en_iod2_bnk4 | RW | 0x1 | <p>IOD 2 to Bank 4 Output Mux Enable.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 4 | en_iod1_bnk2 | RW | 0x1 | <p>IOD 1 to Bank 2 Output Mux Enable.</p> <p>Enables fanout of IOD 1 output clock to Bank 2 output mux.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 3 | en_iod1_bnk1 | RW | 0x1 | <p>IOD 1 to Bank 1 Output Mux Enable.</p> <p>Enables fanout of IOD 1 output clock to Bank 1 output mux.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 2 | en_iod1_bnk0 | RW | 0x1 | <p>IOD 1 to Bank 0 Output Mux Enable.</p> <p>Enables fanout of IOD 1 output clock to Bank 0 output mux.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 1 | en_iod0_bnk1 | RW | 0x1 | <p>IOD 0 to Bank 1 Output Mux Enable.</p> <p>Enables fanout of IOD 0 output clock to Bank 1 output mux.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |
| 0 | en_iod0_bnk0 | RW | 0x1 | <p>IOD 0 to Bank 0 Output Mux Enable.</p> <p>Enables fanout of IOD 0 output clock to Bank 0 output mux.</p> <ul style="list-style-type: none"> ▪ 0x0 = Disabled ▪ 0x1 = Enabled |

Table 83. 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. Set to 1 when the APLL is locked to its reference. <ul style="list-style-type: none"> ▪ 0x0 = Unlocked ▪ 0x1 = Locked |

Table 84. 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. 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. 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. |
| 1 | apll_lol_lmt | RW1C | 0x0 | APLL Loss-of-Lock Counter Threshold Exceeded Status. 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. <ul style="list-style-type: none"> ▪ 0x0 = Loss-of-lock counter has not exceeded the threshold since the last time the bit was cleared ▪ 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. Set to 1 when the APLL lock status transitions from locked to unlocked. |

Table 85. 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. 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 86. ANA_SPARE_CNFG - Analog Spare Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------------|------|---------------|--|
| 15 | regulator_dis | RW | 0x0 | Regulator Disable. 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. <ul style="list-style-type: none"> ▪ 0x0 = Regulators listed above are enabled. ▪ 0x1 = Regulators listed above are disabled. |
| 14 | ana_spare | RW | 0x0 | Reserved |
| 13 | sense_mode_sel | RW | 0x0 | One-shot VDDO Sensing. 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. 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. 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. |
| 10 | BNK6_pwrsel | RW | 0x0 | Bank6 Regulator Select. This bit controls whether to enable or bypass the 1.8V regulator in the output driver. <ul style="list-style-type: none"> ▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator. ▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 9 | BNK5_pwrsel | RW | 0x0 | Bank5 Regulator Select. This bit controls whether to enable or bypass the 1.8V regulator in the output driver. <ul style="list-style-type: none"> ▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator. ▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 8 | BNK4_pwrsel | RW | 0x0 | Bank4 Regulator Select. This bit controls whether to enable or bypass the 1.8V regulator in the output driver. <ul style="list-style-type: none"> ▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator. ▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 7 | BNK3_pwrsel | RW | 0x0 | Bank3 Regulator Select. This bit controls whether to enable or bypass the 1.8V regulator in the output driver. <ul style="list-style-type: none"> ▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator. ▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 6 | BNK2_pwrsel | RW | 0x0 | Bank2 Regulator Select. This bit controls whether to enable or bypass the 1.8V regulator in the output driver. <ul style="list-style-type: none"> ▪ 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator. ▪ 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |

Table 86. ANA_SPARE_CNFG - Analog Spare Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|--------------|------|---------------|--|
| 5 | BNK1_pwrssel | RW | 0x0 | Bank1 Regulator Select. This bit controls whether to enable or bypass the 1.8V regulator in the output driver. <ul style="list-style-type: none"> 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator. 0x1 = Used when VDDO = 1.8V to bypass the 1.8V regulator. |
| 4 | BNK0_pwrssel | RW | 0x0 | Bank0 Regulator Select. This bit controls whether to enable or bypass the 1.8V regulator in the output driver. <ul style="list-style-type: none"> 0x0 = Used when VDDO = 2.5V or 3.3V. Enables the 1.8V voltage regulator. 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 87. ANA_SPARE_STS - Analog Spare Status

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

11.9 IOD Registers

Table 88. 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. 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. <ul style="list-style-type: none"> 0x0 = Divider is powered up 0x1 = Divider is powered down |
| 30 | iod_dis | RW | 0x0 | Integer Output Divider Disable. 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. <ul style="list-style-type: none"> 0x0 = Divider enabled 0x1 = Divider disabled |
| 29 | iod_rst | RW | 0x0 | Integer Output Divider Reset. Resets the integer output divider. <ul style="list-style-type: none"> 0x0 = Divider reset de-asserted 0x1 = Divider reset asserted |
| 28 | iod_squelch | RW | 0x0 | Integer Output Divider Squelch. Synchronously squelches and releases the IOD output clock. <ul style="list-style-type: none"> 0x0 = IOD output is not squelched 0x1 = IOD output is squelched |

Table 88. IOD_INT_CNFG - IOD Integer Ratio Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 27:25 | reserved | RO | 0x0 | Reserved. |
| 24:0 | iod_int | RW | 0x64 | Integer Output Divider Ratio Integer output divider ratio. The minimum value is 14. This register is atomic. When the most significant byte (bit [24]) is written, the new value is applied to the IOD. |

Table 89. 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. 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. 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. |
| 10:9 | iod_sync_group | RW | 0x0 | Integer Output Divider Sync Group. Sets the sync group that this divider belongs to <ul style="list-style-type: none"> ▪ 0x0 = Group0 ▪ 0x1 = Group1 ▪ 0x2 = None ▪ 0x3 = None |
| 8:0 | iod_phase | RW | 0x0 | Integer Output Divider Phase Configuration. 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. The available range is $\pm 0\sim 255$ steps (approximately $\pm 0\sim 20$ ns). 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 . |

11.10 FOD Registers

Table 90. FOD_INT_CNFG - FOD Integer Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|---------------|------|---------------|---|
| 63 | fod_pd | RW | 0x0 | Fractional Output Divider Power Down. 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. <ul style="list-style-type: none"> 0x0 = Divider is powered up 0x1 = Divider is powered down |
| 62 | fod_dis | RW | 0x0 | Fractional Output Divider Disable. 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. <ul style="list-style-type: none"> 0x0 = Divider enabled 0x1 = Divider disabled |
| 61 | fod_rst | RW | 0x0 | Fractional Output Divider Reset. Resets the fractional output divider. <ul style="list-style-type: none"> 0x0 = Divider reset de-asserted 0x1 = Divider reset asserted |
| 60 | fod_acc_reset | RW | 0x0 | Fractional Output Divider Accumulator Reset. Resets the FOD SDM accumulator. <ul style="list-style-type: none"> 0x0 = Accumulator reset de-asserted 0x1 = Accumulator reset asserted |
| 59:26 | fod_frac | RW | 0x0 | Fractional Output Divider Ratio Fraction Portion. Denominator is fixed to 2^{34} . 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. <i>Note:</i> When an FOD has spread-spectrum fod_frac [5:0] <i>must</i> be set to 0x0. |
| 25:9 | fod_2nd_int | RW | 0x0 | Fractional Output Divider Ratio 2nd Integer Portion. 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. Integer divide ratio of first stage (MMD). 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 If the first stage frequency is less than 70MHz, the fod_slow_freq_en bit must be set to 1. 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. <i>Note:</i> Refer to fod_frac for details about serial bus writes to this register field and automatic updates performed by the over-clocking engine. |

Table 91. 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. 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. 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. 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. Sets the sync group that this divider belongs to: <ul style="list-style-type: none"> ▪ 0x0 = Group0 ▪ 0x1 = Group1 ▪ 0x2 = None ▪ 0x3 = None |
| 9:0 | fod_phase | RW | 0x3F0 | Fraction Output Divider Phase Configuration. 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. 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. 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 . |

11.11 SSC Register

Table 92. SSC_CNFG - Spectrum Spreading Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 31 | ssc_en | RW | 0x0 | <p>Spread-Spectrum Clocking Enable.</p> <p>Enable spread spectrum. The spread configuration is determined by the other register fields in this register.</p> <ul style="list-style-type: none"> 0x0 = SSC disabled 0x1 = SSC enabled <p>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.</p> <p><i>Note:</i> 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.</p> |
| 30 | ssc_mode | RW | 0x0 | <p>Spectrum Spreading Mode.</p> <ul style="list-style-type: none"> 0x0 = Down spreading 0x1 = Center spreading |
| 29:24 | reserved | RO | 0x0 | Reserved. |
| 23:16 | ssc_ampl | RW | 0x51 | <p>Spectrum Spreading Amplitude.</p> <p>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%).</p> $\text{ssc_ampl} = \text{spread_percentage} / 100 * 2^{14}$ <p>For example, for 1% spread, set ssc_ampl to $0.01 * 2^{14} = 163$ decimal, or 0xA3.</p> <p>The default value corresponds to a 0.5% down spread at 31.5kHz.</p> |
| 15:0 | ssc_step | RW | 0x2B8C | <p>Spectrum Spreading Step Size.</p> <p>Set ramp step size to get the target modulation rate.</p> <p>For down spread:</p> <ul style="list-style-type: none"> $\text{ssc_step} = \text{ssc_ampl} * 2^{16} * \text{ssc_freq} / 15\text{MHz}$ <p>For center spread:</p> <ul style="list-style-type: none"> $\text{ssc_step} = 2 * \text{ssc_ampl} * 2^{16} * \text{ssc_freq} / 15\text{MHz}$ <p>where:</p> <p>ssc_freq is the target modulation rate from 30kHz to 63kHz 15MHz is the system clock divided by 4, assuming the system clock is 60MHz</p> <p><i>Example 1.</i> For a 32kHz 1% down spread: ssc_ampl = 163 $\text{ssc_step} = 163 * 2^{16} * 32\text{kHz} / 15\text{MHz} = 0x5905$</p> <p><i>Example 2.</i> For a 32kHz ± 0.5% center spread: ssc_ampl = 81 $\text{ssc_step} = 2 * 81 * 2^{16} * 32\text{kHz} / 15\text{MHz} = 0x5879$</p> <p>The default value corresponds to a 0.5% down spread at 31.5kHz.</p> |

11.12 BANK Register

Table 93. 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 | <p>Output Bank Power Down.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = Bank is powered up ▪ 0x1 = Bank is powered down |
| 4:3 | bank_fanout_mode | RW | 0x0 | <p>Output Bank Fan-out Buffer Mode.</p> <p>Configures the bank selection for fanout buffer mode. Only available on banks 4, 5, and 6.</p> <p>The device must be configured for fanout buffer mode by fanout_buf_mode (CLKIN0) / fanout_buf_mode1 (CLKIN1) for this setting to take effect.</p> <ul style="list-style-type: none"> ▪ 0x0 = Normal operation mode. The output bank source clock is selected by output_bank_src. ▪ 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. ▪ 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 | <p>Output Bank Source.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = IOD0 for Banks 0, 1 / clkin2_div (CLKIN1) for Banks 4, 5, 6 ▪ 0x1 = IOD1 for Banks 0, 1, 2 / refin_div (XIN_REF) for Bank 5 ▪ 0x2 = IOD2 for Banks 4, 5, 6 ▪ 0x3 = IOD3 for Banks 5, 6 ▪ 0x4 = FOD0 for Banks 0, 1, 2, 3 ▪ 0x5 = FOD1 ▪ 0x6 = FOD2 for Banks 3, 4, 5, 6 ▪ 0x7 = clkin0_div (CLKIN0) for Banks 4, 5, 6 |

11.13 OUT Registers

Table 94. ODRV_EN - Output Driver Enable

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|---------------|------|---------------|---|
| 7 | out_pd | RW | 0x0 | Output Driver Power Down. Powers down the output clock driver. <ul style="list-style-type: none"> 0x0 = Output driver is powered up 0x1 = Output driver is powered down |
| 6 | out_oe_mode | RW | 0x0 | Output Driver OE Mode. 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. <ul style="list-style-type: none"> 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. 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. 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. <ul style="list-style-type: none"> 0x0 = Group0 0x1 = Group1 0x2 = Group2 0x3 = Group3 0x4 = Group4 0x5 = None 0x7 = None, and exclude global output enables When set to 0x7, allows the output clock to be enabled regardless of: <ul style="list-style-type: none"> APLL lock status (equivalent to out_startup set to 2) goe register bit GOE pin, if one is assigned with gpi_func/gpio_func The output clock can still be disabled by the out_dis register bit. 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 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 will re-align the feedback clock to the reference clock. |
| 2:1 | out_dis_state | RW | 0x3 | OUT Driver Disabled State. Controls the state of OUTx / OUTxb when the output driver is disabled. <ul style="list-style-type: none"> 0x0 = Held High / Low 0x1 = Held Low / High 0x2 = Held Hi-Z / Hi-Z 0x3 = Held Low / Low (except LVDS mode is held High / Low) |
| 0 | out_dis | RW | 0x0 | OUTx and/or OUTxb Driver Disable. 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 . <ul style="list-style-type: none"> 0x0 = OUTx and/or OUTxb Driver is enabled if not disabled by other means 0x1 = OUTx and/or OUTxb Driver is disabled |

Table 95. ODRV_CNFG - Output Driver Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 15 | out_prog7 | RW | 0x0 | Output Driver Programmability Bit 7. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: reserved / flip output polarity / flip output polarity |
| 14 | out_prog6 | RW | 0x0 | Output Driver Programmability Bit 6. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: reserved / reserved / reserved |
| 13 | out_prog5 | RW | 0x0 | Output Driver Programmability Bit 5. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: reserved / cross point lower / cross point tune |
| 12 | out_prog4 | RW | 0x0 | Output Driver Programmability Bit 4. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: reserved / cross point increase for double termination / cross point tune |
| 11 | out_prog3 | RW | 0x0 | Output Driver Programmability Bit 3. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: tristate OUTx / driver impedance -5% / cross point tune |
| 10 | out_prog2 | RW | 0x0 | Output Driver Programmability Bit 2. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: tristate OUTxb / driver impedance +5% / cross point tune |
| 9 | out_prog1 | RW | 0x0 | Output Driver Programmability Bit 1. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: flip OUTx polarity / amplitude -10% / amplitude -10% |
| 8 | out_prog0 | RW | 0x0 | Output Driver Programmability Bit 0. When set to 1, controls the output driver as follows according to CMOS mode / LPHCSL mode / LVDS mode: flip OUTxb polarity / amplitude +5% / amplitude +5% |
| 7 | reserved | RO | 0x0 | Reserved. |
| 6 | out_lpamp | RW | 0x0 | Output Driver LPHCSL Amplitude Control. Controls the amplitude of the output driver when LPHCSL mode is selected. <ul style="list-style-type: none"> ▪ 0x0 = 800mV ▪ 0x1 = 900mV |
| 5 | out_lpsr | RW | 0x1 | Output Driver LPHCSL Slew Rate Control. 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. <ul style="list-style-type: none"> ▪ 0x0 = Slow, 2-4 V/ns ▪ 0x1 = Fast, > 4 V/ns |

Table 95. ODRV_CNFG - Output Driver Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|--|
| 4 | out_lpimp | RW | 0x1 | Output Driver LPHCSL Impedance Control. Controls the output impedance of the output driver when LPHCSL mode is selected. <ul style="list-style-type: none"> 0x0 = 85Ω 0x1 = 100Ω |
| 3:2 | out_cmdrv | RW | 0x3 | Output Driver CMOS Slew Rate Control. 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: <ul style="list-style-type: none"> 0x0 = 4.2 / 2.7 / 1.8 0x1 = 2.2 / 1.5 / 1.8 0x2 = 2.2 / 1.5 / 1.8 0x3 = 3.4 / 2.0 / 1.9 |
| 1:0 | out_mode | RW | 0x0 | Output Driver Type. Selects the output driver type. <ul style="list-style-type: none"> 0x0 = LPHCSL 0x1 = LVDS 0x2 = LVDS 0x3 = CMOS |

11.14 GPI Registers

Table 96. 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 Sets the pin polarity. This bit is ignored if gpi_func configures the pin as a reference clock input. <ul style="list-style-type: none"> 0x0 = Normal sense. Pin functions denoted with a # are active low, others are active high. 0x1 = Inverted sense. Pin functions denoted with a # are active high, others are active low. |
| 8 | gpi_pullup | RW | 0x0 | GPI Pull-up Enable 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. <ul style="list-style-type: none"> 0x0 = Pull-up disabled 0x1 = Pull-up enabled |

Table 96. GPI_CNFG - GPI Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 7 | gpi_pulldn | RW | 0x0 | <p>GPI Pull-down Enable.</p> <p>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.</p> <ul style="list-style-type: none"> ▪ 0x0 = Pull-down disabled ▪ 0x1 = Pull-down enabled |
| 6:0 | gpi_func | RW | 0x7F | <p>GPI Functions.</p> <p>Sets the general purpose input function.</p> <ul style="list-style-type: none"> ▪ 0x0 = OE[0], input, enable output drivers in OE group 0 ▪ 0x1 = OE[1], input, enable output drivers in OE group 1 ▪ 0x2 = OE[2], input, enable output drivers in OE group 2 ▪ 0x3 = OE[3], input, enable output drivers in OE group 3 ▪ 0x4 = OE[4], input, enable output drivers in OE group 4 ▪ 0x5 = PERST#, input, latches CLKIN0 fanout buffer mode clock selection on active edge ▪ 0x6 = GOE, input, enable all output drivers ▪ 0x9 = PERST1#, input, latches XO/IB fanout buffer mode clock selection on active edge ▪ 0xA = Reserved ▪ 0xE = SDI, input (SPI 4-wire mode) ▪ 0xF = SYSREF_IN ▪ 0x10 = GPI, input, input status allowed read back via SSI ▪ 0x7F = GPI function disabled, pin used as reference clock input |

Table 97. GPI_STS - GPI Status

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | gpi_sts | RO | 0x0 | <p>GPI Status.</p> <p>Indicates the status of the GPIO/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.</p> <p>Bit [3] = GPI3 status Bit [2] = GPI2 status Bit [1] = GPI1 status Bit [0] = GPIO status</p> |

11.15 GPIO Registers

Table 98. 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. 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. Sets the value to drive the GPO pin when configured as a general purpose output. |
| 11:10 | gpio_type | RW | 0x1 | GPIO Type. Sets the direction and type following reset. <ul style="list-style-type: none"> ▪ 0x0 = Output (driven high/low), or bidirectional if configured as SDIO by gpio_func ▪ 0x1 = Input (2-level) ▪ 0x2 = Output (open-drain) ▪ 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. 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. <ul style="list-style-type: none"> ▪ 0x0 = Normal sense. Pin functions denoted with a # are active low, others are active high. ▪ 0x1 = Inverted sense. Pin functions denoted with a # are active high, others are active low. |
| 8 | gpio_pullup | RW | 0x0 | GPIO Pull-up Enable. Sets the internal pull-up mode. <ul style="list-style-type: none"> ▪ 0x0 = Pull-up disabled ▪ 0x1 = Pull-up enabled |
| 7 | gpio_pulldn | RW | 0x1 | GPIO Pull-down Enable. Sets the internal pull-down mode. <ul style="list-style-type: none"> ▪ 0x0 = Pull-down disabled ▪ 0x1 = Pull-down enabled |

Table 98. GPIO_CNFG - GPIO Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 6:0 | gpio_func | RW | 0x10 | <p>GPIO Functions.</p> <p>Sets the general purpose input/output function. Refer to pwrn_sel for PWRGD/PWRDN# assignment.</p> <p>0x0 = OE[0], input, enable output drivers in OE group 0 0x1 = OE[1], input, enable output drivers in OE group 1 0x2 = OE[2], input, enable output drivers in OE group 2 0x3 = OE[3], input, enable output drivers in OE group 3 0x4 = OE[4], input, enable output drivers in OE group 4 0x5 = PERST#, input, latches CLKIN0 fanout buffer mode clock selection on active edge 0x6 = GOE, input, enable all output drivers 0x9 = PERST1#, input, latches CLKIN1 fanout buffer mode clock selection on active edge 0xE = SDI, input (SPI 4-wire mode) 0xF = SYSREF_IN, input 0x10 = GPI, input, input status allowed read back via SSI 0x11 = Dynamic CSEL, input, dynamic configuration control. Only valid for GPIO0/1/2. May be tri-level or 2-level. 0x1C = INT, output 0x1D = GPO, output, to control external functions such as LEDs. The output value is set in gpo_ctrl. 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).</p> <p>0x20 = clkIn0 los_sts, output 0x21 = clkIn1 los_sts, output 0x22 = clkIn2 los_sts, output 0x23 = clkIn3 los_sts, output 0x24 = refin los_sts, output 0x25 = apll_lock_sts, output</p> <p>0x28 = clkIn0 los_evt, output 0x29 = clkIn1 los_evt, output 0x2A = clkIn2 los_evt, output 0x2B = clkIn3 los_evt, output 0x2C = refin los_evt, output 0x2D = apll_lol, output</p> <p>0x30 = clkIn0 los_lmt_evt, output 0x31 = clkIn1 los_lmt_evt, output 0x32 = clkIn2 los_lmt_evt, output 0x33 = clkIn3 los_lmt_evt, output 0x34 = refin los_lmt_evt, output 0x35 = apll_lol_lmt, output</p> <p>0x37 = CLKMODE (fanout_clkmode), output 0x38 = device_ready, output 0x39 = CLKMODE1 (fanout_buf_mode1), output</p> |

Table 99. GPIO_STS - GPIO Status

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|--|
| 7:0 | gpio_sts | RO | 0x0 | <p>GPIO Status.</p> <p>Indicates the status of the GPIO0/1/2/3/4 pins without latching and without applying optional polarity inversion (gpio_pol).</p> <p>For GPIO0/1/2, the possible encodings are:</p> <ul style="list-style-type: none"> ▪ 0x0 = Tri-level low, 2-level low ▪ 0x1 = Tri-level mid, 2-level unused ▪ 0x2 = Unused ▪ 0x3 = Tri-level high, 2-level high <p>Bit [7] = GPIO4 status Bit [6] = GPIO3 status Bits [5:4] = GPIO2 status Bits [3:2] = GPIO1 status Bits [1:0] = GPIO0 status</p> |

11.16 EEPROM Registers

Table 100. EEPROM_CNFG - EEPROM Load Configuration

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-----------------|------|---------------|---|
| 63:56 | reserved | RO | 0x0 | Reserved. |
| 55:44 | eeeprom_fall | RW | 0x96 | <p>EEPROM Falling Edge Time.</p> <p>Cycle number (counting down from eeeprom_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.</p> |
| 43:32 | eeeprom_rise | RW | 0x1C2 | <p>EEPROM Rising Edge Time.</p> <p>Cycle number (counting down from eeeprom_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.</p> |
| 31:20 | eeeprom_cycle | RW | 0x258 | <p>EEPROM Cycle Time.</p> <p>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.</p> |
| 19:15 | reserved | RO | 0x0 | Reserved. |
| 14:13 | eeeprom_i2c_drv | RW | 0x0 | <p>I2C/SMBus Drive Strength.</p> <p>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 eeeprom_i2c_speed.</p> <p>0x0 = 1.8V Standard mode (100 kHz) or 2.5V/3.3V standard (100kHz) and Fast mode (400kHz)</p> <ul style="list-style-type: none"> ▪ 0x1 = 1.8V Fast mode (400 kHz) ▪ 0x2 = Reserved ▪ 0x3 = 1.8V/2.5V/3.3V Fast mode plus (1 MHz) |

Table 100. EEPROM_CNFG - EEPROM Load Configuration (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|-------------------------|------|---------------|--|
| 12 | eeeprom_ext_ad dr | RW | 0x0 | EEPROM Extended Address Enable. Allows extended 10-bit addressing with a 1-byte I2C address, if supported by the EEPROM. <ul style="list-style-type: none"> 0x0 = The address is sent outside of the device address. 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. <code>eeeprom_addr_size</code> must be set to 0. |
| 11 | eeeprom_addr_si ze | RW | 0x1 | EEPROM Address Size. Number of address bytes sent to the EEPROM during a read. 0x0 = 1-byte address 0x1 = 2-byte address |
| 10:7 | eeeprom_length | RW | 0x4 | EEPROM Size. Selects the number of bytes in the EEPROM for storing configurations. <ul style="list-style-type: none"> 0x0 = 128B 0x1 = 256B 0x2 = 512B 0x3 = 1KB 0x4 = 2KB 0x5 = 4KB 0x6 = 8KB 0x7 = 16KB 0x8 = 32KB 0x9 = 64KB |
| 6 | eeeprom_load_en | RW | 0x0 | EEPROM Load Enable. 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 <code>eeeprom_load_en</code> is set to 1), OTP user, and EEPROM user (if <code>eeeprom_load_en</code> 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. <i>Warning:</i> If the SDA pin is held low or floating when the I ² C master attempts to read the EEPROM, the I ² C master will wait indefinitely until SDA becomes high before beginning the read request. <ul style="list-style-type: none"> 0x0 = Disabled 0x1 = Enabled |
| 5:4 | eeeprom_i2c_spe ed | RW | 0x0 | EEPROM I2C Speed. Selects the I2C master speed for EEPROM load. When the speed is 400kHz or 1MHz, <code>eeeprom_fall</code> , <code>eeeprom_rise</code> and <code>eeeprom_cycle</code> are internally divided by 4 or 10 respectively to achieve the faster timing. The pad drive strength (<code>eeeprom_i2c_drv</code>) should also be set according to the speed. <ul style="list-style-type: none"> 0x0 = 100kHz 0x1 = 400kHz 0x2 = 1MHz 0x3 = Reserved |
| 3:0 | eeeprom_retry_c ount | RW | 0x4 | EEPROM Load Retry Count. Number of times to attempt to load EEPROM. If <code>eeeprom_bad</code> or load failed (CRC error) after this number of attempts, the <code>eeeprom_load_fail</code> status bit is set. |

Table 101. EEPROM_ADDR - EEPROM Address Configuration

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

Table 102. EEPROM_EVENT - EEPROM Load Event Status

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|----------------------|------|---------------|---|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3 | eeeprom_bad | RW1C | 0x0 | EEPROM Not Detected. When high, indicates the EEPROM did not acknowledge a read access during device startup or a dynamic configuration load. In this case, eeeprom_load_fail is not set. If EEPROM load is disabled (eeeprom_load_en is set to 0), then VC7 will not attempt to read the EEPROM and this bit cannot be set. Cleared by writing it to 1. |
| 2 | eeeprom_config_empty | RW1C | 0x0 | EEPROM Load of Empty Configuration. 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. Cleared by writing it to 1. |
| 1 | eeeprom_load_fail | RW1C | 0x0 | EEPROM Load Failure. 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 (eeeprom_bad is set instead). Cleared by writing it to 1. |
| 0 | eeeprom_crc_err | RW1C | 0x0 | EEPROM Load CRC Error. When high, indicates the EEPROM load encountered one or more CRC errors during device startup or a dynamic configuration load. Cleared by writing it to 1. |

Table 103. EEPROM_ERR_CNT - EEPROM CRC Error Count

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|---------------------|------|---------------|--|
| 7:4 | reserved | RO | 0x0 | Reserved. |
| 3:0 | eeeprom_crc_err_cnt | RW | 0x0 | EEPROM CRC Error Counter. 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. This register can only be written if the block is not clock gated (otp_cg) or held in reset (otp_sw_rst). |

Table 104. 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. 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 105. OTP_EVENT - OTP Event Status

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------------|------|---------------|--|
| 7 | manual_rdy | RW1C | 0x0 | Manual Ready Indicator. When high, indicates a manual request (including Program Assist) completed. Cleared by writing it to 1. |
| 6 | status_latched | RW1C | 0x0 | OTP Latched Status. 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. 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. 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. 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. 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. 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. 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. |

11.17 INT Registers

Table 106. SCRATCH0 - Software Scratch Register 0

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------|------|---------------|---|
| 31:0 | scratch0 | RW | 0x0 | <p>Scratch Register and Power-down Controls.</p> <p>For arbitrary software use.</p> <p>In VC7A+, bits[3:0] have the following function:</p> <ul style="list-style-type: none"> ▪ 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. ▪ 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. ▪ 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 107. INT_EN - Interrupt Enable

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

Table 107. 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. 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. When this field is set to 1, the los0_lmt_int_sts bit contributes to the device interrupt |
| 20 | reserved | RO | 0x0 | Reserved. |
| 19 | apll_lol_lmt_int_en | RW | 0x0 | APLL Loss-of-lock Threshold Exceeded Interrupt Enable. 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. 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. 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. 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. 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. 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. 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. 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. 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. When this field is set to 1, the los0_int_sts bit contributes to the device interrupt. |
| 3:1 | reserved | RO | 0x0 | Reserved. |
| 0 | apll_lol_int_en | RW | 0x0 | APLL Loss-of-Lock Interrupt Enable. When this field is set to 1, the apll_lol_int_sts bit contributes to the device interrupt. |

Table 108. INT_STS - Interrupt Status

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------------------|------|---------------|---|
| 31 | device_int_sts | RO | 0x0 | Device Interrupt Status. 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. 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. 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. 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. Mirrors the REFIN los_lmt_evt event bit |
| 24 | los3_lmt_int_sts | RO | 0x0 | CLKIN3 Monitor LOS Threshold Exceeded Interrupt Status. Mirrors the CLKIN3 los_lmt_evt event bit |
| 23 | los2_lmt_int_sts | RO | 0x0 | CLKIN2 Monitor LOS Threshold Exceeded Interrupt Status. Mirrors the CLKIN2 los_lmt_evt event bit |
| 22 | los1_lmt_int_sts | RO | 0x0 | CLKIN1 Monitor LOS Threshold Exceeded Interrupt Status. Mirrors the CLKIN1 los_lmt_evt event bit |
| 21 | los0_lmt_int_sts | RO | 0x0 | CLKIN0 Monitor LOS Threshold Exceeded Interrupt Status. Mirrors the CLKIN0 los_lmt_evt event bit |
| 20 | reserved | RO | 0x0 | Reserved. |
| 19 | apll_lo_lmt_int_sts | RO | 0x0 | APLL Loss-of-lock Threshold Exceeded Interrupt Status. Mirrors the apll_lo_lmt event bit |
| 18 | reserved | RO | 0x0 | Reserved. |
| 17 | freq3_int_sts | RO | 0x0 | CLKIN3 Frequency Monitor Interrupt Status. Mirrors the CLKIN3 freq_evt event bit |
| 16 | freq2_int_sts | RO | 0x0 | CLKIN2 Frequency Monitor Interrupt Status. Mirrors the CLKIN2 freq_evt event bit |
| 15 | freq1_int_sts | RO | 0x0 | CLKIN1 Frequency Monitor Interrupt Status. Mirrors the CLKIN1 freq_evt event bit |
| 14 | freq0_int_sts | RO | 0x0 | CLKIN0 Frequency Monitor Interrupt Status. 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. Mirrors the REFIN los_evt event bit |
| 7 | los3_int_sts | RO | 0x0 | CLKIN3 Monitor Loss-of-Signal Interrupt Status. Mirrors the CLKIN3 los_evt event bit |
| 6 | los2_int_sts | RO | 0x0 | CLKIN2 Monitor Loss-of-Signal Interrupt Status. Mirrors the CLKIN2 los_evt event bit |

Table 108. INT_STS - Interrupt Status (Cont.)

| Bit Field | Field Name | Type | Default Value | Description |
|-----------|------------------|------|---------------|---|
| 5 | los1_int_sts | RO | 0x0 | CLKIN1 Monitor Loss-of-Signal Interrupt Status. Mirrors the CLKIN1 los_evt event bit |
| 4 | los0_int_sts | RO | 0x0 | CLKIN0 Monitor Loss-of-Signal Interrupt Status. Mirrors the CLKIN0 los_evt event bit |
| 3:1 | reserved | RO | 0x0 | Reserved. |
| 0 | apll_lol_int_sts | RO | 0x0 | APLL Loss-of-lock Interrupt Status. Mirrors the apll_lol event bit |

12. Revision History

| Revision | Date | Description |
|----------|--------------|---|
| 1.01 | Aug 14, 2023 | <ul style="list-style-type: none"> ▪ Updated part number references to RC210xx ▪ Updated device_revision in Table 34 (DEVICE_REV) ▪ Updated the description of en_cap_x2 and en_cap_x1 in Table 53 (XO_CNFG) ▪ Added Table 86 (ANA_SPARE_CNFG) and Table 87 (ANA_SPARE_STS) |
| 1.00 | Oct 26, 2022 | Initial release. |

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

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

Trademarks

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

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.