

Description

The Cascade ClkSoC™ SiT95145 is a single-chip MEMS clock jitter cleaner optimized for the highest level of clock tree integration, consolidating multiple clock ICs and oscillators into a single device. Its low noise quad-PLL architecture and programmable output drivers can deliver up to 10 differential or 20 LVCMOS low jitter clock outputs. This ClkSoC™ supports 4 additional clock inputs with frac-N dividers, enabling virtually any input to output frequency translation configurations from 8 kHz to 2.1 GHz.

The SiT95145 integrates a SiTime third-generation ApexMEMS resonator to completely eliminate the need for external discrete crystal references required by traditional clock generators. This integrated MEMS approach simplifies system design and improves system reliability by eliminating the clock generator's dependency on quartz, along with all quartz related issues:

- No crystal capacitive matching issues that impact clock synthesis accuracy
- Always-reliable startup even at cold temperature and other harsh environmental conditions
- Simpler PCB routing without noise coupling onto a crystal interface which can result in jitter degradation
- No activity dips that are inherent to quartz
- 10 times more vibration resistant

The SiT95145 comes with the companion Cascade GUI software to simplify clock tree design. The device can be shipped with a user-specified, factory pre-programmed default startup configuration. The device configuration can be re-programmed by the user in two-time-programmable non-volatile memory (NVM) during system manufacturing or in-circuit via I2C/SPI for additional BOM flexibility. Details of register mapping can be found in the [Cascade ClkSoC SiT9514x Register Map – Users Manual](#).

Applications

- Clock tree consolidation replacing crystal oscillators (XOs) and buffers
- Low jitter clock frequency translation and generation
- 10G/100G/400G Ethernet clocking
- IEEE 1588 precision time protocol (PTP)
- Synchronous Ethernet clock synchronization, ITU-T G.8262.1
- JESD204B support
- Optical transport network (OTN) clocking for framers, mappers and processors
- FPGA, processor, and memory clocking
- Server, storage, datacenter clocking
- Test and measurement instrumentation
- Broadcast video

Features

- Quad fractional-N PLLs with integrated VCO and loop filter, 93 fs typical RMS phase jitter for SerDes applications
- Wide output frequency support
 - Differential outputs from 8 kHz to 2.1 GHz
 - LVCMOS outputs from 8 kHz to 250 MHz
 - Supports 1 Hz to 2.1 GHz output frequency on one output
- Wide input frequency support
 - Differential input from 8 kHz to 750 MHz
 - LVCMOS input from 8 kHz to 250 MHz
- Individually configurable output formats and VDD supply
 - LVPECL, CML, HCSSL, LVDS or LVCMOS
 - 1.8 V, 2.5 V or 3.3 V
- Flexible input to output frequency translation with jitter attenuation, 4 inputs, 10 outputs
- Programmable jitter attenuation bandwidth for each PLL: 1 mHz to 4 kHz
- Synchronized, holdover or free run operation modes
- Hitless clock inputs with auto or manual switching
 - Sub 50 ps phase build out mode transients
 - Phase propagation with programmable slopes
 - Frequency ramp for plesiochronous clocks
 - Robust and fast cycle slip and frequency step detection for input frequency steps (clean frequency tracking for large frequency steps)
- Locks to gapped clock inputs to support OTN
- Programmable output delay control
- Programmable frequency ramp slopes for switching plesiochronous clocks
- DCO mode via I2C or SPI with 0.005 ppb resolution
- Indicators: Lock Loss, Clock Loss, Frequency Drift
- Repeatable Input to output delays for each power up of chip
 - Zero Delay Buffer mode also possible on any one PLL
 - Output wake up sync with an independent clock also possible
- Industry standard 64-pin 9 x 9 mm PQFN package



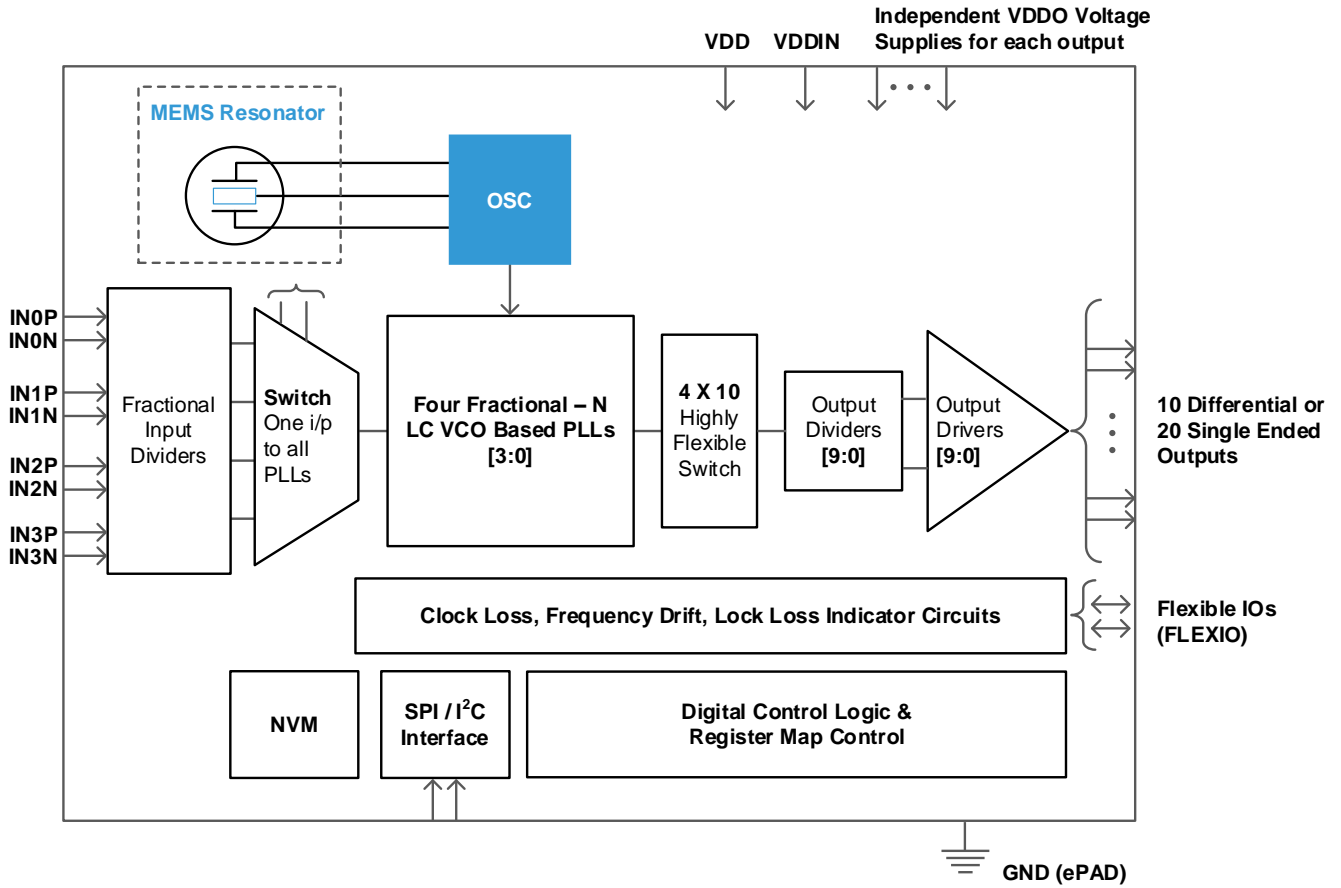


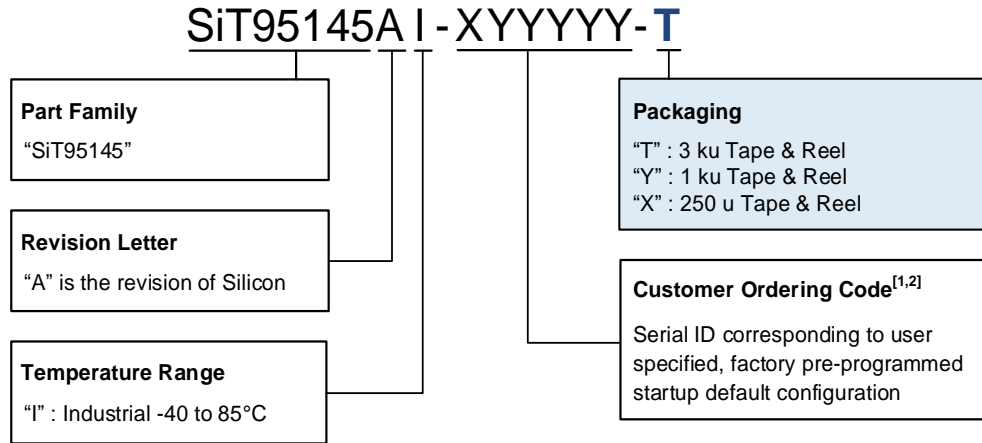
Figure 1. SiT95145 Block Diagram

Table of Contents

| | |
|--|----|
| Description | 1 |
| Applications | 1 |
| Features..... | 1 |
| Ordering Information | 5 |
| Electrical Characteristics..... | 6 |
| “4-16A” Phase Jitter Methodology for SerDes Applications | 21 |
| Functional Description | 23 |
| Frequency Configuration..... | 25 |
| Master and Slaves: Architecture Description and Programming Procedures..... | 28 |
| Input Slave Description | 32 |
| Clock Monitor Slave Description | 32 |
| Fault Monitoring System..... | 32 |
| Clock Loss Monitors | 32 |
| Frequency Drift Monitors | 33 |
| Lock Loss Monitors | 33 |
| Output Slave Description | 33 |
| PLL Bandwidth..... | 33 |
| Fast Lock | 33 |
| Operation Modes | 34 |
| Device Start-Up and Initialization | 35 |
| Free run mode..... | 35 |
| Synchronized mode..... | 35 |
| Holdover mode..... | 35 |
| DCO Mode operation..... | 35 |
| PLL MEMS Oscillator Reference | 35 |
| Inputs..... | 35 |
| Automatic Input Selection..... | 35 |
| Manual Input Selection..... | 35 |
| Hitless Input Switching | 35 |
| Ramped Input Switching | 35 |
| Input Clocks | 36 |
| Clock Monitoring..... | 39 |
| Fault Monitoring System..... | 39 |
| Clock Loss Monitors | 39 |
| Lock Loss Monitors | 40 |
| Outputs | 40 |
| Differential Output Termination Information | 41 |
| LVPECL | 41 |
| LVDS..... | 43 |
| HCSL..... | 45 |
| CML 45 | |
| Output Delay Control | 47 |
| Zero Delay Mode | 47 |
| SiT95145 as device clock and synchronous system reference..... | 49 |
| Programmable Interface (PIF) Top Level View | 50 |
| Serial Programming Interface Description | 52 |
| I ² C protocol..... | 52 |
| SPI Protocol..... | 55 |
| SPI Timing Details..... | 56 |
| SPI Single byte write | 57 |
| SPI Single byte read..... | 57 |
| Alarm Registers’ Information..... | 59 |
| Status and Notify | 59 |
| Examples for Live Status Read Back..... | 62 |
| Dynamic registers to read the various alarm registers in the RealTime page | 62 |
| Examples of Sticky Bit Clearing | 63 |
| Device Initialization for a Non-Programmed Device..... | 64 |
| Monitoring the Status for Master and Slave Pages..... | 74 |
| Programming the Primary E-Fuse | 76 |

| | |
|--|----|
| Configuration Bits to Force Power-up of Digital Slave Subsystems | 76 |
| E-Fuse Lock Configuration Bits | 77 |
| E-Fuse Write Configuration Bits | 78 |
| Configuration Bit to Remove Manual Wake Up for Primary E-Fuse | 79 |
| Pseudo Code: Programming the Primary E-Fuse | 79 |
| Programming the Secondary E-Fuse..... | 82 |
| Configuration Bit to Escape to PROGRAM_CMD State in GENERIC_SYS | 82 |
| Configuration Bit to Change the E-Fuse pointer..... | 82 |
| Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse | 82 |
| Pseudo Code: Programming the Secondary E-Fuse..... | 83 |
| Register Map Details..... | 86 |

Ordering Information



Notes:

- X = "A" and "B" customer device, "C" to "Z" reserved.
 - A: Denotes blank devices;
 - B: Denotes Pre-configured devices, [contact SiTime](#) for the specifics
- Y = 0..9, A..Z for custom serial ID.

Electrical Characteristics

All Min and Max limits in the Electrical Characteristics tables are specified over temperature and rated operating voltage with standard output terminations shown in the termination diagrams. Typical values are at 25°C and nominal supply voltage.

See [Table 1](#) below for absolute maximum ratings. Also see [Table 3](#) for operating temperature.

Table 1. Absolute Maximum Ratings^[3,4]

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|--|--------------------|-------|------|-------|------|--------------------------------|
| Core supply voltage, Analog Input | V _{DDIN} | -0.5 | – | +3.63 | V | |
| Core supply voltage, PLL | V _{DD} | -0.5 | – | +3.63 | V | |
| Output bank supply voltage | V _{DDOX} | -0.5 | – | +3.63 | V | |
| Input voltage, All Inputs | V _{IN} | -0.5 | – | +3.63 | V | |
| I ² C Bus input voltage | V _{INIZC} | -0.5 | – | +3.63 | V | SCLK, SDAT pins |
| SPI Bus input voltage | V _{INSPI} | -0.5 | – | +3.63 | V | |
| Storage Temperature | T _S | -55 | – | +150 | °C | Non-functional, Non-Condensing |
| Programming Temperature | T _{PROG} | +25 | – | +85 | °C | |
| Maximum Junction Temperature in Operation | T _{JCT} | – | – | +125 | °C | |
| Programming Voltage (for Programming the OTP Fuse Memory). | V _{PROG} | 2.375 | 2.5 | 2.625 | V | |
| ESD (human body model) | ESD _{HBM} | – | – | 2000 | V | JESD22A-114 |
| Latch-up | LU | – | – | 100 | mA | JEDEC JESD78D |

Notes:

- Exceeding maximum ratings may shorten the useful life of the device.
- Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or at any other conditions beyond those indicated under the DC Electrical Characteristics is not implied. Exposure to Absolute-Maximum-Rated conditions for extended periods may affect device reliability or cause permanent device damage.

Table 2. Environmental Compliance

| Parameter | Test Conditions | Value | Unit |
|--|---------------------------|--------|------|
| Mechanical Shock Resistance | MIL-STD-883F, Method 2002 | 10,000 | g |
| Mechanical Vibration Resistance | MIL-STD-883F, Method 2007 | 70 | g |
| Soldering Temperature (follow standard Pb free soldering guidelines)(1) | MIL-STD-883F, Method 2003 | 260 | °C |
| Moisture Sensitivity Level | MSL3 @ 260°C | | |
| Electrostatic Discharge (HBM) | HBM, JESD22-A114 | 2,000 | V |
| Charge-Device Model ESD Protection | JESD220C101 | 750 | V |
| Latch-up Tolerance | JESD78 Compliant | | |

Table 3. Recommended Operating Conditions

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|---|-----------------|------|-------|------|------|---|
| Junction temperature | TC | – | – | 125 | °C | |
| Ambient temperature | TA | -40 | – | +85 | °C | Recommended ambient temperature can exceed +85°C if junction temperature is less than or equal to 125°C. Maximum operating temperature is a function of thermal package resistance and power consumption PD: Natural convection: T _{A, MAX} = 125°C – PD * θ _{JA} Heat removal thru exposed pad: T _{BOARD, MAX} = 125°C – PD * θ _{JB} |
| Thermal Resistance Junction to Ambient | θ _{JA} | – | 25.50 | – | °C/W | Still Air |
| | | – | 20.80 | – | °C/W | Air Flow 1 m/s |
| | | – | 19.60 | – | °C/W | Air Flow 2 m/s |
| Thermal Resistance Junction to Case | θ _{JC} | – | 8.70 | – | °C/W | |
| Thermal Resistance Junction to Board | θ _{JB} | – | 7.07 | – | °C/W | |
| Thermal Resistance Junction to Top Center | ψ _{JA} | – | 0.20 | – | °C/W | |

Table 4. DC Electrical Characteristics

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|---|---|---------|------|---------|------|---|
| VDDIN = 3.3 V, VDD = 1.8 V | | | | | | |
| Supply voltage, Analog Input | V _{DDIN} ^[5,12] | 2.375 | 2.5 | 2.625 | V | 2.5-V range: ±5% |
| | | 2.97 | 3,3 | 3.63 | V | 3.3-V range: ±10% |
| Supply voltage, PLL | V _{DD} ^[5,12] | 1.71 | 1.80 | 1.89 | V | 1.8-V range: ±5% |
| | | 2.375 | 2.5 | 2.625 | V | 2.5V range: ±5 % |
| | | 2.97 | 3,3 | 3.63 | V | 3.3-V range: ±10% |
| Supply Voltage, Output Drivers | V _{DDOX} | 1.71 | 1.80 | 1.89 | V | 1.8-V range: ±5% |
| | | 2.375 | 2.50 | 2.625 | V | 2.5-V range: ±5% |
| | | 2.97 | 3.3 | 3.63 | V | 3.3-V range: ±10% |
| Supply Voltage Ramp Rate | T _{RAMP} | 0.100 | – | 50 | ms | Time from 10% to 90% of VDD. See Figure 2 . Power ramps must be monotonic. |
| Total Power Dissipation (2.5V LVDS Outputs @ 156.25M) | P _d | – | 1130 | – | mW | 4 PLLs, 10 Outputs |
| Supply Current, VDDIN | I _{DDIN} ^[6] | – | 20 | 25 | mA | All Four Inputs assumed to be enabled |
| Supply Current, VDD | I _{DD} | – | 320 | 384 | mA | All Four PLLs and 8 Outputs enabled |
| | | – | 345 | 414 | mA | All Four PLLs and 10 Outputs enabled |
| | | – | 170 | 204 | mA | Two PLLs and 4 Outputs enabled |
| | | – | 40 | 48 | mA | LVPECL, output pair terminated 50 Ω to VTT (VDDO – 2 V) |
| Power supply current, VDDOX | I _{DDO} ^[7,8,9,10,11] | – | 28 | 33.6 | mA | LVPECL2, output pair terminated 50 Ω to VTT (VDDO – 2 V) or 0 V without common mode current |
| Power supply current, VDDOX | I _{DDO} ^[7,8,9,10,11] | – | 20 | 24 | mA | CML, output pair terminated 50 Ω to VDDO |
| Power supply current, VDDOX | I _{DDO} ^[7,8,9,10,11] | – | 27 | 32.4 | mA | HCSL, output pair with HCSL termination |
| Power supply current, VDDOX | I _{DDO} ^[7,8,9,10,11] | – | 16 | 19.2 | mA | LVDS, output pair terminated with an AC or DC Coupled diff 100 Ω |
| Power supply current, VDDOX | I _{DDO} ^[7,8,9,10,11] | – | 20 | 24 | mA | LVDS Boost, output pair terminated with an AC or DC Coupled diff 100 Ω |
| Power supply current, VDDOX | I _{DDO} ^[7,8,9,10,11] | – | 15 | 18 | mA | LVC MOS, 250 MHz, 2.5 V output, 5-pF load |
| Logic inputs | V _{LOG} | 0.7*VDD | – | – | V | High Level Logic (Logic State = '1') |
| | | – | – | 0.3*VDD | V | Low Level Logic (Logic State = '0') |

Table 5. Supported Signaling Types and Voltages

| Signaling Type | LVPECL | LVPECL2 | LVDS | LVDS2 (Boosted) | HCSL | CML | LVC MOS |
|----------------|---------------|---------|------|-----------------|------|-----|---------|
| 1.8 V | NOT Supported | OK | OK | NOT Supported | OK | OK | OK |
| 2.5 V | OK | OK | OK | OK | OK | OK | OK |
| 3.3 V | OK | OK | OK | OK | OK | OK | OK |

Notes:

- To use 1.8 V for VDD, series resistor (4.7K) must be used at RSTB, FlexIO (input) and I²C/SPI (input) if the input voltage level is 2.5V/3.3V
- This current can be linearly scaled if smaller than 4 inputs are enabled.
- LVPECL mode provides 6mA of common mode current on each output. LVPECL2 mode does not provide this common mode current.
- A 50 Ω Termination resistor with a DC bias of VDDO – 2 V for LVPECL standards is supported for VDDO = {2.5 V, 3.3 V}.
- IDDOx Output driver supply current specified for one output driver in the table. This includes current in each of the output module that includes output dividers, drivers and clock distributions.
- The LVDS Boost Mode and the LVDS Mode can be used for AC Coupled output terminations. LVDS Boost provides an LVPECL like swing with an AC Coupled 100 Ohm Differential termination.
- Refer to the appropriate later section in the data sheet for various terminations that are supported.
- For efuse programming in SiT95145 parts, VDD along with VDDIN can be set to 2.5 V with no reliability concerns. Refer to [Programming the Primary E-Fuse](#) for VDD/VDDIN voltage information for efuse programming.

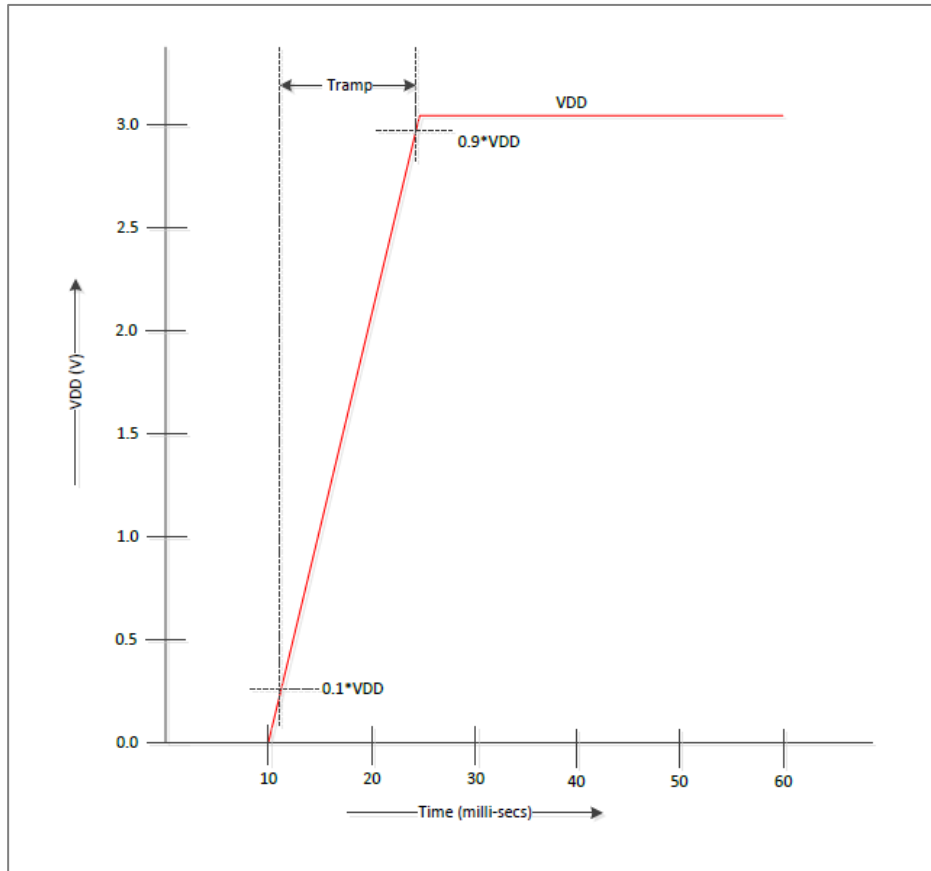


Figure 2. Supply Voltage Ramp

Table 6. Input Clock Characteristics

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|--|-----------------------|-------|------|------|------------|--|
| Differential Input Frequency | $F_{IN,DIFF}$ | 0.008 | – | 2100 | MHz | Differential clock input |
| Single Ended Input Frequency | $F_{IN,SE}$ | 0.008 | – | 250 | MHz | All Single ended clock inputs including LVCMOS |
| Voltage Swing (Differential Amplitude Peak or Single Ended Peak to Peak for the differential signal) ^[13] | $V_{IN,DIFF}$ | 100 | – | – | mV | $f_{IN} < 400$ MHz |
| | | 225 | – | – | mV | 400 MHz < f_{IN} < 750 MHz |
| | | 350 | – | – | mV | 750 MHz < f_{IN} < 2100 MHz |
| Single Ended AC Coupled Inputs (Single Ended Peak to Peak Input) ^[14] | $V_{IN,SE}$ | 500 | – | 3600 | mV | AC Coupled $f_{IN} < 250$ MHz |
| Slew Rate ^[15,16] | SR | 400 | – | – | V/us | |
| Duty Cycle | $DC_{IN,DIFF}$ | 40 | – | 60 | % | |
| Input Capacitance | C_{IN} | – | 0.3 | – | pF | |
| Input Resistance | R_{IN} | – | 15 | – | k Ω | AC Coupled SE |
| | | – | 10 | – | k Ω | Differential |
| Pulsed CMOS Input Buffer — DC-coupled (IN0, IN1, IN2, IN3) | | | | | | |
| Input Frequency | $F_{IN,PULSED_CMOS}$ | 0.008 | – | 250 | MHz | Single ended clock input |
| Input Voltage ^[16] | V_{IH} | 0.8 | – | – | V | |
| | V_{IL} | -0.2 | – | 0.4 | V | |
| Slew Rate ^[15,16] | SR | 400 | – | – | V/us | |
| Duty Cycle | DC | 40 | – | 60 | % | |
| Minimum Pulse Width | PW | 1.6 | – | – | ns | Pulse Input |
| Input Resistance | R_{IN} | – | 30 | – | k Ω | |

Notes:

13. AC Coupled input assumed with series capacitance for differential inputs
14. Single Ended AC Coupled Input Swing requirement (Single Ended Peak to Peak Input) is for optimal noise performance.
15. Resistor termination for differential input followed by series capacitors for each of true and complement differential input connecting to the device pins.
16. LVCMOS single ended is direct coupled on the true input. Connect complement input to ground with a 100 nF capacitor.

Table 7. Serial and Control Input

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Test Condition |
|---------------------|----------|---------------------------------------|------|---------------------------------------|------------|------------------------|
| Input Voltage | V_{IL} | – | – | $0.3 \times V_{DDIO}$ ^[17] | V | |
| | V_{IH} | $0.7 \times V_{DDIO}$ ^[17] | – | – | V | |
| Input Capacitance | C_{IN} | – | 1 | – | pF | |
| Input Resistance | R_{IN} | – | 25 | – | k Ω | |
| Minimum Pulse Width | PW | 100 | – | – | ns | FINC/FLEXIO13, FLEXIO8 |
| Update Rate | F_{UR} | – | – | 1 | μ s | FDEC/FLEXIO13, FLEXIO8 |

Note:

17. VDDIO is the voltage used for all the status outputs and the serial interface. The default voltage for VDDIO can be chosen as either VDDIN or VDD with a hard coded eFuse based selection

Table 8. Output Clock Characteristics

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|--|--|-------|-------|-------|------|---|
| Frequency Stability | F _{stab} | – | – | ±45 | ppm | For output frequencies locked to internal MEMS oscillator; inclusive of initial tolerance, operating temperature, rated power supply voltage, and load variation of 15 pF ± 10%, Contact SiTime for a different stability value |
| 10 Year Aging | F _{10y} | – | ±2 | – | ppm | Ambient temperature of 25°C |
| Differential Output Frequency (PLLA, B, C) | F _{OUT,DIFF} ^[18] | 8k | – | 2100M | Hz | LVPECL, CML, LVDS outputs |
| Differential Output Frequency (PLLD) | | 1 | – | 2100M | Hz | LVPECL, CML, LVDS outputs |
| Differential Output Frequency | F _{OUT,DIFFH} ^[18] | 1 | – | 700M | Hz | HCSL outputs |
| Single Ended Output Frequency | F _{OUN,SE} ^[18] | 1 | – | 250M | Hz | LVC MOS outputs |
| PLL loop bandwidth | F _{BW} | 0.001 | – | 4000 | Hz | Programmable |
| Jitter Peaking | J _{PEAK} | – | – | 0.1 | dB | Meets SONET Jitter Peaking requirements in closed loop |
| Time delay before the Historical average for output Frequency is considered | H _{DELAY} ^[19,20] | 0.035 | 0.5 | 35 | s | Programmable in register map |
| Length of time for which the Average of the frequency is considered | H _{AVG} ^[19,20] | 0.07 | 1 | 70 | s | Programmable in register map |
| Power Supply to I ² C or SPI interface ready | T _{START} | – | – | 10 | ms | No I ² C transaction valid till 10 ms after all power supplies are ramped to 90% of final value. |
| With Speed-Up mode enabled | T _{LOCK} ^[21] | – | 300 | – | ms | Speed-up mode is programmable. This is a Typical number. Actual wake up time depends on fast lock and normal BW settings. |
| DCO Mode Frequency Step Resolution | F _{RES,DCO} ^[22] | – | 0.005 | – | ppb | Frequency Increment or Decrement resolution. This is controlled through the register map. |
| Resolution for output delay | T _{RES} ^[23] | – | 35 | – | ps | Programmable per output clock with this resolution for a total delay of 7.5 ns |
| Maximum Phase Hit | T _{MAX} ^[24] | -50 | ±25 | 50 | ps | Default Hitless Switching Mode (no phase propagation) |
| Uncertainty in Input to Output Delay | ΔT _{DELAY} | -175 | – | 175 | ps | Maximum variation in the static delay from input to output clock between repeated power ups of the chip |
| Pull Range | ω _P | – | 500 | – | ppm | |
| POR to Serial Interface Ready | T _{RDY} | – | – | 15 | ms | |
| Input to Output Delay in ZDB mode (matched pathways on external feedback, IN0 input, IN3 feedback) | T _{ZDELAY} ^[25] | – | 100 | – | ps | |
| Temperature Variation of delay in ZDB mode | T _{ZDELAY,T} ^[25] | – | – | 1 | ps/C | |
| Phase Delay Among Outputs from Different PLLs | T _{OUTDELAY_PS} | – | – | ±250 | ps | In phase sync mode. |
| One free run PLL clock on fuse locked part | T _{START,Special} | – | – | 11 | ms | Using a special mode for fuse locked parts to generate one free run output from one PLL |
| Power supply noise rejection (PSNR) ^[26,27] | PSNR _{VDD} | – | -96 | – | dBc | F _{OUT} = 156.25 MHz, F _{SPUR} = 100 kHz, BW= 100 Hz PSNR on VDD Supply, VDD=3.3 V |
| | | – | -90 | – | dBc | F _{OUT} = 156.25 MHz, F _{SPUR} = 100 kHz, BW= 100 Hz PSNR on VDD Supply, VDD=2.5 V |
| | | – | -75 | – | dBc | F _{OUT} = 156.25 MHz, F _{SPUR} = 100 kHz, BW= 100 Hz PSNR on VDD Supply, VDD=1.8 V |
| | PSNR _{VDDIN} | – | -70 | – | dBc | F _{OUT} = 156.25 MHz, F _{SPUR} = 100 kHz, BW= 100 Hz PSNR on input supply |
| | PSNR _{VDDO} | – | -80 | – | dBc | F _{OUT} = 156.25 MHz, F _{SPUR} = 100 kHz, BW= 100 Hz PSNR on output supply |
| Output-Output Crosstalk ^[28,29,30] | XTALK | – | -75 | – | dBc | 156.25 MHz and 155.52 MHz on adjacent outputs |

Notes:

18. 1 Hz output available only on output OUT0 (OUT0P, OUT0N).
19. Hitless Switching enables PLL to switch between input clocks when the current clock is lost.
 - a. Clock Loss can be defined as 2 / 4 / 8 / 16 consecutive missing pulses.
 - b. Priority list for the input clocks can be set in the register map independently for each PLL.
 - c. Output is truly hitless (no phase transient and 0 ppb relative error in frequency) for exactly same frequency input clocks that are switched.
 - d. Hitless switching support is both revertive and non-revertive.
 - e. Revertive / Non-revertive Support: Assume Clock Input 0 is lost and switch is made to Clock Input 1. Then, PLL reverts to Clock Input 0 when it becomes valid again in Revertive mode. It does not switch back to Clock Input 0 even when it becomes valid again in the non-Revertive mode.
20. PLL enters holdover mode when the active input clock and all spare clocks in the clock priority list for hitless switching are lost.
 - a. Clock Loss can be defined as 2 / 4 / 8 / 16 consecutive missing pulses.
 - b. Programmable Clock Loss settings ensure Gapped Clocks can be supported by choosing higher number of missing pulses as the trigger for clock being invalid.
 - c. Entering holdover mode is supported with the frequency frozen at a historical average determined from the Hdelay and Havg settings.
21. For low PLL Loop Bandwidths, wake up time can be very large unless the speed up feature is used. The speed up feature provides the user options to use a completely independent loop bandwidth for the wake up transitioning to the regular bandwidth after frequency and phase are locked.
 - a. Fast Lock Bandwidth needs to be less than 100 times smaller than the input clock frequency (divided input at PLL phase detector) for stable and bounded (in time) lock trajectory of the PLL.
22. The 0.005 ppb specification is for the smallest frequency step resolution available. Larger frequency step resolutions up to 100 ppm can be used also. The frequency resolution for the DCO mode frequency step is independently programmable for each DCO step.
23. All output clocks from one specific PLL are phase aligned. Relative delay adjustment is then possible on each clock individually as defined by the TRES parameter.
24. This test is for 2 inputs at 8M that are switched to get a 622.08M output.
25. Both input and feedback at 8 MHz with the delays exactly matched and same slew for both for the chip.
26. The PSNR is measured with a 50mVpp sinusoid in series with the supply and checking the spurious level relative to the carrier on the output in terms of phase disturbance impact.
27. Output PSNR measured with LVDS standard which (along with the LVDS boost) are the recommended standards for AC Coupled terminations.
28. Measured across adjacent outputs- All adjacent outputs are covered and the typical value for the worst case output to output coupling is reported.
29. The adjacent output pairs are chosen at 155.52 MHz and 156.25 MHz frequencies.
30. This cross talk between outputs is mainly package dependent therefore terminated outputs are used for reporting these numbers ensuring that there is signal current in the bond wires.

Table 9. Output Serial and Status Pin Characteristics

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Test Condition |
|----------------|-----------------|--|------|-------|------|-------------------------|
| Output Voltage | V _{OH} | V _{VDDIO} ^[31] x 0.75 | – | – | V | I _{OH} = –2 mA |
| | V _{OL} | – | – | 0.125 | V | I _{OL} = 2 mA |

Note:

31. VDDIO is the voltage used for all the status outputs and the serial interface. The default voltage for VDDIO can be chosen as either VDDIN or VDD with a hard coded eFuse based selection.

Table 10. Fault Monitoring Indicators

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|---|---------------------------------------|------|------|-------|--------|---|
| Clock Loss Indicator Thresholds | CL _x ^[32,35] | 2 | 4 | 16 | Pulses | Clock Loss Indicators can be set on any of the four inputs. Loss of 2 / 4 / 8 / 16 consecutive pulses can be used to indicate a clock loss. Programmable in the register map |
| Fine Frequency Drift Indicator Thresholds: Step Size | FD _x ^[33,34,35] | – | ±2 | – | ppm | Frequency drift threshold is programmable in the range with the step size resolution specified. Frequency drift hysteresis is programmable in the range with the step size resolution specified. |
| Fine Frequency Drift Indicator Thresholds: Hysteresis Range | | ±2 | – | ±500 | ppm | |
| Fine Frequency Drift Indicator Thresholds: Range | | ±2 | – | ±500 | ppm | |
| Coarse Frequency Drift Indicator Thresholds | | ±100 | – | ±1600 | ppm | Coarse Drift Indicators programmable from {Up to ±1600 ppm in steps of ± 100 ppm} |
| Lock Loss Indicator Threshold | LL | ±0.2 | – | ±4000 | ppm | Lock Loss Indicator threshold is programmable in the range specified from the following choices for setting and clearing LL: {±0.2, ±0.4} ppm, {±2, ±4} ppm, {±20, ±40} ppm, {±200, ±400} ppm, {±2000, ±4000} ppm |

Notes:

32. Clock Loss Indicators are used for:
 - a. Hitless Switching Triggers
 - b. Update in Status Registers in the register map
33. Coarse and Fine Frequency Drift indicators can be concurrently enabled. This enables the user to detect fast drifting frequencies since detecting fine drifts will take longer measurements.
34. Clock loss and Lock loss indicators are available as alerts on flexible IO pins as described in the functional description section of the data sheet.
35. Clock Loss can be combined with either of the frequency drift monitors (coarse and fine) to trigger the hitless switching event in the PLLs. The trigger for a hitless switching event in the PLL can therefore be either the Clock Loss event or either of Clock Loss or Frequency Drift.

Table 11. Output RMS Jitter in Free Run Mode

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|------------------------------------|--------------------|------|------|------|--------|---|
| 0.012-20 Phase Jitter (legacy) | RMS _{JIT} | – | 127 | – | fs rms | F _{OUT} = 156.25 MHz, single PLL Measured with phase noise analyzer, integrating between 12 kHz and 20 MHz offset frequency. Recommended for SONET OC-48 applications. Includes spurs. |
| 4-16A Phase Jitter ^[36] | RMS _{JIT} | – | 93 | – | fs rms | F _{OUT} = 156.25 MHz, single PLL Measured with phase noise analyzer, extending (flat) phase noise to 3rd harmonic (i.e., 625 MHz offset), folding phase noise below the Nyquist frequency (i.e., 78.125 MHz offset), filtering and integrating from 10 kHz to Nyquist. Filter comprises 4 MHz low pass and 16 MHz high pass cut off frequencies, each with 20 dB/dec roll off. Includes spurs. For more information, refer to “4-16A” Phase Jitter Methodology for SerDes Applications . |
| 4-20 Phase Jitter | RMS _{JIT} | – | 62 | – | fs rms | F _{OUT} = 312.5 MHz, single PLL Measured with phase noise analyzer and applying a 4 MHz high-pass filter with 1 st order roll off, then integrating from 12 kHz to 20 MHz. Includes spurs. |

Note:

36. Recommended for SerDes applications to improve the accuracy of clock jitter analysis, replacing the traditional 12 kHz to 20 MHz brick wall filter when application-specific filter characteristics are not explicitly specified.

Table 12. Output RMS Jitter in Frequency Translation Modes

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|--|-----------------------|------|------|------|--------|---|
| RMS Jitter for 12 kHz – 20 MHz Integration Bandwidth | $RMS_{JIT}^{[37,38]}$ | – | 120 | – | fs rms | $F_{OUT} = 622.08$ MHz, $F_{IN} = 10$ MHz, Single PLL Profile |

Note:

- 37. For best noise performance in jitter attenuation mode, use lowest usable loop bandwidth for the PLL.
- 38. Does not include noise from the input clock to the PLL.

Figure 3 and Figure 4 show representative phase noise measurements at various frequencies.

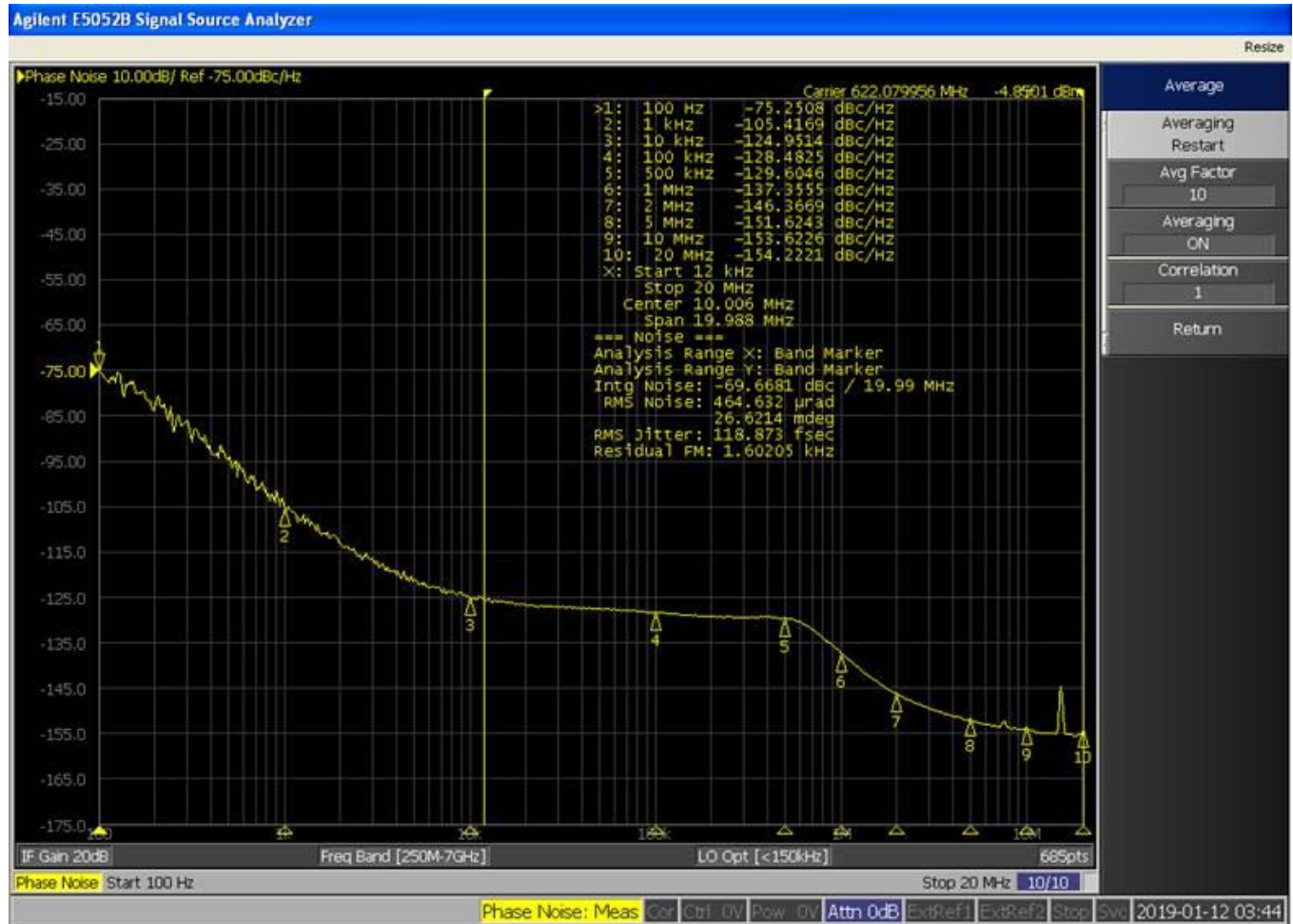


Figure 3. Representative Phase Noise Measurement: $F_{OUT} = 622.08$ MHz, $F_{IN} = 10$ MHz,

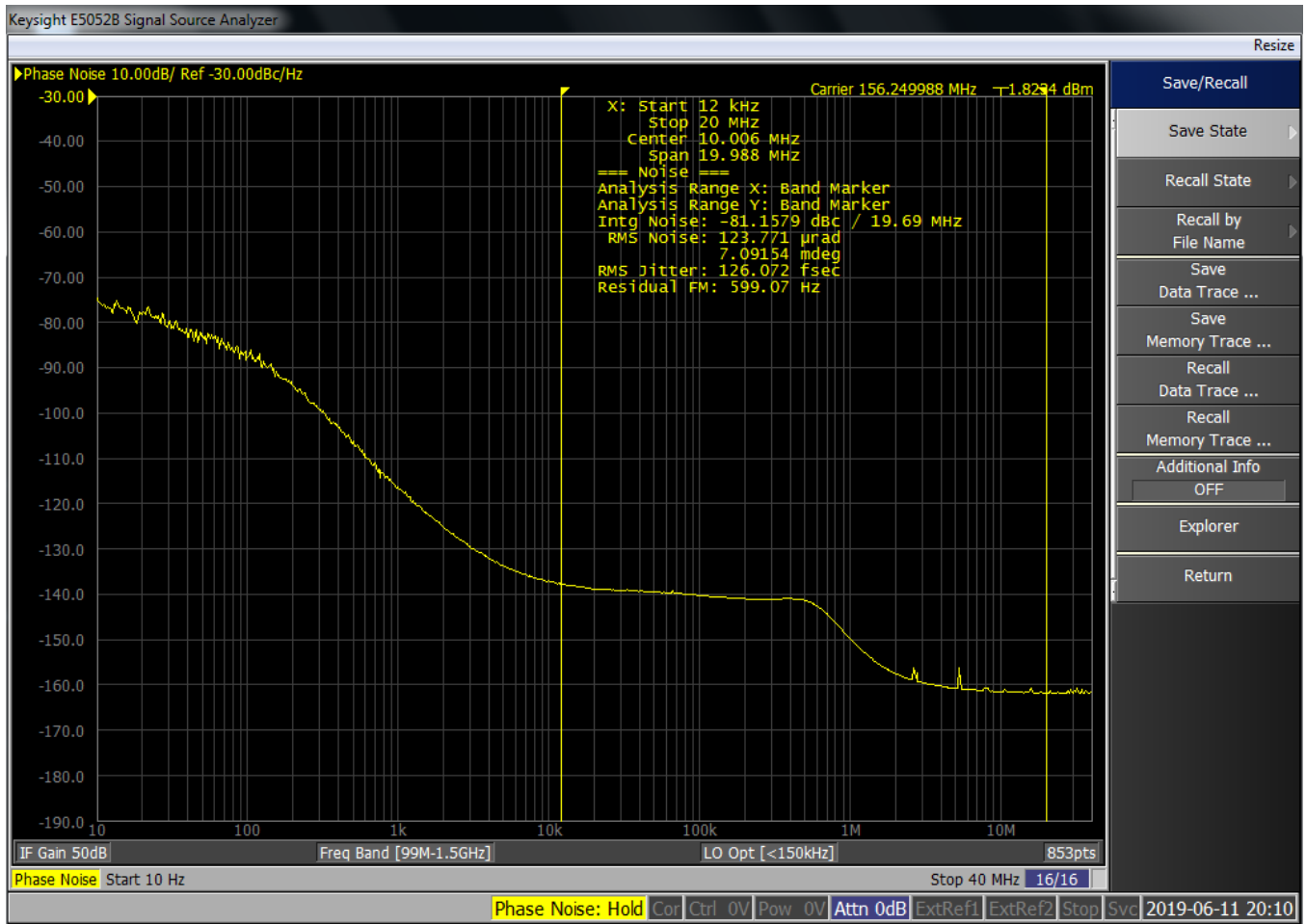


Figure 4. Representative Phase Noise Measurement: $F_{OUT} = 156.25$ MHz, $F_{IN} = 10$ MHz,

Table 13. Close-in Frequency Offset Phase Noise

| Parameter | Symbol | Conditions | Typ. | Unit | Condition |
|---------------------------|--------------------|---------------------------|------|--------|--|
| Low Frequency Phase Noise | PN ^[39] | Offset Frequency = 10 Hz | -91 | dBc/Hz | F _{OUT} = 122.88 MHz, F _{IN} = 10 MHz, PLL BW = 100 Hz |
| | | Offset Frequency = 100 Hz | -113 | | |
| | | Offset Frequency = 1 kHz | -130 | | |
| | | Offset Frequency = 10 kHz | -138 | | |

Note:

39. This is the noise contribution of the chip only excluding the input and reference self contributions.

Table 14. Electrical Specifications for Output Clocks

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|---|----------------------|--------------|-------------|--------------|---------|--|
| DC Electrical Specifications – LVCMOS output (Complementary Out of Phase Outputs or One CMOS Output per Output Driver) | | | | | | |
| Output High Voltage | VOH | VDDO – 0.3 | – | – | V | 4 mA load, VDD = 3.3 V |
| Output High Voltage | VOH | VDDO – 0.4 | – | – | V | 4 mA load, VDD = 1.8 V and 2.5 V |
| Output High Voltage | VOH | VDDO – 0.2 | – | – | V | 100 μ A Load |
| Output Low Voltage | VOL | – | – | 0.3 | V | 4 mA load |
| Output Low Voltage | VOL | – | – | 0.2 | V | 100 μ A Load |
| DC Electrical Specifications – LVDS Outputs (VDDO = 1.8-V, 2.5-V or 3.3-V range) | | | | | | |
| Output Common-Mode Voltage | VOCM | 1.125 | 1.2 | 1.375 | V | VDDO = 2.5 V or 3.3 V range. Applies to LVDS and Boosted LVDS |
| | | 0.74 | 0.87 | 0.96 | | VDDO = 1.8 V |
| Change in VOCM between complementary output states | Δ VOCM | – | – | 50 | mV | |
| Output Leakage Current | IOZ | -20 | – | 20 | μ A | Output Off, VOUT = 0.75 V to 1.75 V |
| DC Electrical Specifications – LVPECL Outputs (VDDO = 2.5-V or 3.3-V range) | | | | | | |
| Output High Voltage | VOH | VDDO – 1.165 | – | VDDO – 0.800 | V | Rterm = 50 Ω to VTT(VDDO – 2.0 V) |
| Output Low Voltage | VOL | VDDO – 2.0 | – | VDDO – 1.55 | V | Rterm = 50 Ω to VTT (VDDO – 2.0 V), w/o common mode current |
| DC Electrical Specifications – HCSSL Outputs (VDD = 1.8-V, 2.5-V or 3.3-V range) | | | | | | |
| Output High Voltage Max | VMAX | – | – | 1150 | mV | Measurement on single-ended signal |
| Output Low Voltage Min | VMIN | -300 | – | – | mV | Measurement on single-ended signal |
| Differential Output High Voltage | VOHDIFF | 150 | – | – | mV | Measurement taken from differential waveform |
| Differential Output Low Voltage | VOLDIFF | – | – | -150 | mV | Measurement taken from differential waveform |
| Absolute Crossing point voltage | VCROSS | 250 | – | 550 | mV | Measurement taken from single ended waveform |
| Variation of VCROSS over all rising clock edges | VCROSS DELTA | – | – | 140 | mV | Measurement taken from single ended waveform |
| DC Electrical Specifications – CML Outputs (VDDO = 1.8-V, 2.5-V or 3.3-V range) | | | | | | |
| Output High Voltage | VOH | VDDO – 0.085 | VDDO – 0.01 | VDDO | V | Rterm = 50 Ω to VDDO |
| Output Low Voltage | VOL | VDDO – 0.6 | VDDO – 0.4 | VDDO – 0.32 | V | Rterm = 50 Ω to VDDO |
| AC Electrical Specifications LVCMOS Output Load: 10 pF < 100 MHz, 7.5 pF < 150 MHz, 5 pF > 150 MHz > 200 MHz | | | | | | |
| Output Frequency | fOUT | 8k | – | 250M | Hz | |
| Output Duty cycle | tDC | 45 | – | 55 | % | Measured at 1/2 VDDO, loaded, fOUT < 100 MHz |
| Output Duty cycle | tDC | 40 | – | 60 | % | Measured at 1/2 VDDO, loaded, fOUT > 100 MHz |
| Rise/Fall time | tRFCMOS | – | – | 2 | ns | VDDO = 1.8 V, 20-80%, Highest Drive setting |
| Rise/Fall time | tRFCMOS | – | – | 1.5 | ns | VDDO = 2.5 V, 20-80%, Highest Drive setting |
| Rise/Fall time | tRFCMOS | – | – | 1.2 | ns | VDDO=3.3 V, 20-80%, Highest Drive setting |
| AC Electrical Specifications (LVPECL, LVDS, CML, HCSSL) | | | | | | |
| Clock Output Frequency | fOUT | 8k | – | 2100M | Hz | |
| PECL Output Rise/Fall Time | tRF | – | – | 350 | ps | 20% to 80% of AC levels. Measured at 156.25 MHz for PECL outputs. |
| CML Output Rise/Fall Time | tRF | – | – | 350 | ps | 20% to 80% of AC levels. Measured at 156.25 MHz for CML outputs |
| LVDS Output Rise/Fall Time | tRF | – | – | 350 | ps | 20% to 80% of AC levels. Measured at 156.25 MHz for LVDS outputs. |
| HCSSL Output Rise/Fall Time | tRF | – | – | – | ps | |
| Output Duty Cycle | tODC | 45 | 50 | 55 | % | Measured at differential 50% level, 156.25 MHz |
| LVDS Output differential peak | VPLVDS | 247 | 350 | 454 | mV | Measured at 156.25 M Output |
| Boosted LVDS differential peak | VPBLVDS | 500 | 700 | – | mV | Measured at 156.25 M Output |
| LVPECL Differential peak | VP _{LVPECL} | 450 | 750 | – | mV | Measured at 156.25 M Output |
| CML Output Differential Peak | VP _{CML} | 250 | – | 600 | mV | Measured at 156.25 M Output |
| HCSSL Differential Peak | VP _{HCSSL} | – | – | – | mV | |

Table 15 below shows the I²C bus timing specifications.

Table 15. I²C Bus Timing Specifications

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|---|--------------------------|------|------|------|------|-----------|
| SCLK clock frequency | f _{SCLK} | – | – | 400 | kHz | |
| Hold time START condition | t _{HD,STA} | 0.6 | – | – | µs | |
| Low period of the SCK clock | t _{LOW} | 1.3 | – | – | µs | |
| High period of the SCK clock | t _{HIGH} | 0.6 | – | – | µs | |
| Setup time for a repeated START condition | t _{SU,STA} | 0.6 | – | – | µs | |
| Data hold time | t _{HD,DAT} | 0 | – | – | µs | |
| Data setup time | t _{SU,DAT} | 100 | – | – | ns | |
| Rise time | t _r | – | – | 300 | ns | |
| Fall time | t _f | – | – | 300 | ns | |
| Setup time for STOP condition | t _{SU,STO} | 0.6 | – | – | µs | |
| Bus-free time between STOP and START conditions | t _{BUF} | 1.3 | – | – | µs | |
| Default is 0x69 if SDO CSB pins are floating Otherwise address is 11010 {SDO},{CSB} | I ² C Address | – | 0x69 | – | | |

Figure 5 shows the I²C timing waveforms.

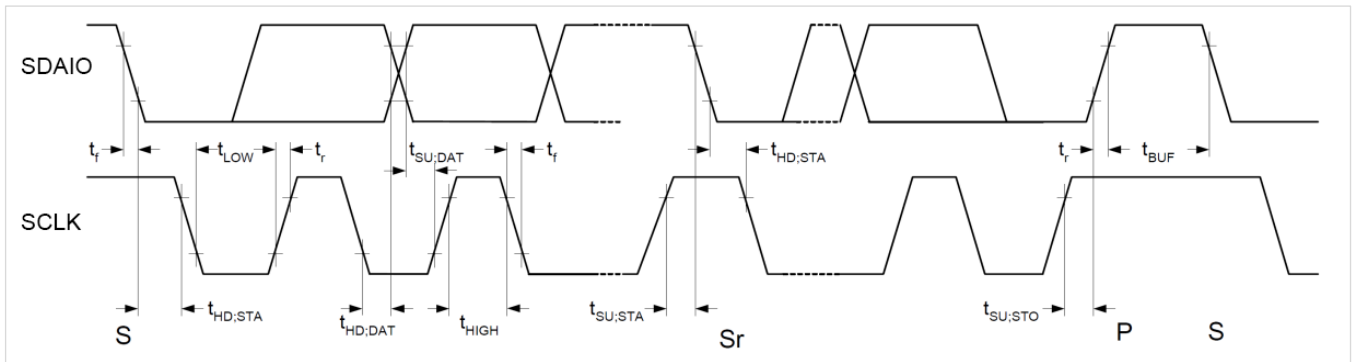


Figure 5. I²C Timing Waveform

Table 16 below shows SPI bus timing specifications.

Table 16. SPI Bus Timing Specifications^[40]

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Condition |
|------------------------|-------------------|------|------|------|------|-----------|
| SCLK clock frequency | f _{SCLK} | – | – | 20 | MHz | |
| Clock pulse width HIGH | t _{CH} | 20 | – | – | ns | |
| Clock pulse width LOW | t _{CL} | 20 | – | – | ns | |
| CSB HIGH time | t _{CS} | 50 | – | – | ns | |
| CSB setup time | t _{CSS} | 25 | – | – | ns | |
| CSB hold time | t _{CSH} | 25 | – | – | ns | |
| Data in setup time | t _{SD} | 10 | – | – | ns | |
| Data in hold time | t _{HD} | 10 | – | – | ns | |
| Output valid | t _{CO} | – | – | 10 | ns | |
| Output setup time | t _{SO} | 10 | – | – | ns | |
| Output hold time | t _{HO} | 10 | – | – | ns | |

Note:

40. When the chip is selected to be in the SPI Mode: The SDO pin is driven to 0 when the CSB is high for the chip in SPI mode.

Figure 6 shows the SPI timing waveforms.

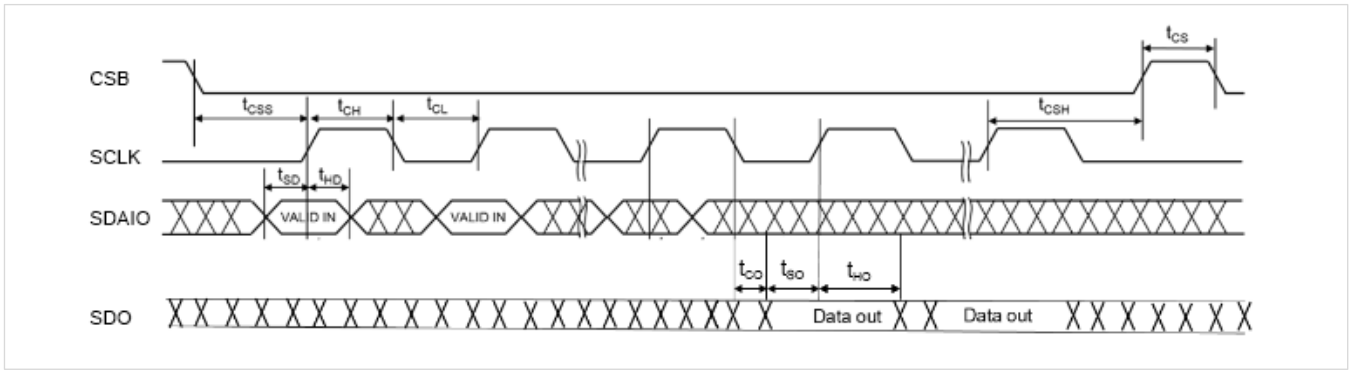


Figure 6. SPI Timing Waveform

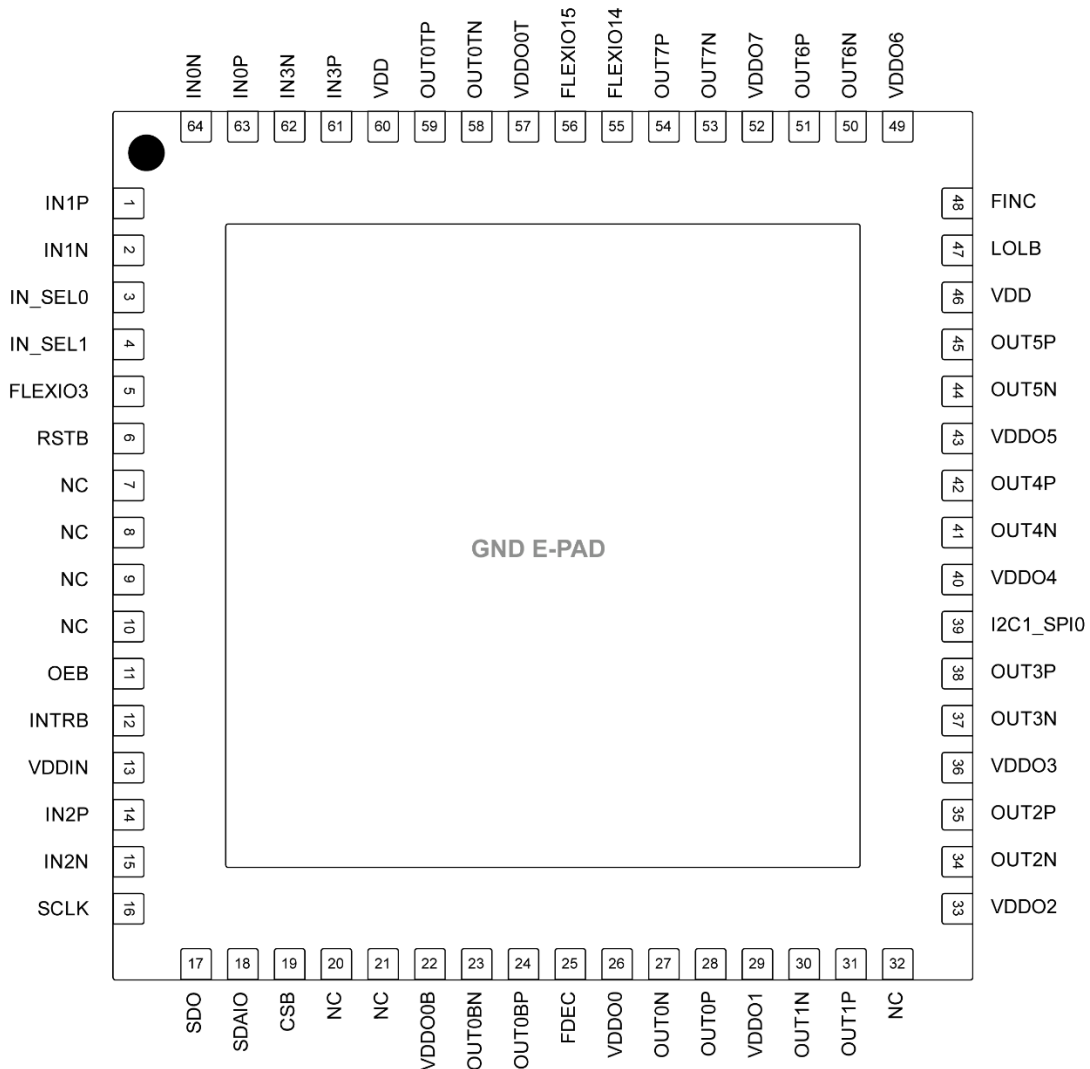


Figure 7. SiT95145 Top View

Table 17. Detailed Pin Description^[41,43,44,45,46]

| Pin Name | I/O Type | Pin # | Pull-up (kΩ) | Pull-down (kΩ) | Function | Comments |
|----------|----------|-------|--------------|----------------|--|---|
| IN0P | Input | 63 | | | True input for IN0 differential pair. Input for LVCMOS IN0 input. Need series external capacitor for differential input. | IN0 / IN1 / IN2 / IN3 inputs can be used for output clock synchronization. An active clock and three spare clocks are chosen for each PLL. There is complete flexibility in the choice of the Active and 4 Spares for each of the 4 PLLs. |
| IN0N | Input | 64 | | | Complement input for IN0 differential pair. Ground with capacitor for LVCMOS IN0 input. Need series external capacitor for differential input. | |
| IN1P | Input | 1 | | | True input for IN1 differential pair. Input for LVCMOS IN1 input. Need series external capacitor for differential input. | |
| IN1N | Input | 2 | | | Complement input for IN1 differential pair. Ground with capacitor for LVCMOS IN1 input. Need series external capacitor for differential input. | |
| IN2P | Input | 14 | | | True input for IN2 differential pair. Input for LVCMOS IN2 input. Need series external capacitor for differential input. | |
| IN2N | Input | 15 | | | Complement input for IN2 differential pair. Ground with capacitor for LVCMOS IN2 input. Need series external capacitor for differential input. | |
| IN3P | Input | 61 | | | True input for IN3 differential pair. Input for LVCMOS IN3 input. Need series external capacitor for differential input. | |
| IN3N | Input | 62 | | | Complement input for IN3 differential pair. Ground with capacitor for LVCMOS IN3 input. Need series external capacitor for differential input. | |
| GND | Power | E-PAD | | | Electrical and Package Ground | Exposed Ground on the bottom E-PAD. |
| OUT0BP | Output | 24 | | | Output 0B True Output or Output 0B LVCMOS | |
| OUT0BN | Output | 23 | | | Output 0B Complement Output or Output 0B LVCMOS | |
| OUT0P | Output | 28 | | | Output 0 True Output or Output 0 LVCMOS | |
| OUT0N | Output | 27 | | | Output 0 Complement Output or Output 0 LVCMOS | |
| OUT1P | Output | 31 | | | Output 1 True Output or Output 1 LVCMOS | |
| OUT1N | Output | 30 | | | Output 1 Complement Output or Output 1 LVCMOS | |
| OUT2P | Output | 35 | | | Output 2 True Output or Output 2 LVCMOS | |
| OUT2N | Output | 34 | | | Output 2 Complement Output or Output 2 LVCMOS | |
| OUT3P | Output | 38 | | | Output 3 True Output or Output 3 LVCMOS | |
| OUT3N | Output | 37 | | | Output 3 Complement Output or Output 3 LVCMOS | |
| OUT4P | Output | 42 | | | Output 4 True Output or Output 4 LVCMOS | |
| OUT4N | Output | 41 | | | Output 4 Complement Output or Output 4 LVCMOS | |
| OUT5P | Output | 45 | | | Output 5 True Output or Output 5 LVCMOS | |
| OUT5N | Output | 44 | | | Output 5 Complement Output or Output 5 LVCMOS | |

Table 17. Detailed Pin Description (continued)

| Pin Name | I/O Type | Pin # | Pull-up (k Ω) | Pull-down (k Ω) | Function | Comments |
|-------------------|----------------|--------|-----------------------|-------------------------|---|--|
| OUT6P | Output | 51 | | | Output 6 True Output or Output 6 LVCMOS | |
| OUT6N | Output | 50 | | | Output 6 Complement Output or Output 6 LVCMOS | |
| OUT7P | Output | 54 | | | Output 7 True Output or Output 7 LVCMOS | |
| OUT7N | Output | 53 | | | Output 7 Complement Output or Output 7 LVCMOS | |
| OUT0TP | Output | 59 | | | Output 0T True Output or Output 0T LVCMOS | |
| OUT0TN | Output | 58 | | | Output 0T Complement Output or Output 0T LVCMOS | |
| VDDIN | Power | 13 | | | Power Supply Voltage pin | Decoupling capacitor close to supply pin required |
| VDD | Power | 46, 60 | | | Power Supply Voltage pin | Multiple Supply Pins, Decoupling capacitor close to each supply pin required |
| INTRB | Output | 12 | | | Active low indicator of programmable sticky notifies. Can be left floating if not used. | |
| RSTB | Input | 6 | 25 | | Active low reset internally pulled up to VDDIO; Pull Up Resistor to VDDIO of fixed value (25 K Ω). Can be left floating . | Active low signal performs a complete reset of the part. See Note 41. |
| OEB | Input | 11 | | 25 | Used to disable (when 1) all the output clocks. Can be left floating or pulled down to GND if not used. | See Note 41. |
| LOLB | Output | 47 | | | Loss of Lock Indicator | |
| FLEXIO3 | Input / Output | 5 | | 25 | Flexible Outputs can be used for programmable status monitoring. Can be left floating or pulled down to GND if not used. | See Note 41. |
| FDEC/ FLEXIO8 | Input | 25 | | 25 | DCO Decrement. | |
| FINC/ FLEXIO13 | Input | 48 | | 25 | DCO Increment. | |
| IN_SEL0 | Input | 3 | | 25 | Input Clock Selection for Manual selection of active clock. Can be left floating or pulled down to GND if not used. | |
| IN_SEL1 | Input | 4 | | 25 | | |
| SDAIO | Input / Output | 18 | 25 | | I ² C Serial Interface Data (SDA) / SPI Input data (SDI) | |
| SCLK | Input | 16 | 25 | | I ² C Serial Interface Clock or SPI Clock Input | |
| CSB/A0 | Input | 19 | 25 | | Chip Select for the SPI Interface. In I ² C mode this is the A0 address pin. | |
| SDO/A1 | Output / Input | 17 | | 25 | Serial Data Output (SPI Interface). In I ² C mode this is the A1 address pin. | |
| I2C1_SPI0 | Input | 39 | 25 | | Choose between SPI(0) and I ² C(1) interface being used | |
| VDDO0B | Power | 22 | | | Output Power Supply for Bank 0B outputs | |
| VDDO0 | Power | 26 | | | Output Power Supply for Bank 0 outputs | |
| VDDO1 | Power | 29 | | | Output Power Supply for Bank 1 outputs | |
| VDDO2 | Power | 33 | | | Output Power Supply for Bank 2 outputs | |
| VDDO3 | Power | 36 | | | Output Power Supply for Bank 3 outputs | |
| VDDO4 | Power | 40 | | | Output Power Supply for Bank 4 outputs | |
| VDDO5 | Power | 43 | | | Output Power Supply for Bank 5 outputs | |
| VDDO6 | Power | 49 | | | Output Power Supply for Bank 6 outputs | |

Table 17. Detailed Pin Description (continued)

| Pin Name | I/O Type | Pin # | Pull-up (kΩ) | Pull-down (kΩ) | Function | Comments |
|----------|----------------|-------------------------|--------------|----------------|--|----------|
| VDDO7 | Power | 52 | | | Output Power Supply for Bank 7 outputs | |
| VDDO0T | Power | 57 | | | Output Power Supply for Bank 0T outputs | |
| FLEXIO14 | Input / Output | 55 | | 25 | Flexible Outputs can be used for programmable status monitoring. Can be left floating or pulled down to GND if not used. | |
| FLEXIO15 | Input / Output | 56 | | 25 | Flexible Outputs can be used for programmable status monitoring. Can be left floating or pulled down to GND if not used. | |
| NC | No Connect | 7, 8, 9, 10, 20, 21, 32 | | | No connect. This pin is not connected to the die. | |

Notes:

41. To use 1.8 V for VDD, series resistor (4.7K) must be used at RSTB, FlexIO (input) and I2C/SPI (input) if the input voltage level is 2.5 V/3.3 V
42. VDDIO is the voltage used for all the status GPIOs and the serial interface. The default voltage for VDDIO can be chosen as either VDDIN or VDD through the TimMaster™ GUI. This needs to be a selection in the base programmable profile itself. Configure VDDIO as VDD by setting 0x23[7] to 0 and VDDIN setting 0x23[7] to 1 on Page 0.
43. All digital input/output GPIOs have an on-chip 25 kΩ pull down resistor to ePAD ground and can be left unconnected if not used.
44. The I2C1_SPI0 pad has an on-chip 25 kΩ pull up resistor to indicate default mode of communication as I2C unless this pin is pulled down on the board to indicate the SPI mode.
45. In I2C mode, the serial data and clock have an on-chip 25 kΩ pull up resistor to VDDIO.
46. The RSTB pin has an on-chip 25 kΩ pull up resistor to VDDIO. Writing 0xFE[0] to 1 with delay addition of 10ms has the same effect as the pulling RSTB pin to GND for chip reset.

Table 18. Input Channel Selection Truth Table

| IN_SEL1 | IN_SEL0 | Input Channel |
|---------|---------|---------------|
| 0 | 0 | IN0 |
| 0 | 1 | IN1 |
| 1 | 0 | IN2 |
| 1 | 1 | IN3 |

“4-16A” Phase Jitter Methodology for SerDes Applications

Proper evaluation of reference clock (refclk) jitter is critical to optimize system performance in high-speed serial links. The traditional 12 kHz to 20 MHz analysis of filtering refclk jitter can mislead designers to select components that degrade rather than improve link performance. Therefore, this datasheet replaces the legacy 12 kHz to 20 MHz filter analysis with an established and more accurate methodology adopted by several industry standards (e.g. PCI Express, CXL, UCIe) and implemented here as “4-16A” phase jitter. A brief overview follows.

Established in 1991 for SONET OC-48 line rates, the traditional 12 kHz to 20 MHz jitter filter served as a golden reference to evaluate refclk jitter for over 30 years. The filter is used is nearly all clock and timing datasheets today. However, the results it provides no longer correlate with system performance and can create suboptimum link performance. Sources of filter error include incorrect corner frequencies, unrealistic brick-wall roll offs and a lack of accounting for aliased phase noise. Errors of tens of femtoseconds are significant today and will become more

significant as data rates increase. For these reasons, we recommend customers adopt the more accurate “4-16A” phase jitter methodology for SerDes applications.

The conventional refclk jitter analysis uses a band-pass filter, as shown in Figure 8, to extract the refclk contribution to jitter observed at the receiver. Historically the refclk jitter filter was arbitrarily applied to phase noise up to an offset equal to the refclk Nyquist frequency. However, this ignores higher-offset phase noise that aliases when the refclk is sampled by a PLL’s digital phase detector. Studies have shown that extending the phase noise data flat to the third harmonic (or, twice the fundamental frequency in the offset-frequency axis) accurately estimates worst-case phase jitter^[47]. Above the third harmonic, phase noise rolls off quickly and can be ignored.

Note:

- 47. “How to evaluation reference-clock phase noise in high-speed serial links”, Signal Integrity Journal.

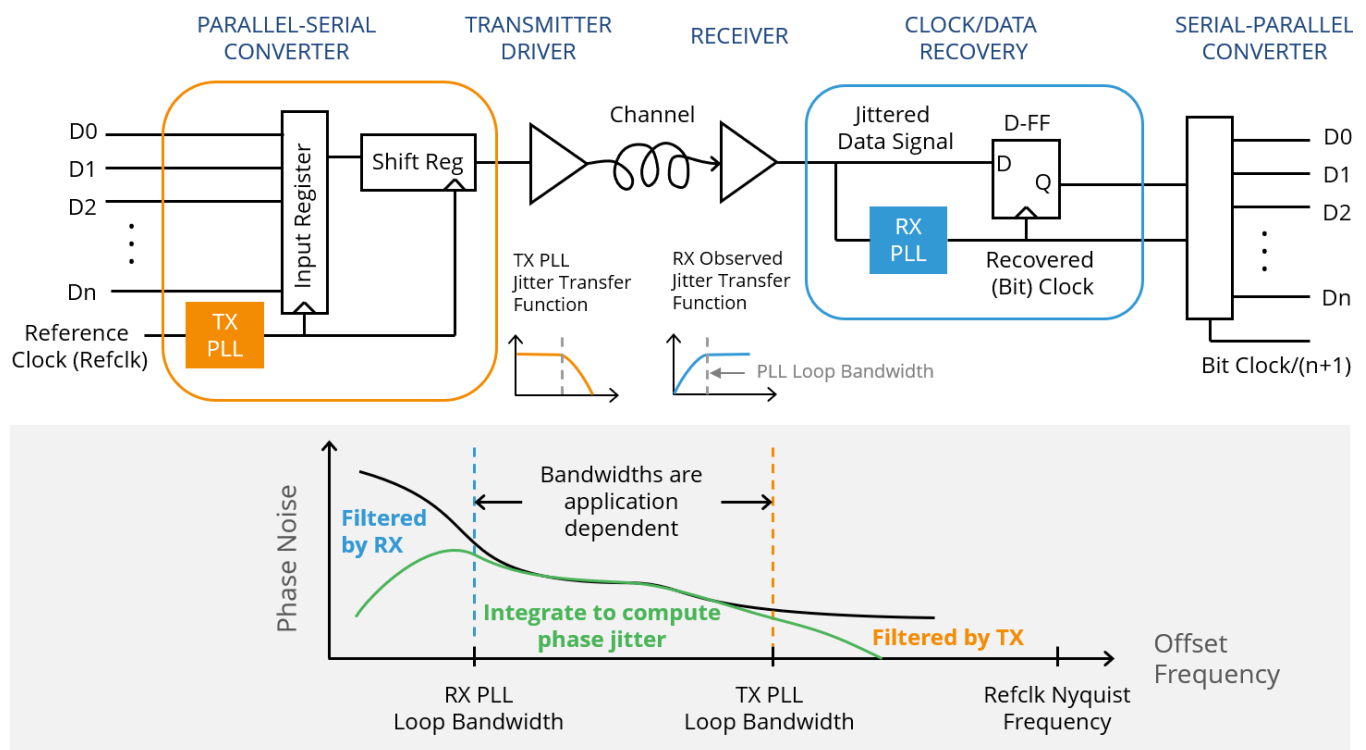


Figure 8. A generic serial link (top) uses a transmit PLL and receive CDR to low and high pass filter, respectively, refclk phase noise. This forms a band-pass system filter (bottom) for computing phase jitter.

The left Figure 9 chart illustrates this methodology of filtering aliased phase noise for a 156.25 MHz clock. Direct phase noise analyzer instruments include an anti-aliasing filter. Thus, to account for aliasing, the phase noise is extended flat to the 3rd harmonic (468.75 MHz in the signal spectrum, or 312.5 MHz in offset frequency spectrum) and the jitter filter is folded across Nyquist-zone boundaries (at

156.25/2, 156.25 and 156.25×3/2 MHz). Then the phase noise data is filtered and integrated to derive phase jitter. The right chart in Figure 9 illustrates a mathematically equivalent process that aliases the extended phase noise below an offset equal to the Nyquist frequency before filtering in the first Nyquist zone^[47].

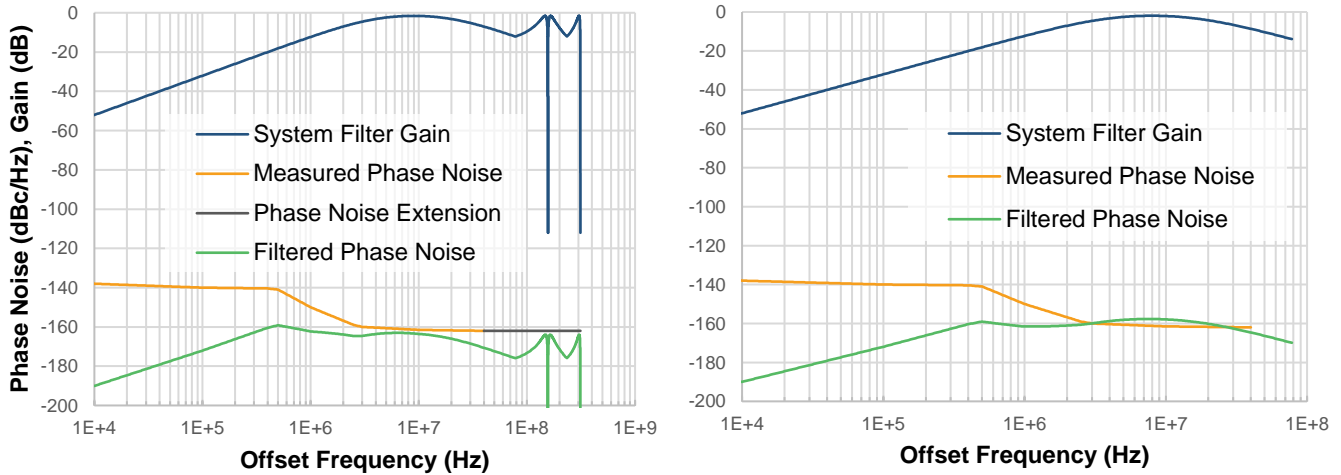


Figure 9. Illustration of two equivalent processes to filter aliased phase noise. The left chart extends (black) the measured phase noise (orange) to the 3rd harmonic, mirrors the filter (blue) across higher Nyquist zones before deriving the filtered phase noise (green). Alternatively, the right chart aliases the extended phase noise (not shown) below the Nyquist frequency before filtering (green). Integrating either green curve yields the same value of phase jitter.

A shorthand label for this methodology is “#-#A” phase jitter where the first and second numbers “#” are replaced with RX CDR and TX PLL bandwidths, respectively, with 20 dB/dec roll offs, and “A” indicates that aliasing is included. For example, “4-16A” phase jitter uses 4 MHz RX and 16 MHz TX bandwidths with aliasing. Here, 4 MHz represents the most common serial standard, Ethernet, which typically specifies a CDR bandwidth of 4 MHz for 10 Gbps and higher link rates, and 16 MHz represents a worst-case

estimate for TX PLL bandwidth (the PLL becomes unstable at higher bandwidths). This shorthand terminology makes it easy to describe variations. For example, “2-10A” phase jitter describes the same methodology but for 2 MHz RX CDR and 10 MHz TX PLL bandwidths. In practice, the exact bandwidths are application dependent, and “4-16A” is simply chosen here to represent the most common application (Ethernet).

Functional Description

SiT95145 is a jitter attenuating frequency translation device that offers four independent fractional PLLs. The output high frequency voltage-controlled oscillator (VCO) associated with each PLL is mapped to the 10 outputs in a very flexible fashion. This offers a very flexible frequency translation arrangement with independent control of each PLL in terms of jitter attenuation, bandwidth control and input clock selection with redundancy.

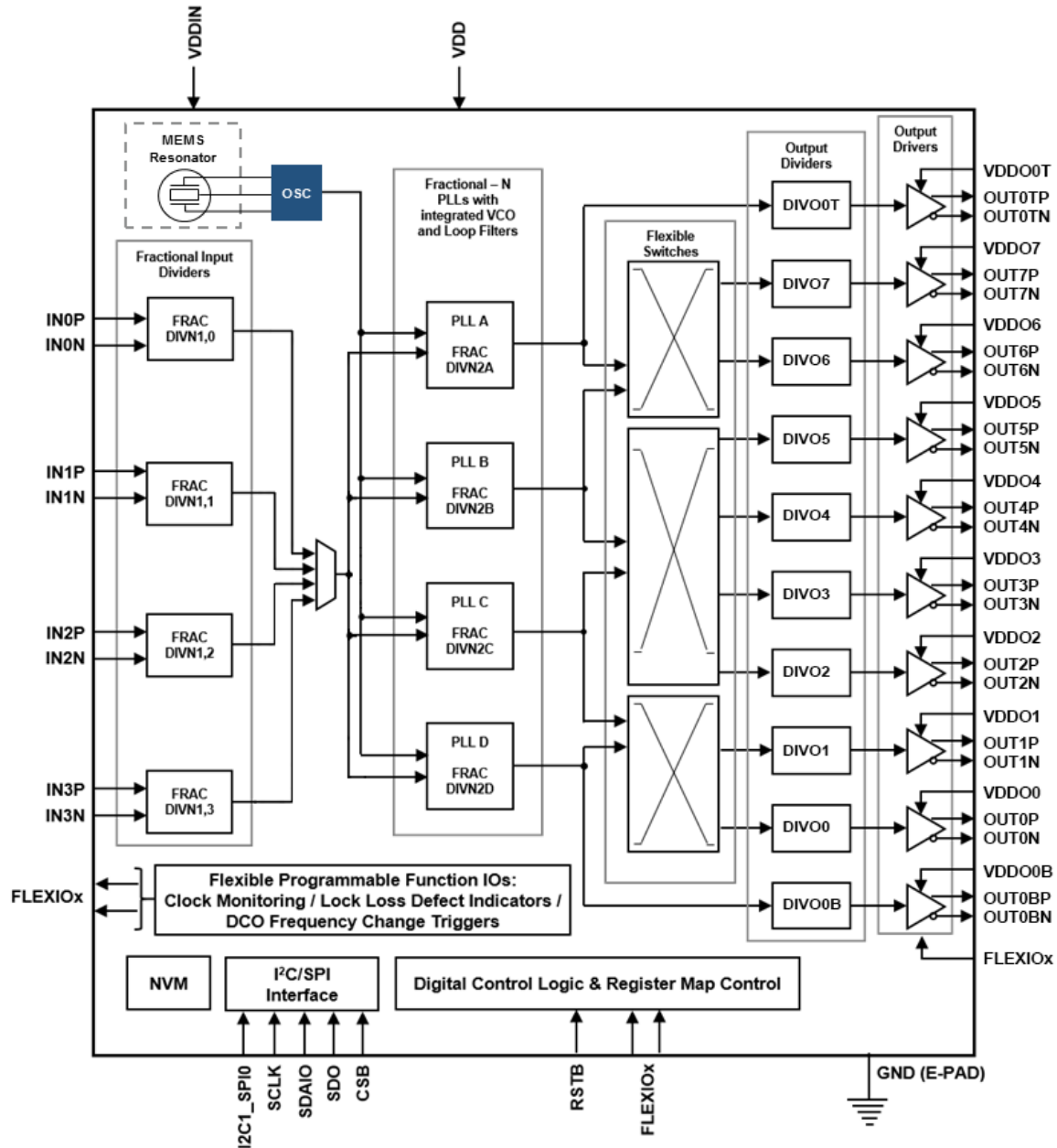


Figure 10. Overall Architecture

The digital architecture of the chip is partitioned into a master digital controller and seven slave controllers. The master controller and each of the seven controllers has an associated volatile programmable interface (PIF). The overall programmable interface PIF structure is a register map that is divided into several pages according to function. Each controller (master and slaves) has an associated unique Page number. Each Page has an independent 8 bit addressable PIF memory.

In all the pages, the last address, FF, holds the current page number and is reserved for changing the page. The current page to be communicated with can be set by writing the page number in hexadecimal form {0x00, 0x01, 0x02, 0x03, 0x0A, 0x0B, 0x0C, 0x0D} corresponding to pages {0, 1, 2, 3, A, B, C, D} in the address FF on any page. [Table 19](#) shows a summary of the PIF contents residing on each page.

Table 19. PIF Description

| Page | What? | Summary of Contents |
|------|--------------|--|
| 0 | Master | All Generic Information related to the chip Chip Configuration details Control for the master sequencer FSM MEMS Oscillator Reference Related Information Fuse Pointer for each of the remaining pages |
| 1 | ClkMon Slave | Clock Loss related function Frequency Drift related function |
| 2 | Input Slave | Input 3 / 2 / 1 / 0 related information (Input type, DIVN1 divider configuration) |
| 3 | Output Slave | Flexible Outputs 0T / 7 / 6 / 5 / 4 / 3 / 2 / 1 / 0 / 0B (ODR Standards, DIVO, Programmable delay configurations for each) |
| | | Fixed Outputs 0T / 0B (ODR Standards, DIVO, Programmable delay configurations for each) |
| A | PLL A Slave | All PLL related functionality |
| B | PLL B Slave | All PLL related functionality |
| C | PLL C Slave | All PLL related functionality |
| D | PLL D Slave | All PLL related functionality |

Frequency Configuration

The hierarchy of the clocks, nomenclature of the various frequency dividers, and the clock translation pathways available on the chip are shown in [Figure 11](#).

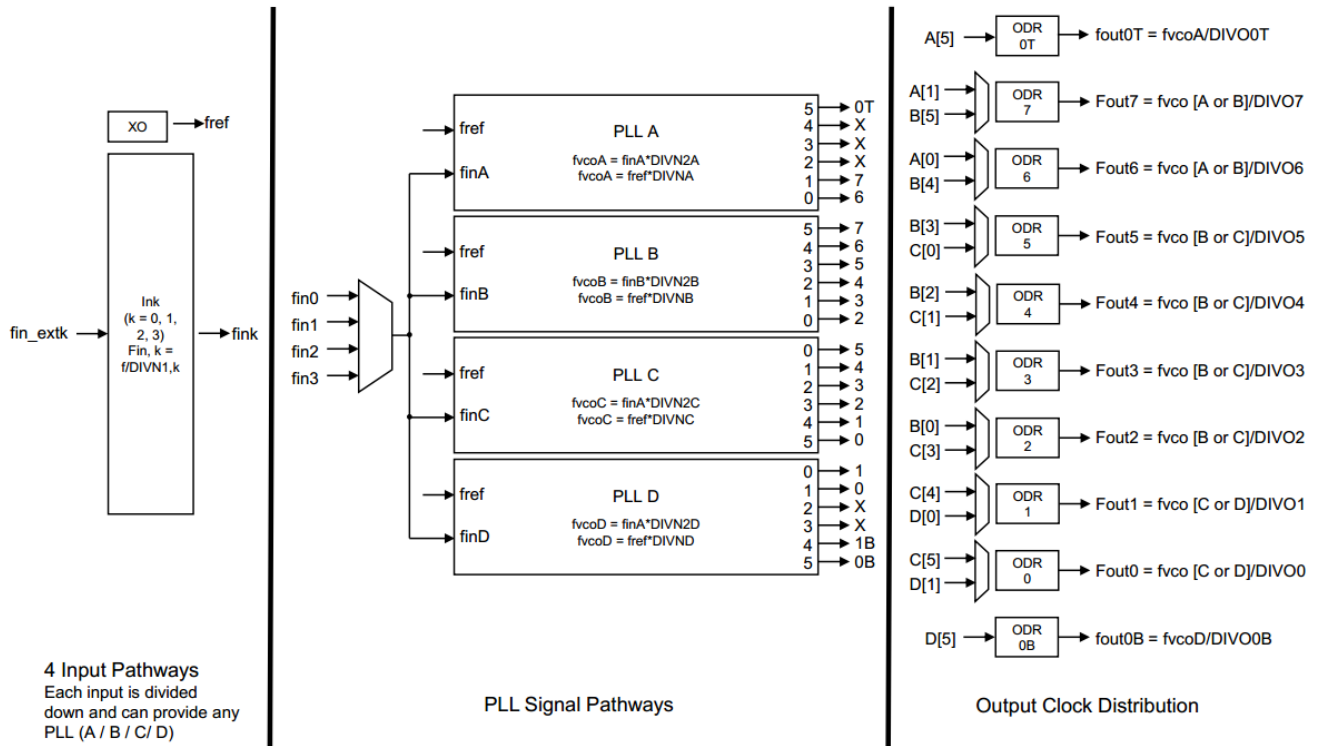


Figure 11. Overall Hierarchy of Clocks

The four input clocks with frequency fin_extk translate to PLL input clocks fin_k following division by the respective input dividers with fractional or integer frequency division ratios $DIVN1k$ where the index $k \in \{0, 1, 2, 3\}$.

See [Figure 12](#). Each of the PLLs chooses one of the four divided input clocks fin_k as its active input clock and sets the priority for up to three spare clocks from the remaining three input clocks if required for hitless switching to a redundant input.

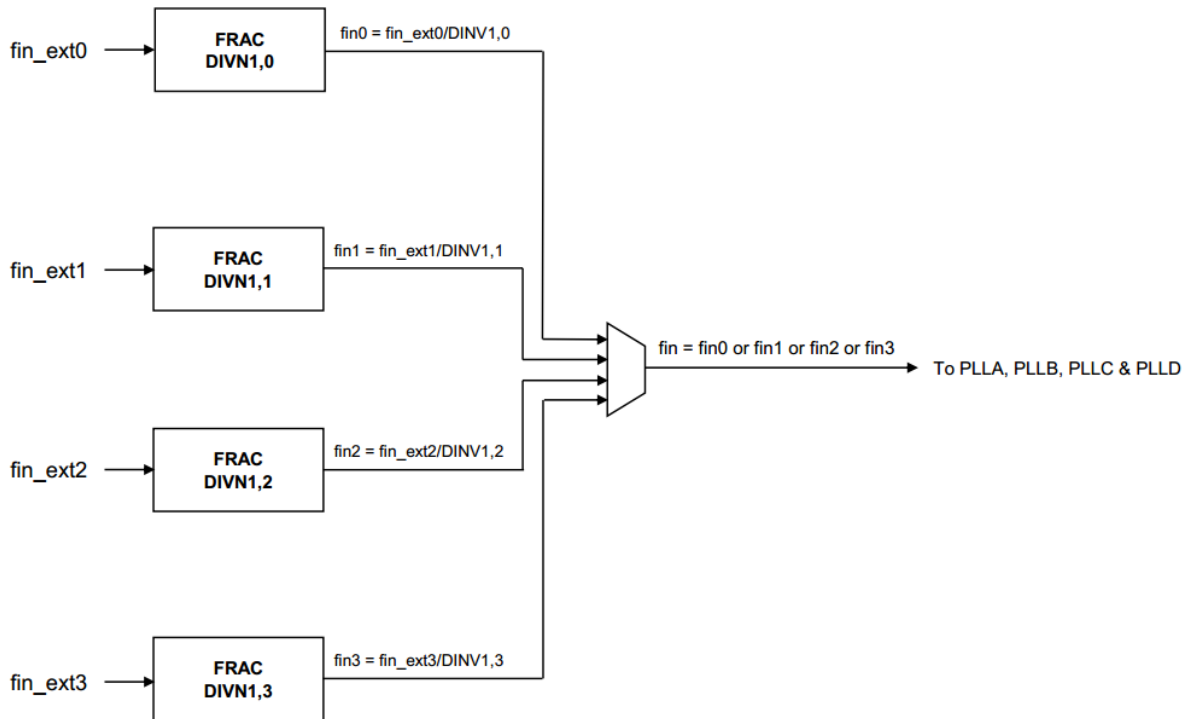


Figure 12. Input Clock Distribution

Each PLL_x ($x \in \{A, B, C, D\}$) has a high frequency VCO whose frequency is determined in the free run mode by $fref$. In the frequency translation synchronized mode, the VCO frequency is corrected from its free run frequency.

Each of the Output Drivers (ODR_j , $j \in \{0:7\}$) chooses an appropriate VCO frequency and divides it using their respective integer divider $DIV0j$ to get the output frequency $foutj$. See [Figure 13](#).

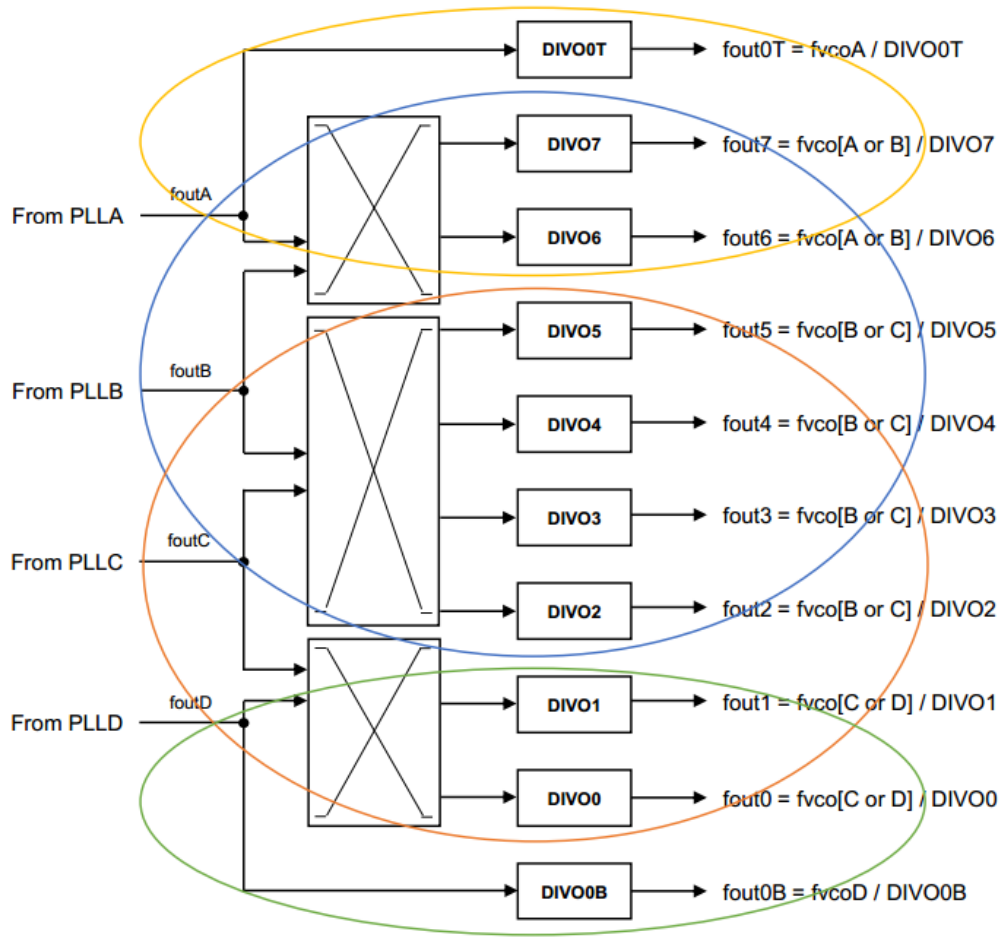


Figure 13. Output Clock Distribution

The choice of the fractional dividers as well as the placement of f_{outj} frequencies at various outputs is defined during chip configuration.

Master and Slaves: Architecture Description and Programming Procedures

The Master controller is the first system to autonomously wake up on the application of power to the chip due to on-chip power on reset circuitry. All generic system information resides in the Master controller memory and it proceeds to wake up the Slaves as required based on this information. The relative wake up sequences of the Master and the various Slaves are described in more detail later in this section after a description of the memory structures. A complete power up of the chip is also emulated with the release of an active low hard reset (RSTB) from pin while selective Master and Slave sub-system resets are enacted from software using the serial interface (I²C / SPI).

The Master memory structure is shown in Figure 14. It contains a one-time programmable non-volatile memory (NVM) that stores the settings for the chip associated with the master controller. The master controller also contains a

volatile PIF bank (NVMCopy) that has an exact copy of the NVM at every chip power up. This volatile PIF (NVMCopy) is the memory that is addressable using the serial interface (I²C / SPI) on Page 0 and can be overwritten from the I²C / SPI interface. The “Chip Settings” is the memory space that is not addressable from the I²C / SPI control and is the actual control for the chip. The NVM contains a two bit “Lock Pattern” that can be set to “10” or “01” to ‘lock’ the chip configuration once the final configuration is determined and wake up of the entire chip is desired in this configuration. Additionally, there is a bit in the NVM that is an active low indicator of a manual wake up. This bit set to “1” along with the ‘lock’ for the configuration leads to an autonomous wake up of the chip using the ‘locked’ configuration. Any number of different configurations can alternatively be tried at all times using only the volatile NVMCopy PIF section. This is useful for evaluations as well as allowing real time programming of the chip in various configurations with complete flexibility. The Master Controller finite state machine (FSM) described later in this section controls the device behavior in accordance with the configuration in this memory structure and as per the wake up mode.

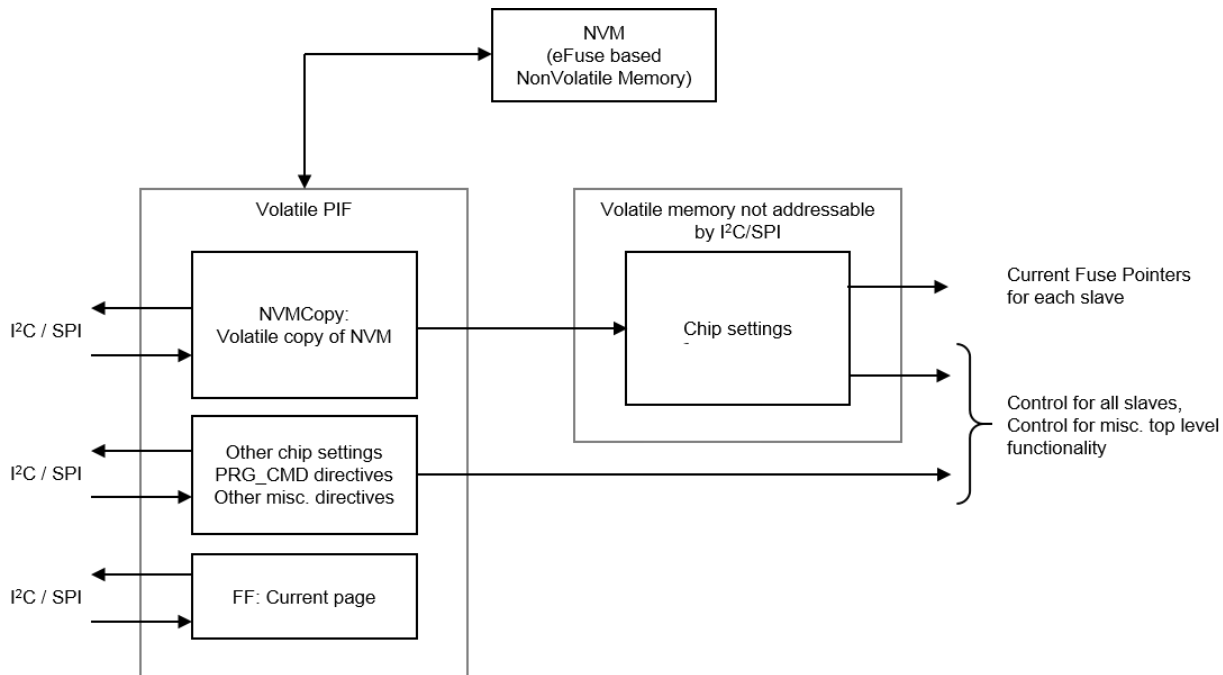


Figure 14. Master Memory Structure

The memory structure for each slave is shown in Figure 15 and is similar in construction to the master controller memory structure with some minor differences. The NVMCopy volatile PIF for the slave is addressable by the serial interface with the unique Page number associated with the slave. The “Slave Settings” is the memory space that is not addressable from the I²C / SPI control and is the actual control for the slave. Each Slave

has a two time programmable NVM by virtue of two copies of the NVM memory. This makes the slave settings two time programmable with the fuse pointer from the master controller determining which of the two NVM banks is used. The presence of two NVM banks is transparent to the slave controller since the current pointer which determines which of the two NVM banks is used is set by the master controller independently.

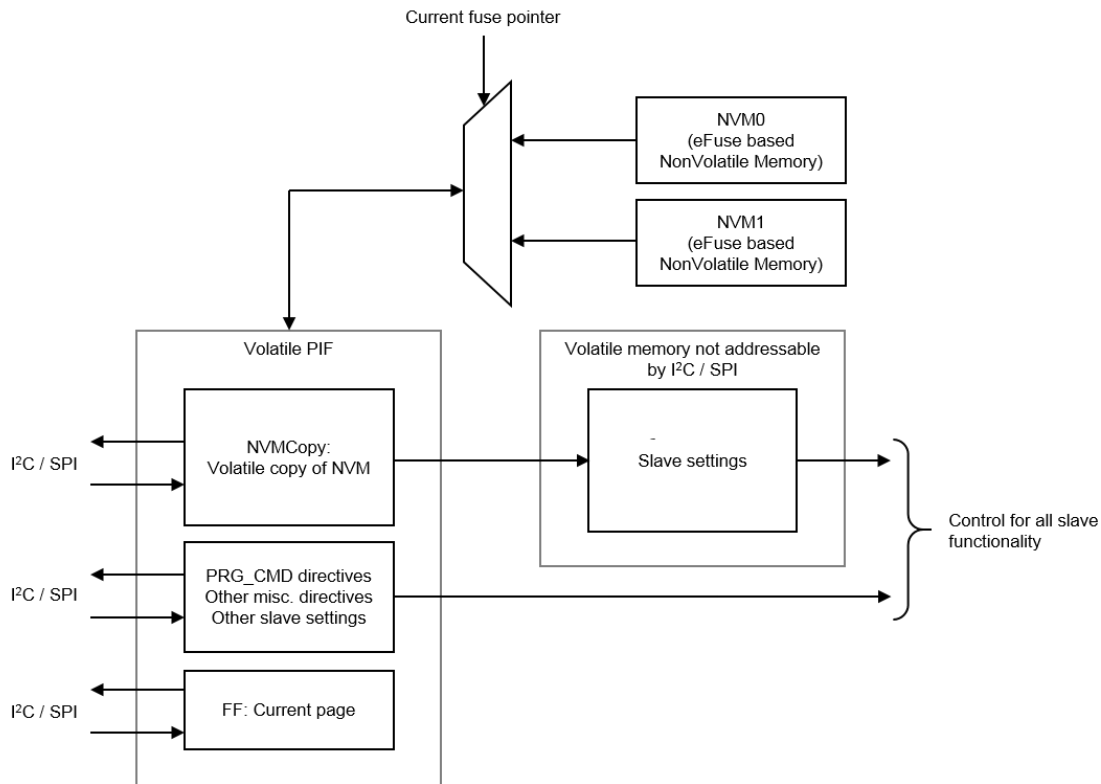


Figure 15. Slave Memory Structure

The Master Wake-Up Finite State Machine (FSM) is shown in more detail in [Figure 16](#). At every power up of the device (or release from hard reset), the power-on-reset circuitry resets all systems and then autonomously releases only the master controller from reset. The NVM contents are copied to the NVMCopy volatile space on Page 0 which is in turn copied to the “Chip Settings”. The master controller now decides if the chip configuration is locked and it is an autonomous wake up of the entire chip or if a manual wake up is desired through the PIF based on the contents in the “Chip Settings”.

In case a “Lock” is detected and an autonomous wake up is desired, the Master controller proceeds to enable the MEMS oscillator and associated fref pathways followed by the Slave systems in a pre-determined sequence. This finally leads the chip to the “Active State” with all desired outputs available as a result of all slave systems released from reset by the master controller. This is according to the requested settings that are programmed in the Master and the Slave NVM banks.

For the case where the final chip settings are not frozen hence the “Lock” pattern is not exercised, the master controller FSM reaches the Program Command Wait State (PRG_CMD). The desired chip settings can be written in the NVMCopy on Page 0 using the serial interface and desired slave sub-systems can be enabled. Several PRG_CMD state directives are available that are

exercisable only in this state (refer to [Figure 16](#)). Using these directives, the desired settings written in NVMCopy can now be copied to “Chip Settings” followed by issuing the directive for the FSM to proceed to the “Active” state where each slave can now be manually written with the desired settings and in turn asked to proceed to its “Active” state.

A similar “Lock” pattern is available in the NVM bank of each slave. The currently used NVM bank for a slave (as determined by the current pointer from the master controller) can be locked for the autonomous wake up of each slave. The slave wake-up FSM is shown in [Figure 17](#) and it similarly has a PRG_CMD state with associated directives. On Proceed to Active state directive on the slave, the slave controller wakes up the various blocks in its sub-system with the correct pre-determined sequence.

The NVM bank for the master and each slave can be programmed with a PRG_CMD directive in that state to lock a configuration / setting specific to the respective sub-system.

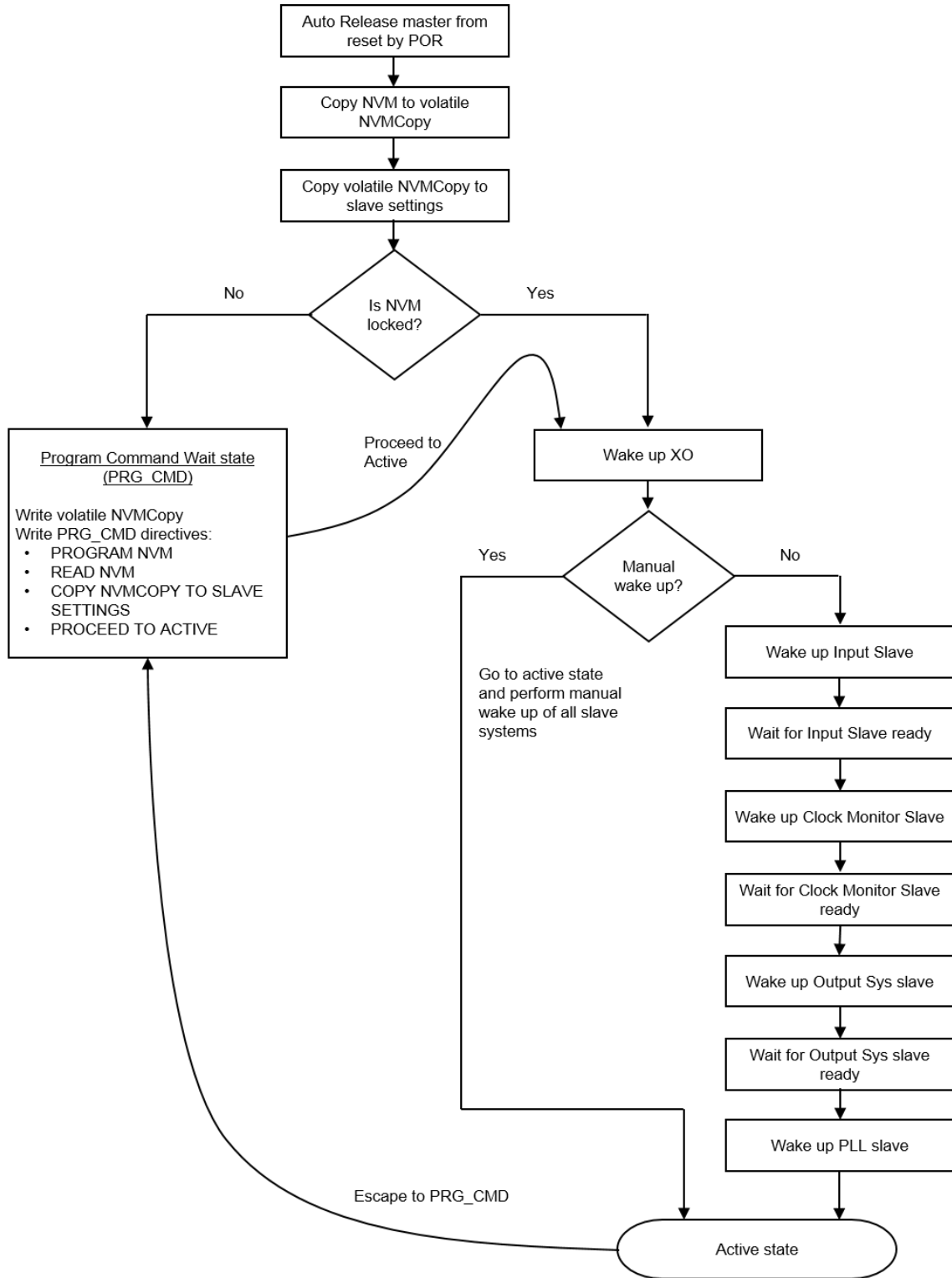


Figure 16. Master Wake-up Finite State Machine

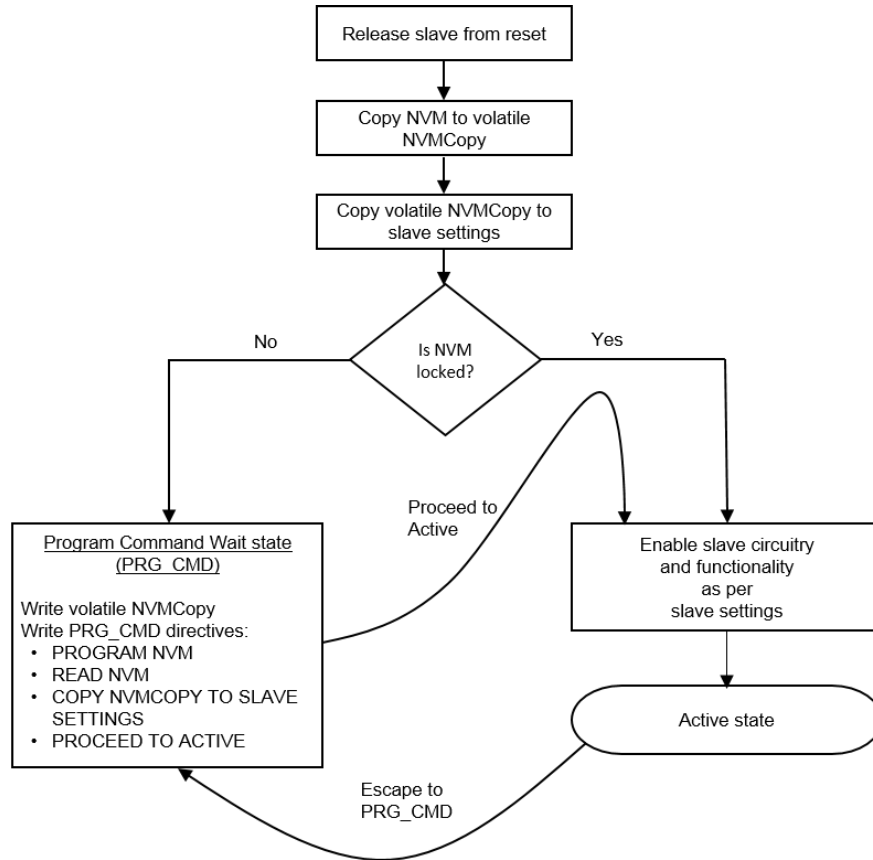


Figure 17. Slave Wake up Finite State Machine

Each FSM (Master and Slaves) allows an escape sequence to go back to PRG_CMD state from its Active State. This can be used to selectively change the settings for that particular sub-system. Such an escape to the PRG_CMD state in the master FSM can be used for example to change current NVM pointers for any of the slaves.

Note that the NVM for the master controller and current NVM for all slaves should be locked after writing desired settings for a completely autonomous wake up of the entire chip. The NVM pointer can then be changed for any slave independently if alternate settings are desired for that slave. In that case, the new NVM is unlocked and can be written with new settings and locked. For evaluations of the chip as well as cases where flexible on-the-fly programmable settings are desired, the chip can be used without engaging the NVM banks at all by using the NVMCopy space for the master and each slave in conjunction with the PRG_CMD directives. It is also possible to lock some of the slaves (to not re-write their settings for each wake up) while use programmable settings for other slaves.

This provides complete flexibility in terms of programming and using the chip in all scenarios.

Input Slave Description

Four independent clock inputs are available on the chip that can be routed to any PLL with complete flexibility. Both single ended and AC coupled differential clock inputs are possible. The input clock receiver settings (to receive a single ended or differential clock) as well as the input clock divider settings are configurable on Page 2 that is assigned to the Input Slave. It is possible to bypass the input clock divider and use the input clock directly as an input to the PLL.

Clock Monitor Slave Description

Various fault monitoring indicators are available on the chip. The Clock Loss and the Frequency Drift indicators are configurable with the Clock Monitor Slave that is accessible on Page 1. The specifications of these fault monitors are indicated in the specifications section of the data sheet.

Defect monitoring on any of the clock monitors can be accessed using multiple techniques. The current status of the defect is available as an Active High defect that can be read from the PIF. The “status” is a current indicator of the defect that is high only during the defect (for example during the time that a Clock Loss event is on-going). Additionally, a sticky indicator of the defect called “Notify” can be enabled in the PIF. In this case, the concerned “notify” bit is high the first time the respective defect occurs and stays high till cleared.

There are multiple FLEXIOs (Flexible IOs) available in the system that can be programmed to monitor individual “notify” signals or a combination of them (as an OR logic). The choice of which fault defect is monitored as an output on the FLEXIO pin is flexible and can be programmed. Additionally there are selected GPIOs that are hard coded for the information for the clock defects.

Fault Monitoring System

The SiT95145 part provides an elaborate arrangement of fault monitoring indicators. There are 4 categories of clock monitoring that are necessary for the chip namely: Clock Loss Monitor (CL), Frequency Drift Monitor (FD), Lock Loss Monitor (LL) and XO Clock Loss Monitor (CL_XO).

Clock Loss (CL) monitors loss of input clocks defined as a pre-determined number of consecutive edges missing.

Frequency Drift (FD) monitors frequency drift of a particular clock against a pre-determined Golden Reference.

Lock Loss (LL) monitors the loss of lock in any PLL by monitoring the difference in frequency between the feedback and input clocks.

XO Clock Loss (CL_XO) monitors the loss of the XO reference that is generated from an external oscillator (XO / TCXO / OCXO).

Each of these categories monitors the health of a particular clock for a certain failure type as illustrated in the name of the clock monitoring category.

For each clock failure observed by the clock monitor block there are two types of indicators provided to the user using the register map:

- 1) Live Failure Bit: There is a bit to indicate the live status of a particular failure. [Status]
- 2) Sticky Failure Bit: For each live failure bit there is a corresponding sticky bit that is set the first time that corresponding failure is encountered and stays set even if the failure has gone away. Only when the user clears the bit does it clear. [Notify]

The status of these can be either read from the register map or from the pins as a dynamic alarm monitoring arrangement. Additionally, sticky notify registers are available which have sticky status read back from the register map for the various defects. These can be selectively chosen to create an INTRB de-assertion on the INTRB pin as well.

An important point to note is that all of the fault monitoring indicators mentioned above that work with respect to the input clock work on the divided input clock post the DIVN1,k dividers. This implies that the fault monitoring indicators use the frequency fink that is input to the PLL ($k \in \{0, 1, 2, 3\}$) post the DIVN1,k divider translation rather than the external frequencies $fin_extk(k \in \{0, 1, 2, 3\})$.

The [Alarm Registers' Information](#) section describes the read back of the alarms for the various fault monitoring arrangements using the chip register map.

Clock Loss Monitors

Each of the 4 inputs (IN0, IN1, IN2, IN3) are monitored for Clock Loss in terms of missing edges to indicate a loss of input signal. The number of edges used to indicate a clock loss (or recovery from a clock loss) is programmable in the Cascade GUI interface allowing for flexibility in choosing these thresholds. In addition there is a programmable “Wait Time” all of which are to be interpreted as follows:

Assertion of Clock Loss-

We declare a CL if “Trigger Edge” number of consecutive edges are missing. The “Trigger Edge” parameter is programmable in the chip GUI.

De-Assertion of Clock Loss-

We declare a ~CL if the clock is back and has less than “Clear Edge” consecutive edges missing. The “Clear Edge” parameter is programmable in the chip GUI.

Wait Time: After the clock is established to have returned, it is ensured that no CL error as defined by the de-assertion threshold occurs for “Val Time” seconds. This valid wait time is programmable using the chip GUI using the “Val Time” parameter which is programmable from the following options: {2m, 100m, 200m, 1} sec. The use of the this valid wait time ensures that sporadic edges in the input clock (such as ones caused by noise on floating nodes or intermittent unstable clock edges) does not de-assert clock loss and it is established over a user determined period of time that the input clock is available and stable.

Frequency Drift Monitors

Any one of the 4 input clocks or the XO clock can be used as the Golden Clock for calculating the frequency drifts of the other 4 clocks. The Golden Clock can be chosen in the GUI and is used as the “0 ppm” Reference Clock for all monitoring. Fine Frequency Drift has a step size of ± 2 ppm.

Fine Frequency Drift has a range of ± 2 to ± 510 ppm and an independent threshold is programmable for “Set” (for setting the FD monitor) and for “Clear” (for clearing the FD monitor).

Fine Frequency Drift has an implicit hysteresis with resolution of ± 2 ppm since the same range is available for the FD assertion and de-assertion. Use of hysteresis prevents unwanted oscillation of the FD monitor output at the decision threshold and is recommended for robust operation. The value of the FD threshold hysteresis is implicit in the choice of the set and clear thresholds.

The Fine Frequency Drift monitors provide precise information for input clock frequency drift. However, since the resolution of the measurement determines time for the measurement – an alternate faster measurement mechanism for drift is needed. This is Coarse Frequency Drift which has coarser measurement but is fast. It is available for cases where the drift is very fast in the input frequency and is programmable from options as shown below.

Coarse Frequency Drift has a step size of ± 100 ppm.

Coarse Frequency Drift has a range of ± 100 to ± 1600 ppm and an independent threshold is programmable for “Set” (for setting the FD monitor) and for “Clear” (for clearing the FD monitor).

Coarse Frequency Drift has an implicit hysteresis with resolution of ± 100 ppm since the same range is available for the FD assertion and de-assertion. Use of hysteresis prevents unwanted oscillation of the FD monitor output at the decision threshold and is recommended for robust operation. The value of the FD threshold hysteresis is implicit in the choice of the set and clear thresholds.

Important Note regarding the above monitors with respect to clock switch in the PLL:

Normally the CL monitor is used for ascertaining a clock is lost for the PLL to switch to a secondary reference or proceed to Holdover. However, the Fine and/or Coarse FD monitors can also be used in addition to the CL monitor to cause a PLL switch. This implements an “OR” logic for the FD Monitors to be used in addition to the CL monitors for triggering a PLL input clock switch or entry to Holdover. This is programmable as an option in the GUI.

Lock Loss Monitors

Lock loss is programmable for each PLL with lock loss triggered if the frequency of the input reference to the PLL phase detection arrangement and the feedback clock to same PLL are different as per the programmed assertion and de-assertion thresholds.

The Set threshold for asserting the LL monitor is programmable from $\{\pm 0.2, \pm 0.4, \pm 2, \pm 4, \pm 20, \pm 40, \pm 200, \pm 400, \pm 2000, \pm 4000\}$ ppm while the Clear threshold for de-asserting the LL monitor is programmable from $\{\pm 0.2, \pm 0.4, \pm 2, \pm 200\}$ ppm. A pre-determined level of hysteresis is implicit by choosing appropriately the set and clear thresholds for the LL monitor.

Additionally from the point of view of LL de-assertion, there is a delay from the point in time that lower than the specified ppm value is achieved to the point where the actual LL is de-asserted to the user such that LL never asserts during this delay period. The choice of this delay is with a timer that ensures that the delay is in line with the BW of the PLL loop. It is fully programmable from the GUI and is useful to ensure complete settling of the PLL without un-necessary toggling before LL de-assertion.

Output Slave Description

The Output Slave accessible on Page 3 is used to configure the output divider (DIVO) and output standard for each output individually. The output load and terminations for each differential output standard are shown in the Output Terminations section of the data sheet. The LVDS and LVDS Boosted modes are recommended for AC coupled termination loads with the termination at the far end. Additionally, an internal termination mode for differential outputs is available where the resistive terminations are internally provided and a differential output is available that can be AC coupled to a clock receiver. The differential clock output pins are shared for LVCMOS outputs as well. LVCMOS outputs can be either enabled on both outputs individually or on any one of the two differential outputs {OUTjP, OUTjN}. The LVCMOS outputs can be used in-phase or out-of-phase on {OUTjP, OUTjN} in case both outputs are chosen. Out of phase LVCMOS toggling on the complementary outputs is recommended for best spur performance.

PLL Bandwidth

Each PLL Slave independently chooses the Bandwidth for jitter attenuation from 1 mHz to 4 kHz. This is the bandwidth that is normally used for steady state operation.

Fast Lock

An independent choice for a fast bandwidth is also available that can be used for speeding up the initial lock. After the PLL lock is achieved and the system is in the synchronized mode, the bandwidth is automatically transitioned to the steady state jitter attenuation bandwidth. This feature avoids the abnormally large wake up times that may be needed for very low PLL bandwidths. For stability considerations of the PLL, the fast lock bandwidth or regular bandwidth for the PLL should be no larger than $1/100^{\text{th}}$ of the input frequency at the input of the PLL (post the DIVN1k dividers).

Operation Modes

There are three distinct modes of operation of the PLL: free run mode, synchronized mode and holdover mode. The frequency of the high frequency VCO in the PLL is determined by the specific mode of operation. The state diagram with the modes of operation shown in Figure 18.

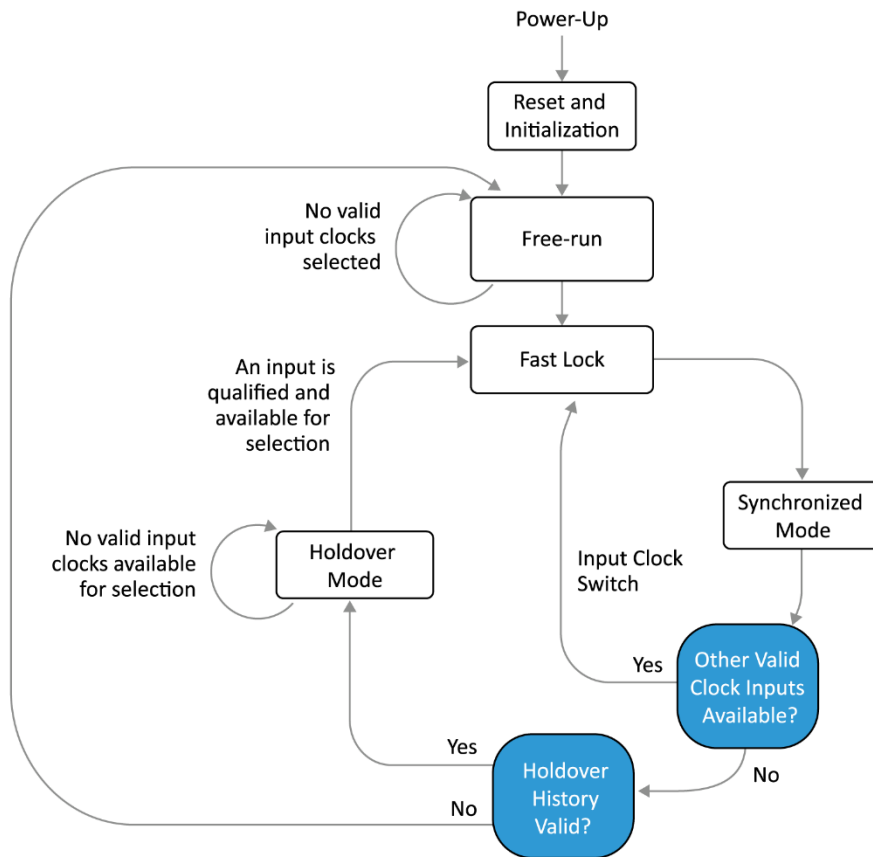


Figure 18. Operation Modes

Device Start-Up and Initialization

After powering up the device begins an initialization period where it downloads default register values and configuration data from NVM. No clocks will be generated until the initialization is done. Communicating with the device through the serial interface is possible once this initialization period is complete. A complete power up of the chip is also emulated with the release of an active low hard reset (RSTB) from pin.

Free run mode

The PLL in the free run mode can be described as a MEMS based oscillator where the output frequency is determined by the relation $fVCOx = DIVNx \cdot fref$. This is the mode of operation before the loop is locked to the selected input clock or the mode of operation for the case none of the input clocks is available.

Synchronized mode

After locking to the chosen input clock, the PLL enters the synchronized mode of operation where the output is now locked to the input frequency with the relation $fVCOx = DIVN2x \cdot finx$. The PLL Loop that synchronizes (locks) the output to the input clock has a programmable loop bandwidth between 1 mHz to 4 kHz and is not affected by static or dynamic drifts in the MEMS oscillator based fref frequency. In synchronized mode, the PLL is able to lock to a Gapped Input clock with some edges missing producing a smooth output clock without any gaps with the requested frequency translation from input to output. Frequency translation ratios in this case should be specified with respect to the average input frequency of the gapped clock rather than the faster instantaneous frequency.

Holdover mode

In case the input clock is lost, the PLL locks to the highest priority spare clock available. If all specified input clocks are lost the PLL remembers the correction based on historical average of the input clock as specified in Table 6 to enter the Holdover mode of operation. When the PLL is exiting the Holdover mode the frequency ramp feature can be enabled that ramps the output frequency of the PLL at a slope that is programmable to one of the following 4 settings: {0.2, 2, 20, 40000} ppm/s. The same ramp settings are used for input switching.

DCO Mode operation

The Digitally Controlled Oscillator (DCO) mode of operation is used for changing the output frequency of a PLL using software control on the serial interface or pin control. A pre-defined change in frequency is programmed in the PIF of the respective PLL. After programming, an increase (FINC/FLEXIO13) or decrease (FDEC/FLEXIO8) command can be given on the PIF of the same PLL to make the change in output frequency effective. A low to high transition (as an edge detect) is used for the trigger of the DCO increment or decrement. Any relative change in frequency from as fine as 5 ppt to as coarse as 100 ppm is available with the DCO mode. DCO mode is available in both free run and synchronized modes of operation.

PLL MEMS Oscillator Reference

An internal MEMS oscillator circuitry is used to produce the fref clock for the system.

Inputs

Four independent clock inputs are available on the device that can be routed to any PLL with complete flexibility. Both single ended and AC coupled differential clock inputs are possible. The input clock receiver settings (to receive a single ended or differential clock) as well as the input clock divider settings are configurable on Page 2 that is assigned to the Input Slave. It is possible to bypass the input clock divider and use the input clock directly as an input to the PLL.

Automatic Input Selection

Each PLL chooses its clock priority in terms of the four input clocks. This is programmed in to the Clock Monitor slave memory. The PLL Slave then looks at Clock Loss status from the Clock Monitor slave to lock to the highest priority available clock to lock. Three spare clocks with an order of priority can be specified in case the highest priority active clock is not available.

Manual Input Selection

A forced manual selection of the active clock with no spares is possible. Less than three spares can also be specified making the clock priority arrangement completely flexible in terms of choosing the input clock for operation.

Hitless Input Switching

Phase Build Out Mode of hitless switching ensures that phase transients are not propagated to the output (the phase difference between redundant input clocks is absorbed by the PLL) and desired MTIE characteristics are seen in the output clock. This is the default mode of hitless switching for the PLL. The transition of input clock for a PLL from one clock to another is hitless in nature (with maximum phase hit limited to be less than 50 ps) for the case of the switched input clocks being same in frequency. Hitless switch is also supported for the switched clocks being fractionally related such that the same frequency can be obtained for both clocks at the input of the PLL using the input clock dividers (DIVN1k).

An alternate mode of hitless switching is the Phase Propagation mode where the phase difference between redundant input clocks is not absorbed by the PLL but is rather propagated to the output. The phase difference that is propagated to the output can either be allowed to propagate as per the PLL bandwidth or can be limited to a phase propagation slope that is programmable to one of the following 3 settings: {10, 40, 160} us/s.

Ramped Input Switching

For redundant input clocks to the PLL that are not exactly the same frequency (plesiochronous clocks), the frequency ramp feature can be enabled that ramps the output frequency of the PLL at a slope that is programmable to one of the following 4 settings: {0.2, 2, 20, 40,000} ppm/s. For redundant input clocks to the PLL that are exactly the same frequency, the frequency ramp feature should not be enabled.

Input Clocks

The recommended Input Clocks termination schemes are shown in figures below.

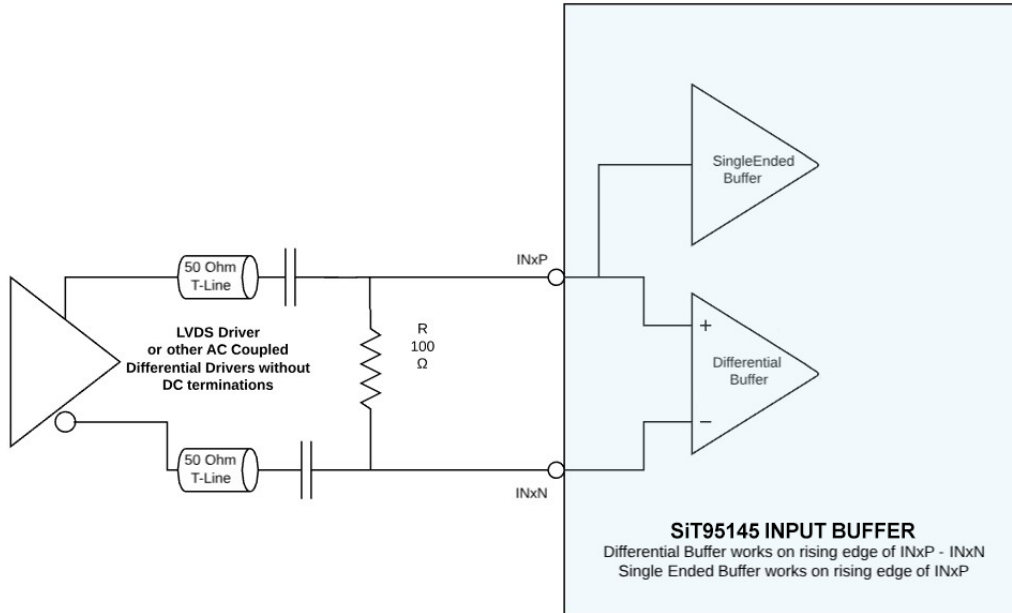


Figure 19. AC Coupled Differential LVDS Input or Other AC Coupled Driver without DC Terminations – Uses Differential Buffer Pathway

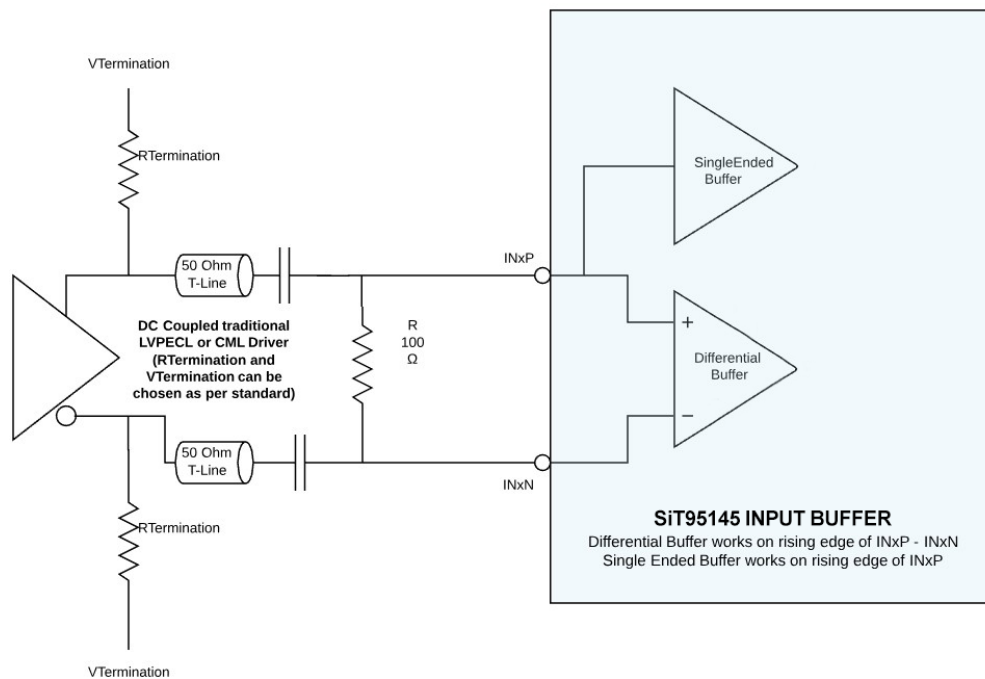


Figure 20. AC Coupled Differential LVPECL or CML (Resistor and Voltage termination is as per the standard) – Uses Differential Buffer Pathway

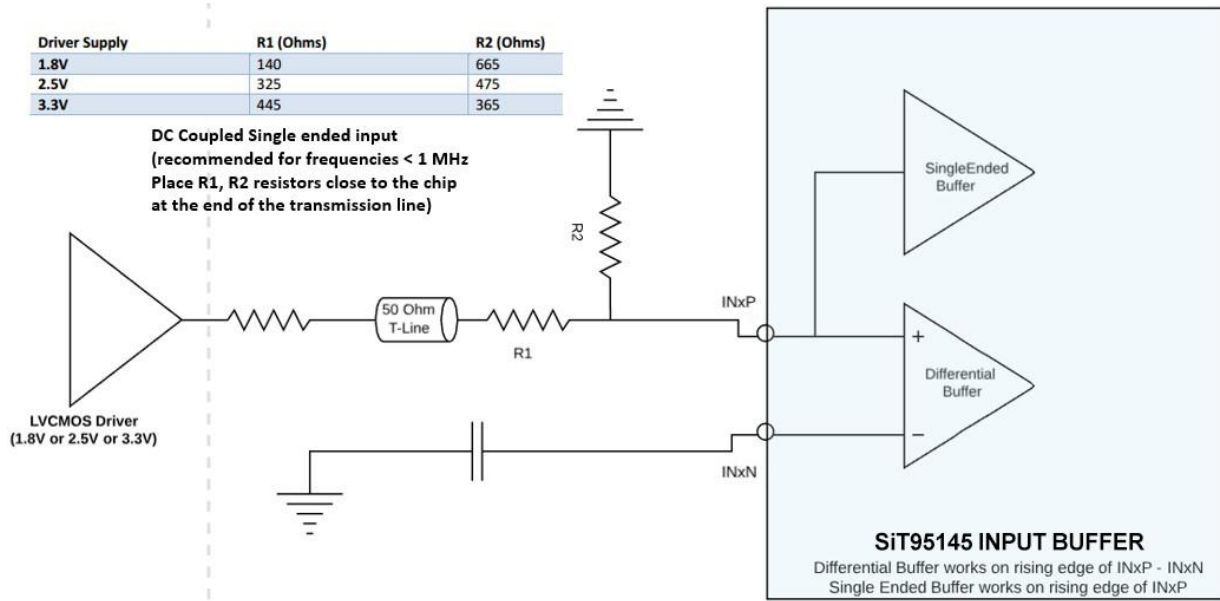


Figure 21. DC Coupled Single Ended Driver – Uses Single Ended Buffer Pathway^[48]

Note:

- 48. Recommended for non-standard duty cycle applications. Please refer above table for the recommended resistor values for frequencies < 1 MHz.

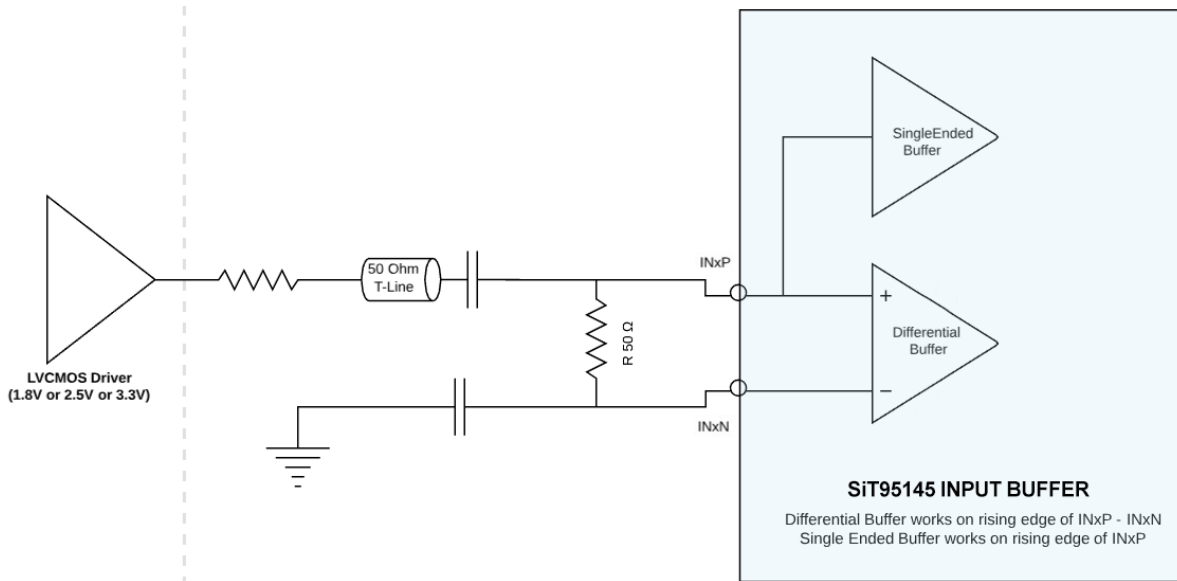


Figure 22. AC Coupled Single Ended Driver with 50 Ω Termination on receiver (chip) side

Note

- 49. Uses Single Ended Buffer pathway in AC coupled mode.

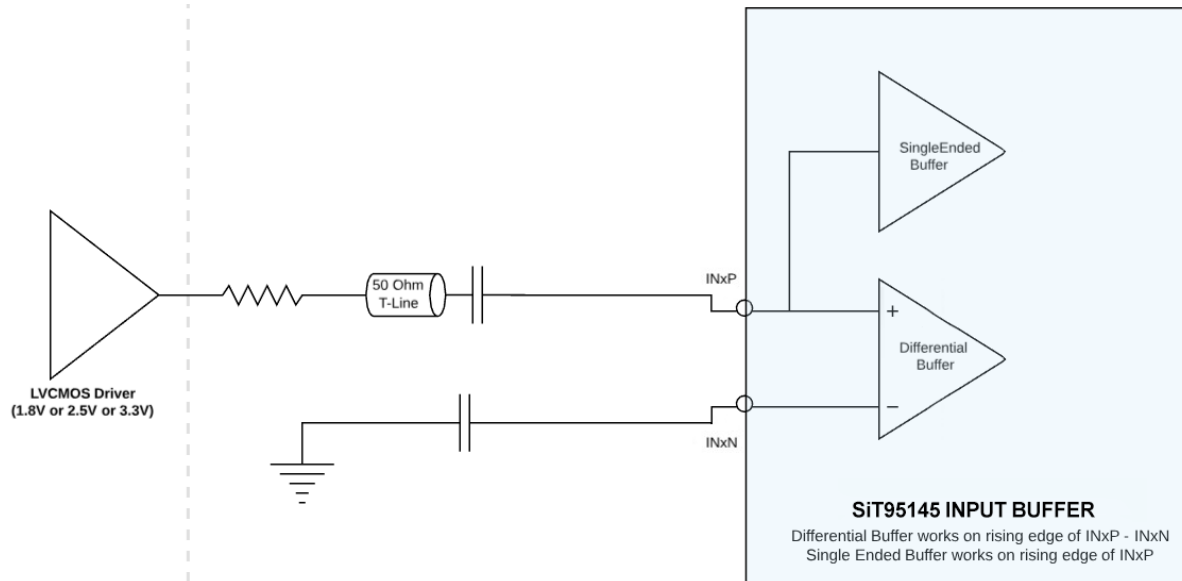


Figure 23. AC Coupled Single Ended LVC MOS input without 50 Ω Termination

Note:

- 50. Uses Single Ended Buffer pathway in AC coupled mode. The LVC MOS driver in this case needs to ensure source termination to match to the transmission line.

Clock Monitoring

Various fault monitoring indicators are available in the device. The Clock Loss indicator is configurable with the Clock Monitor Slave that is accessible on Page 1. The specifications of these fault monitors are indicated in [Table 10](#).

Defect monitoring on any of the clock monitors can be accessed using multiple techniques. The current status of the defect is available as an Active High defect that can be read from the PIF. The “status” is a current indicator of the defect that is high only during the defect (for example during the time that a Clock Loss event is on-going). Additionally, a sticky indicator of the defect called “Notify” can be enabled in the PIF. In this case, the concerned “notify” bit is high the first time the respective defect occurs and stays high till cleared.

PLL Lock Loss is another fault monitor whose specifications are available in [Table 10](#). Various programmable thresholds are available that can be used to detect lock loss in the PLL. Lock loss is indicated by the programmable drift between the frequency of the input clock for the PLL and the divided VCO clock. This defect can be tracked with status, notify and on the FLEXIOs.

There are multiple FLEXIOs (Flexible IOs) available in the system that can be programmed to monitor individual “notify” signals or a combination of them (as an OR logic). The choice of which fault defect is monitored as an output on the FLEXIO pin is flexible and can be programmed.

Fault Monitoring System

The SiT95141 part provides an elaborate arrangement of fault monitoring indicators. There are 4 categories of clock monitoring that are necessary for the chip namely: Clock Loss Monitor (CL) and Lock Loss Monitor (LL).

Clock Loss (CL) monitors loss of input clocks defined as a pre-determined number of consecutive edges missing.

Lock Loss (LL) monitors the loss of lock in any PLL by monitoring the difference in frequency between the feedback and input clocks.

Each of these categories monitors the health of a particular clock for a certain failure type as illustrated in the name of the clock monitoring category.

For each clock failure observed by the clock monitor block there are two types of indicators provided to the user using the register map:

- 1) Live Failure Bit: There is a bit to indicate the live status of a particular failure. [Status]
- 2) Sticky Failure Bit: For each live failure bit there is a corresponding sticky bit that is set the first time that corresponding failure is encountered and stays set even if the failure has gone away. Only when the user clears the bit does it clear. [Notify]

The status of these can be either read from the register map or from the pins as a dynamic alarm monitoring arrangement. Additionally, sticky notify registers are available which have sticky status read back from the register map for the various defects. These can be selectively chosen to create an INTRB de-assertion on the INTRB pin as well.

An important point to note is that all of the fault monitoring indicators mentioned above that work with respect to the input clock work on the divided input clock post the DIVN1,k dividers. This implies that the fault monitoring indicators use the frequency fink that is input to the PLL ($k \in \{0, 1, 2, 3\}$) post the DIVN1,k divider translation rather than the external frequencies fin_extk ($k \in \{0, 1, 2, 3\}$).

The [Alarm Registers' Information](#) section describes the read back of the alarms for the various fault monitoring arrangements using the chip register map.

Clock Loss Monitors

Each of the 4 inputs (IN0, IN1, IN2, IN3) are monitored for Clock Loss in terms of missing edges to indicate a loss of input signal. The number of edges used to indicate a clock loss (or recovery from a clock loss) is programmable in the Cascade GUI interface allowing for flexibility in choosing these thresholds. In addition there is a programmable “Wait Time” all of which are to be interpreted as follows:

Assertion of Clock Loss-

A CL is declared if “Trigger Edge” number of consecutive edges is missing. The “Trigger Edge” parameter is programmable in the chip GUI.

De-Assertion of Clock Loss-

A ~CL is declared if the clock is back and has less than “Clear Edge” consecutive edges missing. The “Clear Edge” parameter is programmable in the chip GUI.

Wait Time: After the clock is established to have returned, it is ensured that no CL error as defined by the de-assertion threshold occurs for “Val Time” seconds. This valid wait time is programmable using the chip GUI using the “Val Time” parameter which is programmable from the following options: {2m, 100m, 200m, 1} sec. The use of this valid wait time ensures that sporadic edges in the input clock (such as ones caused by noise on floating nodes or intermittent unstable clock edges) does not de-assert clock loss and it is established over a user determined period of time that the input clock is available and stable.

Lock Loss Monitors

Lock loss is programmable for each PLL with lock loss triggered if the frequency of the input reference to the PLL phase detection arrangement and the feedback clock to same PLL are different as per the programmed assertion and de-assertion thresholds.

The Set threshold for asserting the LL monitor is programmable from {±0.2, ±0.4, ±2, ±4, ±20, ±40, ±200, ±400, ±2000, ±4000} ppm while the Clear threshold for de-asserting the LL monitor is programmable from {±0.2, ±0.4, ±2, ±200} ppm. A pre-determined level of hysteresis is implicit by choosing appropriately the set and clear thresholds for the LL monitor.

Additionally from the point of view of LL de-assertion, there is a delay from the point in time that lower than the specified ppm value is achieved to the point where the actual LL is de-asserted to the user such that LL never asserts during this delay period. The choice of this delay is with a timer that ensures that the delay is in line with the BW of the PLL loop. It is fully programmable from the GUI and is useful to ensure complete settling of the PLL without un-necessary toggling before LL de-assertion.

Outputs

The Output Slave accessible on Page 3 is used to configure the output divider (DIVO) and output standard for each output individually. The output load and terminations for each differential output standard are shown in the [Differential Output Termination Information](#) section. Additionally, an internal termination mode for differential outputs is available where the resistive terminations are internally provided and a differential output is available that can be AC coupled to a clock receiver. The differential clock output pins are shared for LVCMOS outputs as well. LVCMOS outputs can be either enabled on both outputs individually or on any one of the two differential outputs {OUTjP, OUTjN}. The LVCMOS outputs can be used in-phase or out-of-phase on {OUTjP, OUTjN} in case both outputs are chosen.

Differential Output Termination Information

LVPECL

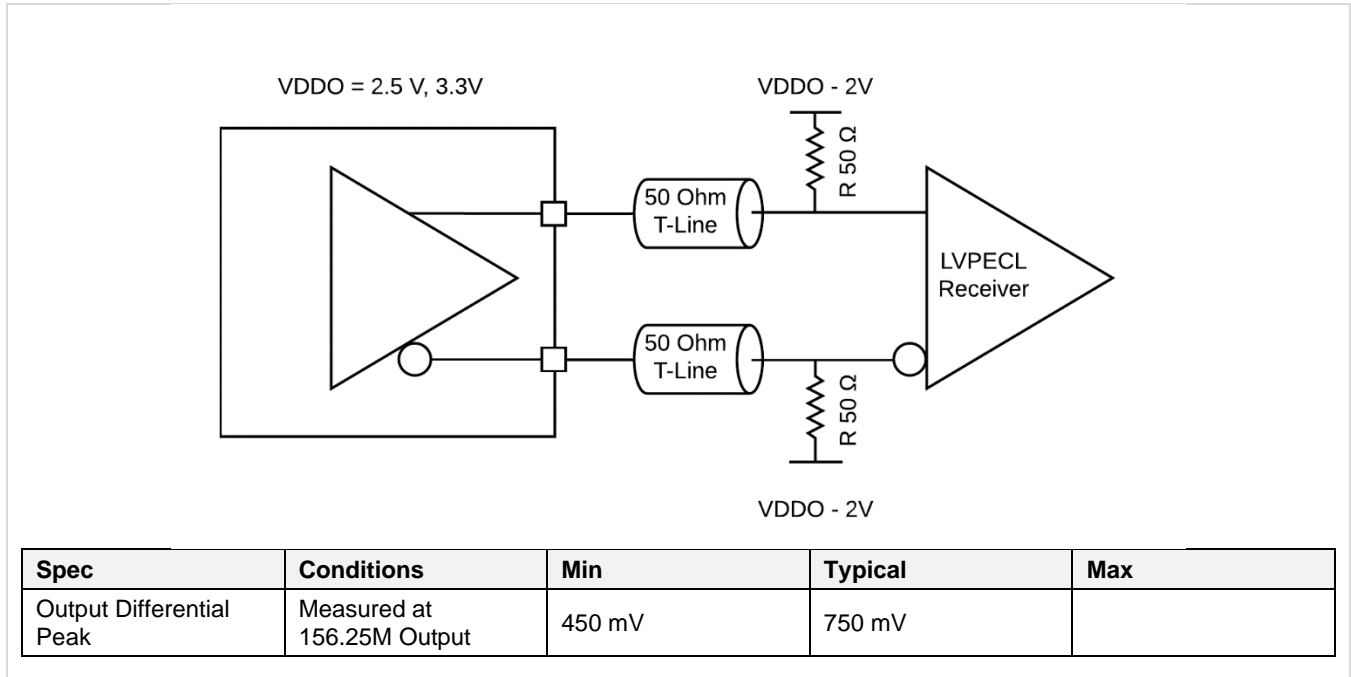


Figure 24. LVPECL with DC-coupled Termination

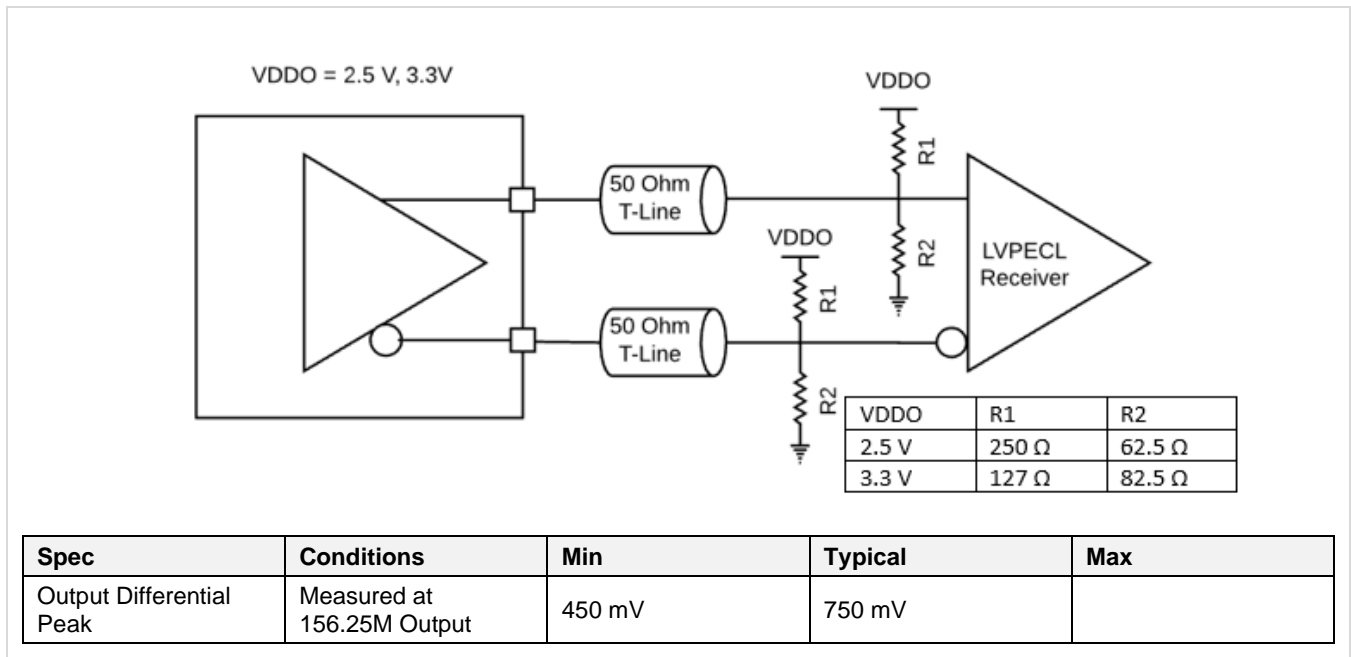


Figure 25. LVPECL DC-coupled Load Termination with Thevenin Equivalent Network

LVPECL (continued)

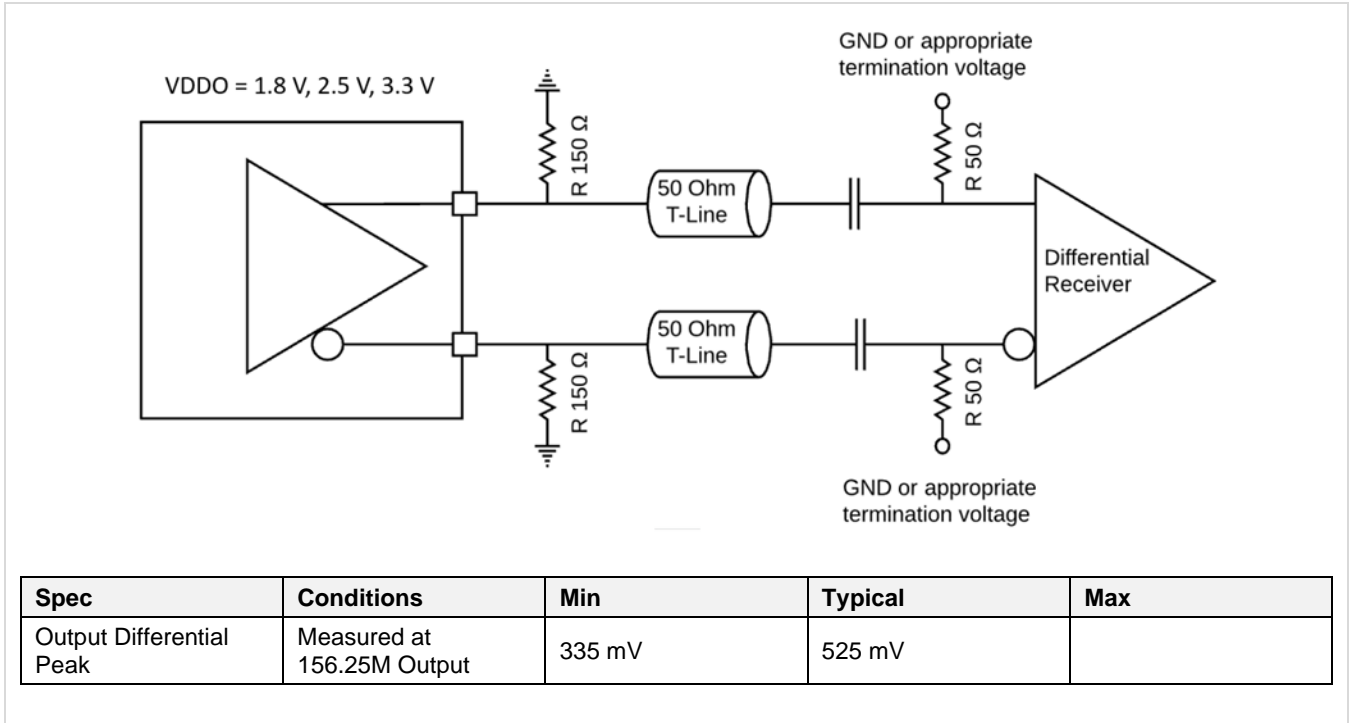


Figure 26. LVPECL2 for the SiT95145 Output Driver with AC Termination at the Load

Differential Output Termination Information (continued)

LVDS

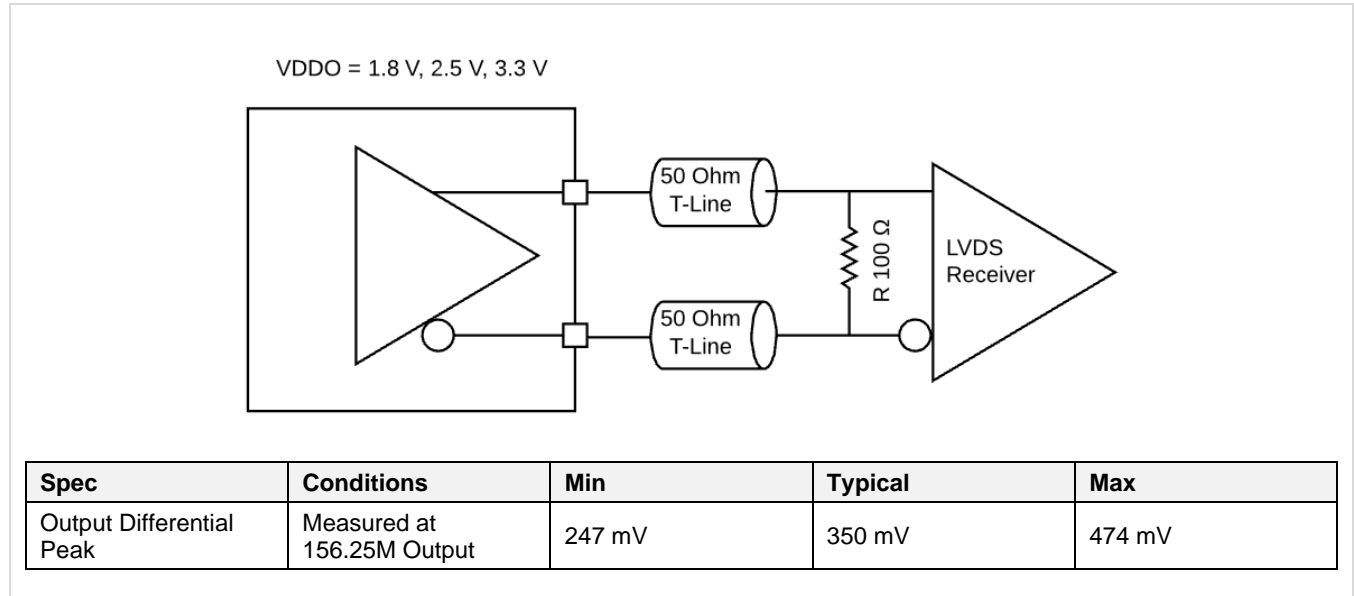


Figure 27. LVDS DC Termination at the Load

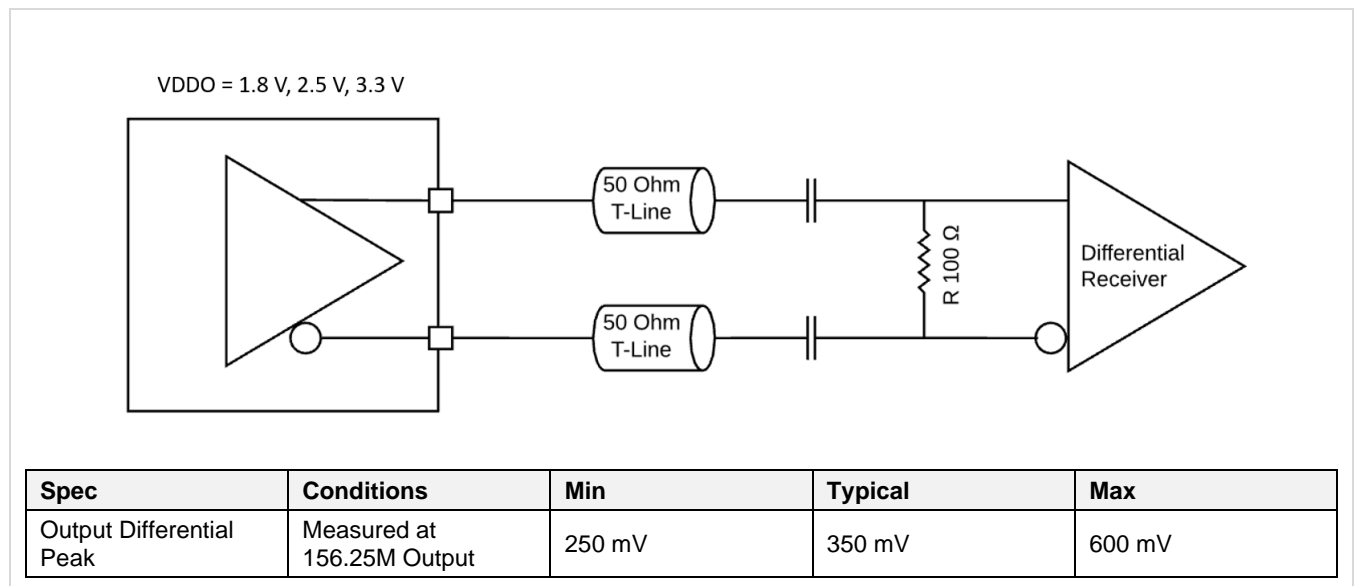


Figure 28. LVDS AC Termination at the Load

LVDS (continued)

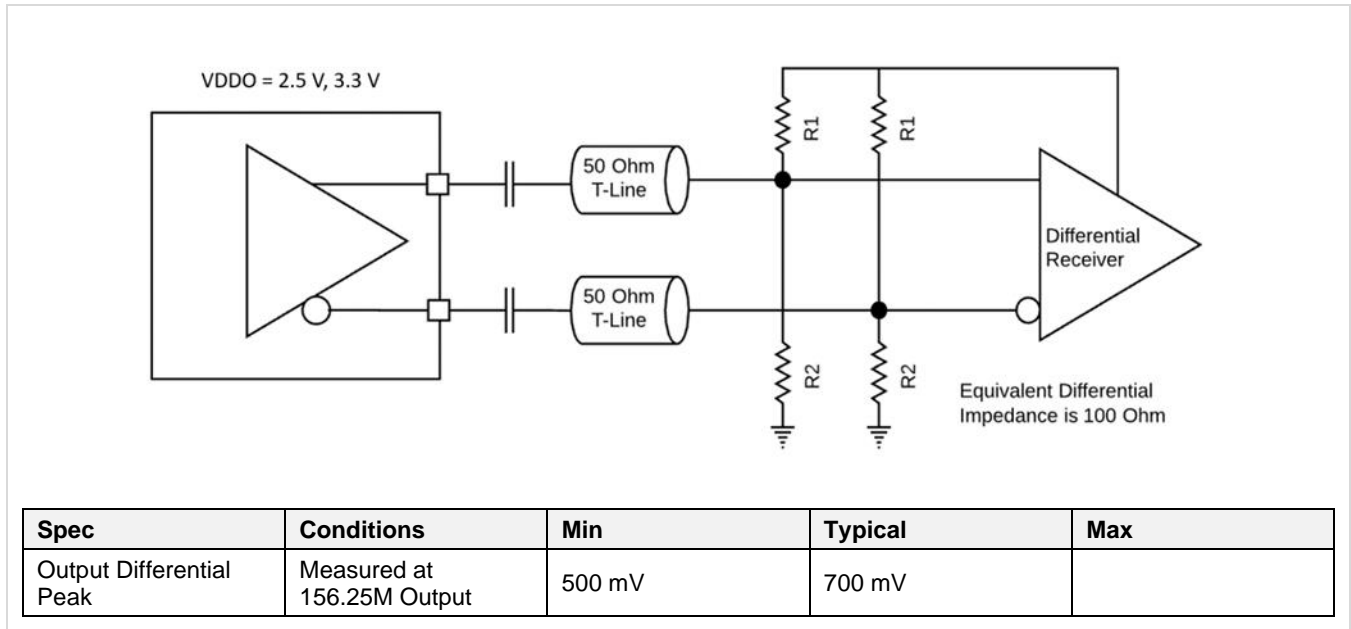


Figure 29. LVDS Boost Standard for the SiT95145 Output Driver with AC Termination at the Load

Differential Output Termination Information (continued)

HCSL

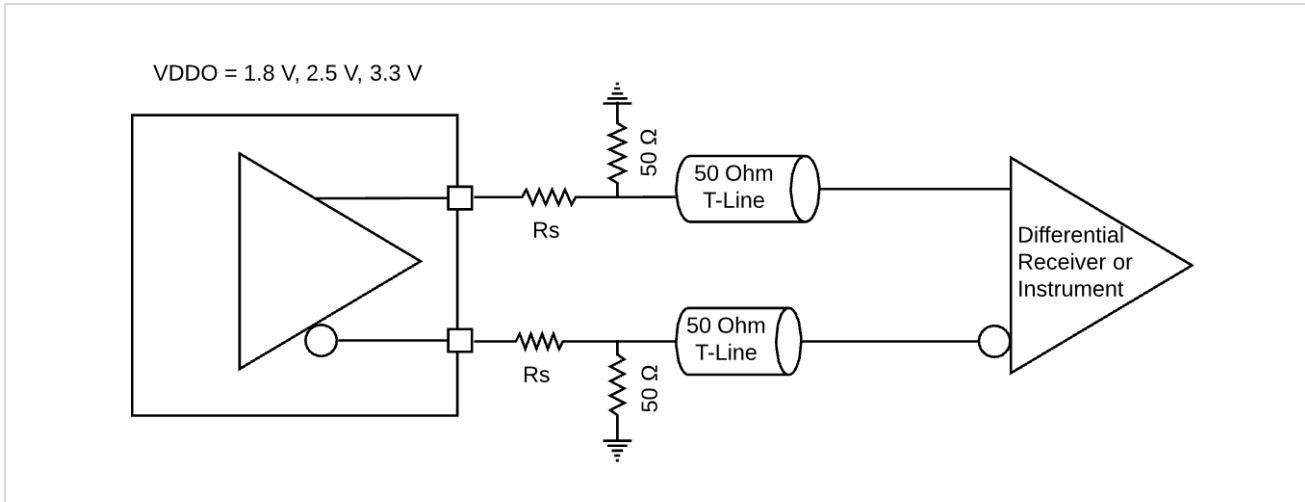
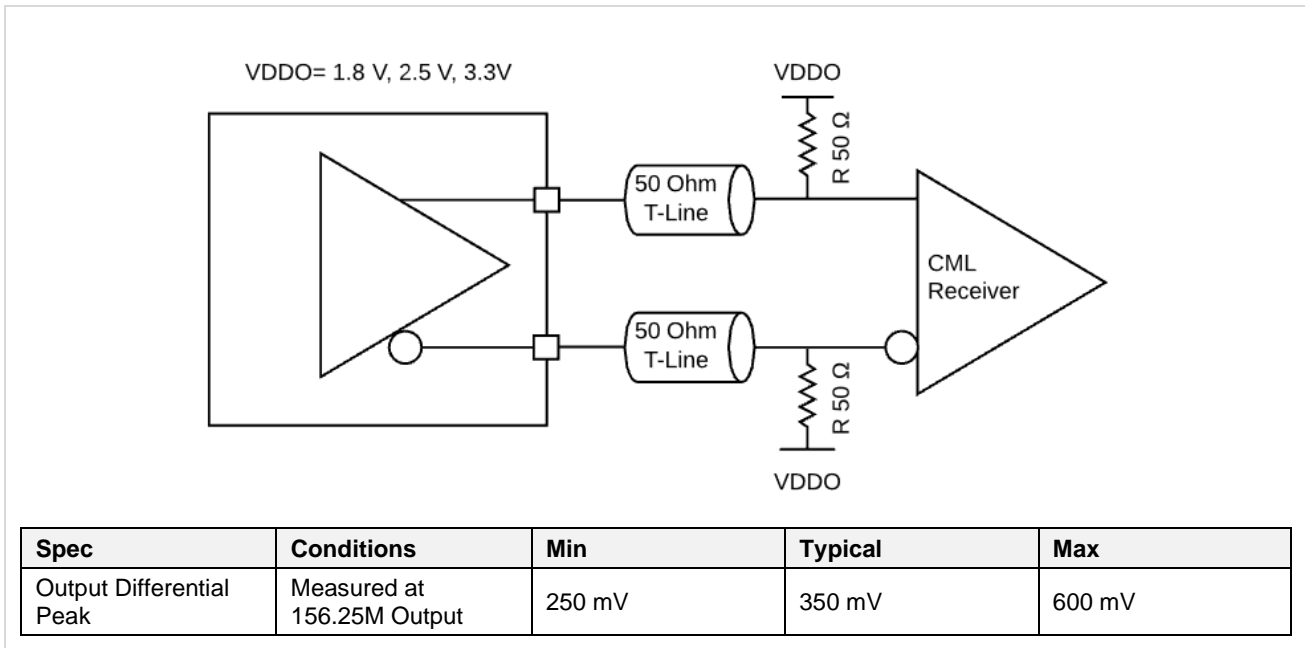


Figure 30. HCSL Interface Termination^[51]

Note:

51. Rs is sometimes used for limiting overshoot, and can be 0 Ohm.

CML



| Spec | Conditions | Min | Typical | Max |
|--------------------------|----------------------------|--------|---------|--------|
| Output Differential Peak | Measured at 156.25M Output | 250 mV | 350 mV | 600 mV |

Figure 31. CML Interface Termination

LVC MOS

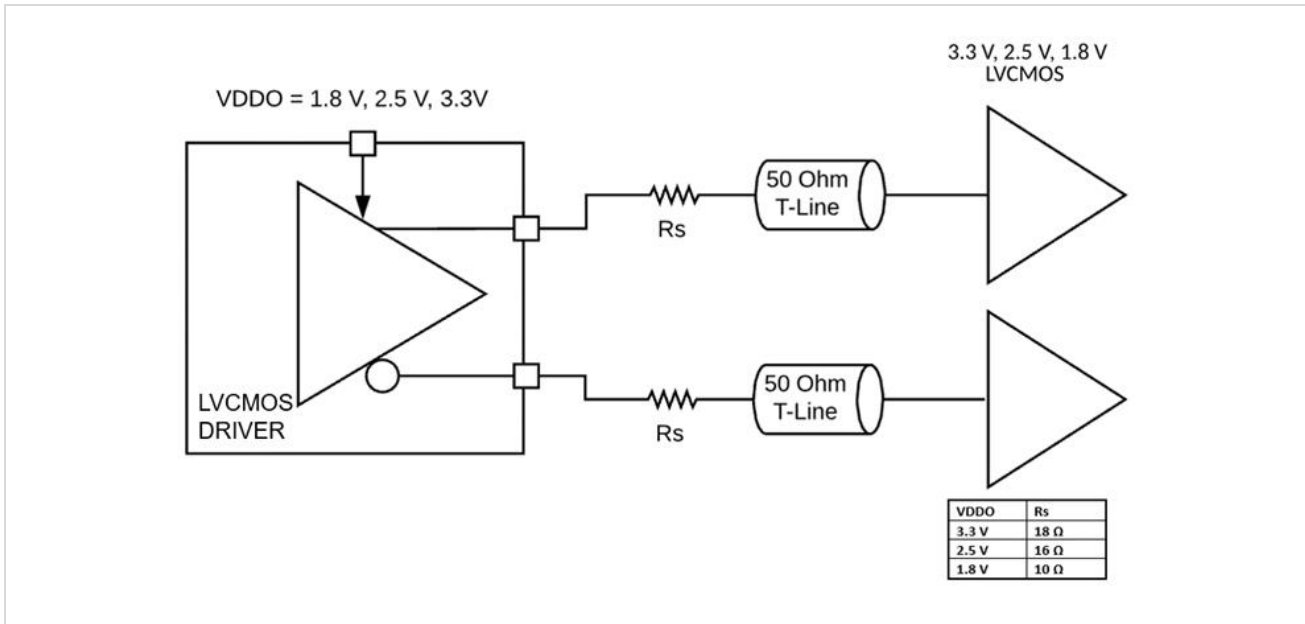


Figure 32. LVC MOS output Termination

Additionally, an internal termination mode for differential outputs is available where the resistive terminations are internally provided and a differential output is available that can be AC coupled to a clock receiver. The differential clock output pins are shared for LVC MOS outputs as well. LVC MOS outputs can be either enabled on both outputs individually or on any one of the two differential outputs {OUTjP, OUTjN}. The LVC MOS outputs can be used in-phase or out-of-phase on {OUTjP, OUTjN} in case both outputs are chosen.

Output Delay Control

By default, all output clocks are phase-aligned. A delay path associated with each of Fractional Input Dividers (see [Figure 33](#)) is available for applications that need a specific output skew configuration. This is useful for PCB trace length mismatch compensation. The resolution of the phase adjustment is 35 ps definable in a range of 7.5 ns. Phase adjustments are register-configurable. All phase delay values are restored to their default values after power-up, hard reset, or a reset using the RSTB pin. Phase delay default values can be written to NVM, allowing a custom phase offset configuration at power-up or after power-on reset, or after a hardware reset using the RSTB pin.

Zero Delay Mode

Zero Delay Mode is used in applications that require minimum delay between the selected input and outputs. Zero Delay Mode is available on any of the 4 PLLs by routing the output clock back to the IN3 input as shown in [Figure 33](#). This ensures minimum delay between the input and output. It can be used for any of the four PLLs. This provides the option to close the feedback loop of the PLL on the PCB and therefore bypasses the internal feedback dividers, thereby cancelling delays introduced by internal dividers and clock distribution pathways. The IN3 input pins are used as the external feedback and one of the outputs from the PLL that is being set up in zero delay mode should be routed to the IN3 differential inputs. It is recommended to use IN0 as the input clock when using IN3 as the external feedback clock in the zero delay mode. The terminations used for IN3 depend on the driver type chosen. The preferred option is to use an LVDS or LVDS boost output ac coupled into a differential 100 Ω termination at the IN3 input side.

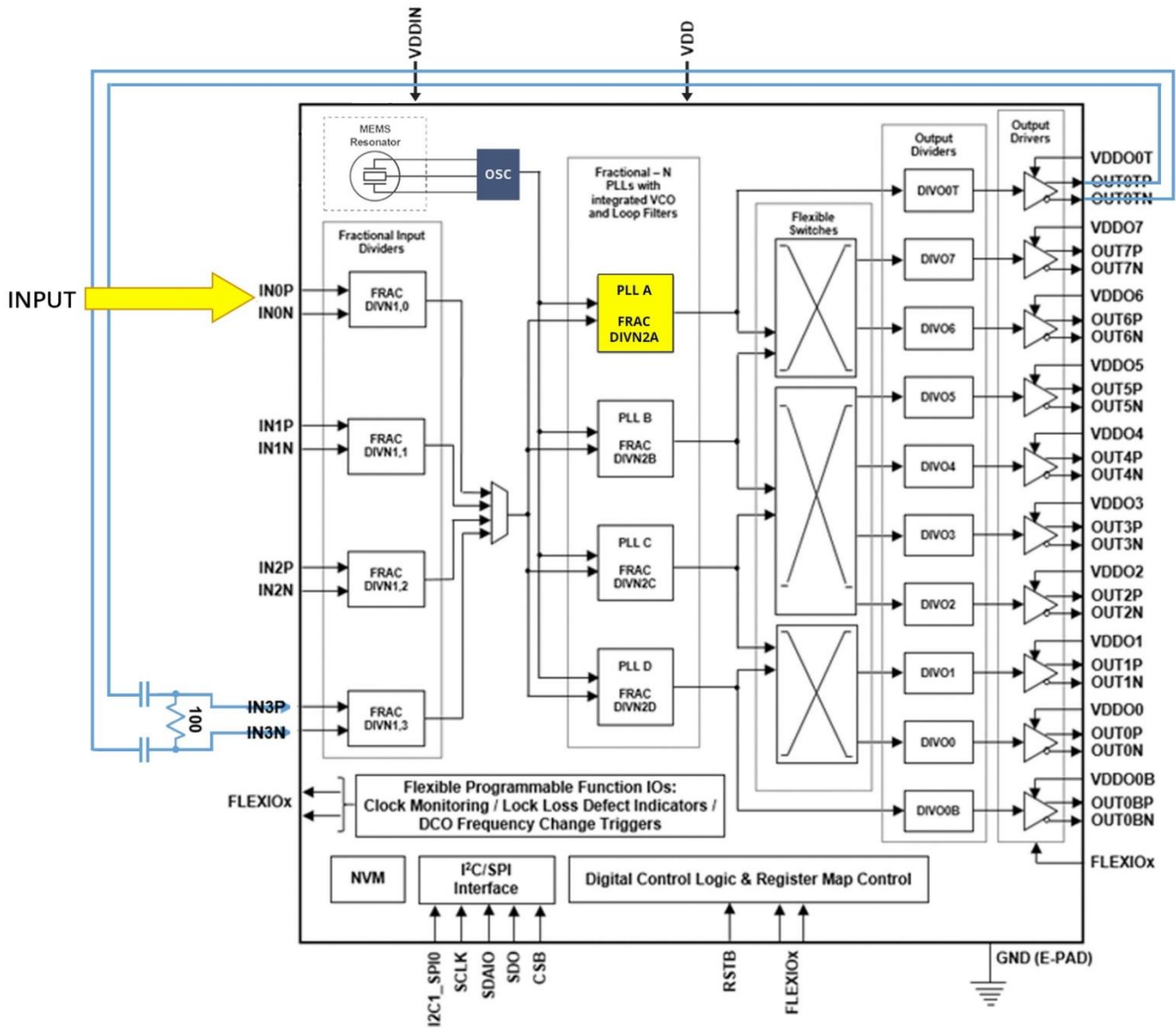


Figure 33. SiT95145 Zero Delay Mode Setup

SiT95145 as device clock and synchronous system reference

SiT95145 serves as the JESD204B clock source for providing timing references with fixed phase relationships. The chip distributes both a device clock (Dev_Clock) and a source synchronous system reference (SYSREF) signal to each device in the link. The converter/logic device generates its local multi-frame clock (LMFC), frame clock, and sampling clock via an internal divider fed from the

Dev_CLK. Each device in the link uses Dev_CLK to sample the SYSREF signal which is in turn used to align the LMFC to achieve synchronization between two devices. In order to maintain a deterministic phase relationship between the clock pair, SYSREF and Dev_Clock must be source-synchronous and SYSREF must meet setup and hold times relative to Dev_CLK. This will ensure that the same Dev_CLK edge, relative to SYSREF, is selected during each successive frame alignment sequence. The high-level block diagram is shown in [Figure 34](#).

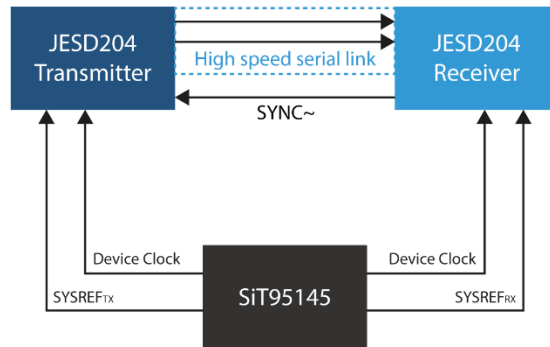


Figure 34. High level block diagram showing a JESD204B interface between an ADC (Transmitter) and an FPGA (Receiver)

Programmable Interface (PIF) Top Level View

Table 20. PIF Overview (Top Level Summary of the Programmable Interface)^[52]

| Page Number | Function | Comments | Fuse repeated twice |
|-------------|--|--|---------------------|
| 0h | Generic | <p><u>2Fh[7:6]: Lock Pattern for the Fuse</u> <u>FFh[7:0]: Current Page Number</u> <u>22h[7:0]: Current Fuse Pointer</u></p> <p>00h - 01h: Customer- Chip Information 02h - 04h: First set of Defect / Notify / Interrupt 06h - 08h: Second set of Defect / Notify / Interrupt 05h: DCO increment/decrement control 0Fh: Program Command Directives and Active Trigger Directives 10h: PLL enable control 11h - 18h: Fuse GPIO (FlexIO) Multiplexed Control 19h: VDD Padding Control and External CLKIN Switch Control 1Ah - 21h: Die ID + Wafer Co-ordinates 22h: Fuse Pointer Generic 24h: Clock Input / Output Enable Control 25h: Clock Output Enable Control 26h: OEB, Clock Output Enable Control Settings 27h - 28h: Masking of sticky bits status for Interrupt generation (INTR_b) 29h: GPIO (FLEXIO) Configuration 2Ah: Fuse Based I2C Addr 2Bh: Calibrations and Misc Settings</p> | NO |
| 1h | Clock Monitor | <p><u>2Fh[7:6]: Customer - Lock Pattern for the Fuse</u> <u>FFh[7:0]: Current Page Number</u></p> <p>00h - 01h: Chip Information 02h - 04h: First set Defect / Notify / Interrupt for Clock Monitor Sub- system. 06h - 08h: Second set of Defect / Notify / Interrupt for Clock Monitor Sub-system 0Fh: Program Command Directives and Active Trigger Directives 10h - 29h, 46h - 48h: Clock Loss Monitor Configuration 2Ah - 45h, 4Ch - 4Fh: Frequency Drift Coarse/Fine Configuration 49h - 4Bh: PLLs Input Clock Priority Information</p> | YES |
| 2h | Input | <p><u>2Fh[7:6]: Customer - Lock Pattern for the Fuse</u> <u>FFh[7:0]: Current Page Number</u></p> <p>00h - 01h: Chip Information 02h - 04h: Defect / Notify / Interrupt for Input Sub-system 0Fh: Program Command Directives and Active Trigger Directives 10h - 19h: CLKIN0 Fuse Configuration (IDR, DIVN0, Clock MUX) 20h - 2Fh: CLKIN1 Fuse Configuration (IDR, DIVN1, Clock MUX) 30h - 3Fh: CLKIN2 Fuse Configuration (IDR, DIVN1, Clock MUX) 40h - 4Fh: CLKIN3 Fuse Configuration (IDR, DIVN1, Clock MUX)</p> | YES |
| 3h | 8 Flexi-Outputs/ 4 Fixed-Output Blocks | <p><u>2Fh[7:6]: Customer - Lock Pattern for the Fuse</u> <u>FFh[7:0]: Current Page Number</u></p> <p>00h - 01h: Chip Information 02h - 04h: Defect / Notify / Interrupt for Output Sub-system 0Fh: Program Command Directives and Active Trigger Directives 10h - 17h: Output Block 0 Fuse Configuration (ODR, DIVO, DIVO- Delay) 18h - 1Fh: Output Block 1 Fuse Configuration (ODR, DIVO, DIVO- Delay) 20h - 27h: Output Block 2 Fuse Configuration (ODR, DIVO, DIVO- Delay) 28h - 2Fh: Output Block 3 Fuse Configuration (ODR, DIVO, DIVO- Delay) 30h - 37h: Output Block 4 Fuse Configuration (ODR, DIVO, DIVO- Delay) 38h - 3Fh: Output Block 5 Fuse Configuration (ODR, DIVO, DIVO- Delay) 40h - 47h: Output Block 6 Fuse Configuration (ODR, DIVO, DIVO- Delay) 48h - 4Fh: Output Block 7 Fuse Configuration (ODR, DIVO, DIVO- Delay) 50h - 57h: Output Block 0T Fuse Configuration (ODR, DIVO, DIVO- Delay) 60h - 67h: Output Block 0B Fuse Configuration (ODR, DIVO, DIVO- Delay)</p> | YES |

Table 20. PIF Overview (Top Level Summary of the Programmable Interface)^[52] (continued)

| Page Number | Function | Comments | Fuse repeated twice |
|--|----------|---|---------------------|
| Ah | PLL A | <u>2Fh[7:6] : Customer - Lock Pattern for the Fuse</u> | YES |
| | | <u>FFh[7:0]: Current Page Number</u> | |
| | | 00h - 01h: Customer- Chip Information | |
| | | 02h - 04h: First set of Defect / Notify / Interrupt for PLLA | |
| | | 06h - 08h: Second set of Defect / Notify / Interrupt for PLLA | |
| | | 05h: Customer- PLL Generic Directives | |
| | | 0Fh: Program Command Directives and Active Trigger Directives | |
| 10h - 2Fh: PLL Fuse Configuration (All PLL specific settings for this PLL) | | | |
| 30h - 37h: Customer-DCO Functionality | | | |

Note:

52. All Addresses in Hexadecimal.

Serial Programming Interface Description

The device has two serial programming interface options, I²C and SPI, for reconfiguring the device settings. The protocol option can be selected through the I2C1_SPI0 pin. A 1/HIGH on the pin sets the device in I²C mode and a 0/LOW in SPI mode.

I²C protocol

The device uses the SDAIO and SCLK pins for a 2-wire serial interface that operates up to 400 Kbps in Read and Write modes. It complies with the I²C bus standard. The I²C access protocol in device is byte access (random access) only for write and both random and sequential access for read.

The I²C serial interface can operate at either Standard rate (100 Kbps) or Fast rate (400 Kbps). For write operation, the device supports only single write operation. For read, the device supports both Single and Multiple read operations.

The device uses a 7-bit slave address, and the default value is 0x69 (right justified). The first 5 bits are static and programmed in the device to be 0b11010. The two least significant bits are controlled by the two address pins A1 (SDO) and A0 (CSB). When these address pins are floating, internal pull-down (for A1) and pull-up (for A0) will set the address to its default value 0x69 or 0b1101001. If the address pins are externally controlled, 3 more addresses are possible: 0x68, 0x6A, and 0x6B.

Additional slave addresses are available by using Register 0x2A at page 0 to change the static portion of the I²C address. Setting the 0x2A[7] bit to 1 enables use of non-default values for the 5 static address bits. These new values would be defined by bits 0x2A[6:2]. A1 and A0 would still control the 2 least significant bits of the address. This allows for a total of 128 supported addresses. To permanently change the I²C address, register 0x2A should be E-Fused.

If the chip is triggered into reset state (by writing a value of 0x01 to Register 0xFE), the 5 default static bits of the I²C address will be used regardless of Register 0x2A state. To exit reset state, a value of 0x00 should be written into the Register 0xFE. In this case, the I²C address will be set by the 5 default MSB bits and the 2 LSB bits controlled by A1 and A0. If these pins are floating, the default slave address 0x69 should be used as the address of the device.

Please refer to Figure 35 for the I²C addressing scheme of this device.

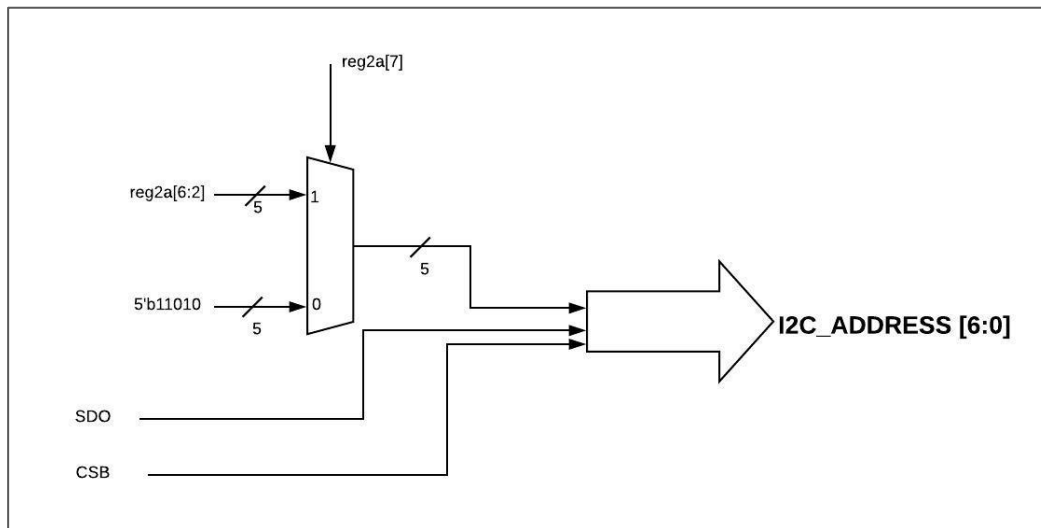


Figure 35. I²C Addressing Scheme for Cascade

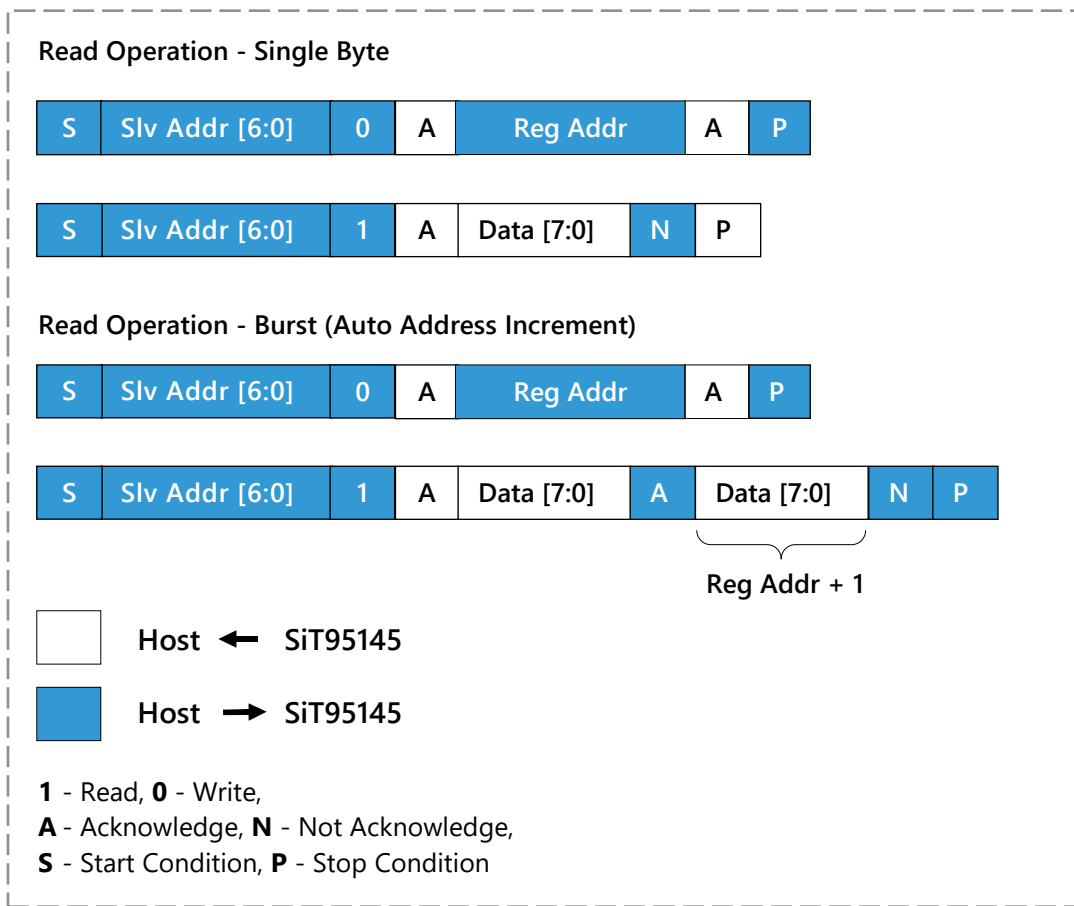


Figure 36. Read Block

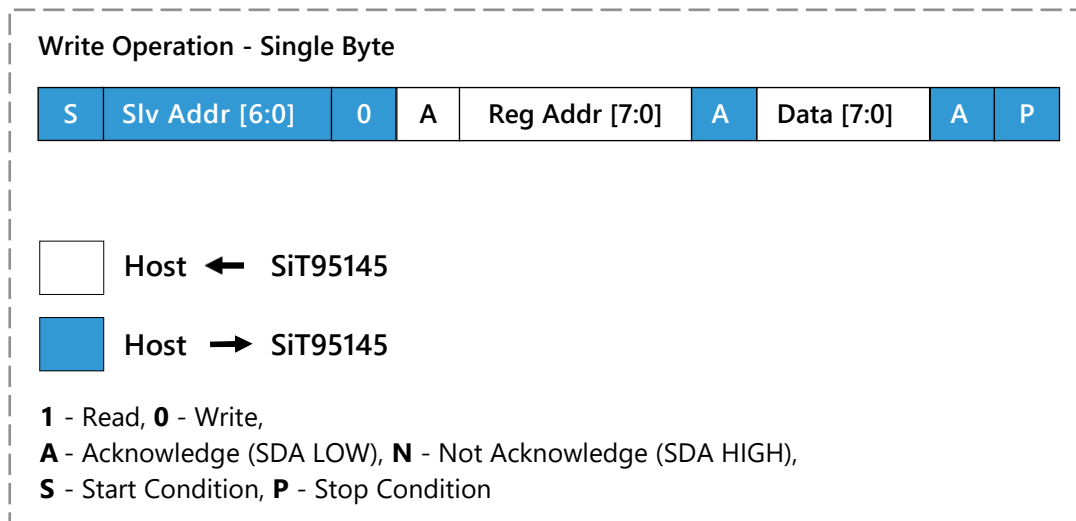


Figure 37. Write Block

■ Single Byte Write

- The master initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 0 (write)
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit register map address
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit data to be written to the register map address specified
- The slave acknowledges by driving zero on the bus
- The master ends the transaction by issuing a stop condition

■ Single Byte Read

- The master initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 0 (write)
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit register map address
- The slave acknowledges by driving zero on the bus
- The master ends the transaction by issuing a stop condition
- The master re-initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 1 (read)
- The slave then writes the 8 bit data to be written to the register map address specified
- The master does not acknowledge this transaction as the slave may assume a multi-byte read operation and there is a risk of slave holding the bus low
- The master ends the transaction by issuing a stop condition

■ Multi Byte Read

The multi-byte read mode is used to read a continuous segment of the register map. The multi-byte read is faster than performing multiple single byte reads as the device address and register map address need not be specified for every byte read from the register map

- The master initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 0 (write)
- The slave acknowledges by driving zero on the bus
- The master then writes the 8 bit register map address
- The slave acknowledges by driving zero on the bus
- The master ends the transaction by issuing a stop condition
- The master re-initiates the transaction by issuing a start condition, writes 7 bit slave address and then the read/write bit is written as 1 (read)
- The slave then writes the 8 bit data to be written to the register map address specified
- The master acknowledges by driving zero on the bus
- The slave automatically increments the register map address and writes the data in at that address to the bus and the master acknowledges
- When all bytes of data are read, master ends the operation by not acknowledging the last read
- The master then ends the transaction by issuing a stop condition

SPI Protocol

The SPI is a four-pin interface with Chip Select (CSB), Serial Input (SDAIO), Serial Output (SDO), and Serial Clock (SCLK) pins. The SPI bus on the device can run at speed up to 20 MHz. The SPI is a synchronous serial interface, which uses clock and data pins for serial access. When I2C1_SPI0 pin is Low, a Low on the CSB pin activates the SPI access. The SPI Timing diagram is shown in [Figure 38](#).

1. The SPI can operate up to 20 MHz for regular write/read operations.
2. The SPI receives serial data from the external master and provides Wr/rdn (set to 0x01h), address and data to the register map during the write operation.
3. The SPI receives serial data from the external master and provides Wr/rdn (set to 0x00h), address to the register map and uses the read data obtained from the register map, serializes the same and transmit to the master.
4. In SiT95145, the total packet size for each SPI transaction is 24 bits where the 8 bits are Wr/rdn (0x01 for write and 0x00 for read), the next 8 bits are address and the last 8 bits are data
5. In SiT95145 for write operation, the master assembles the Wr/rdn byte, address and data for write operation on the falling edge of the spi clock and the slave in the SiT95145 captures the same on the rising edge of the SPI clock. There is no loopback provided here.
6. In SiT95145 for read operation, the master assembles the Wr/rdn byte, address for read operation on the falling edge of the spi clock and the slave in the SiT95145 captures the same on the rising edge of the SPI clock and there is no loopback. The falling edge after the 16th rising SPI clock (i.e. the last address bit), is used by the slave to assemble the first read data which is captured by the master on the 17th edge of the SPI clock. Subsequent 7 more clocks are used for the 7 remaining data bits.
7. In SiT95145 the transmitter always sends data on the falling edge of the SPI clock to be captured in the receiver by the rising edge of the SPI clock. The transmitter can be the master for the whole operation of the write and for the control and address portions of the read. The slave is the transmitter during the data portion of the read cycle.
8. The register can be written to or read from one address at a time. The SPI implemented in SiT95145 does not support burst address write or read operations.

SPI Timing Details

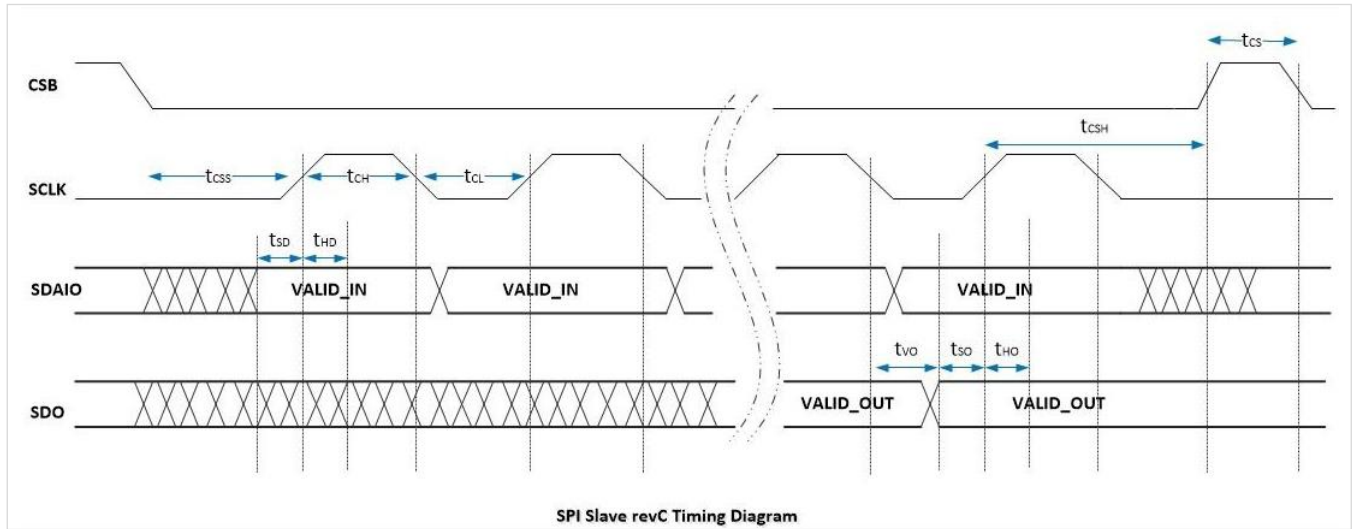


Figure 38. SPI Timing Diagram

Table 21. SPI Timing

| Symbol | Description | Min. | Typ. | Max. | Units |
|------------|------------------------|------|------|------|-------|
| f_{SCLK} | SCLK clock frequency | – | – | 20 | MHz |
| t_{CH} | Clock pulse width HIGH | 20 | | | ns |
| t_{CL} | Clock pulse width LOW | 20 | | | ns |
| t_{CS} | CSB HIGH time | 50 | | | ns |
| t_{CSS} | CSB setup time | 25 | | | ns |
| t_{CSH} | CSB hold time | 25 | | | ns |
| t_{SD} | Data in setup time | 10 | | | ns |
| t_{HD} | Data in hold time | 10 | | | ns |
| t_{VO} | Output valid | | | 10 | ns |
| t_{SO} | Output valid | | | 10 | ns |
| t_{HO} | Output valid | | | 10 | ns |

SPI Single byte write

- The master initiates the transaction by issuing a start condition of pulling `csn_i` to active low
- The master assembles the serial data on the falling edge of the SPI clock so the SPI slave can capture the same on the rising edge of the SPI clock
- The first 7 bits are don't care with the 8th bit being set to 1 to indicate a write operation
- The next 8 bits (second byte) are used for the register map address
- The next 8 bits (third byte) are used for the register map data
- The 24th rising edge of the SPI clock is used to capture the last data bit. The SPI slave then assembles the address, data, enable and `wr_rdn` to the PIF slave block. The inverted version of the next falling edge of the SPI clock is used by the SPI slave to capture the address, data, enable and `wr_rdn` to write to the respective register
- The `csn` is then de-activated (by going high) by the master
- For the next write operation, `CSN` is held high for at least a duration of two SPI clocks following which the entire operation can start again.

SPI Single byte read

- The master initiates the transaction by issuing a start condition of pulling `csn_i` to active low
- The master assembles the serial data on the falling edge of the SPI clock so the SPI slave can capture the same on the rising edge of the SPI clock
- The first 7 bits are don't care with the 8th bit being set to 0 to indicate a read operation
- The next 8 bits (second byte) are used for the register map address
- The next 8 bits (third byte) are used for the register map read data that is supplied by the pif slave block
- The 16th rising edge of the SPI clock is used to capture the last address bit. The SPI slave then assembles the address, enable and `wr_rdn` to the PIF slave block. The slave block then uses the address when enable is high to provide the read back data via a multiplexer. This operation has to be completed within half a SPI clock since the SPI slave has to assemble the first read back data bit on the falling edge of the SPI clock so the SPI slave can capture the same on the next rising edge. After 7 additional clocks, all 8 serial read back data bits are sent out from the SPI slave
- The `CSN` is then de-activated (by going high) by the master
- For the next read operation, `CSN` is held high for at least a duration of two SPI clocks following which the entire operation can start again
- When `CSB` goes high after an SPI transaction is complete, it is recommended that the SPI clock is shut off and stays in the low state till the next SPI transaction starts. During this time, the `SDO` will hold the last transmitted bit. The duration for which the `CSB` is expected to be held high is equivalent of two SPI clocks. If for some reason, the SPI clocks cannot be held static at zero and need to toggle when `CSB` is high, then the `SDO` will output a low after the first rising edge of the SPI clock after the `CSB` has transitioned to high. The duration for which `CSB` is expected to be held high is equivalent of two SPI clock.

Figure 35 shows the SPI Read Operation and Figure 36 shows the SPI Write Operation.

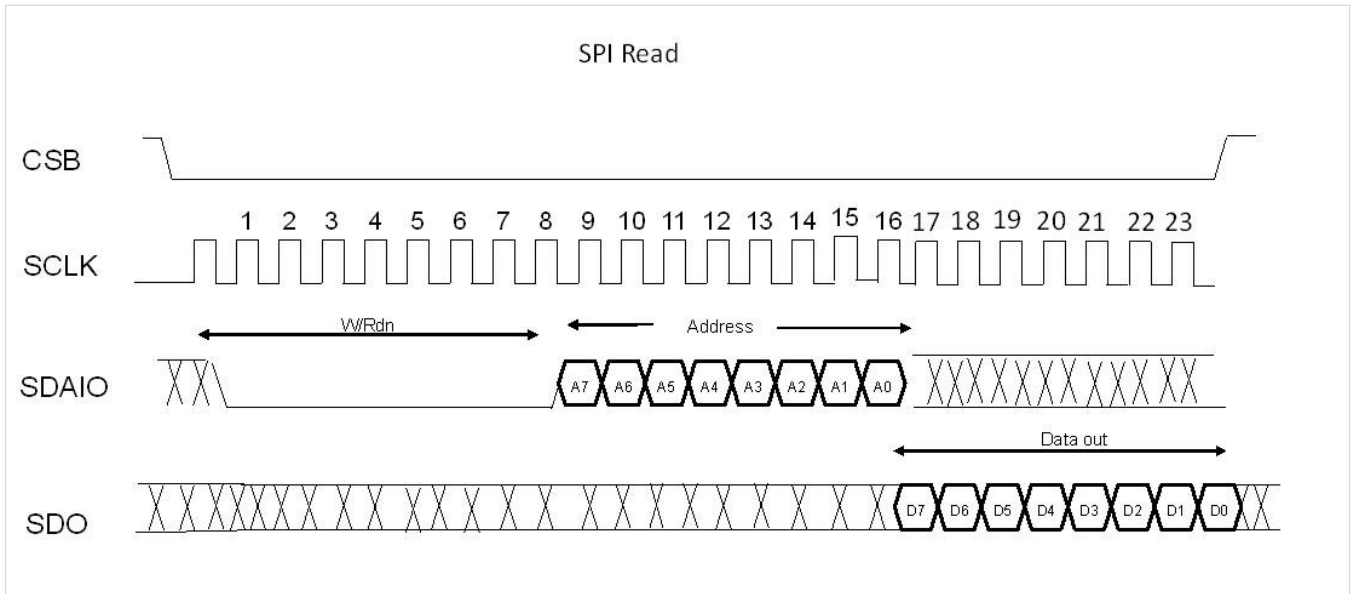


Figure 35. SPI Read Operation

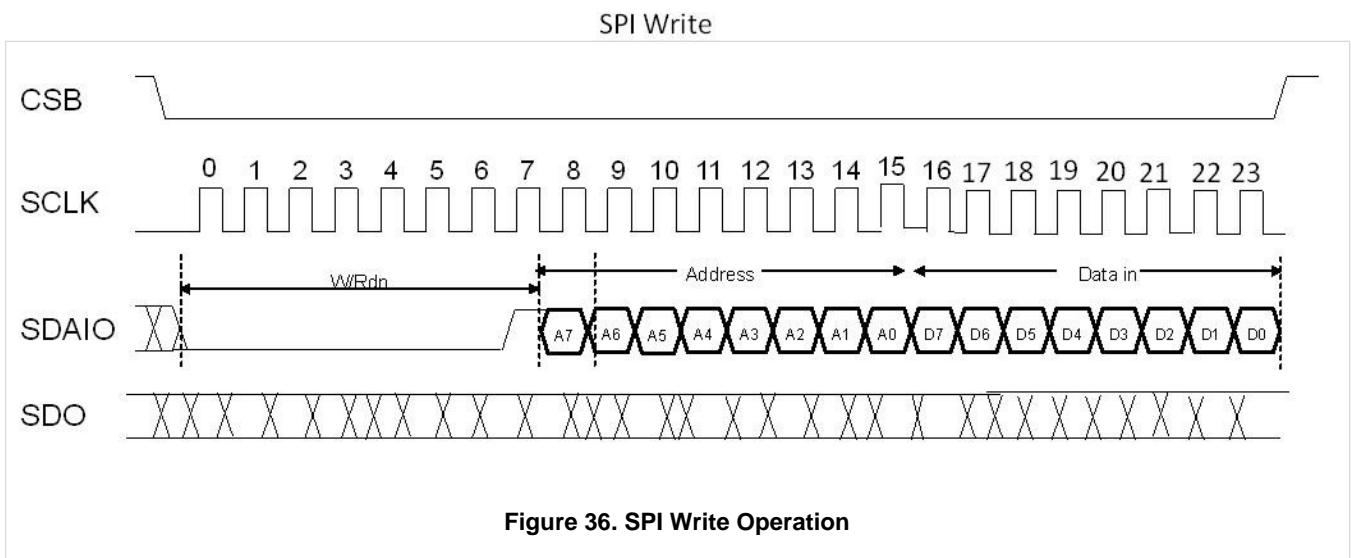


Figure 36. SPI Write Operation

Alarm Registers' Information

Status and Notify

SiT95145 provides various status and notify bits that can be accessed from the register map. Below are the details of the procedure to be followed to access the same.

The alarm registers are a set of three types of registers distributed between the various pages as described in [Table 22](#) below and illustrated with some examples in a later section of this datasheet

- The Status registers are the current dynamic status of a defect. The live status defects are active high with a '1' indicating the defect is present.
- The Notify registers are the sticky bits for a defect. Sometimes, we may get a very short pulse for the status and it may not be possible for the external user to capture the same. Hence a notify register is provided. The notify register is set to 1 whenever there is a rising edge of the corresponding status register. This is a sticky bit and stays at 1 till the user writes a 1 to that specific bit to clear it.
- Each notify has a masking bit to enable or disable its operation. The notify sticky bit operates only if the corresponding masking bit is set to 1. If the masking register bit is set to 0, notify will not be asserted even when status toggles. The default value for the mask register is 0xff so all the notify signals are enabled. Once the user writes a 1 to clear the notify, the notify bit can again go high on the next rising edge of the status.

These registers operate on the internal 4 MHz RC clock. When there is a defect (i.e. status) of any bit in register it gets asserted and de-asserted in a live mode. User can read the corresponding register location to see the current status at any time.. The default value for the mask register is 0xFF so all the notify signals are enabled. Once the user

writes a 1 to clear the notify bit, the notify bit can again go high on the next rising edge of the status. On the other hand, the default value of the notify and masking register is 0xFF – user has to write 0xff to both these registers to clear them at the beginning and use all notifies. The INTRB pin is used as a NOR operation of the selected notifies. The choice of the Notify listing that is used for the INTRB pin is selectable in the GUI when creating the profile. The sticky notifies that are selected and used for INTRB need to be cleared for restoring the INTRB to 1 for further sticky defect monitoring using this pin.

The table below details the name of the alarm, the page it is located in, the address in the page and the bit number in the address. In order to access a page of the register map, the particular page number has to be written to address 0xFF. For instance, to either write to or read from page 1, first the user needs to write to 0xFF a value of 0x01. Following this, any number of write or read operations can be done with page 1.

Table 22. Tabular Listing of Alarm Registers

| S.NO | Name of Signal | Description | Page Number | Register Address | Bit Number |
|------|------------------------------|--|-------------|------------------|------------|
| 1 | xo_clkloss_dynamic_status | Xo clock dynamic status | 00 | 0x02 | 2 |
| 2 | xo_clkloss_dynamic_ntfy | Xo clock dynamic Notify (write 0x04 with 0x02 to enable) | 00 | 0x03 | 2 |
| 3 | plla_lol_dyn_status | pll lol dynamic status | 00 | 0x06 | 0 |
| 4 | pllb_lol_dyn_status | pll lol dynamic status | 00 | 0x06 | 1 |
| 5 | pllc_lol_dyn_status | pll lol dynamic status | 00 | 0x06 | 2 |
| 6 | plld_lol_dyn_status | pll lol dynamic status | 00 | 0x06 | 3 |
| 7 | plla_ho_frz_status | pll ho freeze dynamic status | 00 | 0x06 | 4 |
| 8 | pllb_ho_frz_status | pll ho freeze dynamic status | 00 | 0x06 | 5 |
| 9 | pllc_ho_frz_status | pll ho freeze dynamic status | 00 | 0x06 | 6 |
| 10 | plld_ho_frz_status | pll ho freeze dynamic status | 00 | 0x06 | 7 |
| 11 | plla_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 with 0x01 to enable) | 00 | 0x07 | 0 |
| 12 | pllb_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 with 0x02 to enable) | 00 | 0x07 | 1 |
| 13 | pllc_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 with 0x04 to enable) | 00 | 0x07 | 2 |
| 14 | plld_lol_dyn_ntfy | pll lol dynamic Notify (write 0x08 with 0x08 to enable) | 00 | 0x07 | 3 |
| 15 | plla_ho_frz_ntfy | pll ho freeze dynamic Notify (write 0x08 with 0x10 to enable) | 00 | 0x07 | 4 |
| 16 | pllb_ho_frz_ntfy | pll ho freeze dynamic Notify (write 0x08 with 0x20 to enable) | 00 | 0x07 | 5 |
| 17 | pllc_ho_frz_ntfy | pll ho freeze dynamic Notify (write 0x08 with 0x40 to enable) | 00 | 0x07 | 6 |
| 18 | plld_ho_frz_ntfy | pll ho freeze dynamic Notify (write 0x08 with 0x80 to enable) | 00 | 0x07 | 7 |
| 19 | in0_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | 0x02 | 0 |
| 20 | in1_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | 0x02 | 1 |
| 21 | in2_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | 0x02 | 2 |
| 22 | in3_clock_loss_dyn_status | Clock Loss Dynamic Status | 01 | 0x02 | 3 |
| 23 | in0_clock_loss_fd_dyn_status | Clock Loss with Fd_dynamic Status | 01 | 0x02 | 4 |
| 24 | in1_clock_loss_fd_dyn_status | Clock Loss with Fd_dynamic Status | 01 | 0x02 | 5 |
| 25 | in2_clock_loss_fd_dyn_status | Clock Loss with Fd_dynamic Status | 01 | 0x02 | 6 |
| 26 | in3_clock_loss_fd_dyn_status | Clock Loss with Fd_dynamic Status | 01 | 0x02 | 7 |
| 27 | in0_clock_loss_ntfy | Clock Loss Dynamic Notify (write 0x04 with 0x01 to enable) | 01 | 0x03 | 0 |
| 28 | in1_clock_loss_ntfy | Clock Loss Dynamic Notify (write 0x04 with 0x02 to enable) | 01 | 0x03 | 1 |
| 29 | in2_clock_loss_ntfy | Clock Loss Dynamic Notify (write 0x04 with 0x04 to enable) | 01 | 0x03 | 2 |
| 30 | in3_clock_loss_ntfy | Clock Loss Dynamic Notify (write 0x04 with 0x08 to enable) | 01 | 0x03 | 3 |
| 31 | in0_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x10 to enable) | 01 | 0x03 | 4 |
| 32 | in1_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x20 to enable) | 01 | 0x03 | 5 |
| 33 | in2_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x40 to enable) | 01 | 0x03 | 6 |
| 34 | in3_clock_loss_ntfy | Clock Loss with Fd_dynamic Notify (write 0x04 with 0x80 to enable) | 01 | 0x03 | 7 |
| 35 | in0_fd_fine_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 0 |
| 36 | in1_fd_fine_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 1 |

Table 22. Alarm Register Listing (continued)

| S.NO | Name of Signal | Description | Page Number | Register Address | Bit Number |
|------|-------------------------------|---|-------------|------------------|------------|
| 37 | In2_fd_fine_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 2 |
| 38 | In3_fd_fine_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 3 |
| 39 | in0_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 4 |
| 40 | In1_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 5 |
| 41 | In2_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 6 |
| 42 | In3_fd_coarse_dyn_status | Frequency drift dynamic status | 01 | 0x06 | 7 |
| 43 | in0_fd_fine_ntfy | Frequency drift dynamic notify (write 0x08 with 0x01 to enable) | 01 | 0x07 | 0 |
| 44 | In1_fd_fine_ntfy | Frequency drift dynamic notify (write 0x08 with 0x02 to enable) | 01 | 0x07 | 1 |
| 45 | In2_fd_fine_ntfy | Frequency drift dynamic notify (write 0x08 with 0x04 to enable) | 01 | 0x07 | 2 |
| 46 | In3_fd_fine_ntfy | Frequency drift dynamic notify (write 0x08 with 0x08 to enable) | 01 | 0x07 | 3 |
| 47 | in0_fd_coarse_ntfy | Frequency drift dynamic notify (write 0x08 with 0x10 to enable) | 01 | 0x07 | 4 |
| 48 | In1_fd_coarse_ntfy | Frequency drift dynamic notify (write 0x08 with 0x20 to enable) | 01 | 0x07 | 5 |
| 49 | In2_fd_coarse_ntfy | Frequency drift dynamic notify (write 0x08 with 0x40 to enable) | 01 | 0x07 | 6 |
| 50 | In3_fd_coarse_ntfy | Frequency drift dynamic notify (write 0x08 with 0x80 to enable) | 01 | 0x07 | 7 |
| 51 | fast_lock_dynamic_status_plla | PLL Fast lock dynamic status | 0a | 0x06 | 1 |
| 52 | fast_lock_dynamic_status_pllb | PLL Fast lock dynamic status | 0b | 0x06 | 1 |
| 53 | fast_lock_dynamic_status_pllc | PLL Fast lock dynamic status | 0c | 0x06 | 1 |
| 54 | fast_lock_dynamic_status_plld | PLL Fast lock dynamic status | 0d | 0x06 | 1 |
| 55 | fast_lock_dynamic_ntfy_plla | PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable) | 0a | 0x07 | 1 |
| 56 | fast_lock_dynamic_ntfy_pllb | PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable) | 0b | 0x07 | 1 |
| 57 | fast_lock_dynamic_ntfy_pllc | PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable) | 0c | 0x07 | 1 |
| 58 | fast_lock_dynamic_ntfy_plld | PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable) | 0d | 0x07 | 1 |
| 59 | ho_valid_dynamic_status_plla | PLL Ho Valid dynamic status | 0a | 0x06 | 2 |
| 60 | ho_valid_dynamic_status_pllb | PLL Ho Valid dynamic status | 0b | 0x06 | 2 |
| 61 | ho_valid_dynamic_status_pllc | PLL Ho Valid dynamic status | 0c | 0x06 | 2 |
| 62 | ho_valid_dynamic_status_plld | PLL Ho Valid dynamic status | 0d | 0x06 | 2 |
| 63 | ho_valid_dynamic_ntfy_plla | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable) | 0a | 0x07 | 2 |
| 64 | ho_valid_dynamic_ntfy_pllb | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable) | 0b | 0x07 | 2 |
| 65 | ho_valid_dynamic_ntfy_pllc | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable) | 0c | 0x07 | 2 |
| 66 | ho_valid_dynamic_ntfy_plld | PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable) | 0d | 0x07 | 2 |

Examples for Live Status Read Back

Some examples are presented based on the [Table 22](#) above for reading the live status of the defects.

In the pseudo code presented below:

`wr_cmd(address, data)`: refers to a “Write Command” where the corresponding data is written in to the specified register address

`x = rd_cmd(address)`: refers to a “Read Command” where the corresponding data is read from the specified register address and stored in the variable ‘x’

`y >> x`: denotes a bit wise right shift on the number y by x bit locations

`y << x`: denotes a bit wise left shift on the number y by x bit locations

`&` is the logical AND operation (bit wise)

Dynamic registers to read the various alarm registers in the RealTime page

1. Input Clocks CL and FD Related Real Time live status read back:

`wr_cmd(0xff, 0x01)` Program the CLKMON_SYS page number

Clock Loss dynamic status

`clock_loss_dyn_status = rd_cmd(0x02) & 0xff`

`(clock_loss_dyn_status >> 0) & 0x01` IN0 Status for CL, Read bit position [0]

`(clock_loss_dyn_status >> 1) & 0x01` IN1 Status for CL, Read bit position [1]

`(clock_loss_dyn_status >> 2) & 0x01` IN2 Status for CL, Read bit position [2]

`(clock_loss_dyn_status >> 3) & 0x01` IN3 Status for CL, Read bit position [3]

Frequency Drift dynamic status

`fd_fine_dyn_status = rd_cmd(0x06) & 0x0f`

`fine = (fd_fine_dyn_status >> 0) & 0x01` IN0 Status for Fine FD, Read bit position [0]

`fine = (fd_fine_dyn_status >> 1) & 0x01` IN1 Status for Fine FD, Read bit position [1]

`fine = (fd_fine_dyn_status >> 2) & 0x01` IN2 Status for Fine FD, Read bit position [2]

`fine = (fd_fine_dyn_status >> 3) & 0x01` IN3 Status for Fine FD, Read bit position [3]

`fd_coarse_dyn_status = rd_cmd(0x06) >> 4`

`coarse = (fd_coarse_dyn_status >> 0) & 0x01` IN0 Status for Coarse FD, Read bit position [4]

`coarse = (fd_coarse_dyn_status >> 1) & 0x01` IN1 Status for Coarse FD, Read bit position [5]

`coarse = (fd_coarse_dyn_status >> 2) & 0x01` IN2 Status for Coarse FD, Read bit position [6]

`coarse = (fd_coarse_dyn_status >> 3) & 0x01` IN3 Status for Coarse FD, Read bit position [7]

2. PLL Related Real Time live status read back:

`wr_cmd(0xff, 0x00)` # Program the GENERIC_SYS page number, Page 0

`pll_lol_ho_freeze_dyn_status = rd_cmd(0x06) & 0xff`

PLL Lock Loss dynamic status

`(pll_lol_ho_freeze_dyn_status >> 0) & 0x01` PLLA Status for LL, Read bit position [0]

`(pll_lol_ho_freeze_dyn_status >> 1) & 0x01` PLLB Status for LL, Read bit position [1]

`(pll_lol_ho_freeze_dyn_status >> 2) & 0x01` PLLC Status for LL, Read bit position [2]

`(pll_lol_ho_freeze_dyn_status >> 3) & 0x01` PLLD Status for LL, Read bit position [3]

Holdover Status

`(pll_lol_ho_freeze_dyn_status >> (0 + 4)) & 0x01` PLLA Status for HO, Read bit position [4]

`(pll_lol_ho_freeze_dyn_status >> (1 + 4)) & 0x01` PLLA Status for HO, Read bit position [5]

`(pll_lol_ho_freeze_dyn_status >> (2 + 4)) & 0x01` PLLA Status for HO, Read bit position [6]

`(pll_lol_ho_freeze_dyn_status >> (3 + 4)) & 0x01` PLLA Status for HO, Read bit position [7]

Examples of Sticky Bit Clearing

As described earlier, the sticky notify bits are cleared by writing a '1' to the corresponding notify bit itself. The notify bit by itself is enabled by writing a '1' to the corresponding mask bit.

In the pseudo code presented below,

`rmw_cmd(addr,bit_loc,no_of_bits,data):` denotes the read/modify/write operation where `no_of_bits` number of bits at `bit_loc` location (denoted as 7:0) is replaced with the data at address location `addr`.

```
def clr_intb_XO_CL():
    # This function is used to clear the sticky notify for XO Clock Loss
    #Write the page number
    wr_cmd(0xff, 0)
    # Information to clr ** Page 0: reg03[2]=1 **
    addr    = 0x3
    bit_loc = 2
    no_of_bits = 1
    data    = 1
    rmw_cmd(addr,bit_loc,no_of_bits,data)

def clr_intb_LOL_HO_Freeze():
    # This function is used to clear the sticky notify for loss of lock and holdover notify for all PLLs
    #Write the page number
    wr_cmd(0xff, 0)
    # Information to clr ** Page 0: reg07[7:0] = 0xff **
    addr    = 0x7
    bit_loc = 7
    no_of_bits = 8
    data    = 0xff
    rmw_cmd(addr,bit_loc,no_of_bits,data)

def clr_intb_CL():
    # This function is used to clear the sticky notify for clear clock loss notify
    # Write the page number
    wr_cmd(0xff, 1)
    # Information to clr ** Page1: reg03[3:0]=0x0f **
    addr    = 0x3
    bit_loc = 3
    no_of_bits = 4
    data    = 0x0f
    rmw_cmd(addr,bit_loc,no_of_bits,data)

def clr_intb_drift():
    # This function is used to clear the sticky notify for clear drift notify
    #Write the page number
    wr_cmd(0xff, 1)
    # Information to clr ** Page1: reg07[7:0]=0xff **
    addr    = 0x7
    bit_loc = 7
    no_of_bits = 8
    data    = 0xff
    rmw_cmd(addr,bit_loc,no_of_bits,data)

def clr_intrb():
    """
    This is the main clear function
    which calls the 4 clear functions
    """
    clr_intb_XO_CL()
    clr_intb_LOL_HO_Freeze()
    clr_intb_CL()
    clr_intb_drift()
```

Device Initialization for a Non-Programmed Device

This section describes a device initialization flow chart for an unlocked device. An unlocked device is a device on which the NVM is not programmed and where an autonomous wake up does not happen. It is assumed that the user is using a device that is not programmed for the description in this section.

The SiT95145 device register initialization flowchart for Master Control Page and the slave pages is as below. As explained in the chip functional description in the data sheet, the following is the sequence of the wake up of the various sub systems in the chip.

- First, the master control (Page 0) is initialized and programmed. The Slaves are powered up at this stage.
- Next, the Input System (Page 2) is initialized and programmed for cases where at least one input is enabled.
- Next, the Clock Monitor System (Page 1) is initialized and programmed for cases where at least one input is enabled.
- Next, the Output System (Page 3) is initialized and programmed.
- Finally, the PLLs that are expected to be used for the particular profile (PLLs A, B, C, D correspond to Pages A, B, C, D) are initialized and programmed.

The register 0xFF is written with the Page Number the user would like to access. The chip changes the current page once the user has written the register 0xFF. The page numbers corresponding to each slave and master are described in the respective table in the data sheet. At any point in time the register 0xFF can be read to find out the current page.

Please note that the flow chart in the following pages is to be used together with the sequence of register writes that are obtained from the GUI with the “Dump” button after loading a profile. This file obtained from the GUI describes the set of registers to be written in the exact order and with appropriate delays. This file obtained from the GUI is the master sequence to be followed for programming a part. The [Figures on the following pages](#) describe the flow of the same register write sequence file using flow charts.

Main Page Initialization

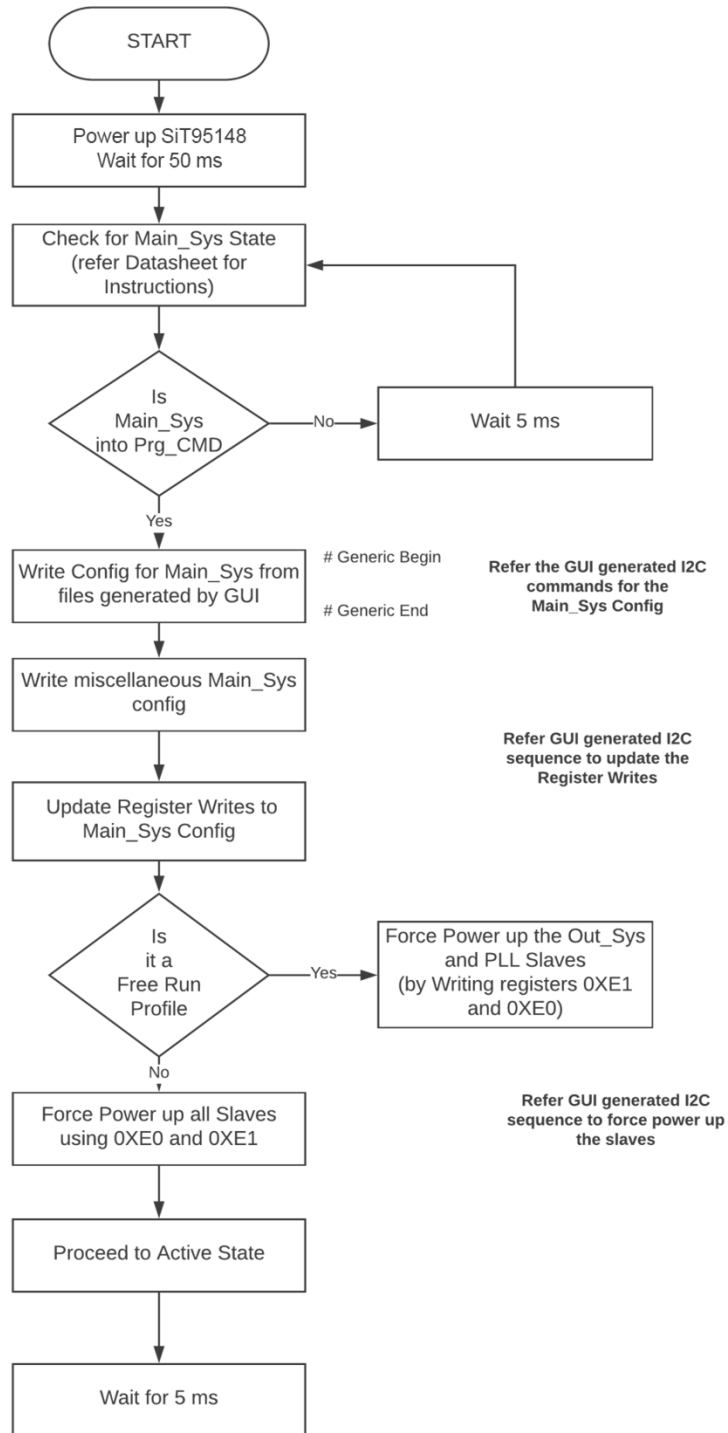


Figure 37. Step 1: Initialize the Main Page – Page 0

One can move to Page 0 by writing 0x00 to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF.

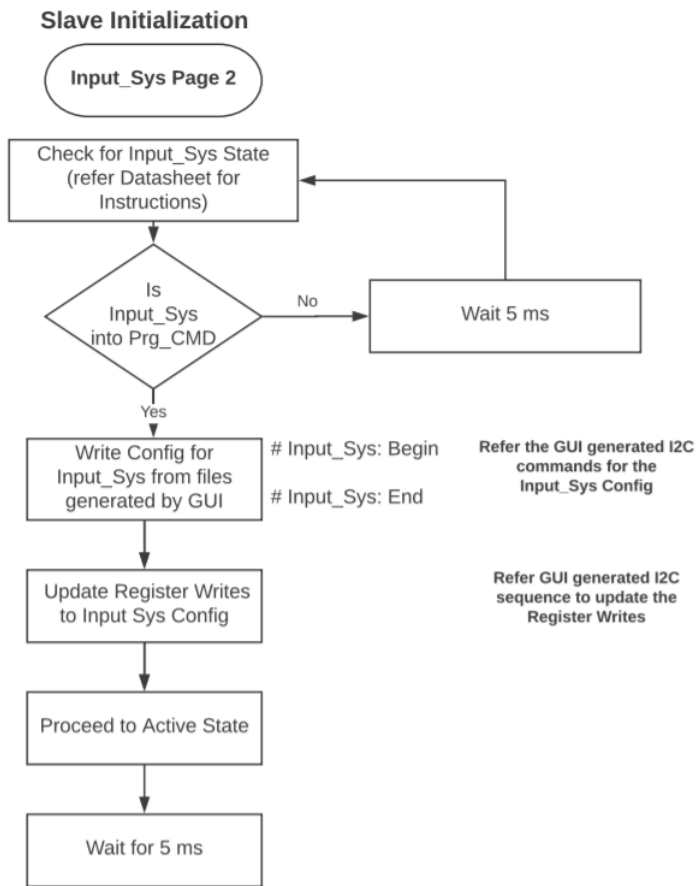


Figure 38. STEP 2: Initialize the Input System Page – Page 2

One can move to Page 2 by writing 0x02 to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF. This page is not initialized for a purely free run profile where no inputs are engaged.

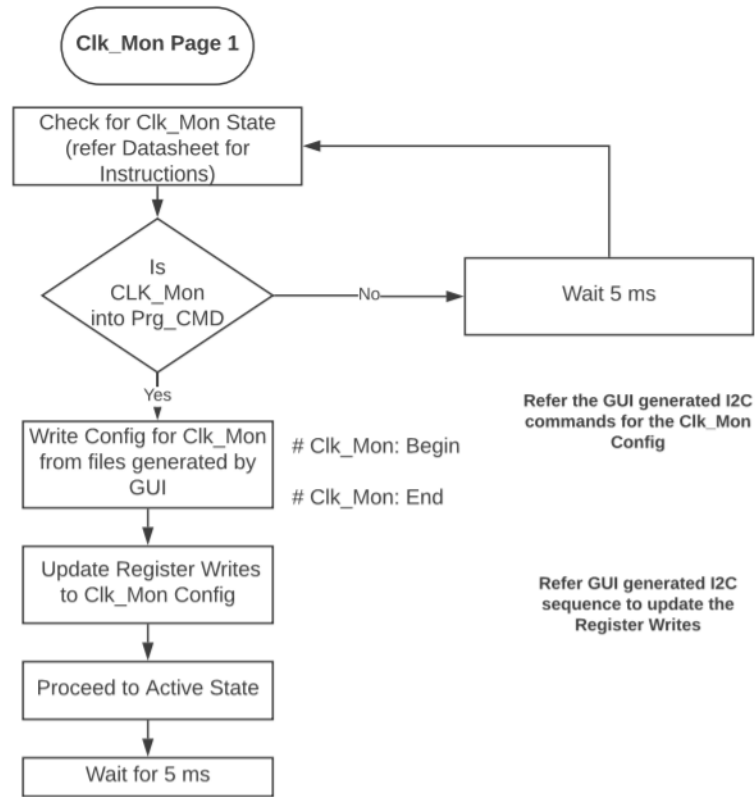


Figure 39. STEP 3: Initialize the Clock Monitor System Page – Page 1

One can move to Page 1 by writing 0x01 to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF. This page is not initialized for a purely free run profile where no inputs are engaged.

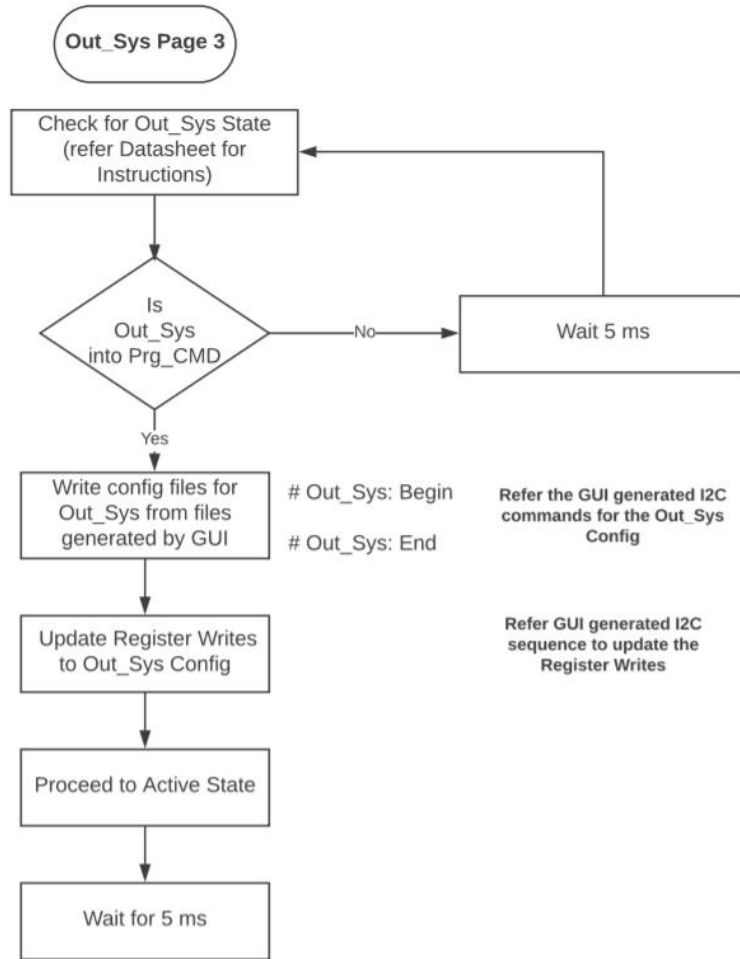


Figure 40. STEP 4: Initialize the Output System Page – Page 3

One can move to Page 3 by writing 0x03 to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF.

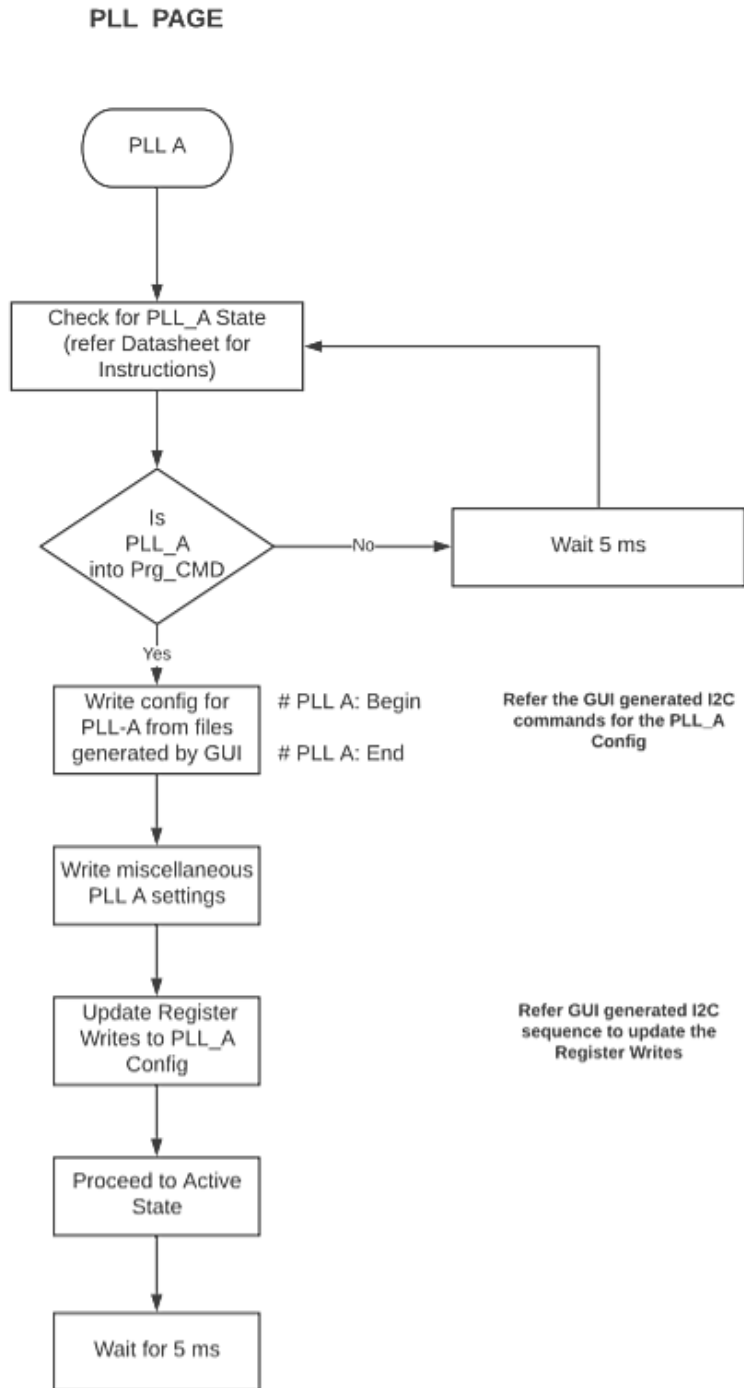


Figure 41. STEP 5: Initialize the PLL A System Page – Page A

One can move to Page A by writing 0x0A to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL A only if it is used in the profile being used.

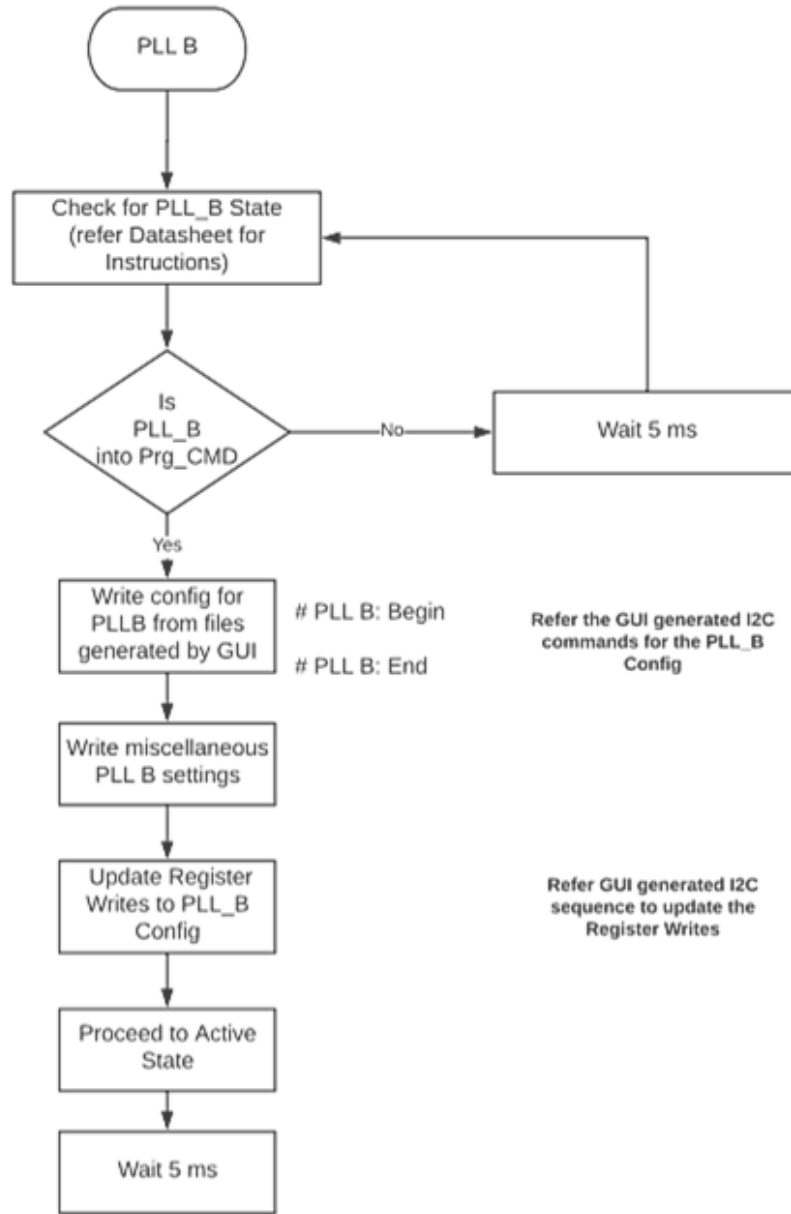


Figure 42. STEP 6: Initialize the PLL B System Page – Page B

One can move to Page B by writing 0x0B to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL B only if it is used in the profile being used.

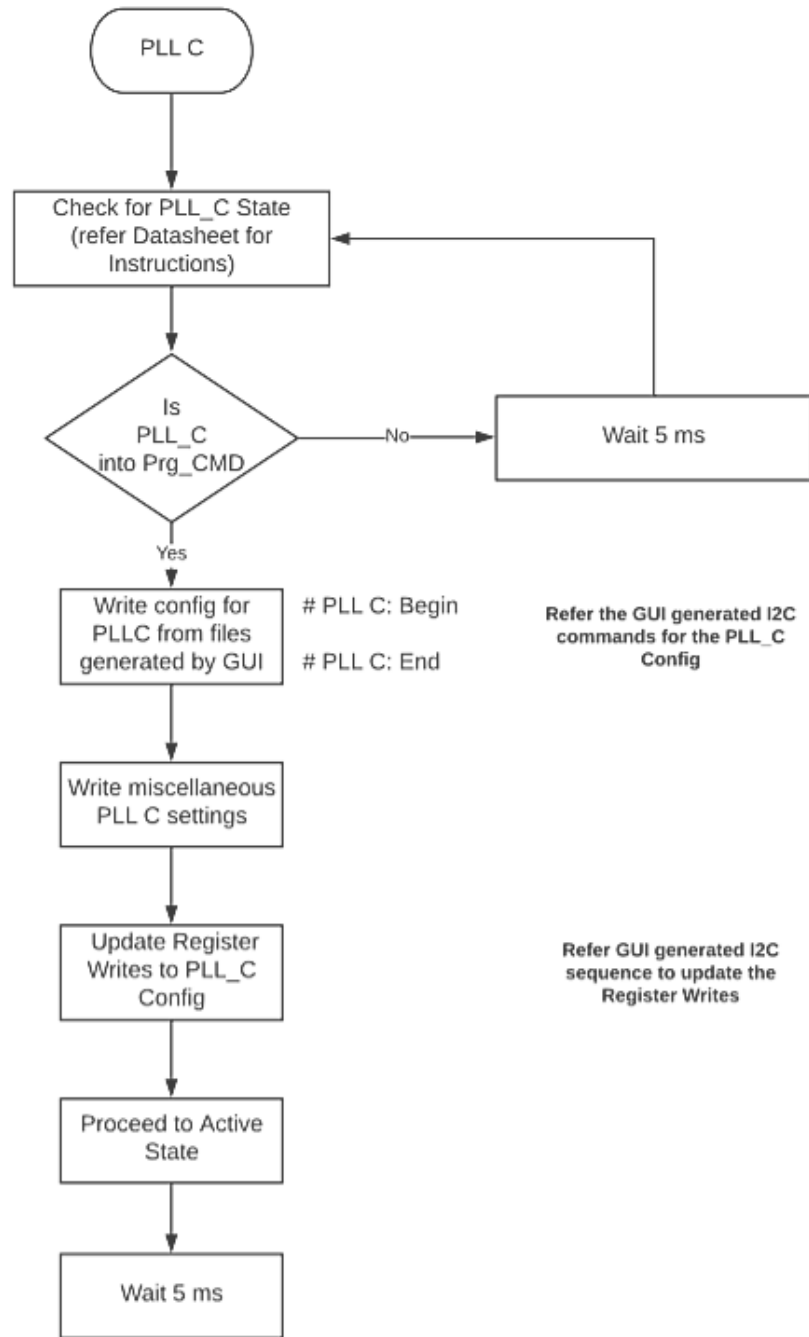


Figure 43. STEP 7: Initialize the PLL C System Page – Page C

One can move to Page C by writing 0x0C to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL C only if it is used in the profile being used.

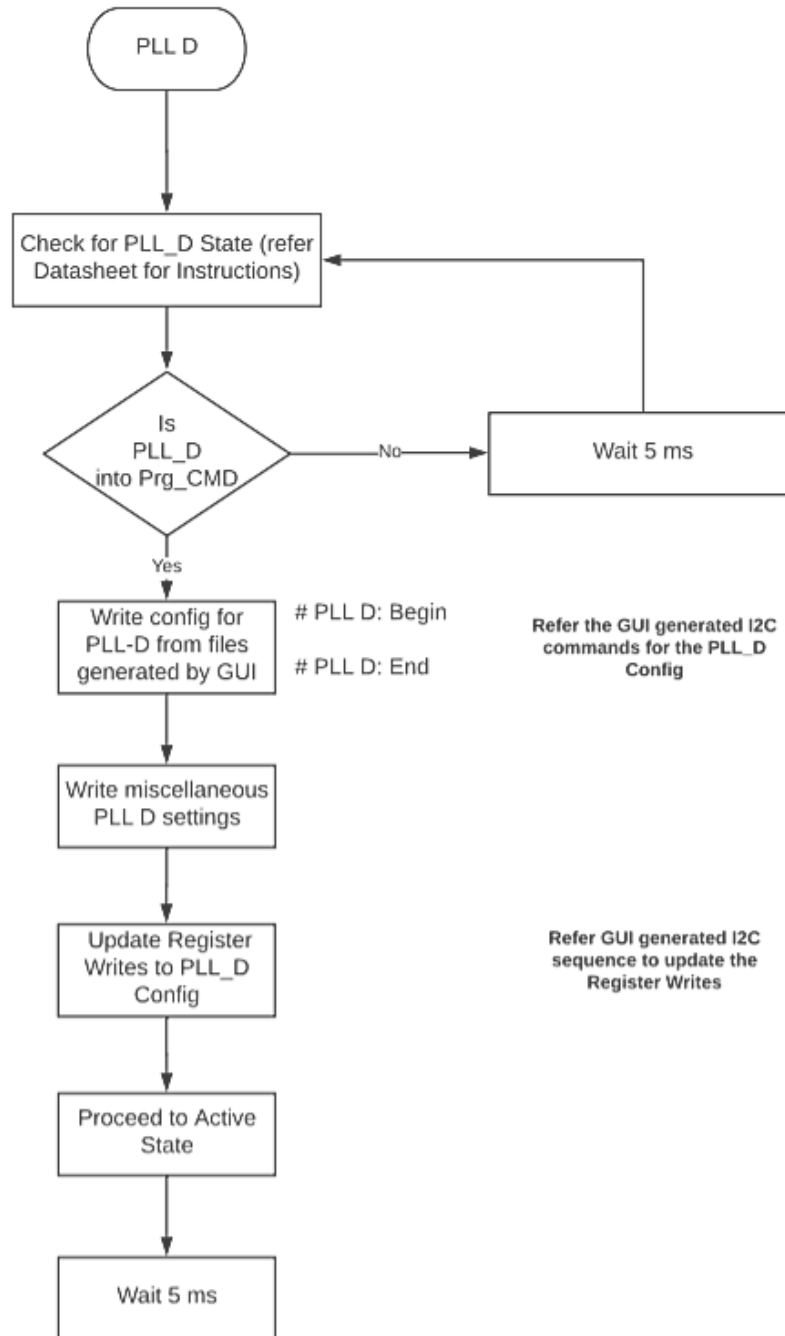


Figure 44. STEP 8: Initialize the PLL D System Page – Page D

One can move to Page D by writing 0x0D to the address 0xFF. Read the current page at any time by reading the contents of the register 0xFF. Initialize PLL D only if it is used in the profile being used.

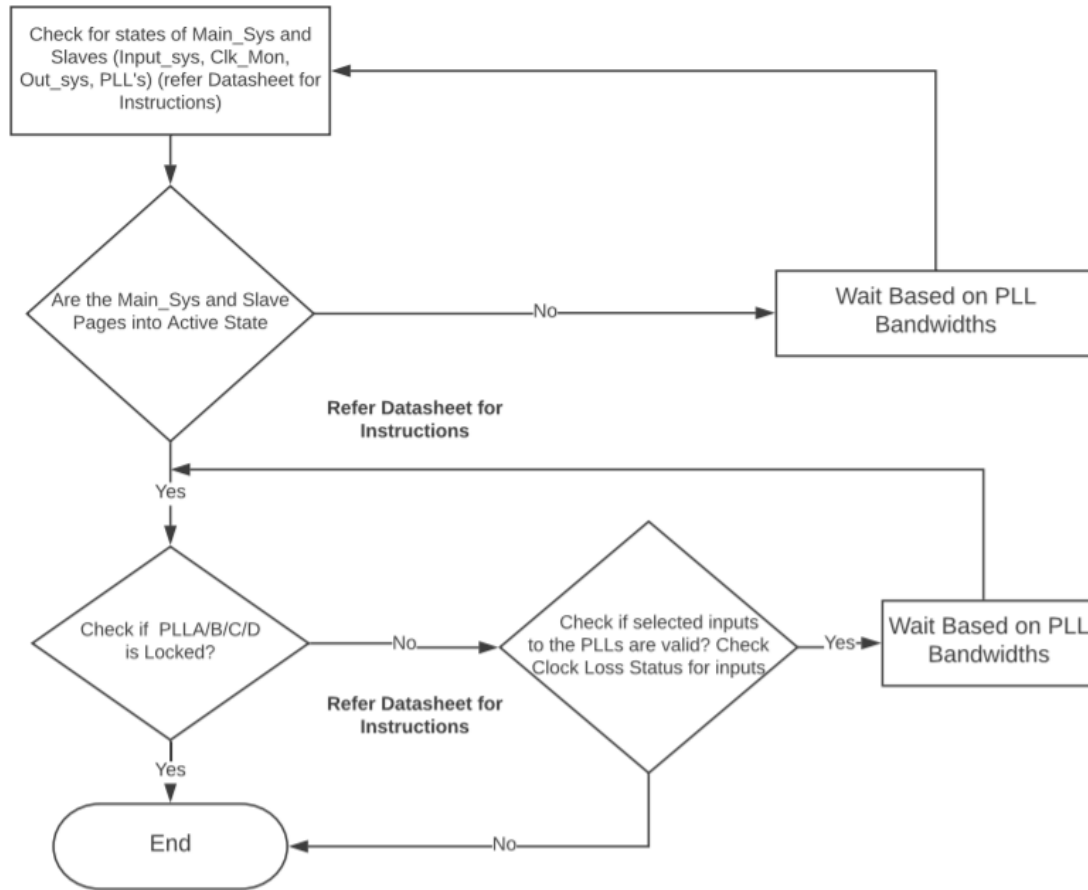


Figure 45. Active State Confirmation

Confirm from each page that was initialized that their respective state machines are in Active State. Check Lock Status of PLLs that are enabled.

Monitoring the Status for Master and Slave Pages

Once the SiT95145 is powered up, the NVM register contents are read, it is preferable to provide sufficient time delay for the read operation.

As explained in the data sheet above, we need to read that the Master Control Page 0 is in the Program Command Wait State (PRG_CMD State) before starting to write to the part. Please use the register write sequence from the GUI generated from the “Dump” button to write into the device.

Once the entire sequence to program the profile has been written, each Page (Pages 0, 1, 2, 3, PLL Pages A, B, C, D based on which PLLs are enabled) can be checked to ensure they are in the ACTIVE state.

The sequence that needs to be followed to check the status of the Master and Slave Pages is **as below**.

MAIN PAGE (0)

- 1) Write 0xFF register to 0x00. # Go to Page 0
- 2) Write 0xD1 register on Page 0 to 0x40.
- 3) Read the register 0xD0.

If read data == 9 (Main Sys is in PRG_CMD state)

read data == 0 (Main Sys is in IDLE state)

read data == 36 (Main Sys is in Active state) # revC

CLK MON PAGE (1)

- 1) Write 0xFF register to 0x01. # Go to Page 1
- 2) Write 0xD1 register on Page 1 to 0x40.
- 3) Read ((0xD0) & 0x0F).

If read data == 5 (CLK MON is in PRG_CMD state)

read data == 0 (CLK MON is in IDLE state)

read data == 12 (CLK MON is in Active state)

INPUT_SYS PAGE (2)

- 1) Write 0xFF register to 0x02. # Go to Page 2
- 2) Write 0xD1 register on Page 2 to 0x40.
- 3) Read ((0xD0) & 0x1F).

If read data == 7 (INPUT_SYS is in PRG_CMD state)

read data == 0 (INPUT_SYS is in IDLE state)

read data == 23 (INPUT_SYS is in Active state)

OUT_SYS PAGE (3)

- 1) Write 0xFF register to 0x03. # Go to Page 3
- 2) Write 0xD1 register on Page 3 to 0x60.
- 3) Read ((0xD0) & 0x1F).

If read data == 12 (OUT_SYS is in PRG_CMD state)

read data == 0 (OUT_SYS is in IDLE state)

read data == 20 (OUT_SYS is in Active state)

PLL PAGE (A, B, C, D)

- 1) Write 0xFF register to 0x0A(PLLA). # (0x0A - PLLA, 0x0B – PLLB, 0x0C-PLLC, 0x0D-PLLD)
- 2) Write 0xD1 register on PLL Page to 0x20.
- 3) Read the register 0xD0.

If read data == 8(PLL PAGE is in PRG_CMD state)

read data == 0 (PLL PAGE is in IDLE state)

read data == 48 (PLL PAGE is in Active state) #revC

Monitoring the Loss of Lock Status for PLL

The sequence that needs to be followed to monitor the dynamic LOL status of the PLL's is as below:

- 1) Write 0xFF register to 0x0A(PLLA). # (0x0A - PLLA, 0x0B – PLLB, 0x0C-PLLC, 0x0D-PLLD)
- 2) Write 0x04 register on PLL Page to 0x01. # Remove the mask for lock loss notify status
- 3) Read ((0x02) & 0x01). # LOL dynamic status_PLLA_B_C_D

If read data == 1 (Loss of Lock is asserted: PLL is not locked)

read data == 0 (PLL is locked)

Monitoring the Hold Over Status for PLLs

The sequence that needs to be followed to monitor the Hold Over status of the PLL's is as below:

- 1) Write 0xFF register to 0x0A(PLLA). # (0x0A - PLLA, 0x0B – PLLB, 0x0C-PLLC, 0x0D-PLLD)
- 2) Read ((0x17)>>7) & 0x01). # To check if the PLL Outer Loop is Enabled/Disabled

If read data == 1

It indicates a Free Run Profile --- The PLL's will always be in Hold Over

read data == 0

It is not a Free Run Profile ----- PLL may or may not be in Hold Over

- 3) Write 0xB3 register to 0x0D.

- 4) Read ((0xb9)>>3) & 0x01).

Read the 3rd bit in B9 register for Hold Over Status Information

If read data == 1

PLL is in Holdover State

read data == 0

PLL is not in Holdover State

Programming the Primary E-Fuse

This section describes primary E-Fuse program configuration for all the pages [GENERIC_SYS, INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS].

VDD=2.5V and VDDIN=2.5V should be used for programming the E-Fuse.

Configuration Bits to Force Power-up of Digital Slave Subsystems

The respective subsystems needs to be powered up before programming the E-Fuse

This section describes GENERIC_SYS page configuration required in register 0xe0 and 0xe1 to enable respective slave subsystems [INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS] as described in [Table 23](#).

Table 23. Configuration Bits to Force Power-up of Digital Slave Subsystems

| S.NO | Page Number | Register Address | Bit Number | Value & It's Description |
|------|-------------|------------------|------------|--|
| 1 | Page 0 | 0xE0 | 0 | 1'h1 (write to bit number 0 in register address 0xE0 with value 1'h1 to enable force overwrite INPUT_SYS) |
| | | | 1 | 1'h1 (write to bit number 1 in register address 0xE0 with value 1'h1 to enable force overwrite CLKMON_SYS) |
| | | | 2 | 1'h1 (write to bit number 2 in register address 0xE0 with value 1'h1 to enable force overwrite OUTPUT_SYS) |
| | | | 3 | 1'h1 (write to bit number 3 in register address 0xE0 with value 1'h1 to enable force overwrite PLLA_SYS) |
| | | | 4 | 1'h1 (write to bit number 4 in register address 0xE0 with value 1'h1 to enable force overwrite PLLB_SYS) |
| | | | 5 | 1'h1 (write to bit number 5 in register address 0xE0 with value 1'h1 to enable force overwrite PLLC_SYS) |
| | | | 6 | 1'h1 (write to bit number 6 in register address 0xE0 with value 1'h1 to enable force overwrite PLLD_SYS) |
| 2 | Page 0 | 0xE1 | 0 | 1'h1 (write to bit number 0 in register address 0xE1 with value 1'h1 INPUT_SYS to be Enabled) |
| | | | 1 | 1'h1 (write to bit number 1 in register address 0xE1 with value 1'h1 CLKMON_SYS to be Enabled) |
| | | | 2 | 1'h1 (write to bit number 2 in register address 0xE1 with value 1'h1 OUTPUT_SYS to be Enabled) |
| | | | 3 | 1'h1 (write to bit number 3 in register address 0xE1 with value 1'h1 PLLA_SYS to be Enabled) |
| | | | 4 | 1'h1 (write to bit number 4 in register address 0xE1 with value 1'h1 PLLB_SYS to be Enabled) |
| | | | 5 | 1'h1 (write to bit number 5 in register address 0xE1 with value 1'h1 PLLC_SYS to be Enabled) |
| | | | 6 | 1'h1 (write to bit number 5 in register address 0xE1 with value 1'h1 PLLD_SYS to be Enabled) |

E-Fuse Lock Configuration Bits

This section describes the location of the two bits to lock the respective slave subsystem E-Fuse by writing into the register 0x2F in all pages as described in [Table 24](#).

Table 24. E-Fuse Lock Configuration Bits for all pages

| S.NO | Page Number | Register Address | Bit Number | Value & It's Description |
|------|-------------|------------------|------------|--|
| 1 | Page 0 | 0x2f | 7:6 | 2'h1 : E-Fuse of GENERIC_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 2 | Page 1 | 0x2f | 7:6 | 2'h1 : E-Fuse of CLKMON_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 3 | Page 2 | 0x2f | 7:6 | 2'h1 : E-Fuse of INPUT_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 4 | Page 3 | 0x2f | 7:6 | 2'h1 : E-Fuse of OUTPUT_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 5 | Page A | 0x2f | 7:6 | 2'h1 : E-Fuse of PLLA_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 6 | Page B | 0x2f | 7:6 | 2'h1 : E-Fuse of PLLB_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 7 | Page C | 0x2f | 7:6 | 2'h1 : E-Fuse of PLLC_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |
| 8 | Page D | 0x2f | 7:6 | 2'h1 : E-Fuse of PLLD_SYS is locked by writing into bit number [7:6] in register address 0x2f with value 2'h1 |

E-Fuse Write Configuration Bits

This section will describe how to program the E-Fuse by writing into the register 0x0F in all the pages as mentioned in [Table 25](#).

Table 25. E-Fuse Write Configuration Bits for all pages

| S.NO | Page Number | Register Address | Bit Number | Value & It's Description |
|------|-------------|------------------|------------|--|
| 1 | Page 0 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in GENERIC_SYS |
| 2 | Page 1 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in CLKMON_SYS |
| 3 | Page 2 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in INPUT_SYS |
| 4 | Page 3 | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in OUTPUT_SYS |
| 5 | Page A | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in PLLA_SYS |
| 6 | Page B | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in PLLB_SYS |
| 7 | Page C | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in PLLC_SYS |
| 8 | Page D | 0x0f | 7:3 | 5'h0C: writing to bit number [7:3] in register address 0x0f with value 5'h0C to do E-Fuse write in PLLD_SYS |

Configuration Bit to Remove Manual Wake Up for Primary E-Fuse

Manual wake-up of slave subsystems is enabled while programming the E-Fuse. As a final step the manual wake-up mode needs to be disabled in the GENERIC_SYS page so that the chip will wake-up autonomously

This section will describe how to remove manual wakeup by writing into a register 0x2f in GENERIC_SYS as mentioned in [Table 26](#).

Table 26. Configuration Bit to Remove Manual Wakeup for Primary E-Fuse

| S.NO | Page Number | Register Address | Bit Number | Value & It's Description |
|------|-------------|------------------|------------|---|
| 1 | Page 0 | 0x2f | 4 | 1'h1 : (writing to bit number 4 in register address 0x2f with value 'h1 to remove manual wake up) |

Pseudo Code: Programming the Primary E-Fuse

VDD and VDDIN supply should be set to 2.5V while programming the E-Fuse

GENERIC_SYS

STEP 1: Write the GENERIC_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x00)
```

STEP 2: Write the GENERIC_SYS NVM Registers configuration

STEP 3: Refer [E-Fuse Write Configuration Bits](#) section described earlier.

Write to bit number [7:3] in register address 0x0f with value 5'hC to program E-Fuse registers of GENERIC_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
```

```
i2c.i2cw(device_address, 0x0f, 0xc0)
```

```
i2c.i2cw(device_address, 0x0f, 0x00)
```

STEP 4: Refer [Configuration Bits to Force Power-up of Digital Slave Subsystems](#) described earlier

Force Enable all slaves [INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS] by programming 0xE0 & 0xE1 both with value 7'h7F

```
i2c.i2cw(device_address, 0xe0, 0x7f)
```

```
i2c.i2cw(device_address, 0xe1, 0x7f)
```

INPUT_SYS

STEP 5: Write the INPUT_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x02)
```

STEP 6: Refer [E-Fuse Lock Configuration Bits](#)

Write the INPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of INPUT_SYS

STEP 7: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0f with value 5'hC to program E-Fuse Registers of INPUT_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
```

```
i2c.i2cw(device_address, 0x0f, 0xc0)
```

```
i2c.i2cw(device_address, 0x0f, 0x00)
```

CLKMON_SYS

STEP 8: Write the CLKMON_SYS page number configuration
`i2c.i2cw(device_address, 0xff, 0x01)`

STEP 9: Refer [E-Fuse Lock Configuration Bits](#)

Write the CLKMON_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of CLKMON_SYS

STEP 10: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of CLKMON_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

OUTPUT_SYS

STEP 11: Write the OUTPUT_SYS page number configuration
`i2c.i2cw(device_address, 0xff, 0x03)`

STEP 12: Refer [E-Fuse Lock Configuration Bits](#)

Write the OUTPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of OUTPUT_SYS

STEP 13: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of OUTPUT_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

PLLA_SYS

STEP 14: Write the PLLA_SYS page number configuration
`i2c.i2cw(device_address, 0xff, 0x0a)`

STEP 15: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLLA_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLA_SYS

STEP 16: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLA_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

PLLB_SYS

STEP 17: Write the PLLB_SYS page number configuration
`i2c.i2cw(device_address, 0xff, 0x0b)`

STEP 18: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLLB_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLB_SYS

STEP 19: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLB_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```


PLL_C_SYS

STEP 20: Write the PLL_C_SYS page number configuration
`i2c.i2cw(device_address, 0xff, 0x0c)`

STEP 21: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLL_C_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLL_C_SYS

STEP 22: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLL_C_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

PLL_D_SYS

STEP 23: Write the PLL_D_SYS page number configuration
`i2c.i2cw(device_address, 0xff, 0x0d)`

STEP 24: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLL_D_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLL_D_SYS

STEP 25: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLL_D_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

Removing Manual Wake-up and Locking GENERIC_SYS E-Fuse

STEP 26: Write the GENERIC_SYS page number configuration
`i2c.i2cw(device_address, 0xff, 0x00)`

STEP 27: Refer [Configuration Bit to Remove Manual Wake Up for Primary E-Fuse](#)

Write to bit number 4 in register address 0x2F with value 1'h1 in GENERIC_SYS to Configuration Bit to Remove Manual Wake Up

```
i2c.i2crmw(device_address, 0x2f, 0x4, 0x1, 0x01) //i2crmw function register 0x2f bit number 4 writing with the value 1'h1
```

Note: `i2crmw(dev_address, register_address, bit_postion, total_bits, value)`

STEP 28: Refer [E-Fuse Lock Configuration Bits](#)

Write to bit number [7:6] in register address 0x2F with value 2'h1 in in GENERIC_SYS to lock the E-Fuse

```
i2c.i2crmw(device_address, 0x2f, 0x7, 0x2, 0x01) //i2crmw function register 0x2f bit number 7:6 writing with the value 2'h1
```

STEP 29: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of GENERIC_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

STEP 30: Reset the chip or recycle Power for the chip to wake-up autonomously based on NVM configuration programmed in the primary E-Fuse.

Programming the Secondary E-Fuse

This section describes Secondary E-Fuse program configuration for all the pages [GENERIC_SYS, INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS].

Configuration Bit to Escape to PROGRAM_CMD State in GENERIC_SYS

With the primary E-Fuse programmed and locked the chip will autonomously wake-up and reach ACTIVE_STATE. For the E-Fuse writes the chip needs to be in PROGRAM_CMD state.

This section will describe how to do escape from ACTIVE_STATE to PROGRAM_CMD state in GENERIC_SYS by writing into the register 0x0F in GENERIC_SYS as mentioned in [Table 27](#).

Table 27. Configuration Bit To Escape to PROGRAM_CMD State in GENERIC_SYS

| S.NO | Page Number | Register Address | Bit Number | Value & It's Description |
|------|-------------|------------------|------------|---|
| 1 | Page 0 | 0x0f | 1 | 1'h1: writing to bit number 1 in register address 0x0f with value 1'h1 to do Escape to PROGRAM_CMD state in GENERIC_SYS |

Configuration Bit to Change the E-Fuse pointer

This section will describe how to point slaves to secondary E-Fuse by writing into a register 0x22 in GENERIC_SYS as mentioned in [Table 28](#).

Table 28. Configuration Bit to change the E-Fuse pointer

| S.NO | Page Number | Register Address | Bit Number | Value & It's Description |
|------|-------------|------------------|------------|---|
| 1 | Page 0 | 0x22 | 7:0 | 8'hFF : (writing to bit number [7:0] in register address 0x22 with value 8'hFF to point slaves [INPUT_SYS,CLKMON_SYS, OUTPUT_SYS,PLLA_SYS, PLLB_SYS,PLLC_SYS,PLLD_SYS] to secondary E-Fuse) |

Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse

This section will describe how to enable manual wakeup for secondary E-Fuse by writing into a register 0x10 in GENERIC_SYS as mentioned in [Table 29](#).

Table 29. Configuration Bit to Enable Manual Wakeup for Secondary E-Fuse

| S.NO | Page Number | Register Address | Bit Number | Value & It's Description |
|------|-------------|------------------|------------|--|
| 1 | Page 0 | 0x10 | 7 | 1'h1 : (writing to bit number 7 in register address 0x10 with value 1'h1 to select manual wake up) |
| 2 | Page 0 | 0x10 | 6 | 1'h0 : (writing to bit number 6 in register address 0x10 with value 1'h0 to select manual wake up) |

Pseudo Code: Programming the Secondary E-Fuse

GENERIC_SYS

STEP 1: Write the GENERIC_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x00)
```

STEP 2: Refer [Configuration Bit To Escape To PROGRAM_CMD State in GENERIC_SYS](#)

Write to bit number 1 in register address 0x0F with value 1'h1 to escape to PROGRAM_CMD state in GENERIC_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0x02)
i2c.i2cw(device_address, 0x0f, 0x00)
```

STEP 3: Refer [Configuration Bit to Change the E-Fuse pointer](#)

Write to bit number [7:0] in register address 0x22 with value 8'hFF to point slaves [INPUT_SYS,CLKMON_SYS,OUTPUT_SYS,PLLA_SYS,PLLB_SYS,PLLC_SYS,PLLD_SYS] to secondary E-Fuse in GENERIC_SYS

```
i2c.i2cw(device_address, 0x22, 0xff)
```

STEP 4: Refer [Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse](#)

Write to bit number 7 in register address 0x10 with value 1'h1 to select for manual wakeup in GENERIC_SYS

`i2c.i2crmw(device_address, 0x10, 0x7, 0x1, 0x01)` //i2crmw function register 0x10 bit number 7 writing with the value 1'h1

STEP 5: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of GENERIC_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

STEP 6: Reset the chip or recycle power for the E-Fuse pointers to get updated

STEP 7: Refer [Configuration Bits to Force Power-up of Digital Slave Subsystems](#)

Force Enable all slaves [INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS] by programming 0xE0 & 0xE1 both with value 7'h7F

```
i2c.i2cw(device_address, 0xe0, 0x7f)
i2c.i2cw(device_address, 0xe1, 0x7f)
```

INPUT_SYS

STEP 8: Write the INPUT_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x02)
```

STEP 9: Refer [E-Fuse Lock Configuration Bits](#)

Write the INPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of INPUT_SYS

STEP 10: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of INPUT_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

CLKMON_SYS

STEP 11: Write the CLKMON_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x01)
```

STEP 12: Refer [E-Fuse Lock Configuration Bits](#)

Write the CLKMON_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of CLKMON_SYS

STEP 13: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of CLKMON_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

OUTPUT_SYS

STEP 14: Write the OUTPUT_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x03)
```

STEP 15: Refer [E-Fuse Lock Configuration Bits](#)

Write the OUTPUT_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of OUTPUT_SYS

STEP 16: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of OUTPUT_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

PLLA_SYS

STEP 17: Write the PLLA_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x0a)
```

STEP 18: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLLA_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLA_SYS

STEP 19: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLA_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

PLLB_SYS

STEP 20: Write the PLLB_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x0b)
```

STEP 21: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLLB_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLB_SYS

STEP 22: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLLB_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

PLL_C_SYS

STEP 23: Write the PLL_C_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x0c)
```

STEP 24: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLL_C_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLL_C_SYS

STEP 25: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLL_C_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

PLL_D_SYS

STEP 26: Write the PLL_D_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x0d)
```

STEP 27: Refer [E-Fuse Lock Configuration Bits](#)

Write the PLL_D_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLL_D_SYS

STEP 28: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of PLL_D_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

Programming GENERIC_SYS E-Fuse to Remove Manual Wake Up in Secondary E-Fuse

STEP 29: Write the GENERIC_SYS page number configuration

```
i2c.i2cw(device_address, 0xff, 0x00)
```

STEP 30: Refer [Configuration Bit To Escape To PROGRAM_CMD State in GENERIC_SYS](#)

Write to bit number 1 in register address 0x0F with value 1'h1 to escape to PROGRAM_CMD state in GENERIC_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0x02)
i2c.i2cw(device_address, 0x0f, 0x00)
```

STEP 31: Write to bit number 6 in register address 0x10 with value 1'h1 in GENERIC_SYS to Configuration Bit to Remove Manual Wake Up in secondary E-Fuse

```
i2c.i2crmw(device_address, 0x10, 6, 0x1, 0x01) //i2crmw function register 0x10 bit number 6 writing with the value 1'h1
```

Note: i2crmw(dev_address,register_address,bit_postion,total_bits,value)

STEP 32: Refer [E-Fuse Write Configuration Bits](#)

Write to bit number [7:3] in register address 0x0F with value 5'hC to program E-Fuse Registers of GENERIC_SYS

```
i2c.i2cw(device_address, 0x0f, 0x00)
i2c.i2cw(device_address, 0x0f, 0xc0)
i2c.i2cw(device_address, 0x0f, 0x00)
```

STEP 33: Reset Chip or recycle power for the chip to wake-up autonomously from the second E-Fuse

Register Map Details

Table 30. PAGE 0: Generic Master System Related registers:
Registers from 10h to 4Fh are equivalent NVMCopy Registers for this Page

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|--------------------|-----------------|-----------|-------------|-------------------|--|
| VERSION_ID | 01h | 7:0 | R | Cxh (SiT95145) | Overall Platform Revision x = 1 or 2 |
| STATUS_1_GENERIC | 02h | 7 | R | 00h | SPARE |
| | | 6 | R | 00h | SPARE |
| | | 5 | R | 00h | SPARE |
| | | 4 | R | 00h | SPARE |
| | | 3 | R | 00h | SPARE |
| | | 2 | R | 00h | Dynamic status for xoclk_loss, |
| | | 1 | R | 00h | SPARE |
| | | 0 | R | 00h | SPARE |
| NOTIFY_1_GENERIC | 03h | 7 | R/W | 1h | SPARE |
| | | 6 | R/W | 1h | SPARE |
| | | 5 | R/W | 1h | SPARE |
| | | 4 | R/W | 1h | SPARE |
| | | 3 | R/W | 1h | SPARE |
| | | 2 | R/W | 1h | Sticky/Notify status for _xoclk_loss, |
| | | 1 | R/W | 1h | SPARE |
| | | 0 | R/W | 1h | SPARE |
| MASKb_1_GENERIC | 04h | 7 | R/W | 1h | SPARE |
| | | 6 | R/W | 1h | SPARE |
| | | 5 | R/W | 1h | SPARE |
| | | 4 | R/W | 1h | SPARE |
| | | 3 | R/W | 1h | SPARE |
| | | 2 | R/W | 1h | Mask bit for NOTIFY_1_GENERIC (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[2]. |
| | | 1 | R/W | 1h | SPARE |
| | | 0 | R/W | 1h | SPARE |
| Directives_GENERIC | 05h | 3:2 | R/W | 00h | 0 : DCO_PLLA 1 : DCO_PLLB 2 : DCO_PLLC 3 : DCO_PLLD |
| | | 1 | R/W | 00h | dco increment commonly used for all plls if 0xe7[7] is set as '1' |
| | | 0 | R/W | 00h | dco decrement commonly used for all plls if 0xe7[7] is set as '1' |
| STATUS_2_GENERIC | 06h | 7 | R | 00h | Dynamic status for plld_ho_freeze |
| | | 6 | R | 00h | Dynamic status for pll_c_ho_freeze |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|-------------------------------|-----------------|-----------|-------------|---------------|--|
| | | 5 | R | 00h | Dynamic status for pll_b_ho_freeze |
| | | 4 | R | 00h | Dynamic status for pll_a_ho_freeze |
| | | 3 | R | 00h | Dynamic status for plld_loss_of_lock |
| | | 2 | R | 00h | Dynamic status for pll_c_loss_of_lock |
| | | 1 | R | 00h | Dynamic status for pll_b_loss_of_lock |
| | | 0 | R | 00h | Dynamic status for pll_a_loss_of_lock |
| NOTIFY_2_GENERIC | 07h | 7 | R/W | 1h | Sticky/Notify status for plld_ho_freeze |
| | | 6 | R/W | 1h | Sticky/Notify status for pll_c_ho_freeze |
| | | 5 | R/W | 1h | Sticky/Notify status for pll_b_ho_freeze |
| | | 4 | R/W | 1h | Sticky/Notify status for pll_a_ho_freeze |
| | | 3 | R/W | 1h | Sticky/Notify status for plld_loss_of_lock |
| | | 2 | R/W | 1h | Sticky/Notify status for pll_c_loss_of_lock |
| | | 1 | R/W | 1h | Sticky/Notify status for pll_b_loss_of_lock |
| | | 0 | R/W | 1h | Sticky/Notify status for pll_a_loss_of_lock |
| MASKb_2_GENERIC | 08h | 7 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[7]. |
| | | 6 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[6]. |
| | | 5 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[5]. |
| | | 4 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[4]. |
| | | 3 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[3]. |
| | | 2 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[2]. |
| | | 1 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[1]. |
| | | 0 | R/W | 1h | Mask bit for NOTIFY_2_GENERIC (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[0]. |
| PRG_Directives_GENERIC | 0Fh | 7:3 | R/W | 0h | PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active |
| | | 2 | R/W | 0h | SPARE |
| | | 1 | R/W | 0h | Escape to the PRG_CMD state from ACTIVE state |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|--------------------|-----------------|-----------|-------------|---------------|--|
| NVMPLEN_GENERIC | 10h | 7 | R/W | 0h | Selects between manual_wake_upb and manual_wake_up2b |
| | | 6 | R/W | 0h | select '0' to enable manual wake up sequence |
| | | 5:4 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)} |
| | | 3:0 | R/W | 0h | Bits {3,2,1,0} correspond to enable of PLL {D,C,B,A}: Active Low. |
| NVMFLEXIO8_GENERIC | 18h | 7 | R/W | 0h | Select the direction of FLEXIO14. '0' – Input, '1' – Output |
| | | 6:4 | R/W | 0h | Mux select to bring out internal signals on FLEXIO14, if programmed as output 3'b000 : 0 3'b001 : Ored_all_notify (clock loss for clk0/1/2/3/ + Freq drift coarse for clk0/1/2/3 + Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b010 : Ored_all_pll_ntfy (Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b011 : Ored_all_clkmon_ntfy ((clock loss for clk0/1/2/3/ + Freq drift coarse for clk0/1/2/3) 3'b100 : Dynamic_Clock_Monitoring_Status 3'b101 : Internal debug 3'b110 : Internal debug 3'b111 : Internal debug |
| | | 3 | R/W | 0h | Select the direction of FLEXIO15. '0' – Input, '1' – Output |
| | | 2:0 | R/W | 0h | Mux select to bring out internal signals on FLEXIO15, if programmed as output 3'b000 : 0 3'b001 : Ored_all_notify (clock loss for clk0/1/2/3/ + Freq drift coarse for clk0/1/2/3 + Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b010 : Ored_all_pll_ntfy (Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b011 : Ored_all_clkmon_ntfy ((clock loss for clk0/1/2/3/ + Freq drift coarse for clk0/1/2/3) 3'b100 : Dynamic_Clock_Monitoring_Status 3'b101 : Internal debug 3'b110 : Internal debug 3'b111 : Internal debug |
| NVMSPPARE1_GENERIC | 19h | 7 | R/W | 0h | if set '1', Enables vdd padding functionality(0x23[7]) for padding selection. Keeping at 0 enables pad rail switch, select 1 to ensure no switch and keep rail to left |
| | | 6 | R/W | 0h | To map the external clock in select and allow the each pll independently to force into holdover mode. Each pll ctrl - refer 0x2B[3:0] |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|------------------------------|-----------------|-----------|-------------|---------------|---|
| FUSE_PTR_GENERIC | 22h | 7:0 | R/W | 0h | One Hot Decode for Fuse Pointer: Bits {7,6,5,4,3,2,1,0} correspond to fuse pointers for Pages 1, 2, 3, 4, A, B, C, D respectively Page 4 is reserved and not used. |
| XO4_GENERIC | 23h | 7 | R/W | 0h | '1' => Set VDDIO as VDDIN & '0' => Set VDDIO as VDD |
| BTOUT_IN_EN_GENERIC | 24h | 7:4 | R/W | 0h | One Hot Input Enable for the Clock Inputs 3:0 which are the 4 Clock inputs defined on Page 2: Active Low |
| | | 3:2 | R/W | 0h | One Hot Output Enable for Bottom Outputs 1:0 which are the 2 Bottom Fixed-Outputs defined on Page 3: Active Low |
| | | 1:0 | R/W | 0h | One Hot Output Enable for Top Outputs 1:0 which are the 2 Top Fixed-Outputs defined on Page 3: Active Low |
| FLEXOUTPUT_EN_GENERIC | 25h | 7:0 | R/W | 0h | One Hot Output Enable for Outputs 7:0 which are the 8 Flex-Outputs defined on Page 3: Active Low |
| OEB_CTRL | 26h | 7:6 | R/W | 0h | Selects which PLL to run in free mode at wake up , this feature is enables by 0x23[5]: 0 : PLLA 1 : PLLB 2 : PLLC 3 : PLLD |
| | | 5:4 | R/W | 0h | Reserved |
| | | 3:2 | R/W | 0h | To program delay between enabling PLLA and other PLLs in fuse locked modes. 2'b00 - 4ms 2'b01 - 40 ms 2'b10 - 400 ms 2'b11 - 4s |
| | | 1 | R/W | 0h | if set as '1' then set PLLX_OEB as '1' if loss of lock status is asserted for respective PLL |
| | | 0 | R/W | 0h | if set as '1' then set PLLX_OEB as '1' if there is XO clock loss |
| INTR_MASK_1_CONFIG | 27h | 7 | R/W | 0h | If set '1', Interrupt will not be generated for DRIFT in freq for clk_in3 |
| | | 6 | R/W | 0h | If set '1', Interrupt will not be generated for DRIFT in freq for clk_in2 |
| | | 5 | R/W | 0h | If set '1', Interrupt will not be generated for DRIFT in freq for clk_in1 |
| | | 4 | R/W | 0h | If set '1', Interrupt will not be generated for DRIFT in freq for clk_in0 |
| | | 3 | R/W | 0h | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in3 |
| | | 2 | R/W | 0h | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in2 |
| | | 1 | R/W | 0h | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in1 |
| | | 0 | R/W | 0h | If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in0 |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------------------|-----------------|-----------|-------------|---------------|---|
| INTR_MASK_2_CONFIG | 28h | 7 | R/W | 0h | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLA |
| | | 6 | R/W | 0h | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLB |
| | | 5 | R/W | 0h | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLC |
| | | 4 | R/W | 0h | If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLD |
| | | 3 | R/W | 0h | If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLA |
| | | 2 | R/W | 0h | If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLB |
| | | 1 | R/W | 0h | If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLC |
| | | 0 | R/W | 0h | If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLD |
| CHIP_FLEXIO_CONFIG | 29h | 7 | R/W | 0h | Reserved |
| | | 6 | R/W | 0h | |
| | | 5 | R/W | 0h | |
| | | 4 | R/W | 0h | |
| | | 3 | R/W | 0h | |
| | | 2:0 | R/W | 0h | 001: SiT95145 |
| I2C_GENERIC | 2Ah | 7 | R/W | 0h | Enable the new I2C Address (if changed from the default 0x69) |
| | | 6:2 | R/W | 0h | I2C Address static bits (when changed from 0b11010) |
| | | 0:1 | R/W | 0h | Reserved |
| OE_PATTERN_GENERIC | 2Fh | 7:6 | R/W | 0h | PATTERN :{'01'/'10'} =] Efuse Locked, {'00'/'11'} =] Efuse NOT Locked |
| | | 5 | R/W | 0h | High Speed Enable for the I2C pads, HS_EN=0 ensures true I2C function (no pull up) for I2C. Don't care for SPI. |
| | | 4 | R/W | 0h | select '0' to enable manual wake up sequence |
| | | 3:0 | R/W | 0h | Reserved |
| OverwriteCLKIN1_2_GENERIC | E7h | 7 | R/W | 0h | EN_COMMON_DCO_DECR |
| RESET_REGISTER | FEh | 7:0 | R/W | 0h | Register FE is a READ / WRITE register used to reset the chip. Writing 0x01 to this register will apply the reset to the digital and this is propagated asynchronously to the digital blocks. On writing a 0 to bit 0 of register FE, the reset is de-asserted and internal to the digital, the various clocks are used to perform the de-assertion reset asynchronization for the appropriate clock domains. |
| PAGE_NUMBER | FFh | 7:0 | R/W | 0h | On all pages register FF is a READ / WRITE register used to change the page number |

Table 31. PAGE 1: Clock Monitor System Related registers:
Registers from 11h to 4Fh are equivalent NVMCopy Registers for this Page

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------|-----------------|-----------|-------------|---------------|--|
| STATUS_CLKMON | 02h | 7 | R | 00h | Dynamic status for clk_in3_loss_OR_with_FD |
| | | 6 | R | 00h | Dynamic status for clk_in2_loss_OR_with_FD |
| | | 5 | R | 00h | Dynamic status for clk_in1_loss_OR_with_FD |
| | | 4 | R | 00h | Dynamic status for clk_in0_loss_OR_with_FD |
| | | 3 | R | 00h | Dynamic status for clk_in3_loss |
| | | 2 | R | 00h | Dynamic status for clk_in2_loss |
| | | 1 | R | 00h | Dynamic status for clk_in1_loss |
| | | 0 | R | 00h | Dynamic status for clk_in0_loss |
| NOTIFY_CLKMON | 03h | 7 | R/W | 1h | Sticky/Notify status for clk_in3_loss_OR_with_FD |
| | | 6 | R/W | 1h | Sticky/Notify status for clk_in2_loss_OR_with_FD |
| | | 5 | R/W | 1h | Sticky/Notify status for clk_in1_loss_OR_with_FD |
| | | 4 | R/W | 1h | Sticky/Notify status for clk_in0_loss_OR_with_FD |
| | | 3 | R/W | 1h | Sticky/Notify status for clk_in3_loss |
| | | 2 | R/W | 1h | Sticky/Notify status for clk_in2_loss |
| | | 1 | R/W | 1h | Sticky/Notify status for clk_in1_loss |
| | | 0 | R/W | 1h | Sticky/Notify status for clk_in0_loss |
| MASKb_CLKMON | 04h | 7 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[7] |
| | | 6 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[6] |
| | | 5 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[5] |
| | | 4 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[4] |
| | | 3 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[3] |
| | | 2 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[2] |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|-------------------------------|-----------------|-----------|-------------|---------------|---|
| | | 1 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[1] |
| | | 0 | R/W | 1h | Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[0] |
| STATUS_FDCOARSE_CLKMON | 06h | 7 | R | 00h | Dynamic status for clk3_freq_coarse_drifted |
| | | 6 | R | 00h | Dynamic status for clk2_freq_coarse_drifted |
| | | 5 | R | 00h | Dynamic status for clk1_freq_coarse_drifted |
| | | 4 | R | 00h | Dynamic status for clk0_freq_coarse_drifted |
| | | 3 | R | 00h | Dynamic status for clk3_freq_fine_drifted |
| | | 2 | R | 00h | Dynamic status for clk2_freq_fine_drifted |
| | | 1 | R | 00h | Dynamic status for clk1_freq_fine_drifted |
| | | 0 | R | 00h | Dynamic status for clk0_freq_fine_drifted |
| NOTIFY_FDCOARSE_CLKMON | 07h | 7 | R/W | 1h | Sticky/Notify status for clk3_freq_coarse_drifted |
| | | 6 | R/W | 1h | Sticky/Notify status for clk2_freq_coarse_drifted |
| | | 5 | R/W | 1h | Sticky/Notify status for clk1_freq_coarse_drifted |
| | | 4 | R/W | 1h | Sticky/Notify status for clk0_freq_coarse_drifted |
| | | 3 | R/W | 1h | Sticky/Notify status for clk3_freq_fine_drifted |
| | | 2 | R/W | 1h | Sticky/Notify status for clk2_freq_fine_drifted |
| | | 1 | R/W | 1h | Sticky/Notify status for clk1_freq_fine_drifted |
| | | 0 | R/W | 1h | Sticky/Notify status for clk0_freq_fine_drifted |
| MASKb_FDCOARSE_CLKMON | 08h | 7 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[7] |
| | | 6 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[6] |
| | | 5 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[5] |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------------------|-----------------|-----------|-------------|---------------|--|
| | | 4 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[4] |
| | | 3 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[3] |
| | | 2 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[2] |
| | | 1 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[1] |
| | | 0 | R/W | 1h | Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[0] |
| FD32_STATUS_COARSE_CLKMON | 0Dh | 7:0 | R/W | 0h | Reserved |
| FD10_STATUS_COARSE_CLKMON | 0Eh | 7:0 | R/W | 0h | Reserved |
| PRG_Directives_CLKMON | 0Fh | 7:3 | R/W | 0h | PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active |
| | | 2 | R/W | 0h | Spare |
| | | 1 | R/W | 0h | Escape to the PRG_CMD state from ACTIVE state |
| CL_REG2_CLKMON | 11h | 7:6 | R/W | 0h | IN3_VAL_TIME: timer setting for deassertion of clock loss for FD3 Values {0,1,2,3} correspond to {2ms, 100ms, 200ms, 1sec} |
| | | 5:4 | R/W | 0h | IN2_VAL_TIME : timer setting for deassertion of clock loss for FD2 Values {0,1,2,3} correspond to {2ms, 100ms, 200ms, 1sec} |
| | | 3:2 | R/W | 0h | IN1_VAL_TIME : timer setting for deassertion of clock loss for FD1 Values {0,1,2,3} correspond to {2ms, 100ms, 200ms, 1sec} |
| | | 1:0 | R/W | 0h | IN0_VAL_TIME : timer setting for deassertion of clock loss for FD0 Values {0,1,2,3} correspond to {2ms, 100ms, 200ms, 1sec} |
| CL3_SET_THR2_CLKMON | 12h | 7:3 | R/W | 0h | SPARE |
| | | 2:0 | R/W | 0h | IN3 19-bit Threshold Value GUI calculates the correct CL register threshold trigger value for IN3 |
| CL3_SET_THR1_CLKMON | 13h | 7:0 | R/W | 0h | |
| CL3_SET_THR0_CLKMON | 14h | 7:0 | R/W | 0h | |
| CL2_SET_THR2_CLKMON | 15h | 7:3 | R/W | 0h | SPARE |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------------|-----------------|-----------|-------------|---------------|---|
| | | 2:0 | R/W | 0h | IN2 19-bit Threshold Value |
| CL2_SET_THR1_CLKMON | 16h | 7:0 | R/W | 0h | GUI calculates the correct CL register threshold trigger value for IN2 |
| CL2_SET_THR0_CLKMON | 17h | 7:0 | R/W | 0h | |
| CL1_SET_THR2_CLKMON | 18h | 7:3 | R/W | 0h | SPARE |
| | | 2:0 | R/W | 0h | IN1 19-bit Threshold Value |
| CL1_SET_THR1_CLKMON | 19h | 7:0 | R/W | 0h | GUI calculates the correct CL register threshold trigger value for IN1 |
| CL1_SET_THR0_CLKMON | 1Ah | 7:0 | R/W | 0h | |
| CL0_SET_THR2_CLKMON | 1Bh | 7:3 | R/W | 0h | SPARE |
| | | 2:0 | R/W | 0h | IN0 19-bit Threshold Value |
| CL0_SET_THR1_CLKMON | 1Ch | 7:0 | R/W | 0h | GUI calculates the correct CL register threshold trigger value for IN0 |
| CL0_SET_THR0_CLKMON | 1Dh | 7:0 | R/W | 0h | |
| CL3_CLR_THR2_CLKMON | 1Eh | 7:3 | R/W | 0h | SPARE |
| | | 2:0 | R/W | 0h | IN3 19-bit Threshold Value |
| CL3_CLR_THR1_CLKMON | 1Fh | 7:0 | R/W | 0h | GUI calculates the correct CL register threshold clear value for IN3 |
| CL3_CLR_THR0_CLKMON | 20h | 7:0 | R/W | 0h | |
| CL2_CLR_THR2_CLKMON | 21h | 7:3 | R/W | 0h | SPARE |
| | | 2:0 | R/W | 0h | IN2 19-bit Threshold Value |
| CL2_CLR_THR1_CLKMON | 22h | 7:0 | R/W | 0h | GUI calculates the correct CL register threshold clear value for IN2 |
| CL2_CLR_THR0_CLKMON | 23h | 7:0 | R/W | 0h | |
| CL1_CLR_THR2_CLKMON | 24h | 7:3 | R/W | 0h | SPARE |
| | | 2:0 | R/W | 0h | IN1 19-bit Threshold Value |
| CL1_CLR_THR1_CLKMON | 25h | 7:0 | R/W | 0h | GUI calculates the correct CL register threshold clear value for IN1 |
| CL1_CLR_THR0_CLKMON | 26h | 7:0 | R/W | 0h | |
| CL0_CLR_THR2_CLKMON | 27h | 7:3 | R/W | 0h | SPARE |
| | | 2:0 | R/W | 0h | IN0 19-bit Threshold Value |
| CL0_CLR_THR1_CLKMON | 28h | 7:0 | R/W | 0h | GUI calculates the correct CL register threshold clear value for IN0 |
| CL0_CLR_THR0_CLKMON | 29h | 7:0 | R/W | 0h | |
| FD_REG1_CLKMON | 2Ah | 7:4 | R/W | 0h | FD COARSE Enable Bit Value of position {3,2,1,0} correspond to Inputs {3,2,1,0} |
| | | 3:0 | R/W | 0h | FD FINE Enable Bit Value of position {3,2,1,0} correspond to Inputs {3,2,1,0} |
| FD_REG2_CLKMON | 2Bh | 7 | R/W | 0h | SPARE |
| | | 6:3 | R/W | 0h | XO divider configuration for FD monitors. GUI configures these dividers |
| | | 2:0 | R/W | 0h | Golden reference clock selection for frequency drift monitors. Value of 4 corresponds to the XO Reference Value of {3,2,1,0} correspond to Inputs {3,2,1,0} |
| FD_REG3_CLKMON | 2Ch | 7:4 | R/W | 0h | FD3 divider configuration for FD monitors. GUI configures these dividers |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---------------------------|-----------------|-----------|-------------|---------------|--|
| | | 3:0 | R/W | 0h | FD2 divider configuration for FD monitors. GUI configures these dividers |
| FD_REG4_CLKMON | 2Dh | 7:4 | R/W | 0h | FD1 divider configuration for FD monitors. GUI configures these dividers |
| | | 3:0 | R/W | 0h | FD0 divider configuration for FD monitors. GUI configures these dividers |
| FXOBYFIN3_LOG2_BAND | 2Eh | 7:4 | R/W | 0h | SPARE |
| | | 3:0 | R/W | 0h | IN3 FD monitor configuration bits calculated by GUI |
| FXOBYFIN2_LOG2_BAND | 2Fh | 7:6 | R/W | 0h | PATTERN :{'01'/'10'} => Efuse Locked, {'00'/'11'} => Efuse NOT Locked |
| FXOBYFIN2_LOG2_BAND | 2Fh | 5:4 | R/W | 0h | SPARE |
| | | 3:0 | R/W | 0h | IN2 FD monitor configuration bits calculated by GUI |
| FXOBYFIN1_FIN0_LOG2_BAND | 30h | 7:4 | R/W | 0h | IN1 FD monitor configuration bits calculated by GUI |
| | | 3:0 | R/W | 0h | IN0 FD monitor configuration bits calculated by GUI |
| PLLA_ACT_SPARE_SEL | 31h | 7:6 | R/W | 0h | PLLA ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 5:4 | R/W | 0h | PLLA SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 3:2 | R/W | 0h | PLLA SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 1:0 | R/W | 0h | PLLA SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| FD3_CLR_THRFINE_CLKMON | 32h | 7:0 | R/W | 0h | IN3 FD clear threshold is calculated as 2X of FD3_CLR_THRFINE_CLKMON |
| FD2_CLR_THRFINE_CLKMON | 33h | 7:0 | R/W | 0h | IN2 FD clear threshold is calculated as 2X of FD2_CLR_THRFINE_CLKMON |
| FD1_CLR_THRFINE_CLKMON | 34h | 7:0 | R/W | 0h | IN1 FD clear threshold is calculated as 2X of FD1_CLR_THRFINE_CLKMON |
| FD0_CLR_THRFINE_CLKMON | 35h | 7:0 | R/W | 0h | IN0 FD clear threshold is calculated as 2X of FD0_CLR_THRFINE_CLKMON |
| FD32_CLR_THRCOARSE_CLKMON | 36h | 7:4 | R/W | 0h | IN3 FD clear threshold is calculated as 100*(FD3_CLR_THRCOARSE_CLKMON +1) |
| | | 3:0 | R/W | 0h | IN2 FD clear threshold is calculated as 100*(FD2_CLR_THRCOARSE_CLKMON +1) |
| FD10_CLR_THRCOARSE_CLKMON | 37h | 7:4 | R/W | 0h | IN1 FD clear threshold is calculated as 100*(FD1_CLR_THRCOARSE_CLKMON +1) |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---------------------------|-----------------|-----------|-------------|---------------|---|
| | | 3:0 | R/W | 0h | IN0 FD clear threshold is calculated as $100 * (\text{FD0_CLR_THRCOARSE_CLKMON} + 1)$ |
| FD32_SET_THRCOARSE_CLKMON | 38h | 7:4 | R/W | 0h | IN3 FD set threshold is calculated as $100 * (\text{SET_FD3_SET_THRCOARSE_CLKMON} + 1)$ |
| | | 3:0 | R/W | 0h | IN2 FD set threshold is calculated as $100 * (\text{FD2_SET_THRCOARSE_CLKMON} + 1)$ |
| FD10_SET_THRCOARSE_CLKMON | 39h | 7:4 | R/W | 0h | IN1 FD set threshold is calculated as $100 * (\text{SET_TH_FD1_SET_THRCOARSE_CLKMON} + 1)$ |
| FD10_SET_THRCOARSE_CLKMON | 39h | 3:0 | R/W | 0h | IN0 FD set threshold is calculated as $100 * (\text{FD0_SET_THRCOARSE_CLKMON} + 1)$ |
| FD3_FGBYFM2_CLKMON | 3Ah | 7:0 | R/W | 0h | IN3 FD monitor configuration bits calculated by GUI |
| FD3_FGBYFM1_CLKMON | 3Bh | 7:0 | R/W | 0h | IN3 FD monitor configuration bits calculated by GUI |
| FD3_FGBYFM0_CLKMON | 3Ch | 7:0 | R/W | 0h | IN3 FD monitor configuration bits calculated by GUI |
| FD2_FGBYFM2_CLKMON | 3Dh | 7:0 | R/W | 0h | IN2 FD monitor configuration bits calculated by GUI |
| FD2_FGBYFM1_CLKMON | 3Eh | 7:0 | R/W | 0h | IN2 FD monitor configuration bits calculated by GUI |
| FD2_FGBYFM0_CLKMON | 3Fh | 7:0 | R/W | 0h | IN2 FD monitor configuration bits calculated by GUI |
| FD1_FGBYFM2_CLKMON | 40h | 7:0 | R/W | 0h | IN1 FD monitor configuration bits calculated by GUI |
| FD1_FGBYFM1_CLKMON | 41h | 7:0 | R/W | 0h | IN1 FD monitor configuration bits calculated by GUI |
| FD1_FGBYFM0_CLKMON | 42h | 7:0 | R/W | 0h | IN1 FD monitor configuration bits calculated by GUI |
| FD0_FGBYFM2_CLKMON | 43h | 7:0 | R/W | 0h | IN0 FD monitor configuration bits calculated by GUI |
| FD0_FGBYFM1_CLKMON | 44h | 7:0 | R/W | 0h | IN0 FD monitor configuration bits calculated by GUI |
| FD0_FGBYFM0_CLKMON | 45h | 7:0 | R/W | 0h | IN0 FD monitor configuration bits calculated by GUI |
| CLKX_HITLESS_SW_SOURCE | 46h | 7:6 | R/W | 0h | 0 : CL3 1 : CL3 + FD3 COARSE 2 : CL3 + FD3 FINE 3 : CL3 + FD3 COARSE + FD3 FINE |
| | | 5:4 | R/W | 0h | 0 : CL2 1 : CL2 + FD2 COARSE 2 : CL2 + FD2 FINE 3 : CL2 + FD2 COARSE + FD2 FINE |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|------------------------|-----------------|-----------|-------------|---------------|---|
| | | 3:2 | R/W | 0h | 0 : CL1 1 : CL1 + FD1 COARSE 2: CL1 + FD1 FINE 3 : CL1 + FD1 COARSE + FD1 FINE |
| | | 1:0 | R/W | 0h | 0 : CL0 1 : CL0 + FD0 COARSE 2: CL0 + FD0 FINE 3 : CL0 + FD0 COARSE + FD0 FINE |
| PLLB_ACT_SPARE_SEL | 49h | 7:6 | R/W | 0h | PLLB ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 5:4 | R/W | 0h | PLLB SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 3:2 | R/W | 0h | PLLB SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 1:0 | R/W | 0h | PLLB SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| PLL_C_ACT_SPARE_SEL | 4Ah | 7:6 | R/W | 0h | PLL_C ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 5:4 | R/W | 0h | PLL_C SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 3:2 | R/W | 0h | PLL_C SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 1:0 | R/W | 0h | PLL_C SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| PLLD_ACT_SPARE_SEL | 4Bh | 7:6 | R/W | 0h | PLLD ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 5:4 | R/W | 0h | PLLD SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 3:2 | R/W | 0h | PLLD SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| | | 1:0 | R/W | 0h | PLLD SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3 |
| FD3_SET_THRFINE_CLKMON | 4Ch | 7:0 | R/W | 0h | IN3 FD set threshold is calculated as 2X of FD3_SET_THRFINE_CLKMON |
| FD2_SET_THRFINE_CLKMON | 4Dh | 7:0 | R/W | 0h | IN2 FD set threshold is calculated as 2X of FD2_SET_THRFINE_CLKMON |
| FD1_SET_THRFINE_CLKMON | 4Eh | 7:0 | R/W | 0h | IN1 FD set threshold is calculated as 2X of FD1_SET_THRFINE_CLKMON |
| FD0_SET_THRFINE_CLKMON | 4Fh | 7:0 | R/W | 0h | IN0 FD set threshold is calculated as 2X FD0_SET_THRFINE_CLKMON_THR |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---------------|-----------------|-----------|-------------|---------------|--|
| PAGE_NUMBER | FFh | 7:0 | R/W | 0h | On all pages register FF is a READ / WRITE register used to change the page number |

Table 32. PAGE 2: Input System Related Registers.
Registers from 10h to 4Fh are equivalent NVMCopy Registers for this Page

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---------------------------|-----------------|-----------|-------------|---------------|--|
| PRG_Directives_INPAGE | 0Fh | 7:3 | R/W | 0h | PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active |
| | | 2 | R/W | 0h | Spare |
| | | 1 | R/W | 0h | Escape to the PRG_CMD state from ACTIVE state |
| CLKIN0_DIVN1_INT1_INPAGE | 10h | 7:0 | R/W | 0h | IN0 DIVN1 Divider Integer Value |
| CLKIN0_DIVN1_INT2_INPAGE | 11h | 7 | R/W | 0h | IN0 DIVN1 Integer Mode of Division |
| | | 6 | R/W | 0h | Enable the Frequency Ramp FD Monitoring |
| | | 5:0 | R/W | 0h | IN0 DIVN1 Divider Integer Value |
| CLKIN0_CFG1_INPAGE | 1Ah | 7:6 | R/W | 0h | IN0 DIVN1 Divider Integer Value |
| | | 5 | R/W | 0h | IN0 Direct Bypass for DIVN1 |
| | | 4 | R/W | 0h | IN0 Single Ended Enable, Default is Differential |
| CLKIN0_CFG2_INPAGE | 1Bh | 7:0 | R/W | 0h | IN0 FD Ramp monitor configuration bits calculated by GUI |
| CLKIN0_CFG3_INPAGE | 1Ch | 7:0 | R/W | 0h | |
| CLKIN0_CFG4_INPAGE | 1Dh | 7:0 | R/W | 0h | |
| CLKIN0_OUTSEL_INPAGE | 1Eh | 7:4 | R/W | 0h | XO divider configuration for FD Ramp monitors. GUI configures these dividers |
| | | 3:0 | R/W | 0h | IN0 divider configuration for FD Ramp monitors. GUI configures these dividers |
| CLKIN0_RAMPDFD_MEAS_COUNT | 1Fh | 7 | R/W | 0h | SPARE |
| | | 6:5 | R/W | 0h | SPARE |
| | | 4:0 | R/W | 0h | Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI. |
| CLKIN1_DIVN1_INT1_INPAGE | 20h | 7:0 | R/W | 0h | IN1 DIVN1 Divider Integer Value |
| CLKIN1_DIVN1_INT2_INPAGE | 21h | 7 | R/W | 0h | IN1 DIVN1 Integer Mode of Division |
| | | 6 | R/W | 0h | SPARE |
| | | 5:0 | R/W | 0h | IN1 DIVN1 Divider Integer Value |
| CLKIN1_CFG1_INPAGE | 2Ah | 7:6 | R/W | 0h | IN1 DIVN1 Divider Integer Value |
| | | 5 | R/W | 0h | IN1 Direct Bypass for DIVN1 |
| | | 4 | R/W | 0h | IN1 Single Ended Enable, Default is Differential |
| CLKIN1_CFG2_INPAGE | 2Bh | 7:0 | R/W | 0h | IN1 FD Ramp monitor configuration bits calculated by GUI |
| CLKIN1_CFG3_INPAGE | 2Ch | 7:0 | R/W | 0h | |
| CLKIN1_CFG4_INPAGE | 2Dh | 7:0 | R/W | 0h | |
| | | 3:0 | R/W | 0h | |
| CLKIN1_RAMPDFD_MEAS_COUNT | 2Fh | 7:6 | R/W | 0h | PATTERN :{'01'/'10'} =] Efuse Locked , {'00'/'11'} =] Efuse NOT Locked |
| | | 4:0 | R/W | 0h | Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI. |
| CLKIN2_DIVN1_INT1_INPAGE | 30h | 7:0 | R/W | 0h | IN2 DIVN1 Divider Integer Value |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---------------------------|-----------------|-----------|-------------|---------------|---|
| CLKIN2_DIVN1_INT2_INPAGE | 31h | 7 | R/W | 0h | IN2 DIVN1 Integer Mode of Division |
| | | 5:0 | R/W | 0h | IN2 DIVN1 Divider Integer Value |
| CLKIN2_CFG1_INPAGE | 3Ah | 7:6 | R/W | 0h | IN2 DIVN1 Divider Integer Value |
| | | 5 | R/W | 0h | IN2 Direct Bypass for DIVN1 |
| | | 4 | R/W | 0h | IN2 Single Ended Enable, Default is Differential |
| CLKIN2_CFG2_INPAGE | 3Bh | 7:0 | R/W | 0h | IN2 FD Ramp monitor configuration bits calculated by GUI |
| CLKIN2_CFG3_INPAGE | 3Ch | 7:0 | R/W | 0h | |
| CLKIN2_CFG4_INPAGE | 3Dh | 7:0 | R/W | 0h | |
| CLKIN2_OUTSEL_INPAGE | 3Eh | 7:4 | R/W | 0h | Internal FD Ramp related settings |
| | | 3:0 | R/W | 0h | IN2 divider configuration for FD Ramp monitors. GUI configures these dividers |
| | | 4:0 | R/W | 0h | Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI. |
| CLKIN3_DIVN1_INT1_INPAGE | 40h | 7:0 | R/W | 0h | IN3 DIVN1 Divider Integer Value |
| CLKIN3_DIVN1_INT2_INPAGE | 41h | 7 | R/W | 0h | IN3 DIVN1 Integer Mode of Division |
| | | 5:0 | R/W | 0h | IN3 DIVN1 Divider Integer Value |
| CLKIN3_CFG1_INPAGE | 4Ah | 7:6 | R/W | 0h | IN3 DIVN1 Divider Integer Value |
| | | 5 | R/W | 0h | IN3 Direct Bypass for DIVN1 |
| | | 4 | R/W | 0h | IN3 CLKIN3 Single Ended Enable, Default is Differential |
| CLKIN3_CFG2_INPAGE | 4Bh | 7:0 | R/W | 0h | IN3 FD Ramp monitor configuration bits calculated by GUI |
| CLKIN3_CFG3_INPAGE | 4Ch | 7:0 | R/W | 0h | |
| CLKIN3_CFG4_INPAGE | 4Dh | 7:0 | R/W | 0h | |
| CLKIN3_OUTSEL_INPAGE | 4Eh | 7:6 | R/W | 0h | SPARE |
| | | 5:4 | R/W | 0h | SPARE |
| | | 3:0 | R/W | 0h | IN3 divider configuration for FD Ramp monitors. GUI configures these dividers |
| CLKIN3_RAMPDFD_MEAS_COUNT | 4Fh | 4:0 | R/W | 0h | Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI. |
| PAGE_NUMBER | FFh | 7:0 | R/W | 0h | On all pages register FF is a READ / WRITE register used to change the page number |

Table 33. PAGE 3: Output System and Output Dividers Related Registers.
Registers from 18h to 67h are equivalent NVMCopy Registers for this Page

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description | | | | | | | | | | | | | | | | |
|-------------------------------|-----------------|-----------|---|---------------|---|---|---|------|------|---|---|------|------|---|---|------|------|---|---|------|------|
| PRG_Directives_OUTPAGE | 0Fh | 7:3 | R/W | 0h | PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active | | | | | | | | | | | | | | | | |
| | | 2 | R/W | 0h | Spare | | | | | | | | | | | | | | | | |
| | | 1 | R/W | 0h | Escape to the PRG_CMD state from ACTIVE state | | | | | | | | | | | | | | | | |
| | | 2 | R/W | 0h | | | | | | | | | | | | | | | | | |
| | | 1 | R/W | 0h | | | | | | | | | | | | | | | | | |
| | | 0 | R/W | 0h | | | | | | | | | | | | | | | | | |
| DIVO1_DIV2_OUTPAGE | 18h | 7:4 | R/W | 0h | Spare | | | | | | | | | | | | | | | | |
| | | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] | | | | | | | | | | | | | | | | |
| DIVO1_DIV1_OUTPAGE | 19h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] | | | | | | | | | | | | | | | | |
| DIVO1_DIV0_OUTPAGE | 1Ah | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] | | | | | | | | | | | | | | | | |
| DIVO1_PROG1_OUTPAGE | 1Bh | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N <table border="0"> <tr> <td>0</td> <td>0</td> <td>CLKP</td> <td>CLKN</td> </tr> <tr> <td>0</td> <td>1</td> <td>CLKP</td> <td>CLKP</td> </tr> <tr> <td>1</td> <td>0</td> <td>CLKN</td> <td>CLKN</td> </tr> <tr> <td>1</td> <td>1</td> <td>CLKN</td> <td>CLKP</td> </tr> </table> | 0 | 0 | CLKP | CLKN | 0 | 1 | CLKP | CLKP | 1 | 0 | CLKN | CLKN | 1 | 1 | CLKN | CLKP |
| | | 0 | 0 | CLKP | CLKN | | | | | | | | | | | | | | | | |
| 0 | 1 | CLKP | CLKP | | | | | | | | | | | | | | | | | | |
| 1 | 0 | CLKN | CLKN | | | | | | | | | | | | | | | | | | |
| 1 | 1 | CLKN | CLKP | | | | | | | | | | | | | | | | | | |
| 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 | | | | | | | | | | | | | | | | | | |
| DIVO1_PROG0_OUTPAGE | 1Ch | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code | | | | | | | | | | | | | | | | |
| DIVO1_MISC2_OUTPAGE | 1Dh | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. | | | | | | | | | | | | | | | | |
| | | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) | | | | | | | | | | | | | | | | |
| DIVO2_DIV2_OUTPAGE | 20h | 7:4 | R/W | 0h | Spare | | | | | | | | | | | | | | | | |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------------|-----------------|-----------|-------------|---------------|---|
| | | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] |
| DIVO2_DIV1_OUTPAGE | 21h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] |
| DIVO2_DIV0_OUTPAGE | 22h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] |
| DIVO2_PROG1_OUTPAGE | 23h | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP |
| | | 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO2_PROG0_OUTPAGE | 24h | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO2_MISC2_OUTPAGE | 25h | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |
| | | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO3_DIV2_OUTPAGE | 28h | 7:4 | R/W | 0h | Spare |
| | | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] |
| DIVO3_DIV1_OUTPAGE | 29h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] |
| DIVO3_DIV0_OUTPAGE | 2Ah | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] |
| DIVO3_PROG1_OUTPAGE | 2Bh | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP |
| | | 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---------------------|-----------------|-----------|---|---------------|---|
| DIVO3_PROG0_OUTPAGE | 2Ch | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO3_MISC2_OUTPAGE | 2Dh | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |
| DIVO3_MISC2_OUTPAGE | 2Dh | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO3_MISC0_OUTPAGE | 2Fh | 7:6 | R/W | 0h | PATTERN :{'01'/'10'} =] Efuse Written, {'00'/'11'} =] Efuse NOT Written |
| | | 4 | R/W | 0h | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code |
| | | 3 | R/W | 0h | {0, 0, 0, 0} External Termination, Differential Output |
| | | 2 | R/W | 0h | {0, 0, 1, 0} Internal Pull Up, Differential Output |
| | | 1 | R/W | 0h | {0, 0, 0, 1} Internal Pull Dn, Differential Output |
| | | 0 | R/W | 0h | {0, 1, 0, 0} CMOS On OutP, Nothing on OutN |
| | | 2 | R/W | 0h | {1, 0, 0, 0} Nothing on OutP, CMOS on OutN |
| | | 1 | R/W | 0h | {1, 1, 0, 0} CMOS on OutP, CMOS on OutN |
| DIVO4_DIV2_OUTPAGE | 30h | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] |
| DIVO4_DIV1_OUTPAGE | 31h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] |
| DIVO4_DIV0_OUTPAGE | 32h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] |
| DIVO4_PROG1_OUTPAGE | 33h | 7:6 | R/W | 0h | CMOS Driver Phase Selection |
| | | | | | phase_sel<1> phase_sel<0> ODR_P ODR_N |
| | | | | | 0 0 CLKP CLKN |
| | | | | | 0 1 CLKP CLKP |
| | | | | | 1 0 CLKN CLKN |
| | 1 1 CLKN CLKP | | | | |
| 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 | | |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---|-----------------|-----------|---|---------------|--|
| DIVO4_PROG0_OUTPAGE | 34h | 7:0 | R/W | 0h | Programmable Output Delay |
| | | | | | PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number |
| | | | | | PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO4_MISC2_OUTPAGE | 35h | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |
| | | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: |
| | | | | | 0b010: DC Coupled CML |
| | | | | | 0b011: DC Coupled HCSL |
| | | | | | 0b000: LVDS (Can be AC Coupled or DC Coupled) |
| | | | | | 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) |
| | | | | | 0b001: DC Coupled LVPECL (with Common mode current) |
| 0b101: DC Coupled LVPECL2 (without Common mode current) | | | | | |
| DIVO4_MISC0_OUTPAGE | 37h | 4 | R/W | 0h | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code |
| | | 3 | R/W | 0h | {0, 0, 0, 0} External Termination, Differential Output |
| | | | | | {0, 0, 1, 0} Internal Pull Up, Differential Output |
| | | | | | {0, 0, 0, 1} Internal Pull Dn, Differential Output |
| | | 2 | R/W | 0h | {0, 1, 0, 0} CMOS On OutP, Nothing on OutN |
| | | 1 | R/W | 0h | {1, 0, 0, 0} Nothing on OutP, CMOS on OutN |
| 0 | R/W | 0h | {1, 1, 0, 0} CMOS on OutP, CMOS on OutN | | |
| DIVO5_DIV2_OUTPAGE | 38h | 7:4 | R/W | 0h | Spare |
| | | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] |
| DIVO5_DIV1_OUTPAGE | 39h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] |
| DIVO5_DIV0_OUTPAGE | 3Ah | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] |
| DIVO5_PROG1_OUTPAGE | 3Bh | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------------|-----------------|-----------|-------------|---------------|---|
| | | 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO5_PROG0_OUTPAGE | 3Ch | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO5_MISC3_OUTPAGE | 3Dh | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |
| DIVO5_MISC3_OUTPAGE | 3Dh | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO6_DIV2_OUTPAGE | 40h | 7:4 | R/W | 0h | spare |
| | | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] |
| DIVO6_DIV1_OUTPAGE | 41h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] |
| DIVO6_DIV0_OUTPAGE | 42h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] |
| DIVO6_PROG1_OUTPAGE | 43h | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP |
| | | 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO6_PROG0_OUTPAGE | 44h | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO6_MISC2_OUTPAGE | 45h | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|---------------------|-----------------|-----------|-------------|---------------|---|
| | | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO6_MISC0_OUTPAGE | 47h | 7:6 | R/W | 0h | DIVO Divider for 0B output: Bits [33:32] |
| | | 4 | R/W | 0h | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code |
| | | 3 | R/W | 0h | {0, 0, 0, 0} External Termination, Differential Output |
| | | 2 | R/W | 0h | {0, 0, 1, 0} Internal Pull Up, Differential Output |
| | | 1 | R/W | 0h | {0, 0, 0, 1} Internal Pull Dn, Differential Output |
| | | 0 | R/W | 0h | {0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN |
| DIVO7_DIV2_OUTPAGE | 48h | 7:4 | R/W | 0h | spare |
| | | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] |
| DIVO7_DIV1_OUTPAGE | 49h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] |
| DIVO7_DIV0_OUTPAGE | 4Ah | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] |
| DIVO7_PROG1_OUTPAGE | 4Bh | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP |
| | | 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO7_PROG0_OUTPAGE | 4Ch | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO7_MISC2_OUTPAGE | 4Dh | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|-----------------------------|-----------------|-----------|-------------|---------------|---|
| | | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO7_MISC0_OUTPAGE | 4Fh | 7:6 | R/W | 0h | DIVO Divider for 0B output: Bits [31:30] |
| | | 4 | R/W | 0h | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code |
| | | 3 | R/W | 0h | {0, 0, 0, 0} External Termination, Differential Output |
| | | 2 | R/W | 0h | {0, 0, 1, 0} Internal Pull Up, Differential Output |
| | | 1 | R/W | 0h | {0, 0, 0, 1} Internal Pull Dn, Differential Output |
| | | 0 | R/W | 0h | {0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN |
| DIVO0T_DIV2_OUTPAGE | 50h | 7:4 | R/W | 0h | spare |
| | | 3:0 | R/W | 0h | DIVO Divider for this output: Bits [19:16] |
| DIVO0T_DIV1_OUTPAGE | 51h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [15:8] |
| DIVO0T_DIV0_OUTPAGE | 52h | 7:0 | R/W | 0h | DIVO Divider for this output: Bits [7:0] |
| DIVO0T_PROG1_OUTPAGE | 53h | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP |
| | | 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO0T_PROG0_OUTPAGE | 54h | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO0T_MISC2_OUTPAGE | 55h | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------|-----------------|-----------|-------------|---------------|---|
| | | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO0T_MISC0_OUTPAGE | 57h | 7:6 | R/W | 0h | DIVO Divider for 0B output: Bits [29:28] |
| | | 4 | R/W | 0h | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code |
| | | 3 | R/W | 0h | {0, 0, 0, 0} External Termination, Differential Output |
| | | 2 | R/W | 0h | {0, 0, 1, 0} Internal Pull Up, Differential Output |
| | | 1 | R/W | 0h | {0, 0, 0, 1} Internal Pull Dn, Differential Output |
| | | 0 | R/W | 0h | {0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN |
| DIVO0B_DIV2_OUTPAGE | 60h | 7:0 | R/W | 0h | DIVO Divider for 0B output: Bits [23:16] |
| DIVO0B_DIV1_OUTPAGE | 61h | 7:0 | R/W | 0h | DIVO Divider for 0B output: Bits [15:8] |
| DIVO0B_DIV0_OUTPAGE | 62h | 7:0 | R/W | 0h | DIVO Divider for 0B output: Bits [7:0] |
| DIVO0B_PROG1_OUTPAGE | 63h | 7:6 | R/W | 0h | CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP |
| | | 2:0 | R/W | 0h | Single Ended Driver Programming of strength: Use 0b111= 0d7 |
| DIVO0B_PROG0_OUTPAGE | 64h | 7:0 | R/W | 0h | Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code |
| DIVO0B_MISC2_OUTPAGE | 65h | 4:3 | R/W | 0h | VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output. |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|-----------------------------|-----------------|-----------|-------------|---------------|--|
| | | 2:0 | R/W | 0h | DRV_TYPE : Output Driver Standard: 0b010: DC Coupled CML 0b011: DC Coupled HCSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current) |
| DIVO0B_MISC0_OUTPAGE | 67h | 7:6 | R/W | 0h | DIVO Divider for 0B output: Bits [25:24] |
| | | 4 | R/W | 0h | PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code |
| | | 3 | R/W | 0h | {0, 0, 0, 0} External Termination, Differential Output |
| | | 2 | R/W | 0h | {0, 0, 1, 0} Internal Pull Up, Differential Output |
| | | 1 | R/W | 0h | {0, 0, 0, 1} Internal Pull Dn, Differential Output |
| | | 0 | R/W | 0h | {0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN |
| PAGE_NUMBER | FFh | 7:0 | R/W | 0h | On all pages register FF is a READ / WRITE register used to change the page number |

Table 34. Page A: PLL A Related Registers (Similar for Pages B, C, D).
The Registers from 10h to 2Fh are equivalent NVMCopy Registers for this Page. Fractional Divider Registers are not valid for DIVN2 for PLLA but valid for PLLs B, C, D.

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|--------------------------------|-----------------|----------------------|-------------|---------------|--|
| STATUS_PLLA | 02h | 7 | R | 0h | SPARE |
| | | 6 | R | 0h | SPARE |
| | | 5 | R | 0h | SPARE |
| | | 4 | R | 0h | SPARE |
| | | 3 | R | 0h | SPARE |
| | | 2 | R | 0h | SPARE |
| | | 1 | R | 0h | SPARE |
| | | 0 | R | 0h | Dynamic status of Loss of lock for PLLA |
| NOTIFY_PLLA | 03h | 7 | R/W | 1h | SPARE |
| | | 6 | R/W | 1h | SPARE |
| | | 5 | R/W | 1h | SPARE |
| | | 4 | R/W | 1h | SPARE |
| | | 3 | R/W | 1h | SPARE |
| | | 2 | R/W | 1h | SPARE |
| | | 1 | R/W | 1h | SPARE |
| | | 0 | R/W | 1h | Sticky/Notify status Loss of lock for PLLA |
| MASKb_PLLA | 04h | 7 | R/W | 1h | SPARE |
| | | 6 | R/W | 1h | SPARE |
| | | 5 | R/W | 1h | SPARE |
| | | 4 | R/W | 1h | SPARE |
| | | 3 | R/W | 1h | SPARE |
| | | 2 | R/W | 1h | SPARE |
| | | 1 | R/W | 1h | SPARE |
| | | 0 | R/W | 1h | Mask bit for NOTIFY_PLLA (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[0] |
| Directives_GENERIC_PLLA | 05h | 7 | R/W | 0h | Reserved |
| | | 6 | R/W | 0h | Spare |
| | | 5 | R/W | 1h | DLPF co-efficient selection provided by the GUI |
| | | 4 | R/W | 0h | Force external clock in switch |
| | | 3 | R/W | 0h | Force the PLL in holdover mode |
| | | 2 | R/W | 0h | Large change for resetting entire DIVO system: Edge triggered |
| | | 0 | R/W | 0h | Spare |
| | | STATUS_1_PLLA | 06h | 7 | R |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|----------------------------|-----------------|-----------|-------------|---------------|--|
| | | 6 | R | 0h | SPARE |
| | | 5 | R | 0h | SPARE |
| | | 4 | R | 0h | SPARE |
| | | 3 | R | 0h | Dynamic status for cycle slip detection |
| | | 2 | R | 0h | Dynamic status for indicating holdover window is valid |
| | | 1 | R | 0h | Dynamic status for fast lock mode |
| | | 0 | R | 0h | Dynamic status for holdover |
| NOTIFY_1_PLLA | 07h | 7 | R/W | 1h | SPARE |
| | | 6 | R/W | 1h | SPARE |
| | | 5 | R/W | 1h | SPARE |
| | | 4 | R/W | 1h | SPARE |
| | | 3 | R/W | 1h | Sticky/Notify status for cycle slip detection |
| | | 2 | R/W | 1h | Sticky bit indicating holdover window is valid |
| | | 1 | R/W | 1h | Sticky/Notify status for fast lock mode |
| | | 0 | R/W | 1h | Sticky/Notify status for holdover |
| MASKb_1_PLLA | 08h | 7 | R/W | 1h | SPARE |
| | | 6 | R/W | 1h | SPARE |
| | | 5 | R/W | 1h | SPARE |
| | | 4 | R/W | 1h | SPARE |
| | | 3 | R/W | 1h | Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[3] |
| | | 2 | R/W | 1h | Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[2] |
| | | 1 | R/W | 1h | Mask bit for NOTIFY_1_PLLA FASTLOCK (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[1] |
| | | 0 | R/W | 1h | Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[0] |
| PRG_Directives_PLLA | 0Fh | 7:3 | R/W | 0h | PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active |
| | | 2 | R/W | 0h | Spare |
| | | 1 | R/W | 0h | Escape to the PRG_CMD state from ACTIVE state |
| PPATH_PLLA | 10h | 7:5 | R/W | 0h | DLPF Settings from the GUI |
| | | 4:0 | R/W | 0h | |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description | |
|--------------------------|-----------------|-----------|-------------|---------------|--|--|
| IPATH1_PLLA | 11h | 7:3 | R/W | 0h | | |
| | | 2:0 | R/W | 0h | | |
| IPATH2_PLLA | 12h | 7:5 | R/W | 0h | | |
| | | 4:0 | R/W | 0h | | |
| FASTLOCK_PPATH_PLLA | 13h | 7:5 | R/W | 0h | | |
| | | 4:0 | R/W | 0h | | |
| FASTLOCK_IPATH1_PLLA | 14h | 7:3 | R/W | 0h | | |
| | | 2:0 | R/W | 0h | | |
| FASTLOCK_IPATH2_PLLA | 15h | 7:5 | R/W | 0h | | |
| | | 4:0 | R/W | 0h | | |
| CYCLESIP_MISC_CTRL_PLLA | 16h | 7 | R/W | 0h | | CP gain configuration settings provided by GUI |
| | | 6 | R/W | 0h | | |
| | | 5 | R/W | 0h | PLL loop filter configuration provided by GUI | |
| | | 4 | R/W | 0h | PLL loop filter configuration provided by GUI | |
| | | 3 | R/W | 0h | ZDB related setting provided by GUI | |
| | | 2:1 | | 0h | Cycle Slip detector threshold settings computed by GUI | |
| | | 0 | R/W | 0h | Enable for cycle slip detector for DLPF | |
| DIVNINT_PLLA | 17h | 7 | R/W | 0h | Master Disable for PLLA in sync mode and all associated functions | |
| | | 6:0 | R/W | 0h | DIVN Integer part computed by GUI | |
| MISCXO_PLLA | 18h | 7 | R/W | 0h | Enable ADC Dither | |
| | | 6 | R/W | 0h | Offset Enable in the PLL Charge Pump | |
| | | 5 | R/W | 0h | Select ACTIVE clock manually from manual input select pins when in manual active select mode | |
| | | 4 | R/W | 0h | DIVN Integer part computed by GUI | |
| | | 3:1 | R/W | 0h | Charge Pump Binning for VCO KV | |
| | | 0 | R/W | 0h | 0: Fractional Mode; 1: Integer Mode | |
| DLPF_PHI_CORRECTION_PLLA | 19h | 7:6 | R/W | 0h | Phase averager configuration settings | |
| | | 5:2 | R/W | 0h | Phase measurement filter BW setting (3 is spare) | |
| | | 1 | R/W | 0h | Phase Correction Setting: Use default from GUI | |
| DIVNFRAC2_PLLA | 1Ah | 7:0 | R/W | 0h | DIVN Fractional part computed by GUI | |
| DIVNFRAC3_PLLA | 1Bh | 7:0 | R/W | 0h | DIVN Fractional part computed by GUI | |
| DIVNFRAC4_PLLA | 1Ch | 7:0 | R/W | 0h | DIVN Fractional part computed by GUI | |
| DIVN2_INT1_PLLA | 1Dh | 7:0 | R/W | 0h | DIVN2 Integer part computed by GUI | |
| DIVN2_INT2_PLLA | 1Eh | 7:0 | R/W | 0h | DIVN2 Integer part computed by GUI | |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|-----------------|-----------------|-----------|-------------|---------------|--|
| DIVN2_INT3_PLLA | 1Fh | 7 | R/W | 0h | Enable Revertive switching for input clock switching |
| | | 6 | R/W | 0h | Force manual selection of ACTIVE clock |
| | | 5 | R/W | 0h | Force Integer mode for the DIVN2 DSM |
| | | 4:0 | R/W | 0h | DIVN2 Integer part computed by GUI |
| DIVN2_FRN1_PLLA | 20h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| DIVN2_FRN2_PLLA | 21h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| DIVN2_FRN3_PLLA | 22h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| DIVN2_FRN4_PLLA | 23h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD1_PLLA | 24h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD2_PLLA | 25h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD3_PLLA | 26h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| DIVN2_FRD4_PLLA | 27h | 7:0 | R/W | 0h | DIVN2 Fractional part computed by GUI |
| OUTPUT_EN_PLLA | 28h | 7:6 | R/W | 0h | PSEQ_SYNC_DELAY 0: 500us 1: 8.17ms 2: 131.07ms 3: 2.09sec |
| | | 5:0 | R/W | 0h | One Hot Output Enable for Outputs 5:0 |
| LL_REG1_PLLA | 29h | 7:6 | R/W | 0h | Programmable Phase Propagation Mode slope for the PLL 00: Use the Bandwidth 01: 10 usec/sec 10: 40 usec/sec 11: 160 usec/sec |
| | | 5 | R/W | 0h | Wait for Input Clock in power up in the PLL wake-up sequence |
| | | 4 | R/W | 0h | Use Fast Lock BW for exit from holdover to latch on fast to the new clock |
| | | 3:1 | R/W | 0h | Loss of Lock Delay = $(2^{(26-(2*LLDELAYTIMER))}/4M)$. Wait for this delay time before announcing LL de-assertion |
| | | 0 | R/W | 0h | LL Clear Threshold {LL_CLR_VALUE_PLLA[1], 0x2A[0]} 2'b00 : 0.2 PPM 2'b01 : 0.4 PPM 2'b10 : 2 PPM 2'b11 : 200 PPM |
| LL_REG2_PLLA | 2Ah | 7:5 | R/W | 0h | LL_SET_VALUE_PLLA[7:4] 0 : 0.2 PPM 1 : 0.4 PPM |
| | | 4 | R/W | 0h | 2 : 2 PPM 3 : 4 PPM 4 : 20 PPM 5 : 40 PPM |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|------------------------------|-----------------|-----------|-------------|---------------|---|
| | | | | | 6 : 200 PPM 7 : 400 PPM 8 : 2000 PPM 9 : 4000 PPM 10: 0.2 PPM 11: 0.2 PPM 12: 0.2 PPM 13: 0.2 PPM 14: 0.2 PPM 15: 0.2 PPM |
| | | 3:1 | R/W | 0h | PLL_PSEQ_DELAY 0: 500us 1: 2ms 2: 8.19ms 3: 32.76ms 4: 131.07ms 5: 524.28ms 6: 2.09sec 7: 8.38sec |
| | | 0 | R/W | 0h | LL Clear Threshold {0x29[0], LL_CLR_VALUE_PLLA[0]} 2'b00 : 0.2 PPM 2'b01 : 0.4 PPM 2'b10 : 2 PPM 2'b11 : 200 PPM |
| HOLDOVER2_PLLA | 2Bh | 7:5 | R/W | 0h | Holdover Tdelay settings for the PLL computed by the GUI |
| | | 4 | R/W | 0h | Enable Zero Delay Buffer mode with feedback clock routed from the PCB on in3 |
| | | 3:1 | R/W | 0h | Holdover Average settings for the PLL computed by the GUI |
| | | 0 | R/W | 0h | Enable output clock sync on an independent input coming from IN3 |
| | 2Ch | 7 | R/W | 0h | Cycle Slip Detector related default |
| | | 6 | R/W | 0h | 1: Enable Phase Propagation during input clock switch 0 : Enable Phase Build Out Mode where the phase difference between input clocks is absorbed by the PLL |
| | | 5 | R/W | 1h | Dither configuration for DIVN2 DSM |
| | | 4 | R/W | 0h | Enable revert to spare input clock during clock switching |
| | | 3:1 | R/W | 0h | DLPF related constant from the GUI |
| | | 0 | R/W | 0h | Internal voltage programming: Use default from GUI Profile |
| DECIMATION_RATIO_PLLA | 2Dh | 7 | R/W | 0h | Enable fast lock mode based on loss of lock status |
| | | 6:3 | R/W | 0h | Internal Rate Change factors in DLPF computed by GUI |

| Register Name | Register Number | Bit Range | Access Type | Default Value | Description |
|--------------------------|-----------------|-----------|-------------|---------------|---|
| | | 2:0 | R/W | 0h | Internal Rate Change factors in DLDPF computed by GUI |
| ONEBYR2_PLLA | 2Eh | 7:3 | R/W | 0h | Internal Rate Change factors in DLDPF computed by GUI |
| | | 2:0 | R/W | 0h | Internal Rate Change factors in DLDPF computed by GUI |
| LOCKPATTERN_PLLA | 2Fh | 7:6 | R/W | 0h | PATTERN :{'01'/'10'} =] Efuse Locked, {'00'/'11'} =] Efuse NOT Locked |
| | | 5:4 | R/W | 0h | Program the frequency ramp slope from the following- 00: 0.2 ppm/s, 01: 2 ppm/s, 10: 20 ppm/s, 11: 200 ppm/s |
| | | 3 | R/W | 0h | Enable the frequency ramp feature |
| | | 2:0 | R/W | 0h | Internal Frequency Divider : Computed by the GUI |
| DCO_FRAC1_PLLA | 31h | 7:0 | R/W | 0h | DCO fractional control code |
| DCO_FRAC2_PLLA | 32h | 7:0 | R/W | 0h | DCO fractional control code |
| DCO_FRAC3_PLLA | 33h | 7:0 | R/W | 0h | DCO fractional control code |
| DCO_FRAC4_PLLA | 34h | 7:0 | R/W | 0h | DCO fractional control code |
| DCO_FUNCTION_PLLA | 35h | 7:4 | R/W | 0h | Reserved |
| | | 3 | R/W | 0h | DCO Increment in Frequency from registers |
| | | 2 | R/W | 0h | DCO Decrement in Frequency from registers |
| | | 1 | R/W | 0h | Enable DCO free run mode |
| | | 0 | R/W | 0h | DCO Mask for this PLL |
| DCO_BUMP2_PLLA | 36h | 7:0 | R/W | 0h | DCO integer control code |
| DCO_BUMP3_PLLA | 37h | 7:6 | R/W | 0h | DCO integer control code |
| | | 5 | R/W | 0h | Enable DCO sync mode |
| | | 4 | R/W | 0h | SPARE |
| | | 3:0 | R/W | 0h | SPARE |
| PAGE_NUMBER | FFh | 7:0 | R/W | 0h | On all pages register FF is a READ / WRITE register used to change the page number |

Package Dimensions and Patterns

Package Size – Dimensions (Unit: mm)

(TOP VIEW)

(BOTTOM VIEW)

(SIDE VIEW)

| | SYMBOL | MIN | NOM | MAX |
|------------------------|--------|-----------|-------|-------|
| TOTAL THICKNESS | A | 0.800 | 0.850 | 0.900 |
| STAND OFF | A1 | 0.000 | 0.035 | 0.050 |
| BODY SIZE | X | D | | |
| | Y | E | | |
| EXPOSED PAD | X | J | 5.000 | 5.100 |
| | Y | K | 5.000 | 5.100 |
| LEAD WIDTH | b | 0.200 | 0.250 | 0.300 |
| LEAD LENGTH | L | 0.350 | 0.400 | 0.450 |
| LEAD PITCH | e | 0.500 BSC | | |
| PACKAGE TOLERANCE | aaa | 0.100 | | |
| MOLD FLATNESS | bbb | 0.100 | | |
| COPLANARITY | ccc | 0.080 | | |
| NOTE | | | | |
| 1. ALL DIMENSION IN MM | | | | |

| PKG INFO | | DRAWING NO. | |
|----------------------|-----------|-------------------------|-------|
| 64L PQFN | | POD-054-PQFN-064-C09090 | |
| 9.000x9.000x0.850 mm | | REV | SHEET |
| DATE | 2020/7/22 | A01 | 01 |

Recommended Land Pattern (Unit: mm)

Note : All units in mm.

| | PKG INFO | | SPL DRAWING NO. | |
|--------------------|------------|---------------------------|-----------------|------------------------|
| | | 64L QFN 9.000x9.000 mm | | SPL-054-QFN-064-C09090 |
| DATE 2020/11/09 | REV A00 | | | SHEET 01 |

Table 35. Revision History

| Revisions | Release Date | Change Summary |
|-----------|--------------|--|
| 0.1 | Nov 30, 2018 | Preliminary release |
| 0.2 | Feb 15, 2019 | Updated Overall Architecture diagram Figure 10 Updated SPI Operations Figures |
| 0.3 | Jul 9, 2019 | Added Register Map sections |
| 0.31 | Aug 31, 2019 | Fixed typos Added "Y" 1ku tape and reel order code |
| 0.9 | Jun 23, 2020 | I ² C address setting description added Added Low Wander Mode and Sysref sections Updated figures and numbering Added additional Notes Updated Features and Description Added "X" tape and reel order code Added pin pull-up and pull-down resistor values Updated supply current for VDDIN Updated PSNR, VDD, VDDIN values Added frequency stability and aging specifications to Output Clock Characteristics table |
| 0.91 | Aug 2, 2020 | Formatting changes Updated Ordering Information |
| 0.92 | Aug 29, 2020 | Changed VCROSS specification Changed rev table date format |
| 0.93 | Sep 28, 2020 | Formatting updates Updated I2C Protocol section |
| 0.94 | 2 Feb 2021 | Formatting and trademarks update |
| 0.95 | 6 Oct 2021 | Added Table 18: Input Channel Selection Truth Table Rearranged sections |
| 2.0 | 21 Mar 2022 | Final Release |
| 2.2 | 29 Sep 2022 | Updated notes for DC Electrical Characteristics Added Voltage Ramp data and diagrams Updated I ² C addressing instructions Fixed typos |
| 2.3 | 14 Nov 2022 | Updated version ID Added specs for LVDS common mode voltage at 1.8 V Fixed register maps Minor corrections to diagrams & wording |
| 2.31 | 21 Nov 2022 | Fixed typos in "Monitoring the Status for Master and Slave Pages" |
| 2.33 | 22 May 2023 | Updated trademarks and page 1 references links |
| 2.35 | 26 Jul 2023 | Updated LVDS (Vocm, Table 13) and Output Voltage (Vol, Table 9) specifications |
| 2.37 | 30 Aug 2023 | Added "4-16A" integration bandwidth jitter |
| 2.38 | 25 Mar 2024 | Fixed typo to correct MSL3 |

SiTime Corporation, 5451 Patrick Henry Drive, Santa Clara, CA 95054, USA | Phone: +1-408-328-4400 | Fax: +1-408-328-4439

© SiTime Corporation 2018-2024. The information contained herein is subject to change at any time without notice. SiTime assumes no responsibility or liability for any loss, damage or defect of a Product which is caused in whole or in part by (i) use of any circuitry other than circuitry embodied in a SiTime product, (ii) misuse or abuse including static discharge, neglect or accident, (iii) unauthorized modification or repairs which have been soldered or altered during assembly and are not capable of being tested by SiTime under its normal test conditions, or (iv) improper installation, storage, handling, warehousing or transportation, or (v) being subjected to unusual physical, thermal, or electrical stress.

Disclaimer: SiTime makes no warranty of any kind, express or implied, with regard to this material, and specifically disclaims any and all express or implied warranties, either in fact or by operation of law, statutory or otherwise, including the implied warranties of merchantability and fitness for use or a particular purpose, and any implied warranty arising from course of dealing or usage of trade, as well as any common-law duties relating to accuracy or lack of negligence, with respect to this material, any SiTime product and any product documentation. Products sold by SiTime are not suitable or intended to be used in a life support application or component, to operate nuclear facilities, or in other mission critical applications where human life may be involved or at stake. All sales are made conditioned upon compliance with the critical uses policy set forth below.

CRITICAL USE EXCLUSION POLICY

BUYER AGREES NOT TO USE SITIME'S PRODUCTS FOR ANY APPLICATION OR IN ANY COMPONENTS USED IN LIFE SUPPORT DEVICES OR TO OPERATE NUCLEAR FACILITIES OR FOR USE IN OTHER MISSION-CRITICAL APPLICATIONS OR COMPONENTS WHERE HUMAN LIFE OR PROPERTY MAY BE AT STAKE.

SiTime owns all rights, title and interest to the intellectual property related to SiTime's products, including any software, firmware, copyright, patent, or trademark. The sale of SiTime products does not convey or imply any license under patent or other rights. SiTime retains the copyright and trademark rights in all documents, catalogs and plans supplied pursuant to or ancillary to the sale of products or services by SiTime. Unless otherwise agreed to in writing by SiTime, any reproduction, modification, translation, compilation, or representation of this material shall be strictly prohibited.