SiT95141 Cascade ClkSoC™

Low Jitter, 10-output MEMS Clock Generator

Description

The Cascade ClkSoC™ SiT95141 is a single-chip MEMS clock generator 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 provide up to 10 differential or 20 LVMCOS 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 SiT95141 integrates SiTime's 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 and associated 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/frequency jumps inherent to quartz
- 10 times more vibration resistant

The SiT95141 is configured using a companion Cascade GUI software application to simplify clock tree design. The device can also be shipped with a user-specified, factory pre-programmed default startup configuration. The device configuration can be re-programmed twice using two banks of one-time-programmable (OTP) memory during manufacturing or configured in-system via I2C/SPI for additional BOM flexibility. Details of register mapping can be found in the Cascade ClkSoC SiT9514x Register Map – Users Manual.

Features

- Quad Fractional-N PLLs with integrated VCO and loop filter, 93 fs typical RMS phase jitter for SerDes applications
- Flexible input to output frequency translation with 4 inputs and 10 outputs
- 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 2.1 GHz
  - LVCMOS input from 8 kHz to 250 MHz
- Individually configurable output formats and supply
  - LVPECL, CML, HCSL, LVDS or LVCMOS
  - 1.8 V, 2.5 V or 3.3 V
- Synchronized or free run operation modes
- Redundant clock inputs with manual switching
- DCO mode via I2C or SPI with 0.005 ppb resolution
- Programmable Output Delay Control
- Optional zero delay mode on one PLL
- Fixed 4 kHz PLL bandwidth
- Indicators: Lock Loss, Clock Loss, Frequency Drift
- Industry standard 64-pin 9 x 9 mm package

Applications

- Clock tree consolidation replacing crystal oscillators (XOs) and buffers
- Low jitter clock frequency translation and generation
- 10G/100G/400G Ethernet clocking
- 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



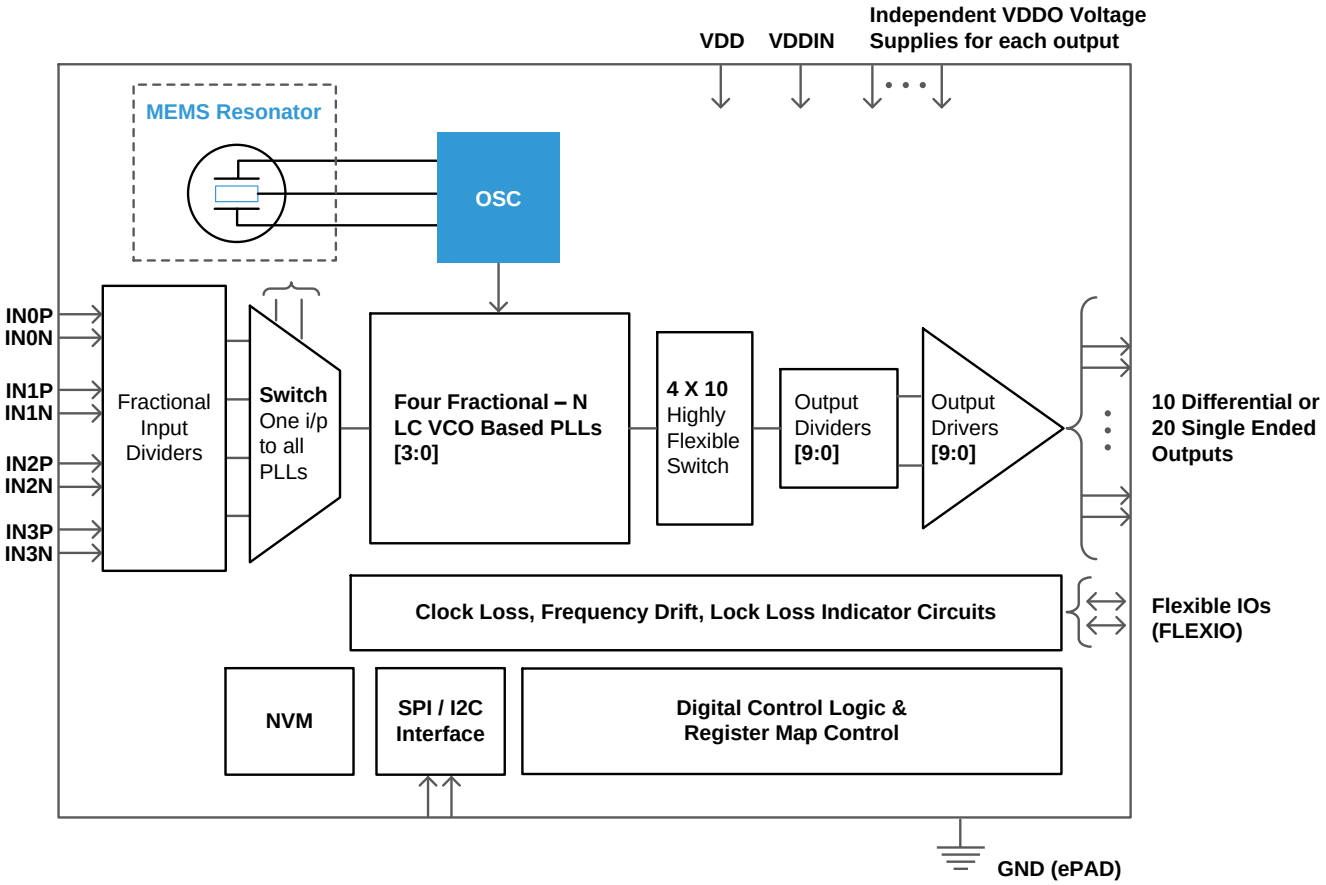
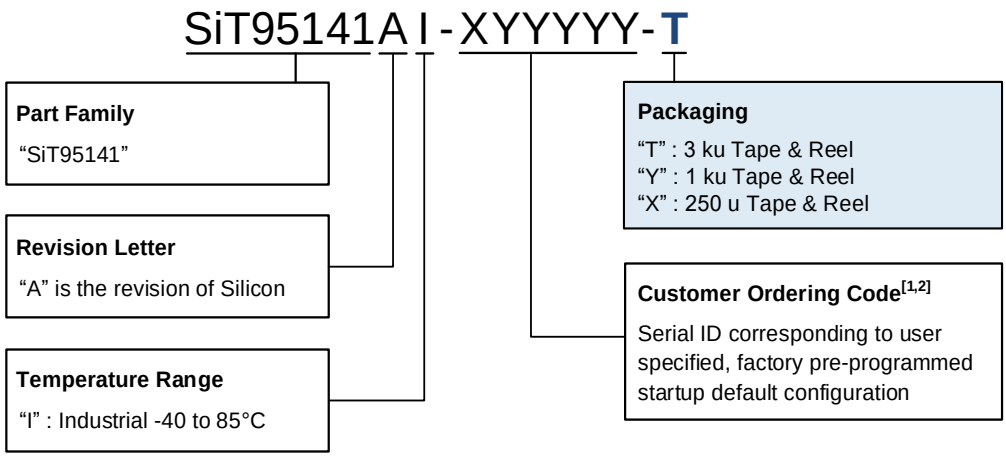


Figure 1. SiT95141 Block Diagram

Table of Contents

Description 1
Features 1
Applications 1
Ordering Information 4
Electrical Characteristics 5
"4-16A" Phase Jitter Methodology for SerDes Applications 19
Functional Description 21
Frequency Configuration 23
Master and Slaves: Architecture Description and Programming Procedures 26
Input Slave Description 30
Output Slave Description 30
PLL Bandwidth 30
    Input Clocks 31
Clock Monitor Description 34
    Fault Monitoring System 34
    Clock Loss Monitors 34
    Lock Loss Monitors 34
Outputs 35
Differential Output Termination 36
    LVPECL 36
    LVDS 38
    HCSL 40
    CML 40
PLL Description 42
    Free run mode 42
    Synchronized mode 42
    Holdover mode 42
Manual Input Switching 42
Zero Delay Mode 42
MEMS Oscillator Reference 42
DCO Mode operation 42
Programmable Interface Top Level View 44
Serial Programming Interface Description 46
    I^2C protocol 46
    SPI Protocol 49
    SPI Timing Details 50
    SPI Single byte write 51
    SPI Single byte read 51
Alarm Registers' Information 53
    Status and Notify 53
Examples for Live Status Read Back 54
    Dynamic registers to read the various alarm registers in the RealTime page 54
Examples of Sticky Bit Clearing 55
Device Initialization for a Non-Programmed Device 56
Monitoring the Status for Master and Slave Pages 66
Programming the Primary E-Fuse 68
    Configuration Bits to Force Power-up of Digital Slave Subsystems 68
    E-Fuse Lock Configuration Bits 69
    E-Fuse Write Configuration Bits 69
    Configuration Bit to Remove Manual Wake Up for Primary E-Fuse 70
    Pseudo Code: Programming the Primary E-Fuse 70
Programming the Secondary E-Fuse 74
    Configuration Bit to Escape to PROGRAM_CMD State in GENERIC_SYS 74
    Configuration Bit to Change the E-Fuse pointer 74
    Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse 74
    Pseudo Code: Programming the Secondary E-Fuse 75
Register Map Details 79

Ordering Information



Notes:

1. X = "A" and "B" customer device, "C to "Z" reserved.
   a. A: Denotes blank devices;
   b. B: Denotes Pre-configured devices, contact SiTime for the specifics
2. 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	Relative to GND
I^2C Bus input voltage	V_{INI2C}	-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	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:

3. Exceeding maximum ratings may shorten the useful life of the device.
4. 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	MSL1 @ 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. Operating Temperature

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Ambient temperature	TA	-40	–	+85	°C	
Junction temperature	TJ	–	–	+125	°C	
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 shows the DC electrical characteristics.

Table 4. DC Electrical Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
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	2.5-V 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.
Supply Current, VDDIN	I_{DDIN}[6]	–	18	25	mA	All Four Inputs assumed to be enabled
Supply Current, VDD	I_{DD}	–	340	408	mA	All Four PLLs and All 10 Outputs enabled (Maximum current mode)
Power supply current, VDDOX	I_{DDO}[7,8,9,10,11]	–	40	48	mA	LVPECL, output pair terminated 50 Ω to V_{TT} (V_{DDO} – 2 V).
		–	28	34	mA	LVPECL2, output pair terminated 50 Ω to V_{TT} (V_{DDO} – 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 V_{DDO}
Power supply current, VDDOX	I_{DDO}[7,8,9,10,11]	–	27	33	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	LVCMOS, 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	LVCMOS
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:

5. To use 1.8 V for VDD, series resistor (4.7K) must be used at RSTB, FlexIO (input) and I^2C/SPI (input) if the input volatge level is 2.5V/3.3V
6. This current can be linearly scaled if smaller than 4 inputs are enabled.
7. LVPECL mode provides 6 mA of common mode current on each output. LVPECL2 mode does not provide this common mode current.
8. A 50 Ω Termination resistor with a DC bias of VDDO – 2 V for LVPECL standards is supported for VDDOx = {2.5 V, 3.3 V}.
9. 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.
10. The LVDS Boost Mode 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 Ω Differential termination.
11. Refer to the appropriate later section in the data sheet for various terminations that are supported.
12. For efuse programming in SiT95141 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.

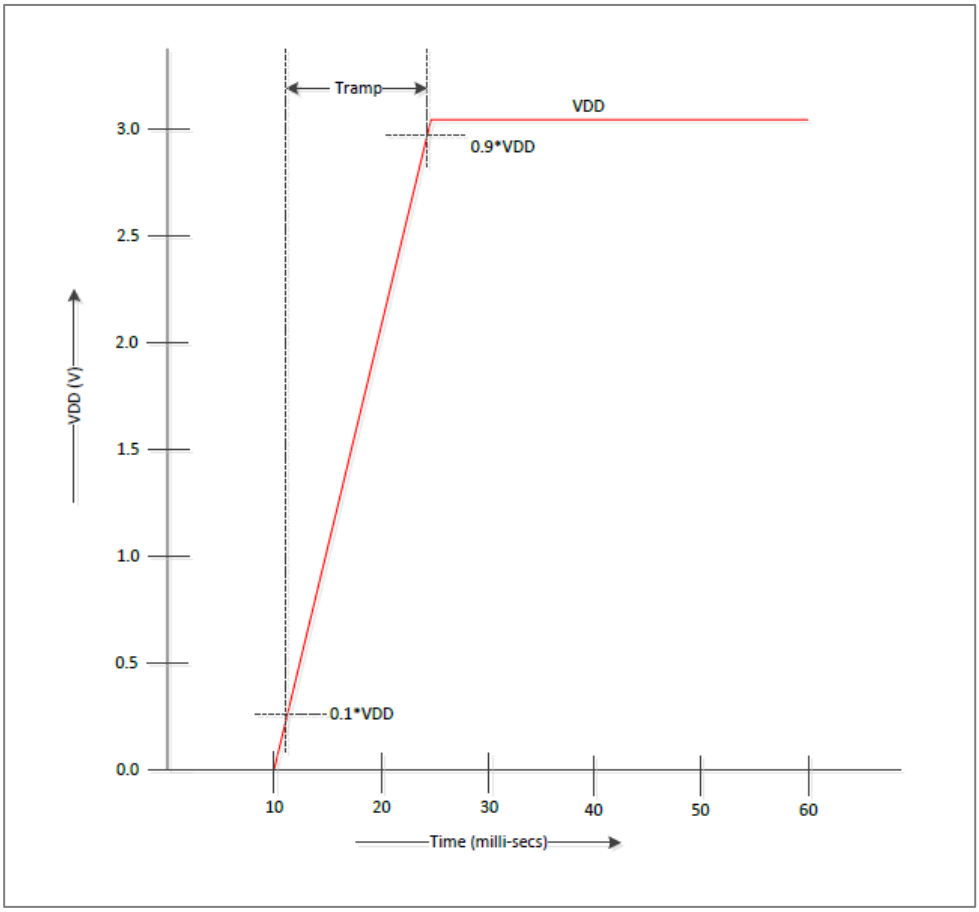


Figure 2. Supply Voltage Ramp

Table 6. Input Clock Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Differential Input Frequency	F_{IN}	0.008	–	2100	MHz	Differential clock input
		0.008	–	250	MHz	All Single-ended signals (including LVCMOS)
Voltage Swing (Differential Amplitude Peak or Single Ended Peak to Peak for the differential signal)[13]	V_{IN}	100	–	–	mV	AC-coupled f_{IN} < 400 MHz
		225	–	–	mV	400 MHz < AC-coupled f_{IN} < 750 MHz
		350	–	–	mV	750 MHz < AC-coupled 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	
Input Duty Cycle	DC	40	–	60	%	Measured at crossover point
Input Capacitance	C_{IN}	–	0.3	–	pF	
Input Resistance	R_{IN}	–	15	–	kΩ	AC Coupled SE
		–	10	–	kΩ	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	–	kΩ	

Notes:

13. AC Coupled input assumed with series capacitance for differential inputs or single ended AC Coupled inputs. Swing requirement at device pins.
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 Clock Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Input Voltage	V_{IL}	–	–	0.3 x V_{DDIO} [17]	V	
	V_{IH}	0.7 x V_{DDIO} [17]	–	–	V	
Input Capacitance	C_{IN}	–	1	–	pF	
Input Resistance	R_{IN}	–	25	–	kΩ	
Minimum Pulse Width	PW	100	–	–	ns	FINC, FDEC
Update Rate	F_{UR}	–	–	1	µs	FINC, FDEC

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%
10 Year Aging	F_10y	–	±2	–	ppm	Ambient temperature of 25°C
Differential Output Frequency	$F_{OUT,DIFF}$[18]	1	–	2100M	Hz	LVPECL, CML, LVDS outputs
Differential Output Frequency	$F_{OUT,DIFFH}$[18]	1	–	700M	Hz	HCSL outputs
Single Ended Output Frequency	$F_{OUT,SE}$[18]	1	–	250M	Hz	LVCMOS outputs
PLL Loop Bandwidth	F_{BW}	4000	4000	4000	Hz	Constant
Jitter Peaking	J_{PEAK}	–	–	0.1	dB	Meets SONET Jitter Peaking requirements in closed loop
Power Supply to I²C or SPI interface ready	T_{START}	–	–	10	ms	No I²C transaction valid till 10 ms after all power supplies are ramped to 90% of final value.
DCO Mode Frequency Step Resolution	$F_{RES,DCO}$[19]	–	0.005	–	ppb	Frequency Increment or Decrement resolution. This is controlled by accessing the control registers.
Resolution for Output Delay	T_{RES}[20]	–	35	–	ps	Programmable per output clock with this resolution for a total delay of 7.5 ns
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
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}[21]	–	100	–	ps	
Temperature Variation of delay in ZDB mode	$T_{ZDELAY, TMP}$[21]	–	–	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 parts	$T_{START, Special}$	–	–	11	ms	Using a special mode for fuse locked parts to generate one free run output from one PLL
Power Supply Rejection Ratio[22,23]	$PSNR_{VDD}$	–	-96	–	dBc	F_{OUT} = 156.25 MHz, F_{SPUR} = 100 kHz, BW= 100 Hz PSNR on VDD Supply, VDD = 3.3 V
		–	-90	–		F_{OUT} = 156.25 MHz, F_{SPUR} = 100 kHz, BW= 100 Hz PSNR on VDD Supply, VDD = 2.5 V
		–	-75	–		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 VDDIN Supply
	$PSNR_{VDDO}$	–	-80	–	dBc	F_{OUT} = 156.25 MHz, F_{SPUR} = 100 kHz, BW= 100 Hz PSNR on VDDO Supply
Output-Output Crosstalk[24,25,26]	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). Range supported is 8 kHz to 2100 MHz for all the other outputs.
19. 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.
20. All output clocks from one specific PLL are phase aligned. Relative delay adjustment is then possible on each clock individually as defined by the T_{RES} parameter.
21. Both input and feedback at 8MHz with the delays exactly matched and same slew for both
22. 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.
23. Output PSNR measured with LVDS standard which (along with the LVDS boost) are the recommended standards for AC Coupled terminations.
24. Measured across adjacent outputs- All adjacent outputs are covered and the typical value for the worst case output to output coupling is reported.
25. The adjucent output pairs are chosen at 155.52 MHz and 156.25 MHz frequencies.
26. This crosstalk 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_{DDIO} [27] x 0.75	–	–	V	I_{OH} = –2 mA
	V_{OL}	–	–	0.125	V	I_{OL} = 2 mA

Note:

27. 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 shows the fault monitoring indicators.

Table 10. Fault Monitoring Indicators

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Clock Loss Indicator Thresholds	CL_X[28,29]	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
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:

28. Clock Loss Indicators are used for updating Status Registers in the register map:
29. 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.

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[30]	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 1st order roll off, then integrating from 12 kHz to 20 MHz. Includes spurs.

Note:

30. 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}[30,31]	–	120	–	fs	F_{OUT} = 622.08 MHz, F_{IN} = 10 MHz, Single PLL Profile

Note:

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

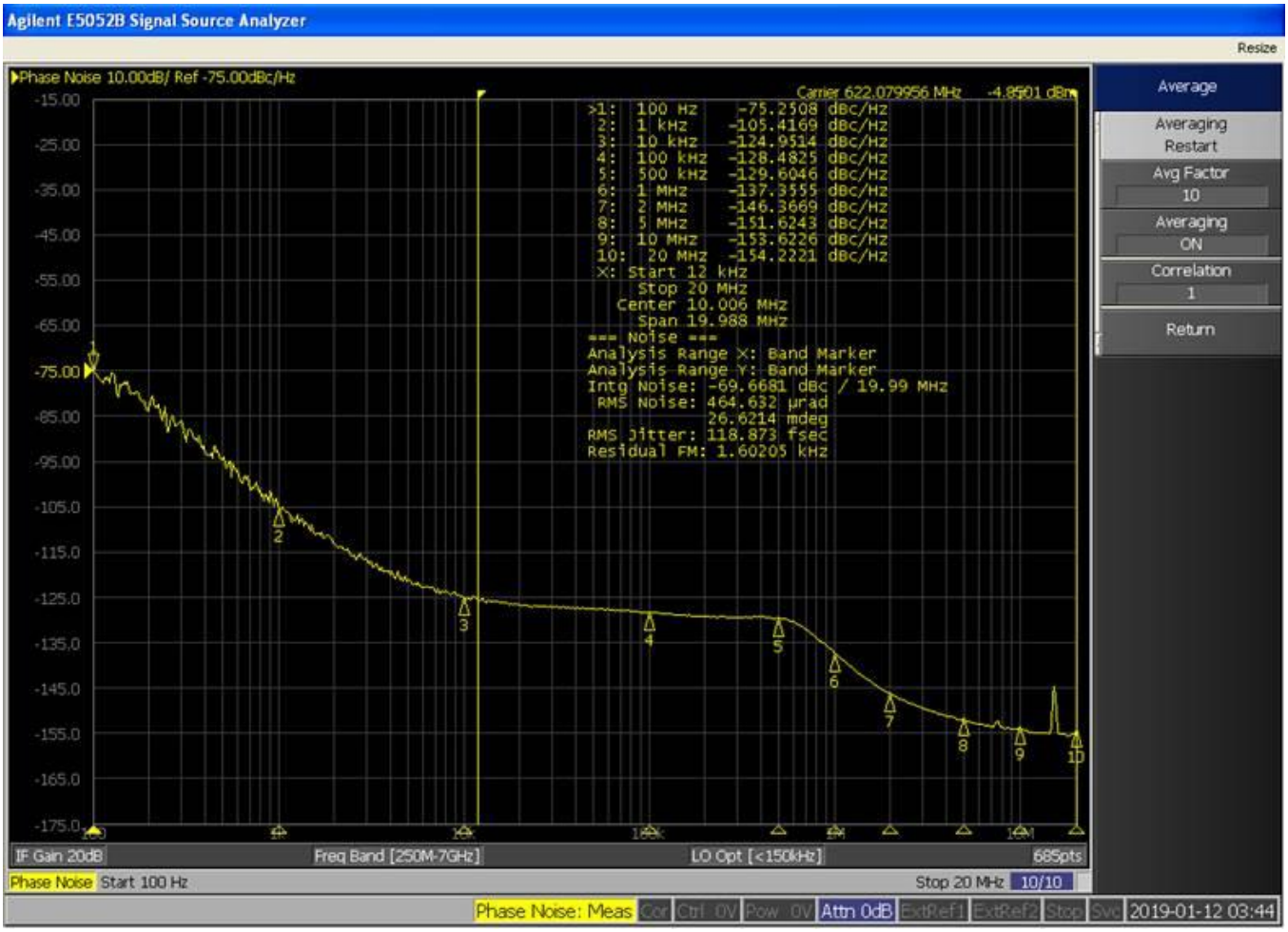


Figure 3. Representative Phase Noise Measurement: F_{OUT} = 622.08 MHz, F_{IN} = 10 MHz,



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[32]	Offset Frequency = 10 Hz	-91	dBc/Hz	F_{OUT} = 122.88 MHz, F_{in} = 10 MHz
		Offset Frequency = 100 Hz	-113		
		Offset Frequency = 1 kHz	-130		
		Offset Frequency = 10 kHz	-138		

Note:

33. This is the noise contribution of the chip only without including 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 uA Load
Output Low Voltage	VOL	–	–	0.3	V	4 mA load
Output Low Voltage	VOL	–	–	0.2	V	100 µA 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	ΔVOCM	–	–	50	mV	
Output Leakage Current	IOZ	-20	–	20	µA	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 – 1.45		Rterm = 50 Ω to VTT(VDDO – 2.0 V), w/o common mode current
DC Electrical Specifications – HCSL 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 Voltage	VP	300	–	–	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 – 0.3	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)						
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.
Output Duty Cycle	tODC	45	50	55	%	Measured at differential 50% level, 156.25 MHz
LVDS Output differential peak	VP_{LVDS}	247	350	454	mV	Measured at 156.25M Output
Boosted LVDS differential peak	VPB_{LVDS}	500	700	–	mV	Measured at 156.25M Output
LVPECL Differential peak	VP_{LVPECL}	450	750	–	mV	Measured at 156.25M Output
CML Output Differential Peak	VP_{CML}	250	–	600	mV	Measured at 156.25M Output

Table 14 below shows the I^2C 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}	I2C Address	–	0x69	–		

Figure 5 shows the I²C timing waveforms.

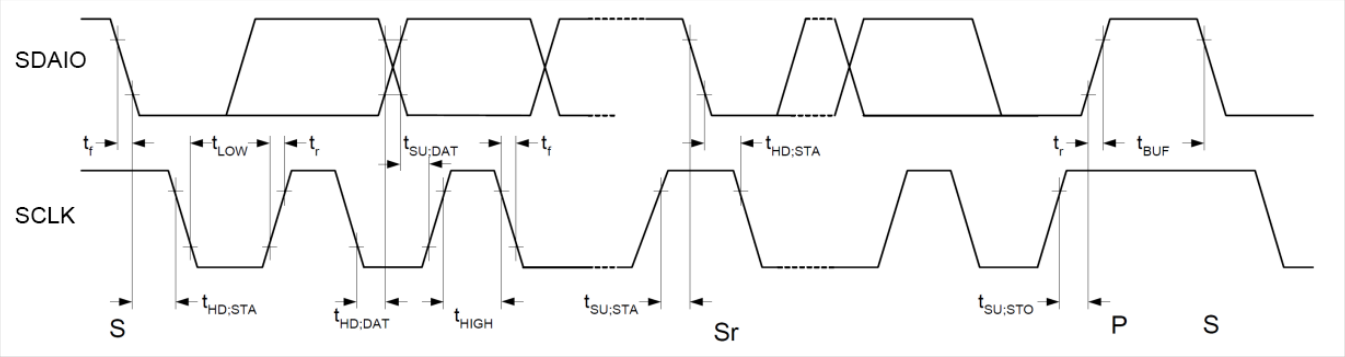


Figure 5. I²C Timing Waveform

Table 16. SPI Bus Timing Specifications[33]

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:

34. 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.

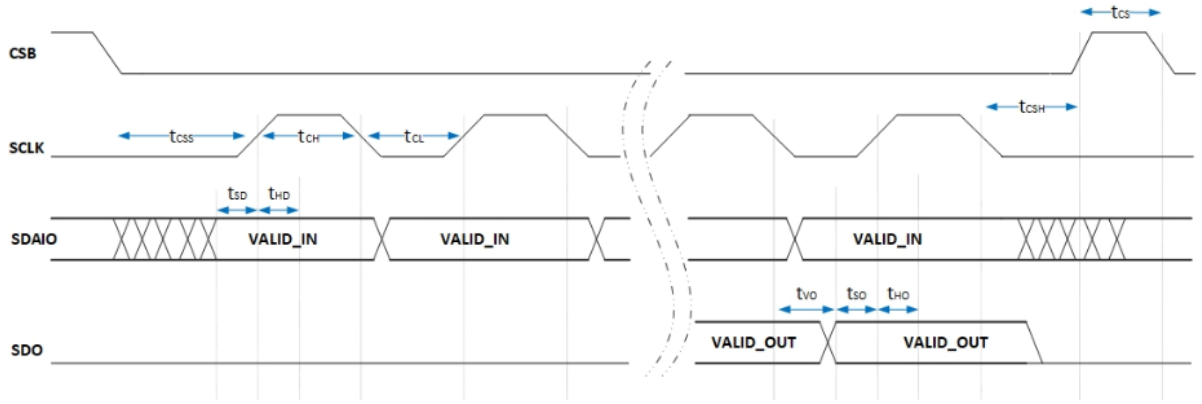


Figure 6. SPI Timing Waveform

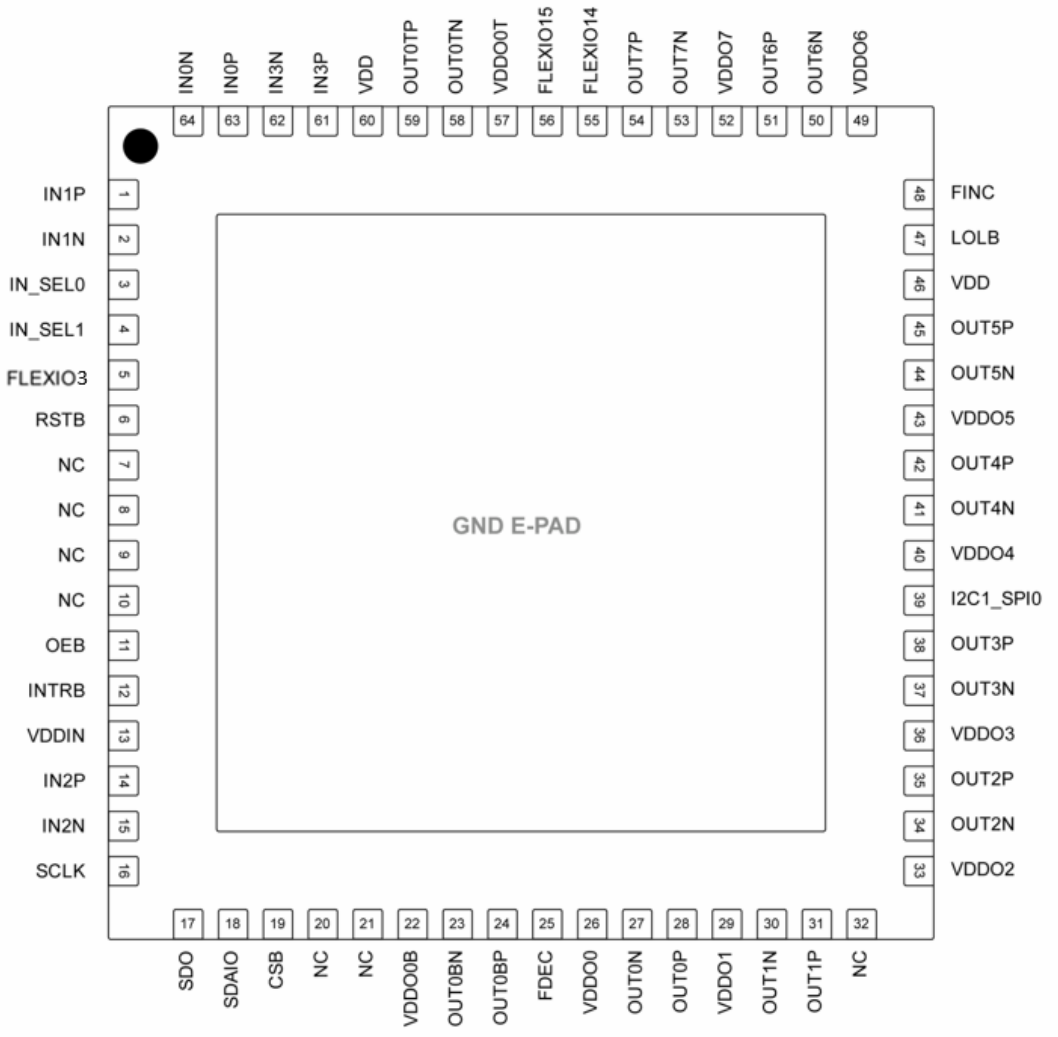


Figure 7. SiT95141 Top View

Table 17. Detailed Pin Description[35,36,37,38,39,40]

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 such that the same choice holds for all 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.	LVPECL, LVDS, HCSL, CML and LVCMOS support.
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.	
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	Input/Output	12				
RSTB	Input	6	25		Active low reset internally pulled up to VDDIO; Pull Up Resistor to VDDIO of fixed value (25 KΩ). Can be left floating	Active low signal performs a complete reset of the part. See Note 34.
OEB	Input	11		25	Used to disable (when 1) all the output clocks	See Note 34.

Table 17. Detailed Pin Description (continued)[35,36,37,38,39,40]

Pin Name	I/O Type	Pin #	Pull-up (kΩ)	Pull-down (kΩ)	Function	Comments
LOLB	Input/Output	47			Loss of Lock Indicator (NOR value of all PLLs' LOL active high indicators comes out on the LOLb pin)	
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 34.
FDEC	Input/Output	25		25	DCO decrement	
FINC	Input/Output	48		25	DCO increment	
IN_SEL0	Input	3		25	Input Clock Selection for Manual selection of active clock	
IN_SEL1	Input	4		25		
SDAIO	Input/Output	18	25		I^2C Serial Interface Data (SDA) / SPI Input data (SDI)	
SCLK	Input	16	25		I^2C Serial Interface Clock or SPI Clock Input. Pull Up Resistor to VDDIO of fixed value (25 KΩ)	
CSB/A0	Input	19	25		Chip Select for the SPI Interface. In I^2C mode this is the A0 address pin.	
SDO/A1	Output / Input	17		25	Serial Data Output (SPI Interface). In I^2C mode this is the A1 address pin.	
I2C1_SPI0	Input	39	25		Choose between SPI(0) and I^2C(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	
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	
FLEXIO15	Input/Output	56		25		
NC	No Connect	7, 8, 9, 10, 20, 21, 32			No connect. This pin is not connected to the die.	

Notes:

35. To use 1.8 V for VDD, series resistor (4.7K) must be used at RSTB, FlexIO (input) and I^2C/SPI (input) if the input volatge level is 2.5V/3.3V
36. 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 programmable GUI.
37. All digital input/output GPIOs (FLEXIOs) have an on-chip 25 kΩ pull down resistor to ePAD ground (unless mentioned otherwise) and can be left unconnected if not used.
38. The I2C1_SPI0 pad has a an on-chip 25 kΩ pull up resistor to indicate default mode of communication as I^2C unless this pin is pulled down on the board to indicate the SPI mode.
39. In I^2C mode, the serial data and clock have an on-chip 25 kΩ pull up resistor to VDDIO.
40. The RSTB pin has an on-chip 25 kΩ pull up resistor to VDDIO.
41. SDO and CSB pins are used as A1 and A0 address bits for I^2C Slave address setting in I^2C mode. Default Slave address value is 0x69. SDO (A1) and CSB (A0) pins have an on-chip 25 kΩ pull down resistor for pulling these ones to ground if they are floating (not connected) in the I^2C mode.

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[42]. Above the third harmonic, phase noise rolls off quickly and can be ignored.

Note:

42. "How to evaluation reference-clock phase noise in high-speed serial links", Signal Integrity Journal.

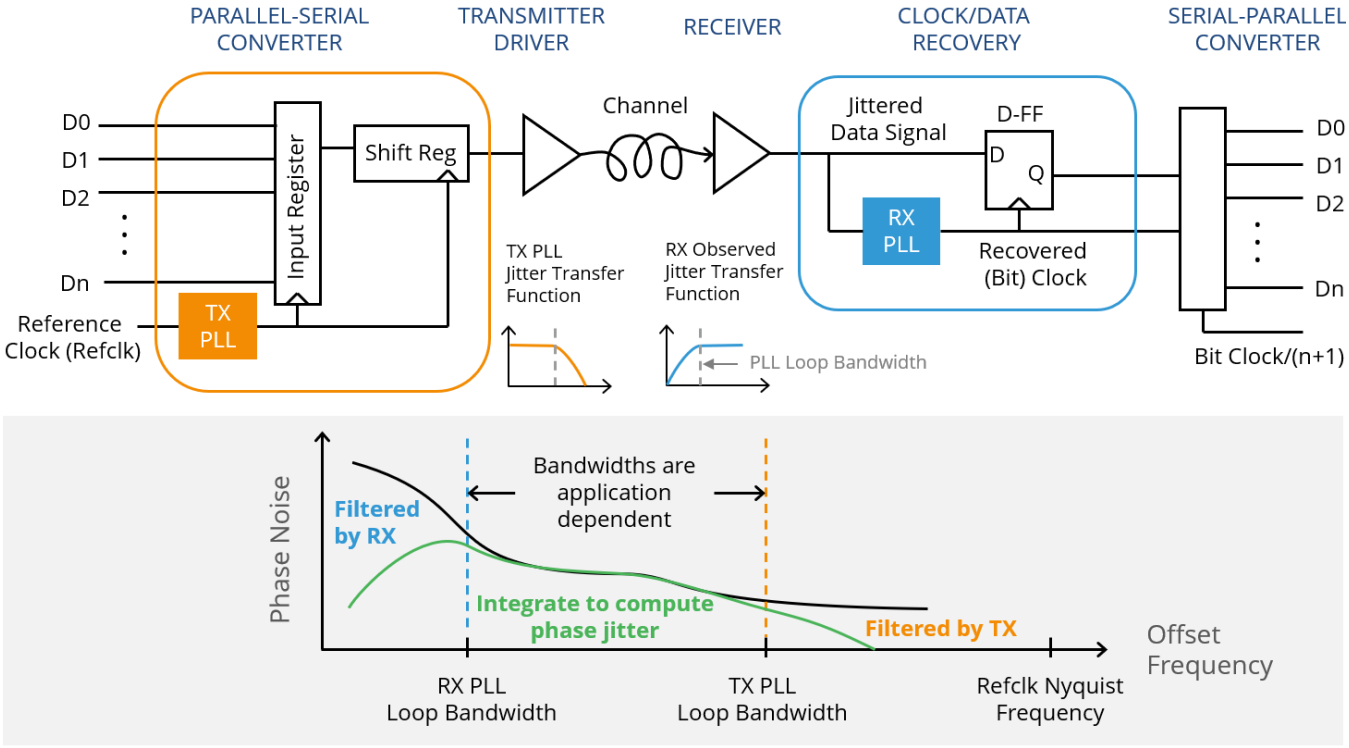


Figure 8. A generic serial link (top) uses a transmit PLL and receive CDR to low and high pass filter, respectively, refclk phase noise. This forms a band-pass system filter (bottom) for computing phase jitter.

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[42].

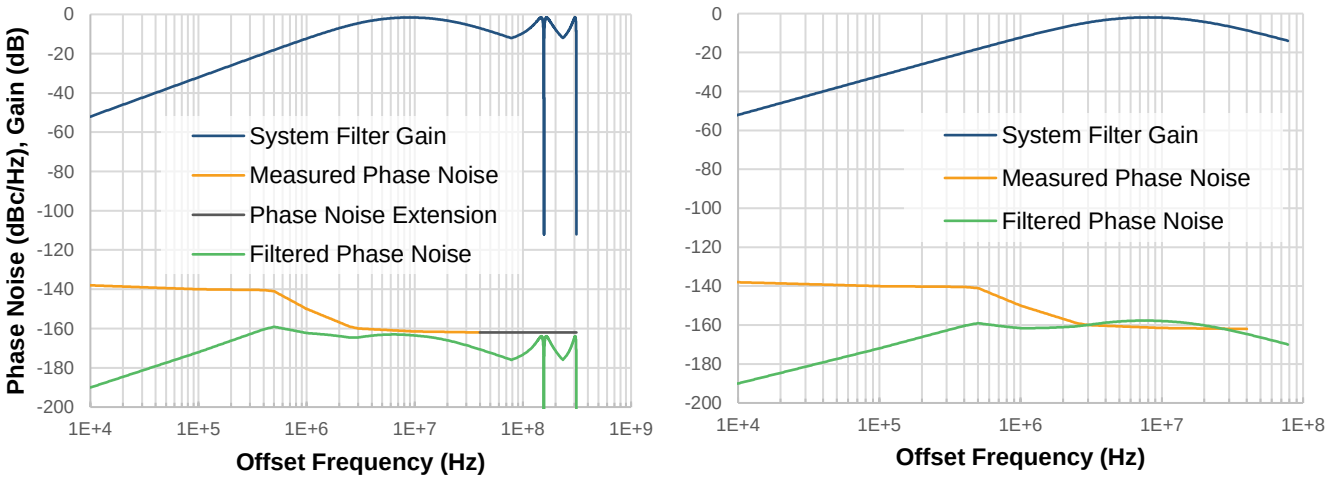


Figure 9. Illustration of two equivalent processes to filter aliased phase noise. The left chart extends (black) the measured phase noise (orange) to the 3rd harmonic, mirrors the filter (blue) across higher Nyquist zones before deriving the filtered phase noise (green). Alternatively, the right chart 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

The SiT95141 is a high-performance clock generator that offers four PLLs for 4 frequency translation pathways from the same input. The four clock inputs map to all of the four PLLs such that clock priority is the same across the 4 PLLs. This creates an arrangement that provides up to 4 fractional translations from one input at any given time. The output high frequency voltage-controlled oscillators (VCO) associated with each PLL are mapped to the 10 outputs in a very flexible fashion. This offers a very flexible frequency translation arrangement.

Figure 8 shows a high-level architecture diagram of SiT95141.

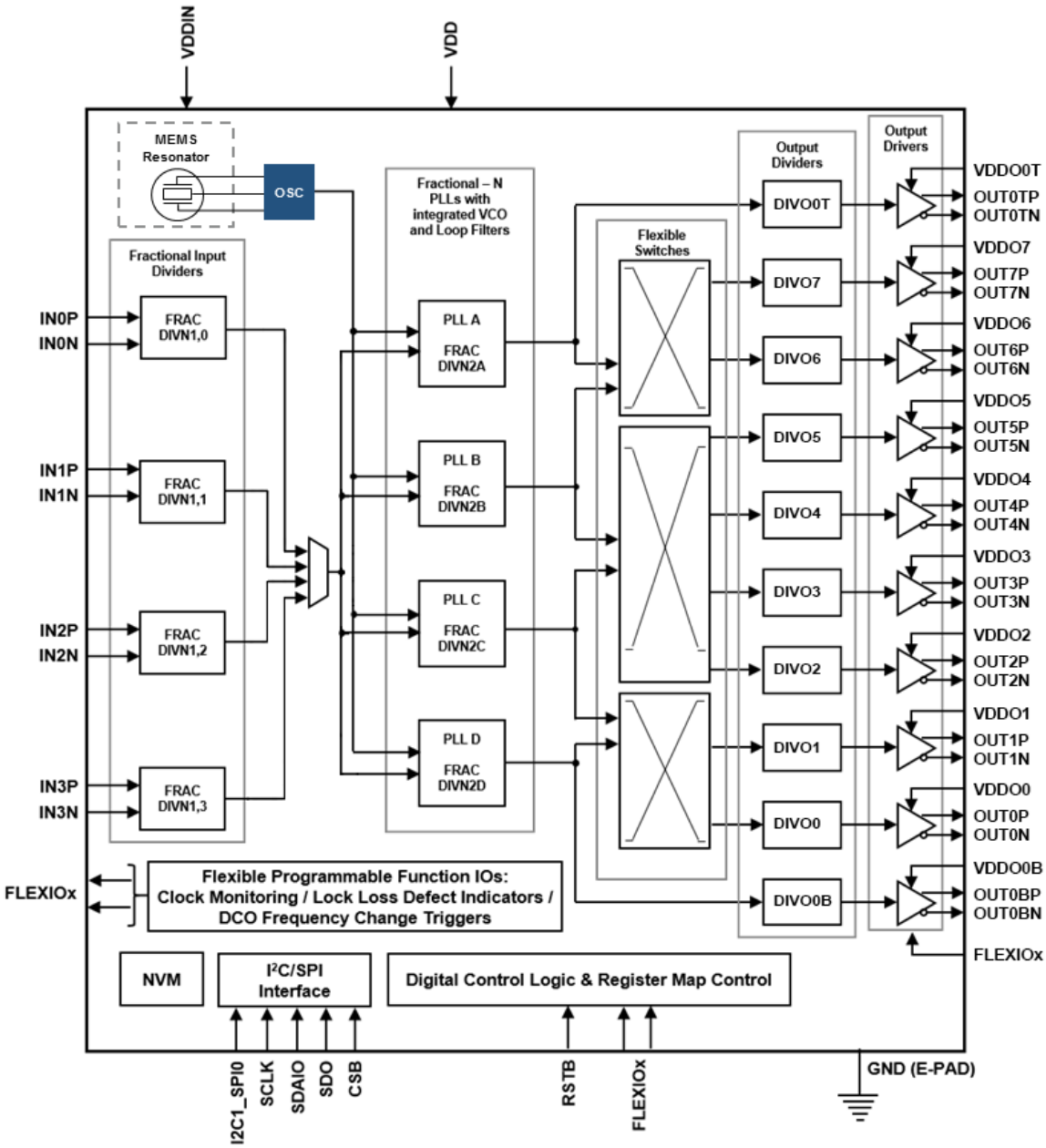


Figure 10. Overall Architecture

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 PIF structure is a register map that is divided into several pages according to the 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 18 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 device Device Configuration details Control for the master sequencer FSM Fuse Pointer for each of the remaining pages
1	ClkMon Slave	Clock Loss related function Frequency Drift related func tion
2	Input Slave	Input 3 / 2 / 1 / 0 related information (Input type, DIVN1 divider configuration)
3	Output Slave	Flexible Outputs 7 / 6 / 5 / 4 / 3 / 2 / 1 / 0 (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 device are shown in Figure 9.

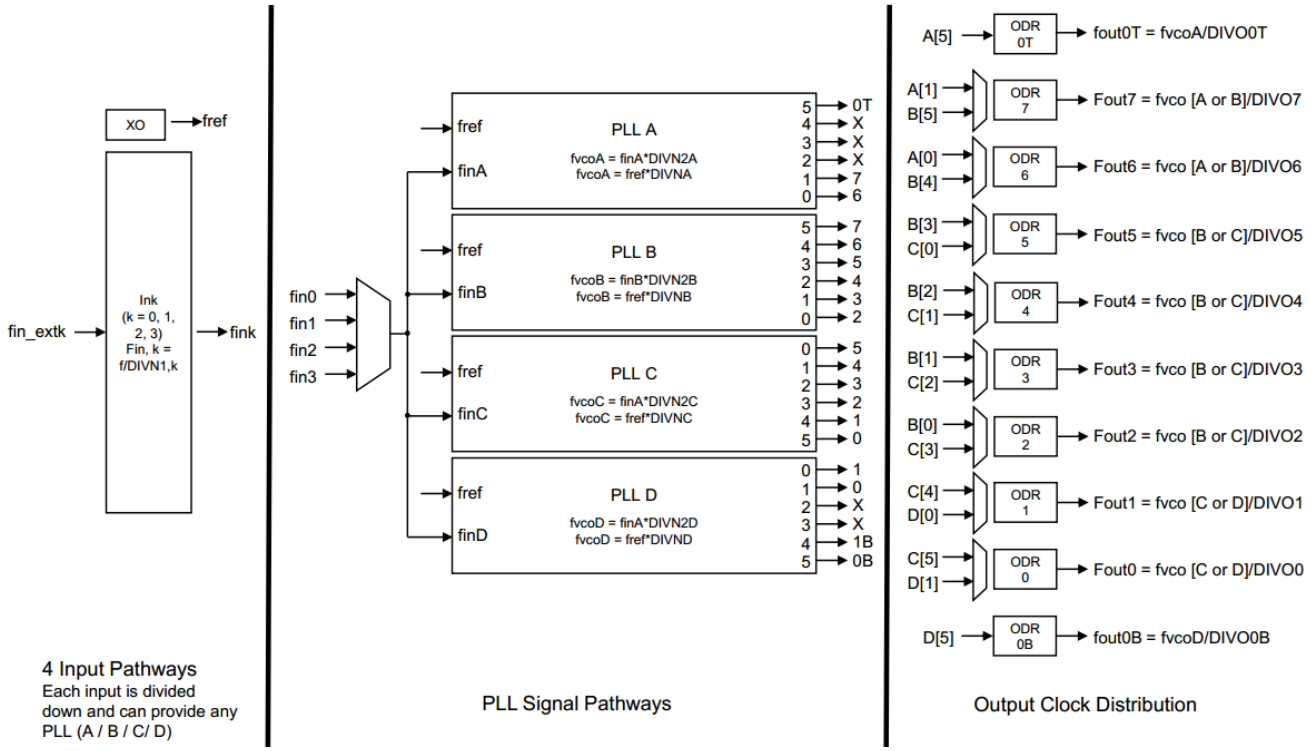


Figure 11. Overall Hierarchy of Clocks

The four input clocks with frequency fin_ext*k* translate to PLL input clocks fin*k* following division by the respective input dividers with fractional or integer frequency division ratios DIVN1*k* where the index *k* ϵ {0, 1, 2, 3}. See Figure 10. All of the PLLs choose one of the four divided input clocks fin*k* as its active input clock.

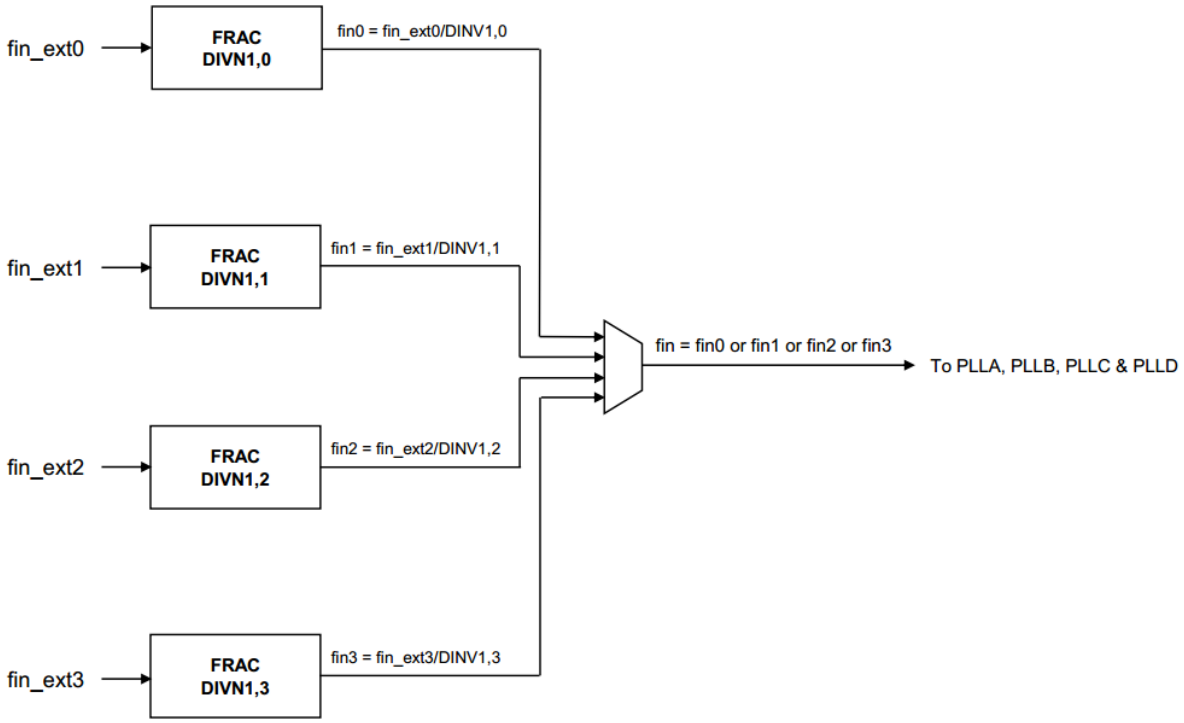


Figure 12. Input Clock Distribution

Each PLL*x* (*x* ϵ {A, B, C, D}) has a high frequency VCO whose frequency is determined in the free run mode by fref with relation fVCXOx = DIVNx*fref. In the frequency translation synchronized mode, the VCO frequency is corrected from its free run frequency to satisfy the relation fVCO*x* = DIVN2*x**fin*x* where fin*x* is chosen from one of fin*k* input clocks per the desired input clock priority for PLLx.

Nominally the fractional dividers DIVN*x* and DIVN2*x* are chosen such that the relation DIVN*x**fref = DIVN2*x**fin,x = fVCO*x* is satisfied.

Each of the ten Output Drivers (ODRj, j ϵ {0, 1, 2, 3, 4, 5, 6, 7 ,0B ,0T}) chooses an appropriate VCO frequency and divides it using their respective integer divider DIVOj to get the output frequency foutj. See Figure 11.

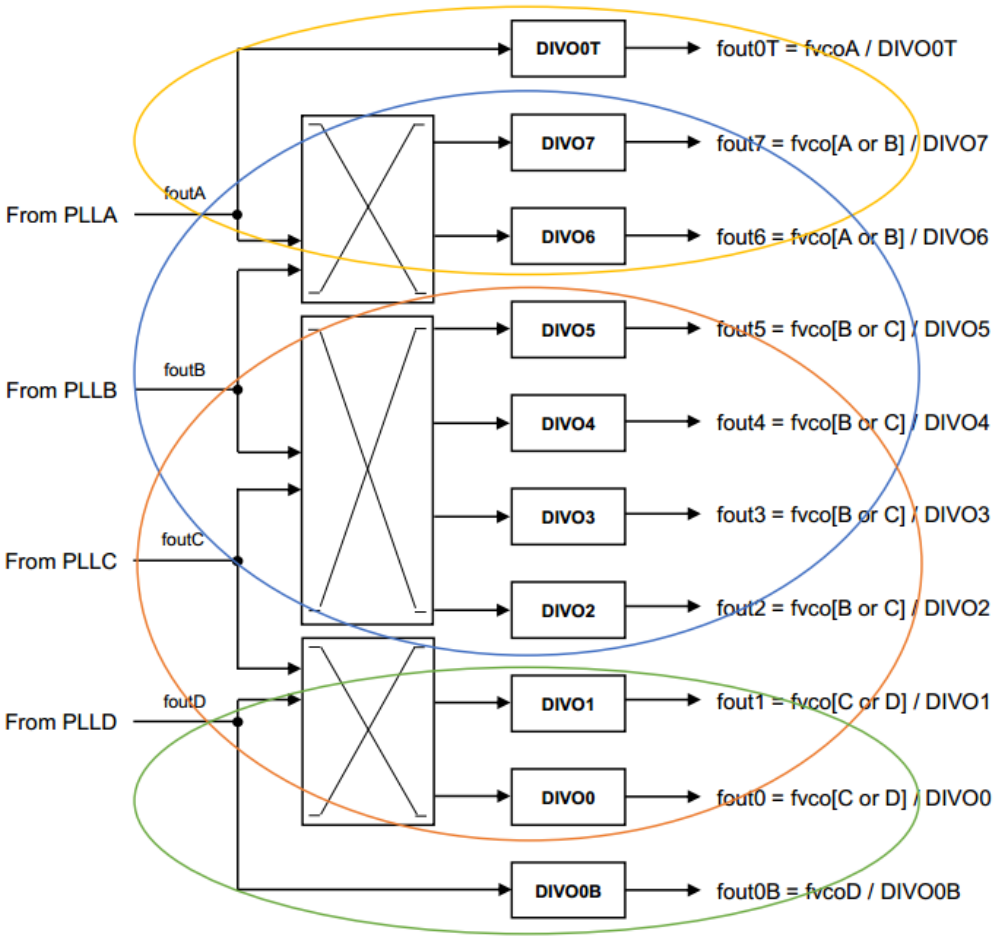


Figure 13. Output Clock Distribution

The choice of the fractional dividers as well as the placement of foutj 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 (I^2C / SPI).

The Master memory structure is shown in Figure 12. 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 (I^2C / SPI) on Page 0 and can be overwritten from the I^2C / SPI interface. The "Chip Settings" is the memory space that is not addressable from the I^2C / 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.

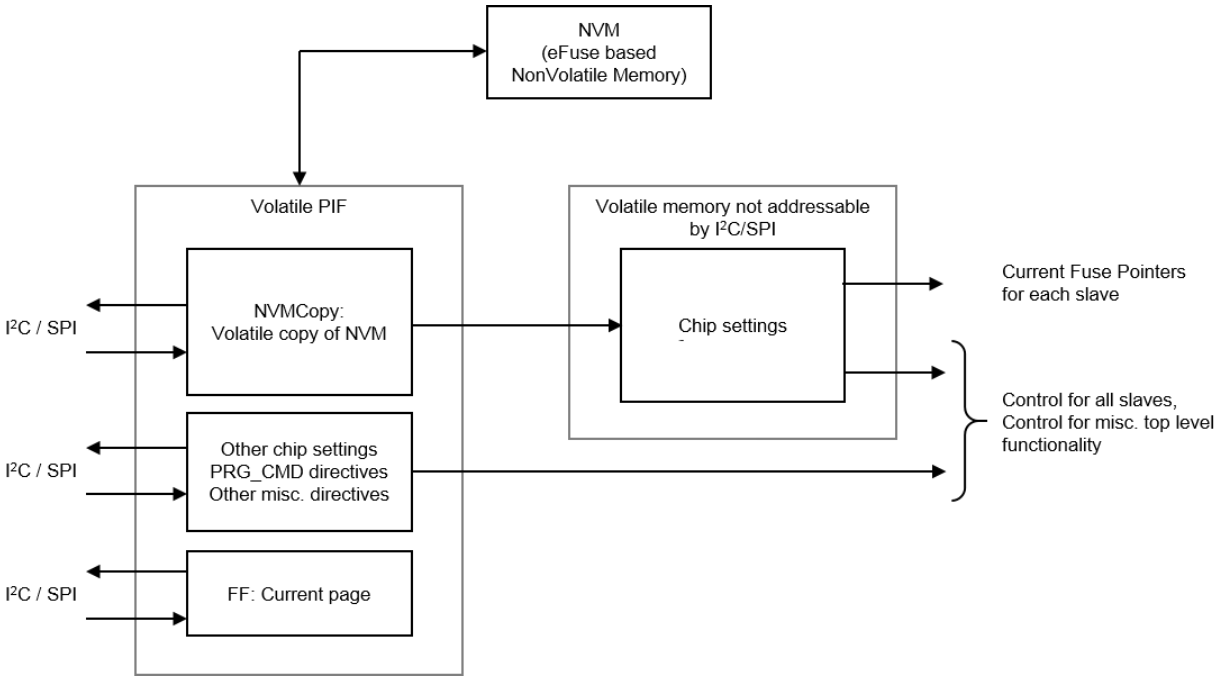


Figure 14. Master Memory Structure

The memory structure for each slave is shown in Figure 13 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 I^2C / 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.

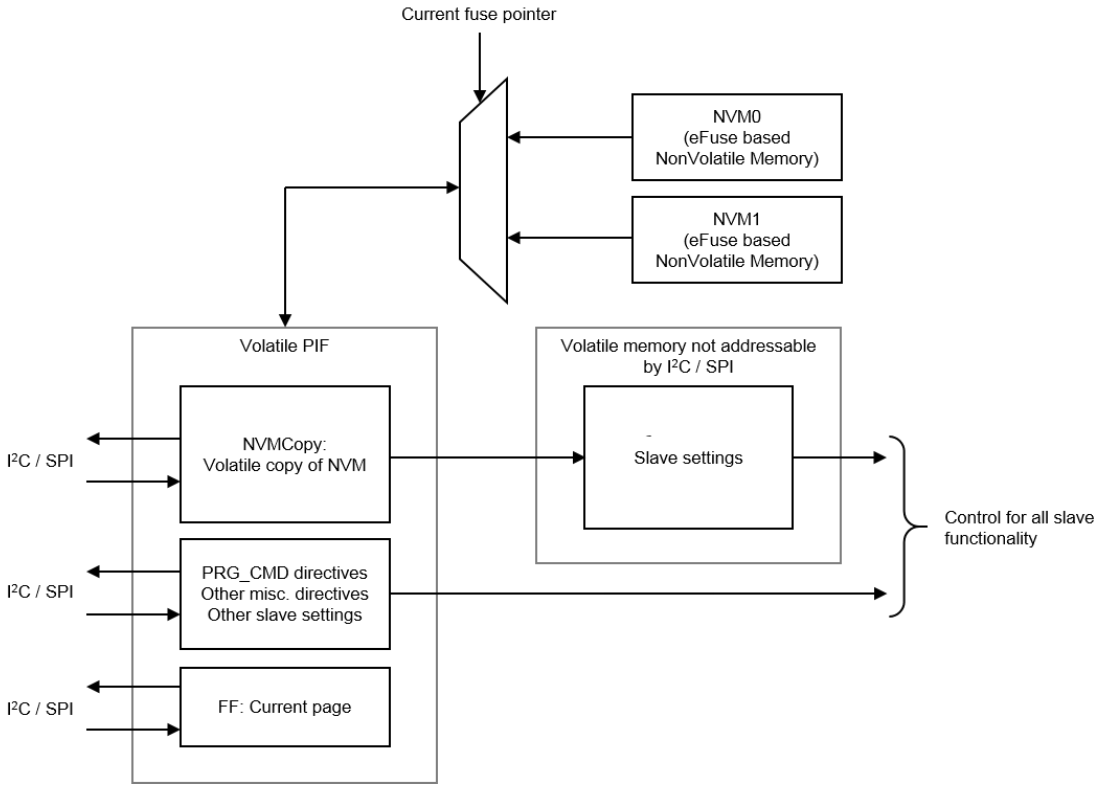


Figure 15. Slave Memory Structure

The Master Wake-Up Finite State Machine (FSM) is shown in more detail in Figure 14. 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 14). 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 15 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.

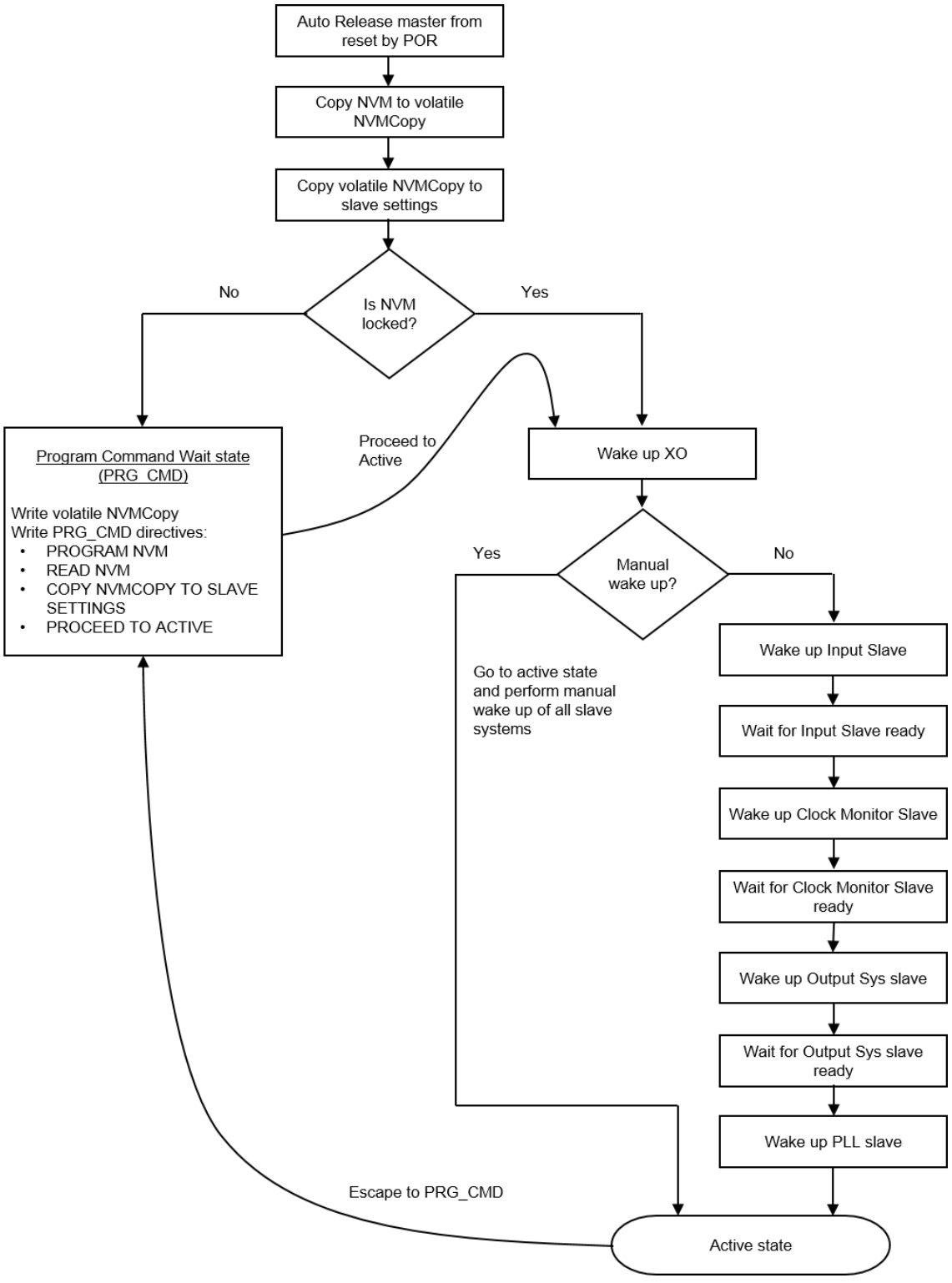


Figure 16. Master Wake-up Finite State Machine

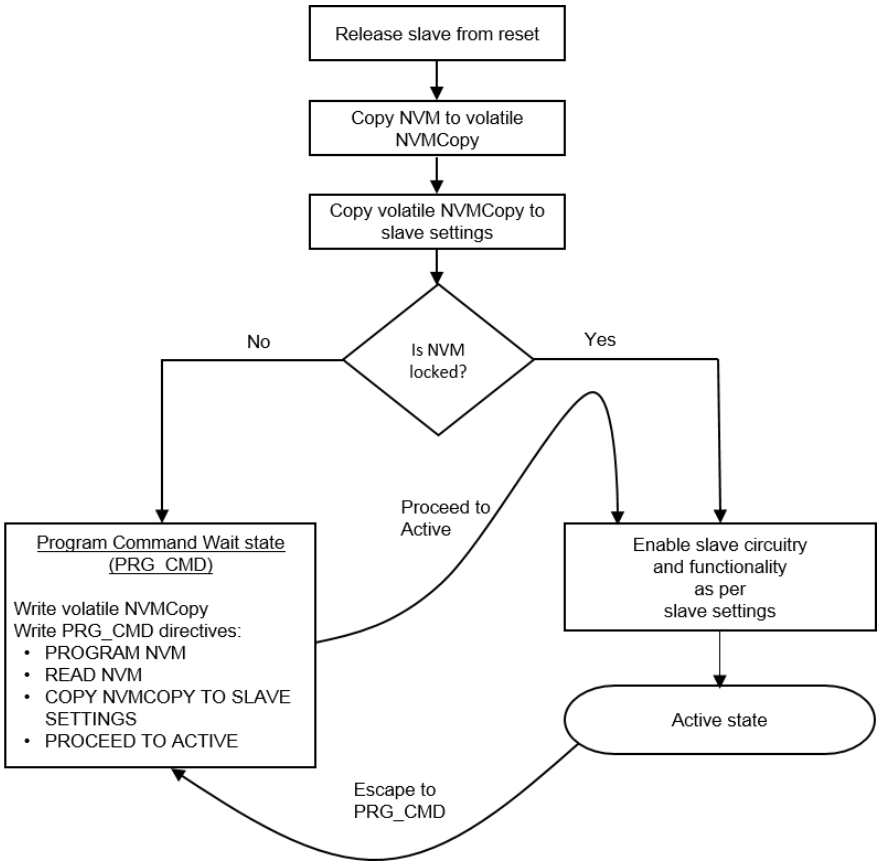


Figure 17. Slave Wake up Finite State Machine

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 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.

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 Bandwidth is 4 kHz. This is the bandwidth that is used for steady state operation.

Input Clocks

The recommended Input Clocks termination schemes are shown on the figures below.

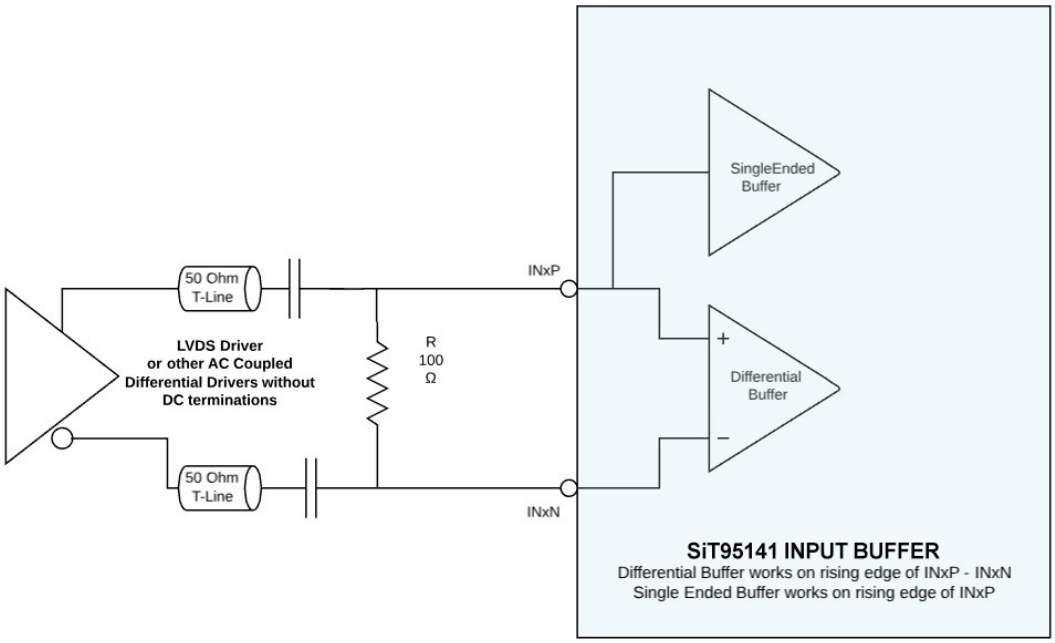


Figure 18. AC Coupled Differential LVDS Input or Other AC Coupled Driver without DC Terminations – Uses Differential Buffer Pathway

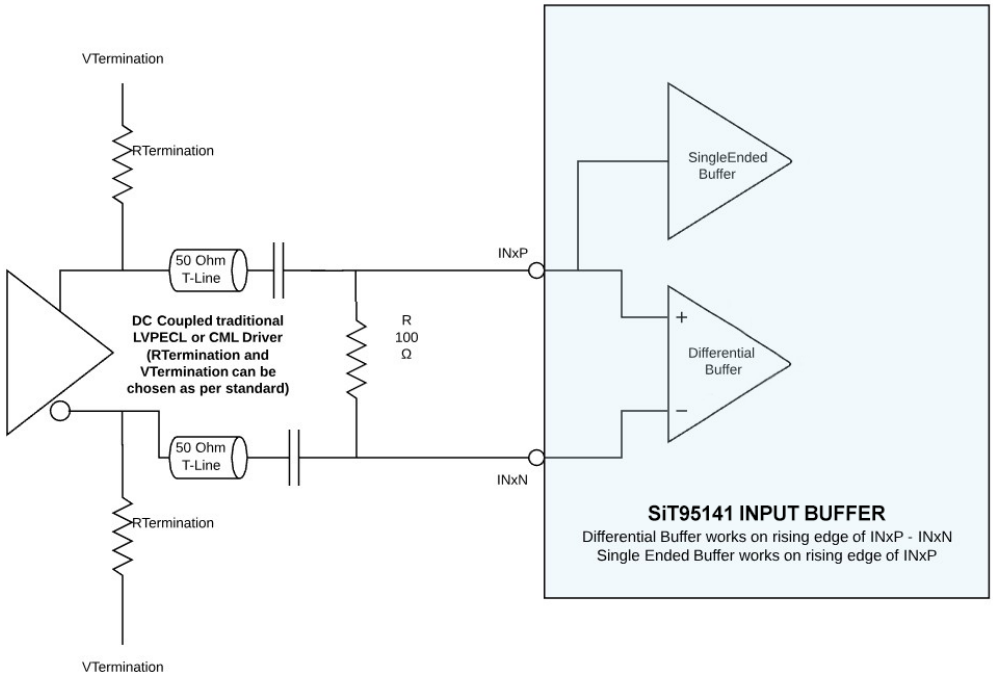


Figure 19. AC Coupled Differential LVPECL or CML (Resistor and Voltage termination is as per the standard) – Uses Differential Buffer Pathway

Driver Supply	R1 (Ohms)	R2 (Ohms)
1.8V	140	665
2.5V	325	475
3.3V	445	365

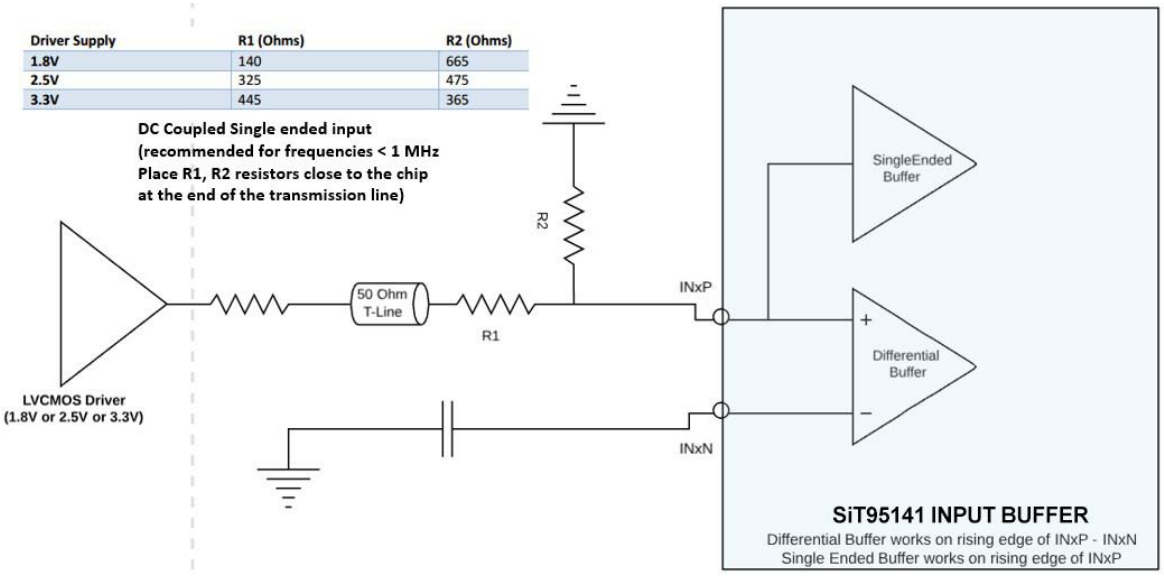


Figure 20. DC Coupled Single Ended Driver – Uses Single Ended Buffer Pathway[41]

Note:

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

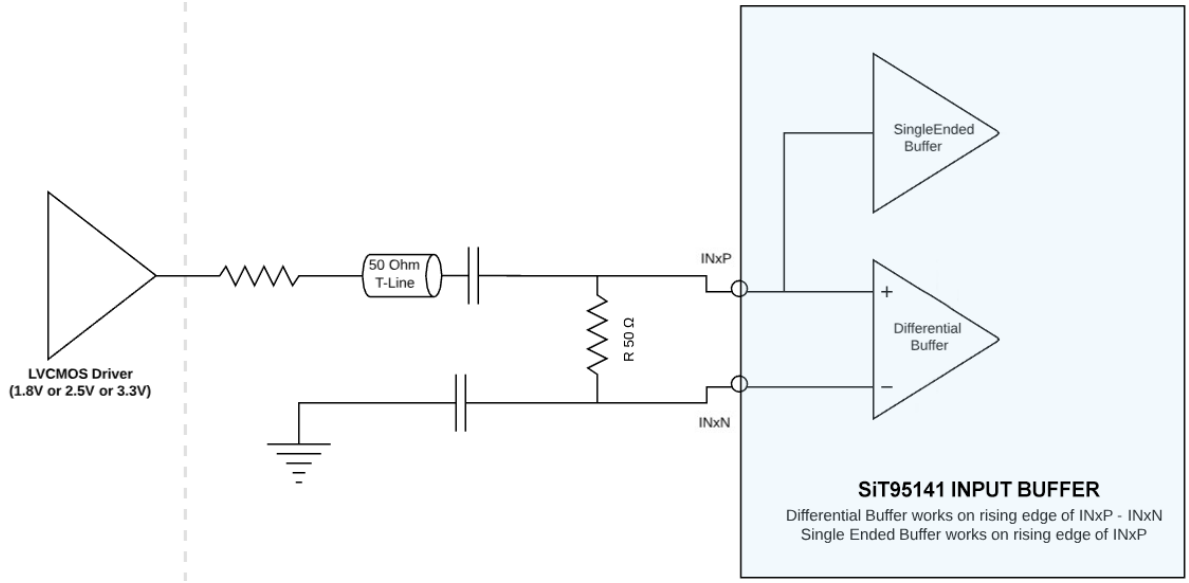


Figure 21. AC Coupled Single Ended Driver with 50 Ω Termination on receiver (chip) side

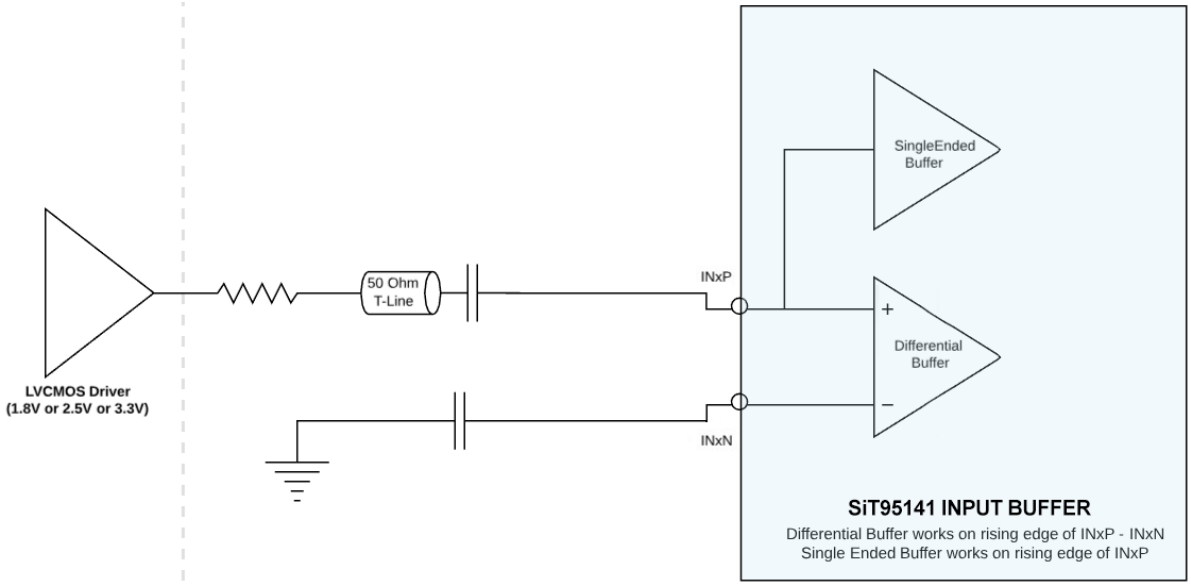


Figure 22. AC Coupled Single Ended LVCMOS input without 50 Ω Termination

Clock Monitor Description

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 ϵ {0, 1, 2, 3}) post the DIVN1,k divider translation rather than the external frequencies fin_extk (k ϵ {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 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.

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.

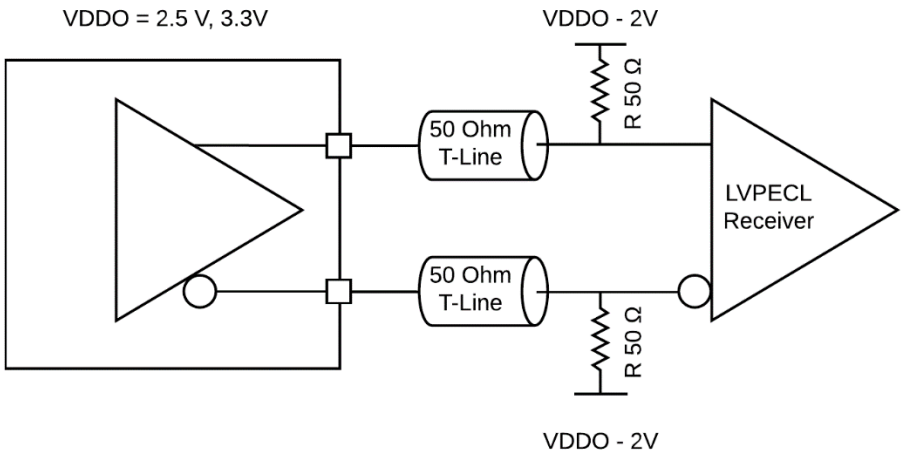
Additionally from the point of view of LL de-assertion, there is a delay from the point in time that lower than the specified ppm value is achieved to the point where the actual LL is de-asserted to the user such that LL never asserts during this delay period. The choice of this delay is with a timer 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 SiT95141 supports 10 differential output drivers which can be independently configured as differential or LVCMOS. The output load and terminations for each differential output standard are shown below.

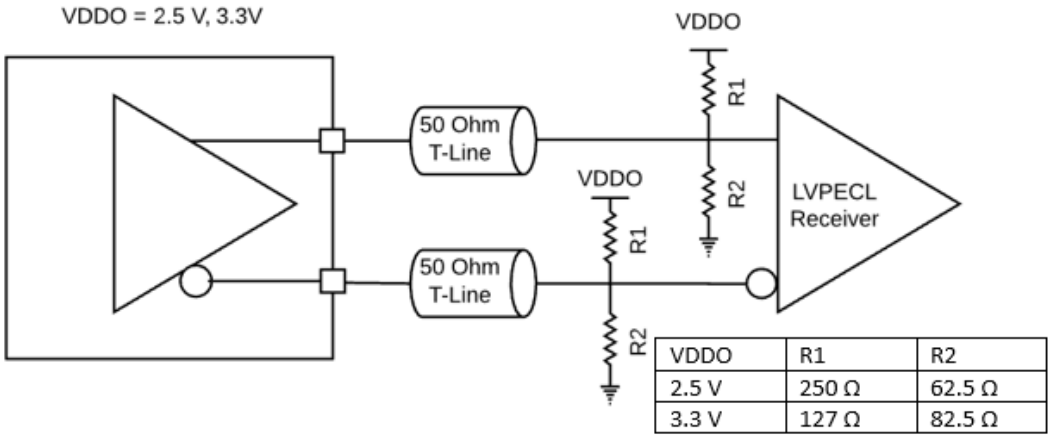
Differential Output Termination

LVPECL



Spec	Conditions	Min	Typical	Max
Output Differential Peak	Measured at 156.25M Output	450 mV	750 mV	

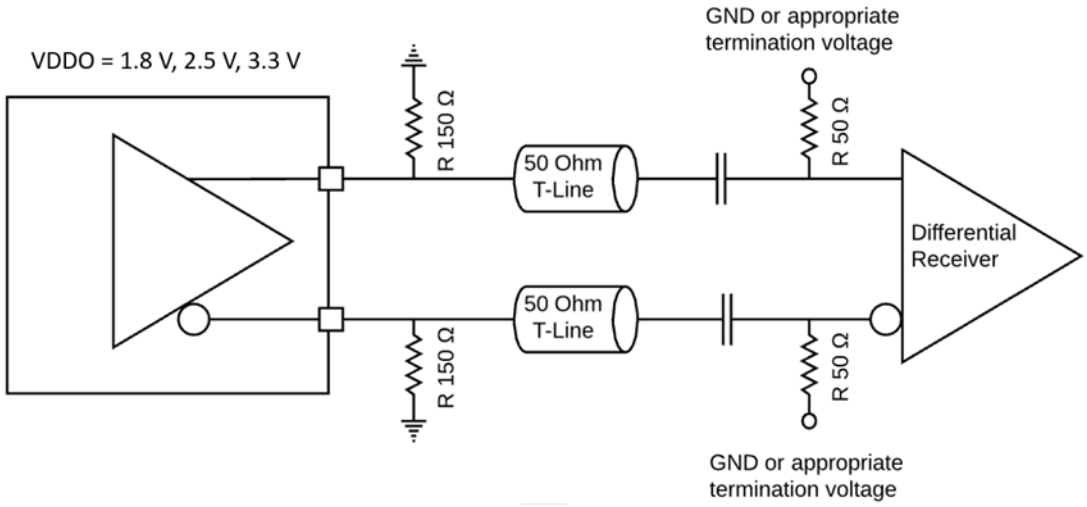
Figure 23. LVPECL with DC-coupled Termination



Spec	Conditions	Min	Typical	Max
Output Differential Peak	Measured at 156.25M Output	450 mV	750 mV	

Figure 24. LVPECL DC-coupled Load Termination with Thevenin Equivalent Network

LVPECL (continued)

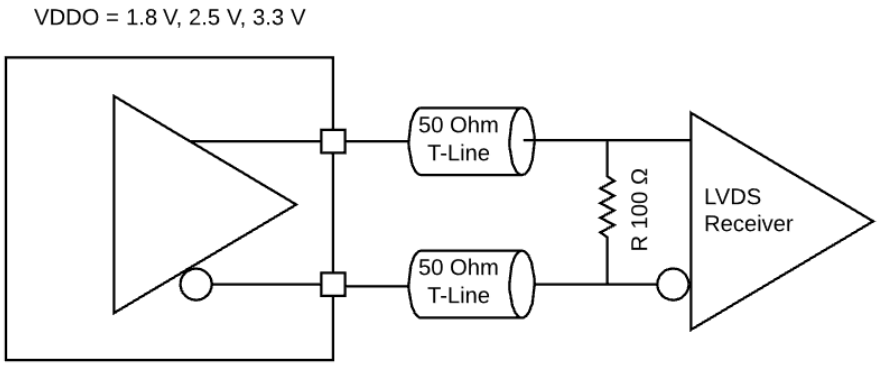


Spec	Conditions	Min	Typical	Max
Output Differential Peak	Measured at 156.25M Output	335 mV	525 mV	

Figure 25. LVPECL2 for the SiT95141 Output Driver with AC Termination at the Load

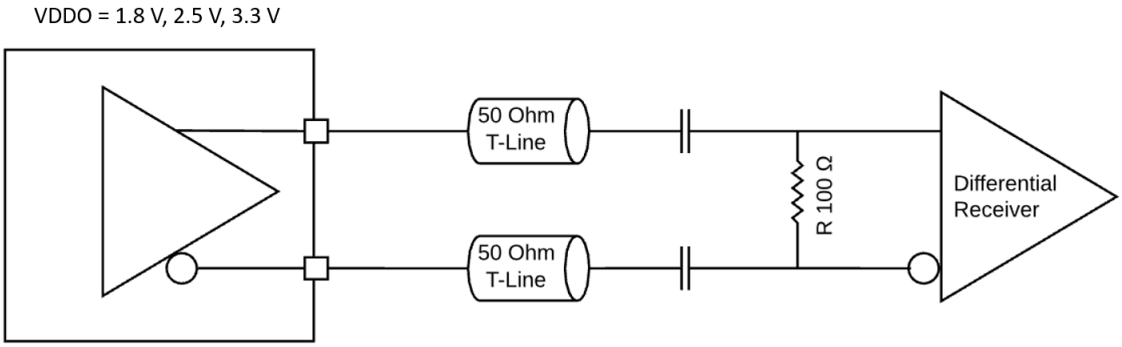
Differential Output Termination Information (continued)

LVDS



Spec	Conditions	Min	Typical	Max
Output Differential Peak	Measured at 156.25M Output	247 mV	350 mV	474 mV

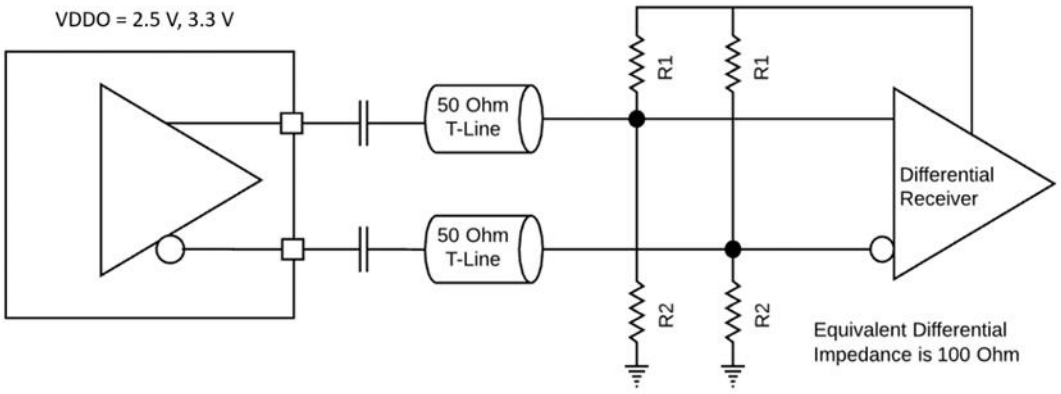
Figure 26. LVDS DC Termination at the Load



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

Figure 27. LVDS AC Termination at the Load

LVDS (continued)



Spec	Conditions	Min	Typical	Max
Output Differential Peak	Measured at 156.25M Output	500 mV	700 mV	

Figure 28. LVDS Boost Standard for the SiT95141 Output Driver with AC Termination at the Load

Differential Output Termination Information (continued)

HCSL

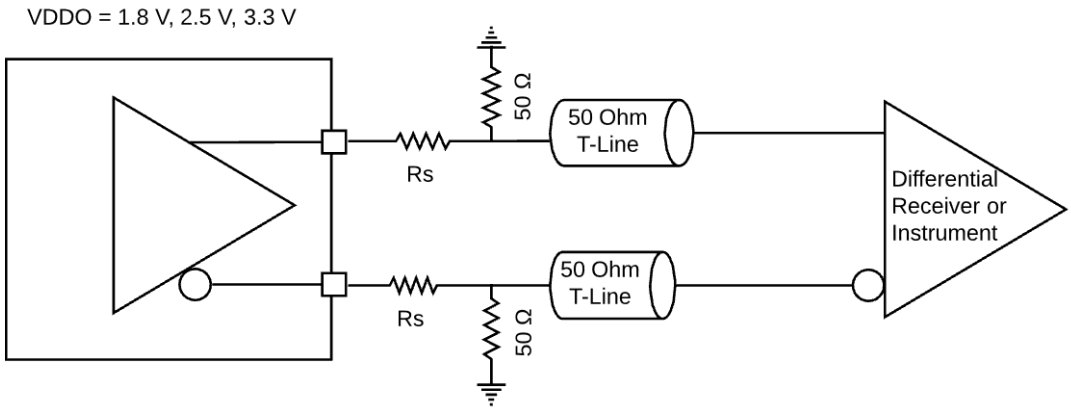
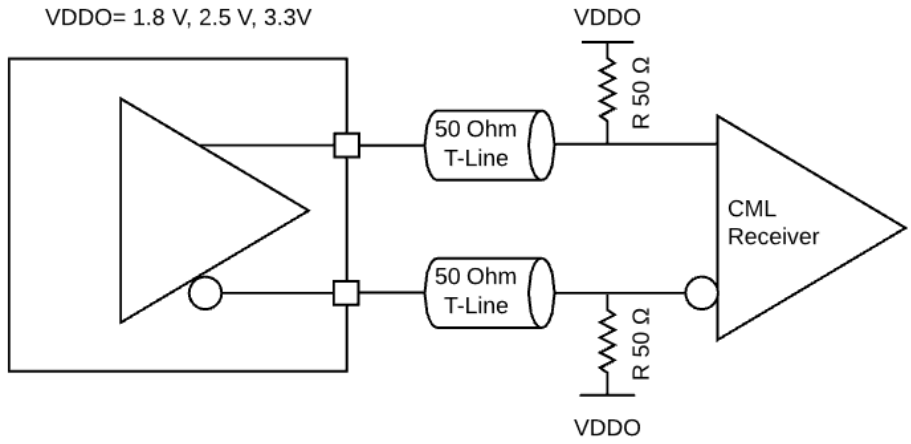


Figure 29. HCSL Interface Termination[42]

Note:

44. R_S 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 30. CML Interface Termination

LVCMOS

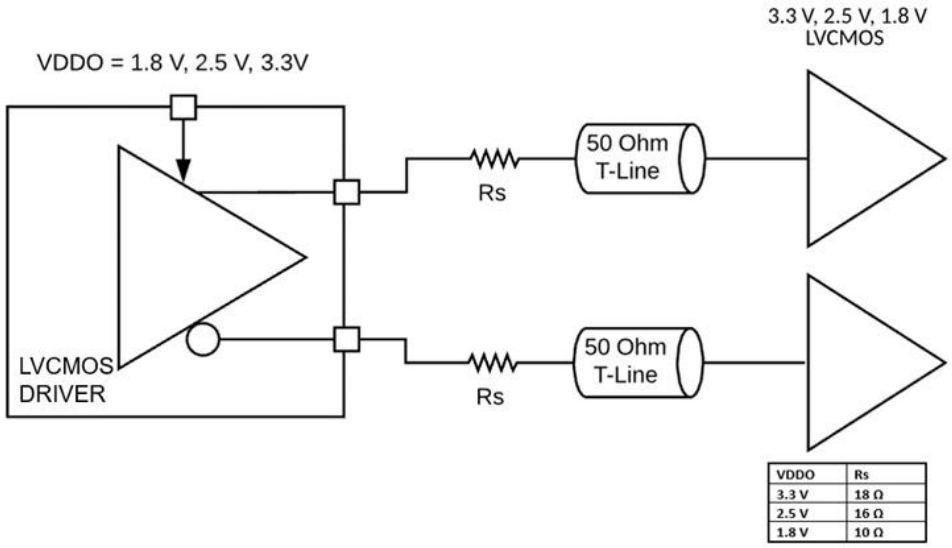


Figure 31. LVCMOS output Termination

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 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.

PLL Description

All settings with respect to each PLLx ($x \in \{A, B, C, D\}$) are accessible on the respective Page {A, B, C, D}. There are two distinct modes of operation of the PLL: free run mode and synchronized mode. The frequency of the high frequency VCO in the PLL is determined by the specific mode of operation.

The VCO frequency is then divided down to get the output frequency on the ODR as described with relation to the overall hierarchy of clocks described along with Figure 8 earlier.

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*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 of the input clocks is not available. 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*finx. In case the input clock is lost, the PLL locks to the highest priority spare clock available.

Synchronized mode

In synchronized mode, the PLL is also 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 the Output Clock Characteristics table) 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.

Manual Input Switching

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.

Zero Delay Mode

Zero Delay Mode is used when 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 30. This ensures minimum delay between the input and output. It can be used for one of the four PLLs at any time. A zero delay mode is available and can be configured for any one of the PLLs in the chip. This provides the option to close the feedback loop of the PLL on the PCB and therefore bypasses the internal feedback dividers, thereby canceling the 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.

MEMS Oscillator Reference

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

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 (FUP) or decrease (FDN) command can be given on the PIF of the PLL, in order 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.

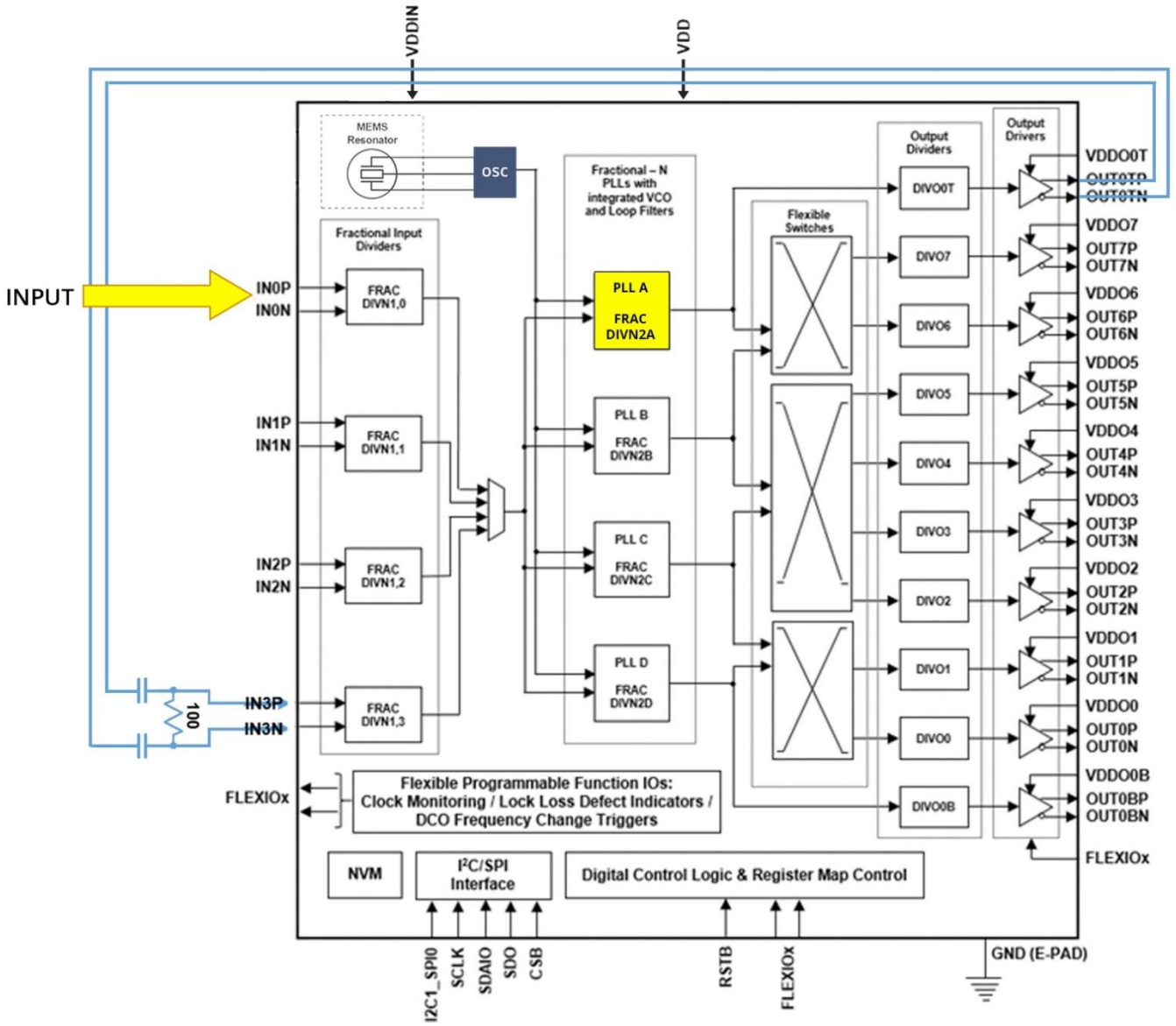


Figure 32. SiT95141 Zero Delay Mode Setup

Programmable Interface Top Level View

Table 20. PIF Overview (Top Level Summary of the Programmable Interface)[43]

Page Number	Function	Comments	Fuse repeated twice
0h	Generic	**2Fh[7:6]: Lock Pattern for the Fuse** **FFh[7:0]: Current Page Number** **22h[7:0]: Current Fuse Pointer** 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 Padring 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	NO
1h	Clock Monitor	**2Fh[7:6] : Customer - Lock Pattern for the Fuse** **FFh[7:0]: Current Page Number** 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	YES
2h	Input	**2Fh[7:6] : Customer - Lock Pattern for the Fuse** **FFh[7:0]: Current Page Number** 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)	YES
3h	8 Flexi-Outputs/ 4 Fixed-Output Blocks	**2Fh[7:6] : Customer - Lock Pattern for the Fuse** **FFh[7:0]: Current Page Number** 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) 58h - 5Fh: Output Block 1T Fuse Configuration (ODR, DIVO, DIVO- Delay) 60h - 67h: Output Block 0B Fuse Configuration (ODR, DIVO, DIVO- Delay) 68h - 6Fh: Output Block 1B Fuse Configuration (ODR, DIVO, DIVO- Delay)	YES

Table 19. PIF Overview (Top Level Summary of the Programmable Interface)[43] (continued)

Page Number	Function	Comments	Fuse repeated twice
Ah	PLL A	**2Fh[7:6] : Customer - Lock Pattern for the Fuse** FFh[7:0]: Current Page Number 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	YES

Note:

45. 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 31 for the I²C addressing scheme of this device.

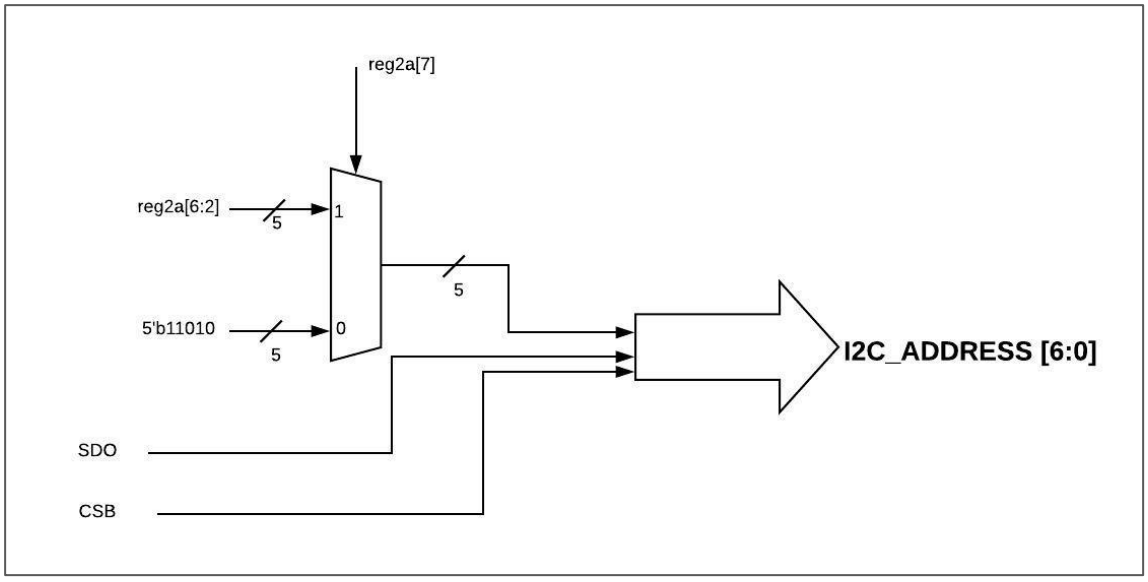


Figure 33. I²C Addressing Scheme for Cascade

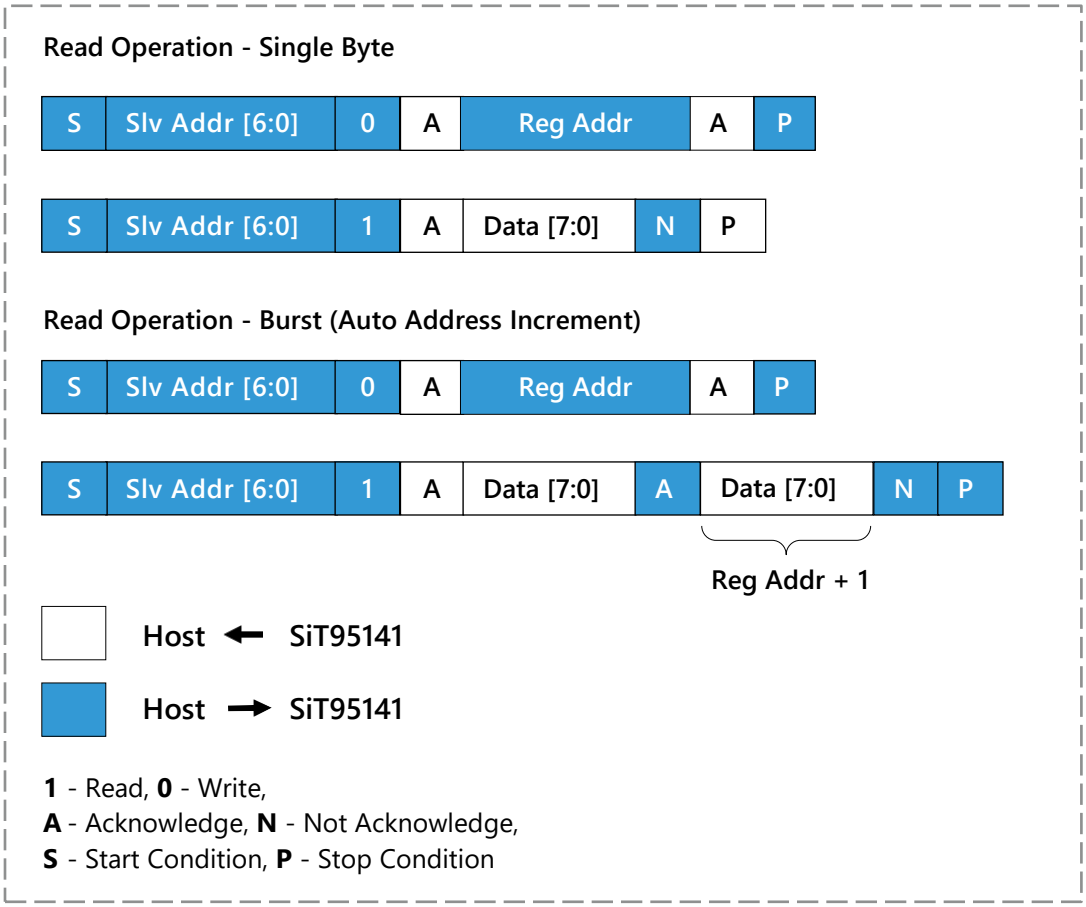


Figure 34. Read Block

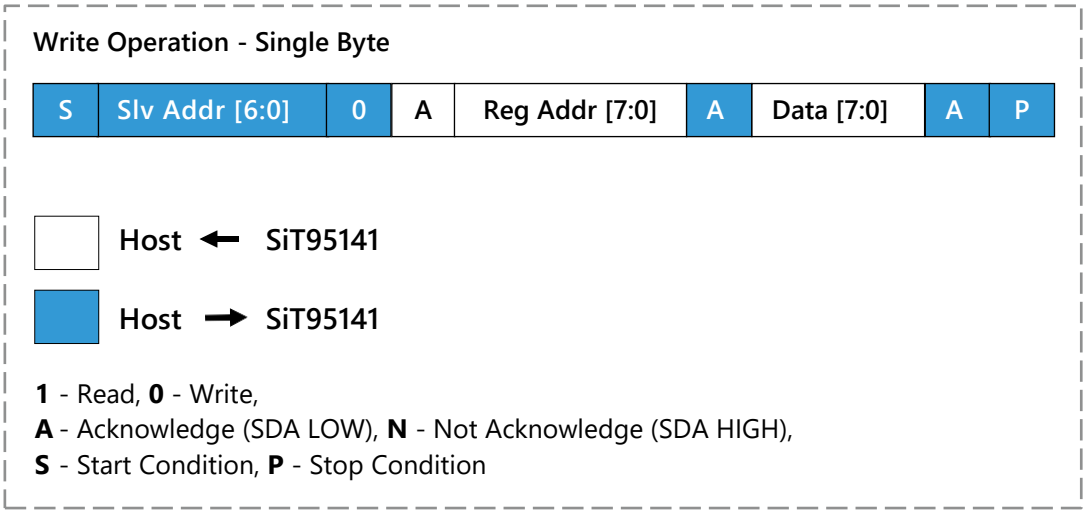


Figure 35. Write Block

- **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 34.

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 SiT95141, 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 SiT95141 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 SiT95141 captures the same on the rising edge of the SPI clock. There is no loopback provided here.
6. In SiT95141 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 SiT95141 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 SiT95141 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 SiT95141 does not support burst address write or read operations.

SPI Timing Details

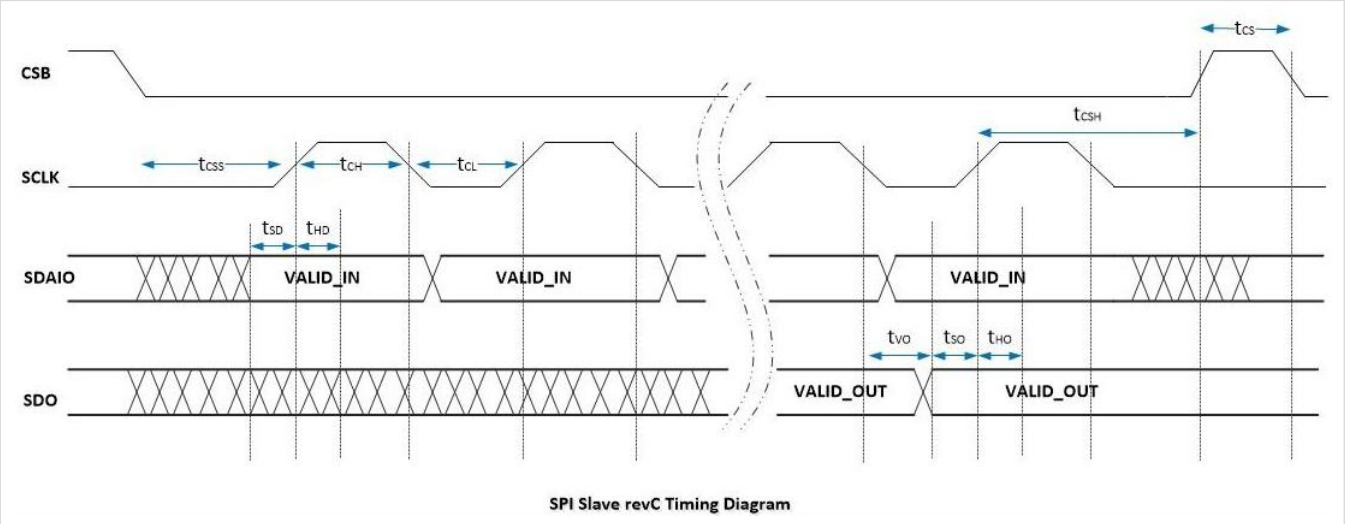


Figure 36. SPI Timing Diagram

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 the 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.

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

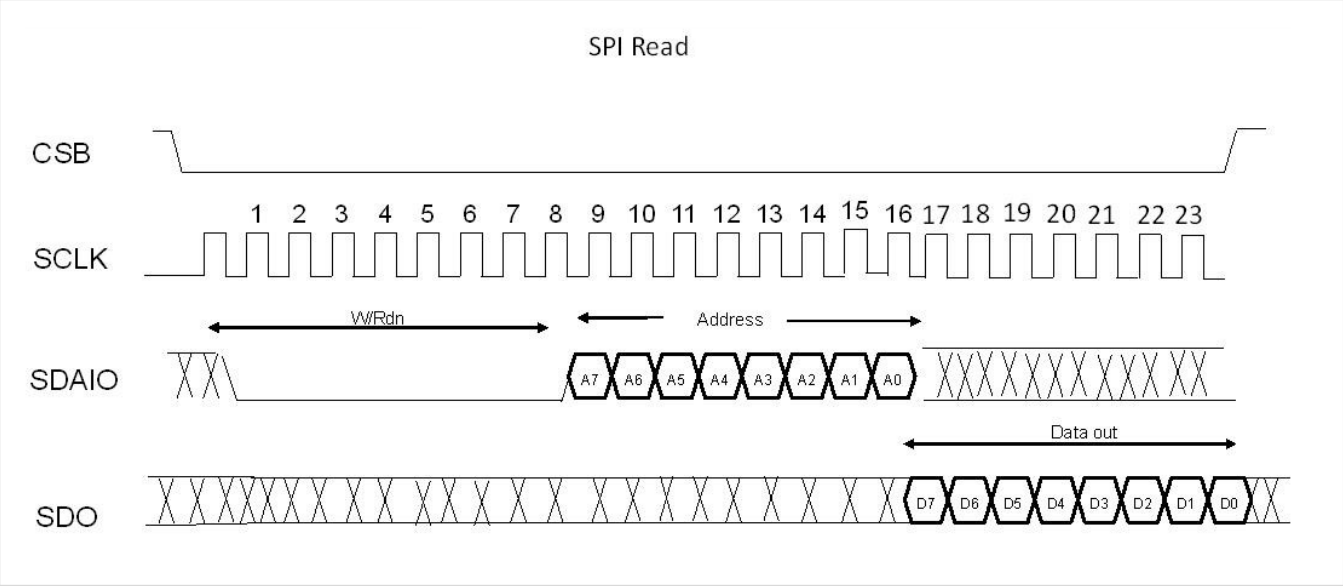


Figure 37. SPI Read Operation

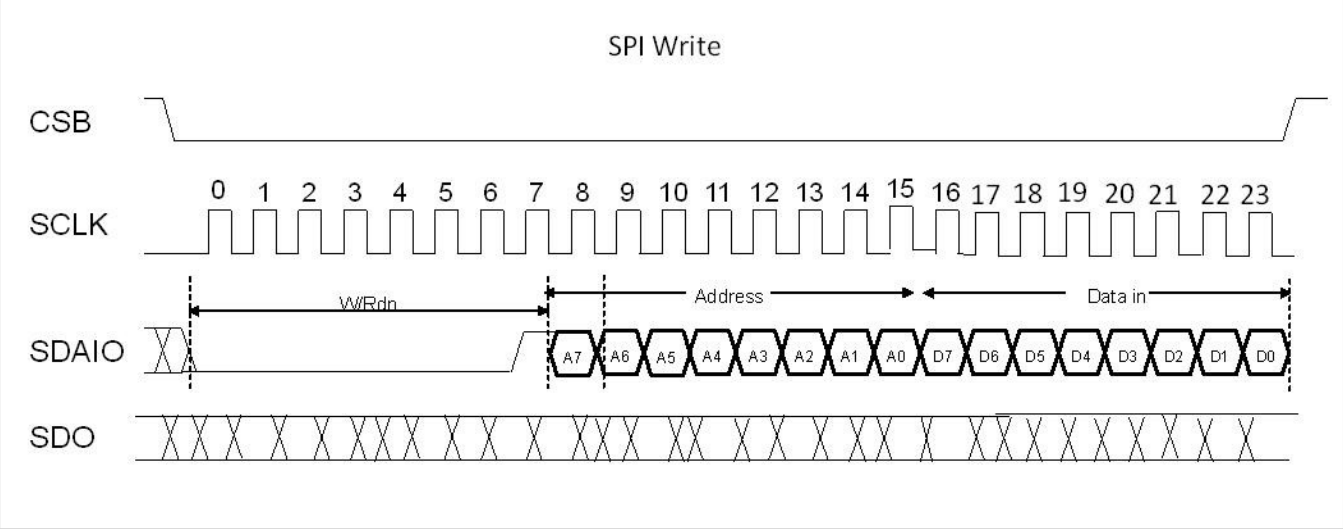


Figure 38. SPI Write Operation

Alarm Registers' Information

Status and Notify

SiT95141 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 21 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 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. Alarm Register Listing

S.NO	Name of signal	Description	Page Number	Register Address	Bit Number
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
11	plla_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x01 to enable selectively, 0xff to enable all notifies in 0x07)	00	0x07	0
12	pllb_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x02 to enable enable selectively, 0xff to enable all notifies in 0x07)	00	0x07	1
13	pllc_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x04 to enable enable selectively, 0xff to enable all notifies in 0x07)	00	0x07	2
14	plld_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x08 to enable enable selectively, 0xff to enable all notifies in 0x07)	00	0x07	3
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
27	In0_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x01 to enable selectively, 0xff to enable all notifies in 0x03)	01	0x03	0
28	In1_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x02 to enable selectively, 0xff to enable all notifies in 0x03)	01	0x03	1
29	In2_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x04 to enable selectively, 0xff to enable all notifies in 0x03)	01	0x03	2
30	In3_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x08 to enable selectively, 0xff to enable all notifies in 0x03)	01	0x03	3

Examples for Live Status Read Back

Some examples are presented based on the Table 21 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 Clock Loss Related Real Time live status read back example:

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]

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_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)

    '''
    This is the main clear function

    '''    clr_intb_CL()
```

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 SiT95141 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.

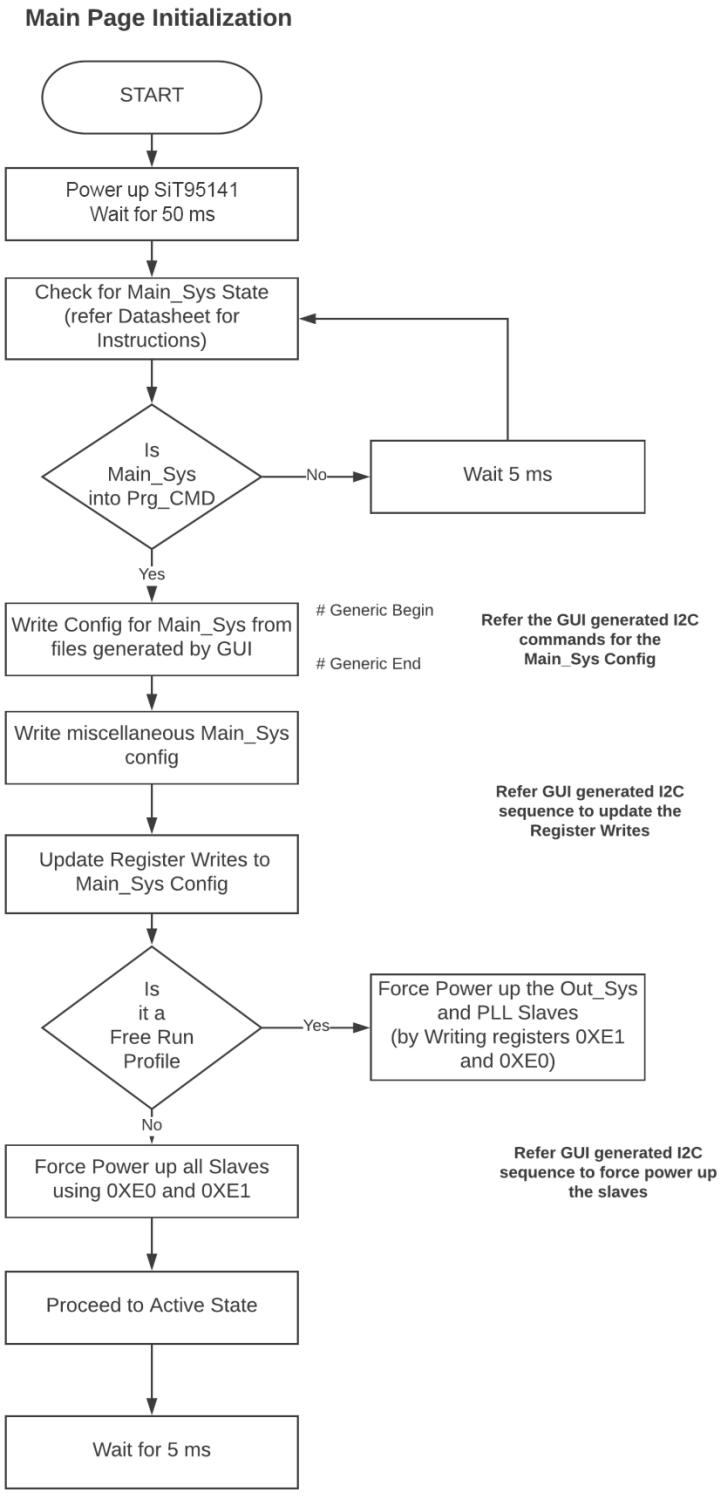


Figure 39. Step 1: Initialize the Main Page – Page 0

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.

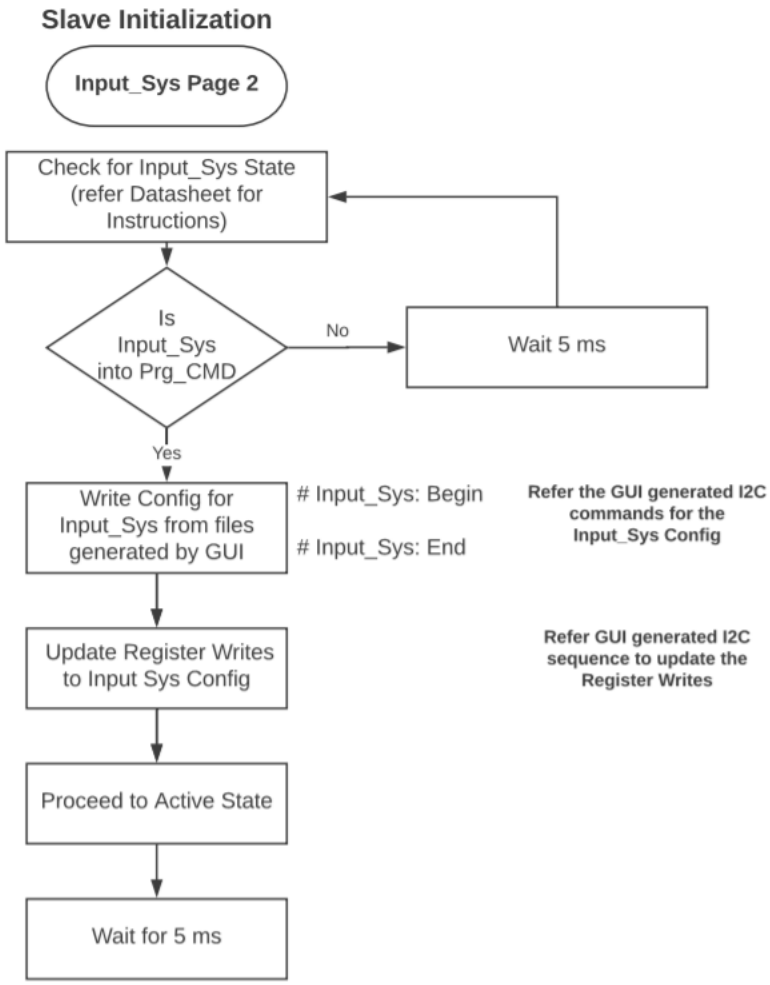


Figure 40. STEP 2: Initialize the Input System Page – Page 2

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.

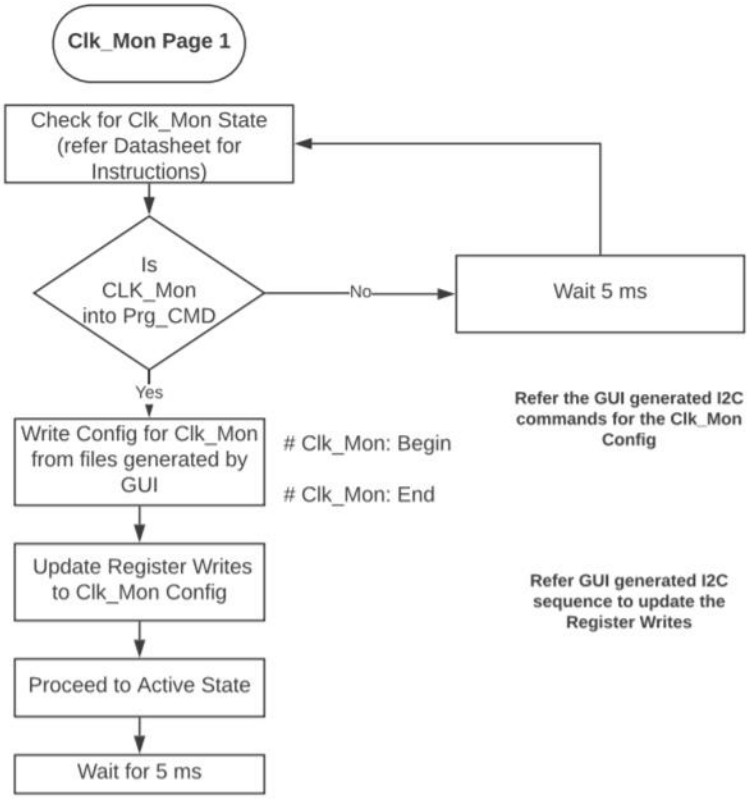


Figure 41. STEP 3: Initialize the Clock Monitor System Page – Page 1

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.

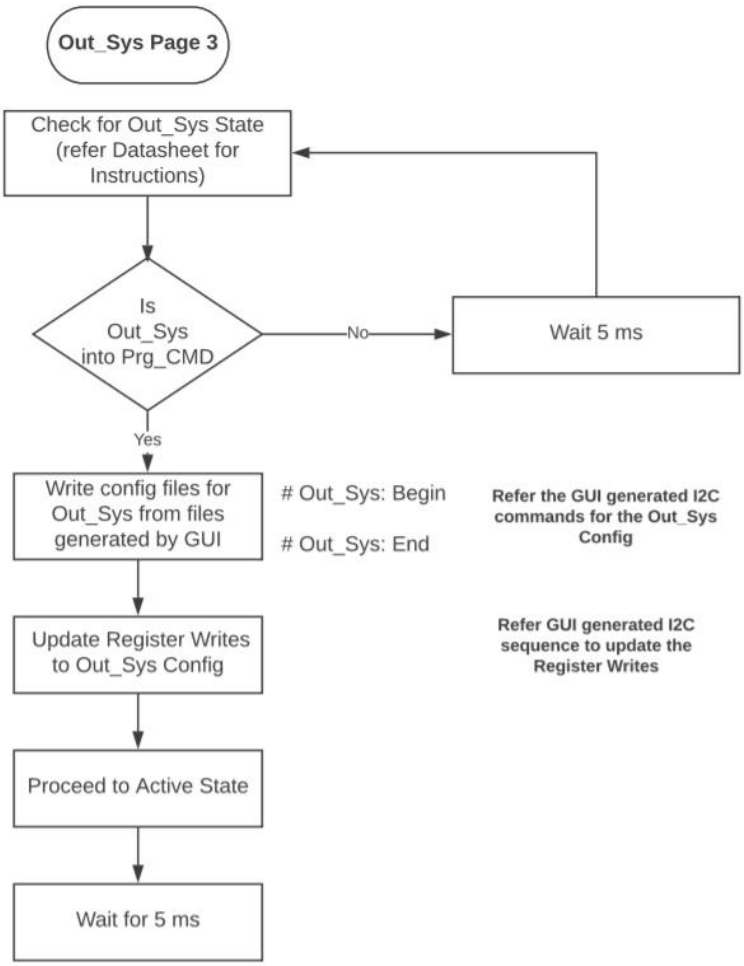


Figure 42. STEP 4: Initialize the Output System Page – Page 3

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.

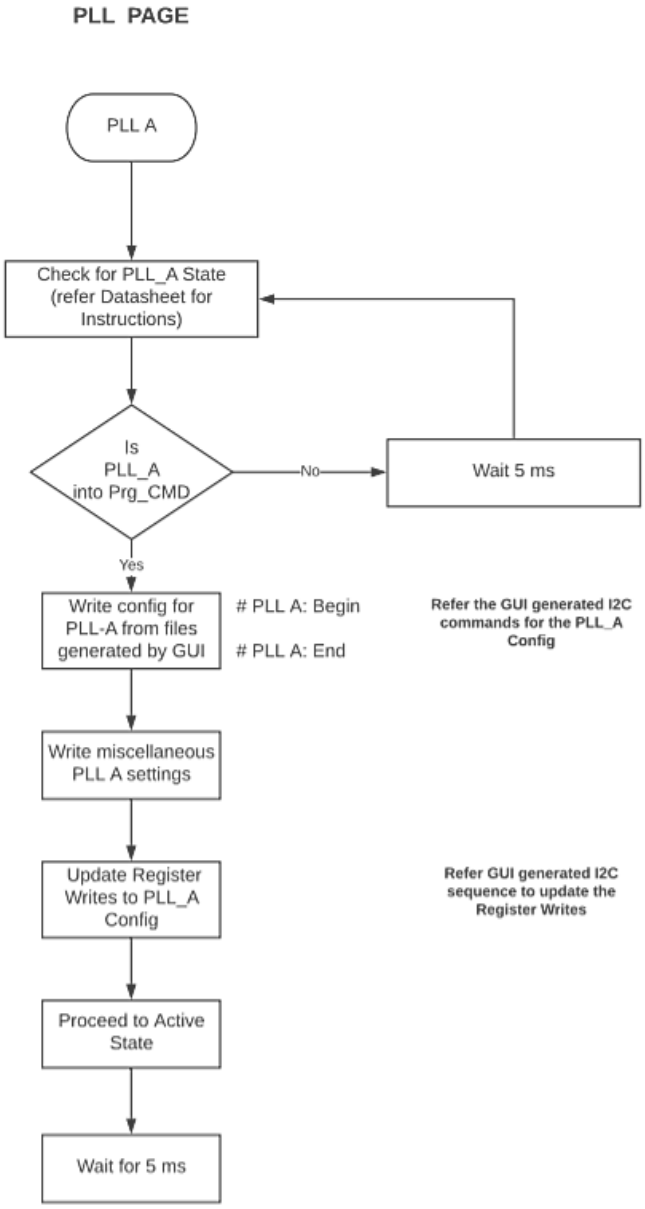


Figure 43. STEP 5: Initialize the PLL A System Page – Page A

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.

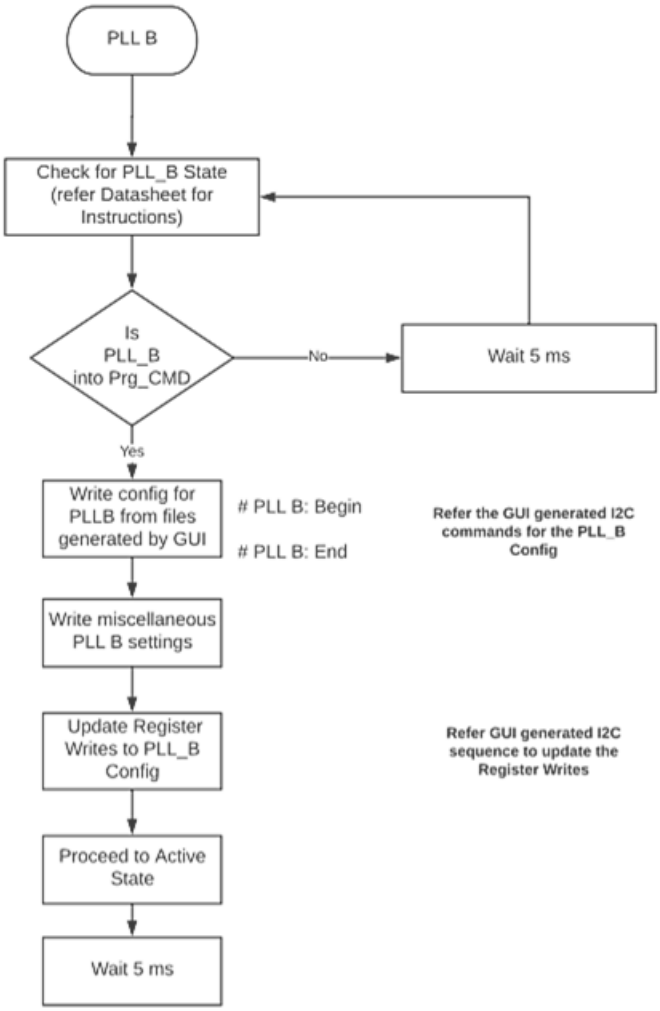


Figure 44. STEP 6: Initialize the PLL B System Page – Page B

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.

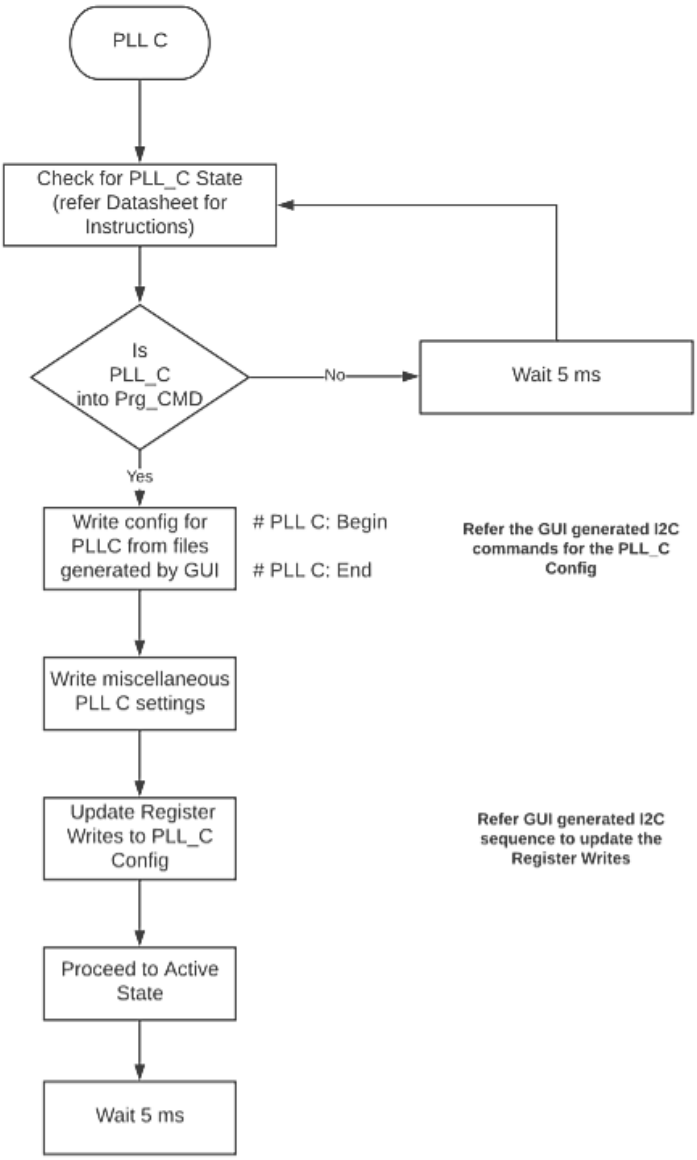


Figure 45. STEP 7: Initialize the PLL C System Page – Page C

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.

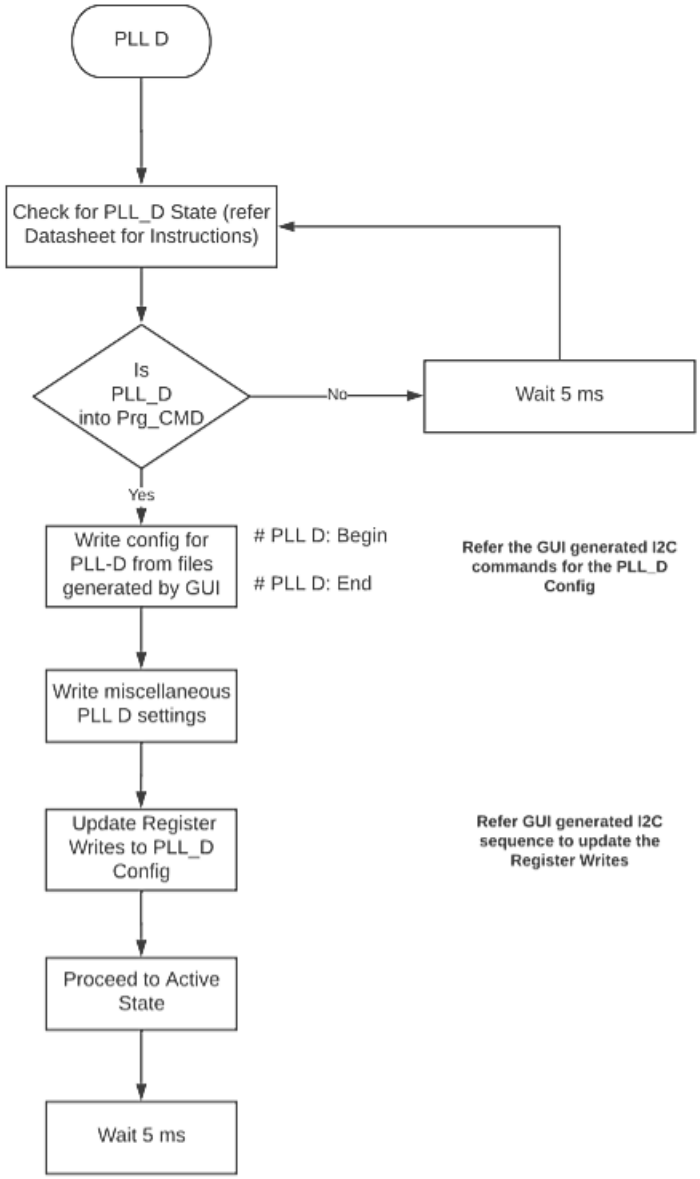


Figure 46. STEP 8: Initialize the PLL D System Page – Page D

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.

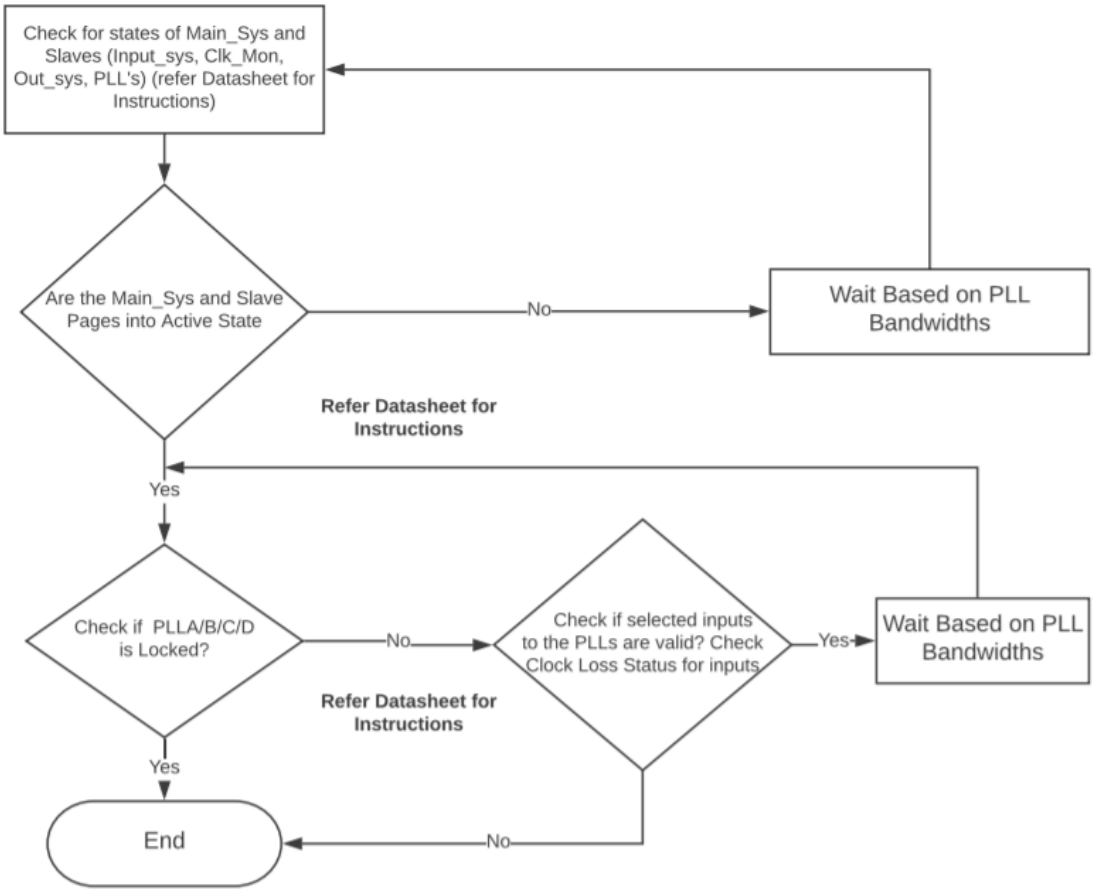


Figure 47 . Active State Confirmation

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 SiT95141 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 in to 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)

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 = 2.5 V and VDDIN = 2.5 V 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 22.

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 value1'h1PLLD_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 23.

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 24.

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 25.

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)
```

PLLC_SYS

STEP 20: Write the PLLC_SYS page number configuration

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

STEP 21: Refer E-Fuse Lock Configuration Bits

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

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

PLLD_SYS

STEP 23: Write the PLLD_SYS page number configuration

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

STEP 24: Refer E-Fuse Lock Configuration Bits

Write the PLLD_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLD_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 PLLD_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 26.

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 27.

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 28.

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)
```

PLLC_SYS

STEP 23: Write the PLLC_SYS page number configuration

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

STEP 24: Refer E-Fuse Lock Configuration Bits

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

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

PLLD_SYS

STEP 26: Write the PLLD_SYS page number configuration

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

STEP 27: Refer E-Fuse Lock Configuration Bits

Write the PLLD_SYS NVM Registers configuration and write 0x2F register for bit number [7:6] with 2'h1 to lock E-Fuse of PLLD_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 PLLD_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 (SiT95141)	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	Dynamic status for rccal_done
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	7:4			Reserved
		3	R	00h	Dynamic status for plld_loss_of_lock
		2	R	00h	Dynamic status for pllc_loss_of_lock
		1	R	00h	Dynamic status for pllb_loss_of_lock
		0	R	00h	Dynamic status for plla_loss_of_lock
NOTIFY_2_GENERIC	07h	7:4			Reserved

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		3	R/W	1h	Sticky/Notify status for plld_loss_of_lock
		2	R/W	1h	Sticky/Notify status for pllc_loss_of_lock
		1	R/W	1h	Sticky/Notify status for pllb_loss_of_lock
		0	R/W	1h	Sticky/Notify status for plla_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
NVMPLLEN_GENERIC	10h	7	R/W	0h	Selects between maual_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)}

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
NVMPLLEN_GENERIC	10h	3:0	R/W	0h	Bits {3,2,1,0} correspond to enable of PLL {D,C,B,A}: Active Low.
NVMFLEXIO8_GEffNERIC	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 clkin0/1/2/3/ Loss of lock for PLL A/B/C/D 3'b010 : Ored_all_pll_ntfy (Loss of lock for PLL A/B/C/D 3'b011 : Ored_all_clkmon_ntfy ((clock loss for clkin0/1/2/3 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 clkin0/1/2/3 + Loss of lock for PLL A/B/C/D3'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 clkin0/1/2/3 3'b100 : Dynamic_Clock_Monitoring_Status 3'b101 : Internal debug 3'b110 : Internal debug 3'b111 : Internal debug
NVMSPARE1_GENERIC	19h	7	R/W	0h	if set '1', Enables vdd padring functionality(0x23[7]) for padring selection. Keeping at 0 enables pad rail switch, select 1 to ensure no switch and keep rail to left
		6	R/W	0h	Reserved
		5:4	R/W	0h	00: DCO Function, 01: DIVO Large Trigger Change, 10: DIVO Small Trigger Change
		3:0	R/W	0h	One hot PLL Selection {3, 2, 1, 0} correspond to {D, C, B, A}
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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
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	Reserved
		6	R/W	0h	Reserved
		5	R/W	0h	Reserved
		4	R/W	0h	Reserved
		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
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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		5	R/W	0h	If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLC
		4	R/W	0h	Reserved
		3	R/W	0h	Reserved
		2	R/W	0h	Reserved
		1	R/W	0h	Reserved
		0	R/W	0h	Reserved
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	010: SiT95141
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 synchronization 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
		6	R	00h	Dynamic status for clk_in2_loss
		5	R	00h	Dynamic status for clk_in1_loss
		4	R	00h	Dynamic status for clk_in0_loss
		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	Reserved
		6	R/W	1h	Reserved
		5	R/W	1h	Reserved
		4	R/W	1h	Reserved
		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]

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		2	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[2]
		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]
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]
		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]

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		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
		2:0	R/W	0h	IN2 19-bit Threshold Value

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
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	
FXOBYFIN2_LOG2_BAND	2Fh	7:6	R/W	0h	PATTERN :{'01'/'10'} => Efuse Locked, {'00'/'11'} => Efuse NOT Locked
		5:0	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

Table 32. PAGE 2: Input System Related Registers.

Registers from 10h to 4Fh are equivalent NVMCopy Registers for this Page

Reg 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	Reserved
		5:0	R/W	0h	IN0 DIVN1 Divider Integer Value
CLKIN0_DIVN1_FRACN1_INPAGE	12h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Numerator
CLKIN0_DIVN1_FRACN2_INPAGE	13h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Numerator
CLKIN0_DIVN1_FRACN3_INPAGE	14h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Numerator
CLKIN0_DIVN1_FRACN4_INPAGE	15h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Numerator
CLKIN0_DIVN1_FRACD1_INPAGE	16h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Denominator
CLKIN0_DIVN1_FRACD2_INPAGE	17h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Denominator
CLKIN0_DIVN1_FRACD3_INPAGE	18h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Denominator
CLKIN0_DIVN1_FRACD4_INPAGE	19h	7:0	R/W	0h	IN0 DIVN1 Divider Fractional Value Denominator
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_RAMPFD_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

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
CLKIN1_DIVN1_FRACN1_INPAGE	22h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Numerator
CLKIN1_DIVN1_FRACN2_INPAGE	23h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Numerator
CLKIN1_DIVN1_FRACN3_INPAGE	24h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Numerator
CLKIN1_DIVN1_FRACN4_INPAGE	25h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Numerator
CLKIN1_DIVN1_FRACD1_INPAGE	26h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Denominator
CLKIN1_DIVN1_FRACD2_INPAGE	27h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Denominator
CLKIN1_DIVN1_FRACD3_INPAGE	28h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Denominator
CLKIN1_DIVN1_FRACD4_INPAGE	29h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Denominator
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_RAMPFD_MEAS_COUNT	2Fh	7:6	R/W	0h	PATTERN :{'01'/'10'} =] Efuse Locked , {'00'/'11'} =] Efuse NOT Locked
		4:0	R/W	0h	Reserved
CLKIN2_DIVN1_INT1_INPAGE	30h	7:0	R/W	0h	IN2 DIVN1 Divider Integer Value
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_DIVN1_FRACN1_INPAGE	32h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACN2_INPAGE	33h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACN3_INPAGE	34h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACN4_INPAGE	35h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACD1_INPAGE	36h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Denominator
CLKIN2_DIVN1_FRACD2_INPAGE	37h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Denominator
CLKIN2_DIVN1_FRACD3_INPAGE	38h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Denominator
CLKIN2_DIVN1_FRACD4_INPAGE	39h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Denominator
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

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
CLKIN2_OUTSEL_INPAGE	3Eh	7:4	R/W	0h	Allows multiple trigger addressing within a single window of FD Ramp drift monitor for selected PLL [3:0] - {PLLA,PLLB, PLLC, PLLD}
		3:0	R/W	0h	Reserved
		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_DIVN1_FRACN1_INPAGE	42h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Numerator
CLKIN3_DIVN1_FRACN2_INPAGE	43h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Numerator
CLKIN3_DIVN1_FRACN3_INPAGE	44h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Numerator
CLKIN3_DIVN1_FRACN4_INPAGE	45h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Numerator
CLKIN3_DIVN1_FRACD1_INPAGE	46h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Denominator
CLKIN3_DIVN1_FRACD2_INPAGE	47h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Denominator
CLKIN3_DIVN1_FRACD3_INPAGE	48h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Denominator
CLKIN3_DIVN1_FRACD4_INPAGE	49h	7:0	R/W	0h	IN3 DIVN1 Divider Fractional Value Denominator
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_OUTSEL_INPAGE	4Eh	7:6	R/W	0h	SPARE
		5:4	R/W	0h	SPARE
		3:0	R/W	0h	Reserved
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 10h 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
DIVO0_DIV2_OUTPAGE	10h				
		3:0	R/W	0h	DIVO Divider for this output: Bits [19:16]
DIVO0_DIV1_OUTPAGE	11h	07:00	R/W	0h	DIVO Divider for this output: Bits [15:8]
DIVO0_DIV0_OUTPAGE	12h	07:00	R/W	0h	DIVO Divider for this output: Bits [7:0]
DIVO0_PROG1_OUTPAGE	13h	07:06	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
		02:00	R/W	0h	Single Ended Driver Programming of strength: Use 0b111= 0d7
DIVO0_PROG0_OUTPAGE	14h	07:00	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
DIVO0_MISC2_OUTPAGE	15h	04:03	R/W	0h	VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output.
		02:00	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)

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
					0b101: DC Coupled LVPECL2 (without Common mode current)
DIVO0_MISC0_OUTPAGE	17h	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
DIVO1_DIV2_OUTPAGE	18h				
		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
					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)

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
					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)
DIVO1_MISC0_OUTPAGE	1Fh	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
DIVO2_DIV2_OUTPAGE	20h				
		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
		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)
		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
DIVO3_DIV2_OUTPAGE	28h				
		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
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,

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
					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.
		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
					{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
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
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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
					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
		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
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_MISC2_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.
		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)
DIVO5_MISC0_OUTPAGE	3Fh	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
DIVO6_DIV2_OUTPAGE	40h				
		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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
					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.
		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
					{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
DIVO7_DIV2_OUTPAGE	48h				
		3:00	R/W	0h	DIVO Divider for this output: Bits [19:16]

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
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.
		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
					{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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		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
DIVO0T_DIV2_OUTPAGE	50h				
		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.
		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
					{0, 0, 0, 0} External Termination,

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		3	R/W	0h	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
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.
		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	SPARE
		4	R/W	0h	PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
					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
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

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	Dynamic status for pseq_prg_restart_ntfy_1
		3	R	0h	Dynamic status for pseq_prg_restart_ntfy_0
		2	R	0h	Dynamic status for porb_line_not_ok_defect
		1	R	0h	Dynamic status for fcal_done
		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	Sticky/Notify status for pseq_prg_restart_ntfy_1
		3	R/W	1h	Sticky/Notify status for pseq_prg_restart_ntfy_0
		2	R/W	1h	Sticky/Notify status for porb_line_not_ok_defect
		1	R/W	1h	Sticky/Notify status for fcal_done
		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	Mask bit for NOTIFY_PLLA (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[4]
		3	R/W	1h	Mask bit for NOTIFY_PLLA (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[3]
		2	R/W	1h	Mask bit for NOTIFY_PLLA (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[2]
		1	R/W	1h	Mask bit for NOTIFY_PLLA (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[1]
		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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		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	0h	SPARE
		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	Reserved
		1	R/W	1h	Reserved
		0	R/W	1h	Reserved
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]
Sysref_sel_PLLA	09h	7:6	R/W	0h	Reserved
		5:0	R/W	0h	Selection of Sysref signal based on one hot active high encoding here
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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		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	
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	
CYCLESLIP_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 filer configuration provided by GUI
		4	R/W	0h	PLL loop filer 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
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
DIVN2_INT3_PLLA	1Fh	7	R/W	0h	Reserved
		6	R/W	0h	Force manual selection of ACTIVE clock

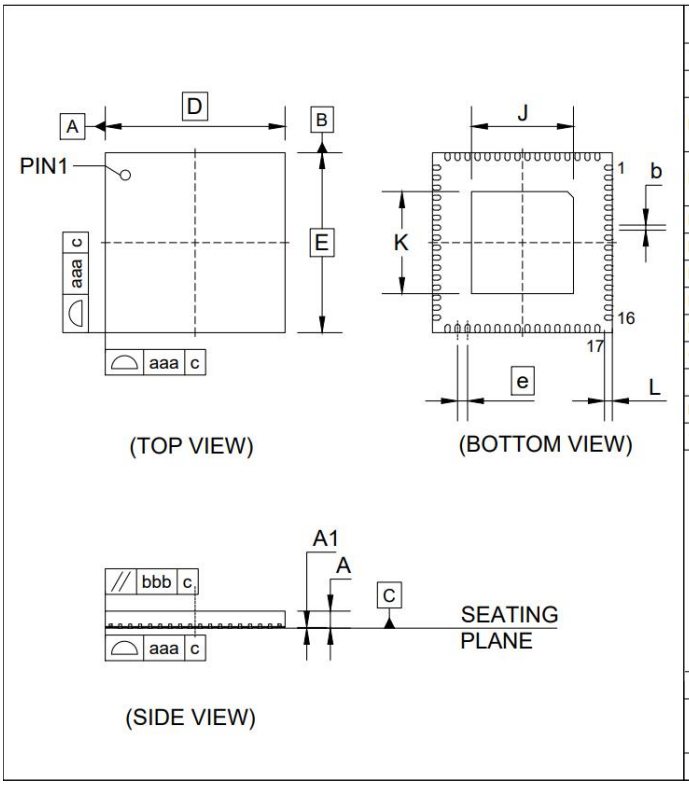
Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		5	R/W	0h	Force Integer mode for the DIVN2 DSM
		4:0	R/W	0h	DIVN2 Integer part computed by GUI
DIVN2_FRN1_PLLA	20h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRN2_PLLA	21h	7:0	R/W	0h	DIVN2 Fractional part computed 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	Reserved
		5:0	R/W	0h	One Hot Output Enable for Outputs 5:0
LL_REG1_PLLA	29h	7:6	R/W	0h	Reserved
		5	R/W	0h	Wait for Input Clock in power up in the PLL wake-up sequence
		4	R/W	0h	Reserved
		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 2 : 2 PPM 3 : 4 PPM 4 : 20 PPM 5 : 40 PPM 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
		4	R/W	0h	

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
		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
HOLDOVER1_PLLA	2Bh	7:5	R/W	0h	Reserved
		4	R/W	0h	Enable Zero Delay Buffer mode with feedback clock routed from the PCB on in3
		3:1	R/W	0h	Reserved
		0	R/W	0h	Enable output clock sync on an independent input coming from IN3
HOLDOVER2_PLLA	2C	7	R/W	0h	Cycle Slip Detector related default
		6	R/W	0h	Reserved
		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	Reserved
		6:3	R/W	0h	Internal Rate Change factors in DLPF computed by GUI
		2:0	R/W	0h	Internal Rate Change factors in DLPF computed by GUI
ONEBYR2_PLLA	2Eh	7:3	R/W	0h	Internal Rate Change factors in DLPF computed by GUI
		2:0	R/W	0h	Internal Rate Change factors in DLPF 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	Reserved
		3	R/W	0h	Reserved
		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

Register Name	Register Number	Bit Range	Access Type	Default Value	Description
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)

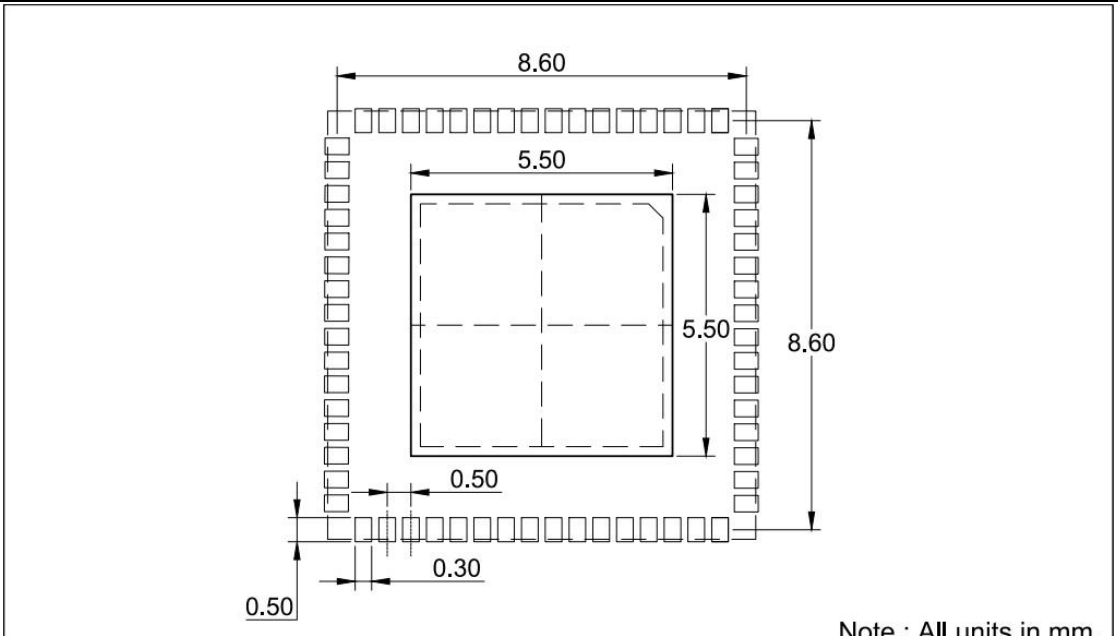


		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.800	0.850	0.900
STAND OFF		A1	0.000	0.035	0.050
BODY SIZE	X	D		9.000 BSC	
	Y	E		9.000 BSC	
EXPOSED PAD	X	J	5.000	5.100	5.200
	Y	K	5.000	5.100	5.200
LEAD WIDTH		b	0.200	0.250	0.300
LEAD LENGTH		L	0.350	0.400	0.450
LEAD PITCH		e		0.500 BSC	
PACKAGE TOLERNCE		aaa		0.100	
MOLD FLATNESS		bbb		0.100	
COPLANARITY		ccc		0.080	

NOTE
1. ALL DIMENSION IN MM

PKG INFO		DRAWING NO.	
64L PQFN 9.000x9.000x0.850 mm		POD-054-PQFN-064-C09090	
		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		REV	SHEET
2020/11/09		A00	01

Table 35. Revision History

Revisions	Release Date	Change Summary
0.1	30 Nov 2018	Preliminary release
0.2	15 Feb 2019	Updated Figure 8 Overall Architecture diagram Updated SPI Operations Figures
0.21	1 Aug 2019	Corrected typos in pin description for OUT pins
0.22	2 Aug 2019	Reorganized OUTx and VDDOx pin labels to make it clearer
0.3	19 Aug 2019	Added Register Map sections
0.31	30 Sep 2019	Fixed typos Added 1ku “Y” tape and reel order code
0.4	23 Mar 2020	I^2C address setting description added Revised Pin numbering and Reg Maps, updated figures
0.9	22 May 2020	Added “X” tape and reel option for 250u Formatting changes Clarified descriptions Added pin pull-up and pull-down resistor values Updated supply current for VDDIN Updated PSNR values Revised registers Added stability specifications
0.91	1 Aug 2020	Formatting changes Updated Ordering Information
0.92	29 Aug 2020	Changed VCROSS specification Changed rev table date format
0.93	13 Oct 2020	Addition of Cascade Platform™ Updated I2C Protocol section
0.94	2 Feb 2021	Formatting and trademarks update
0.95	6 Oct 2021	Added Table 17: 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 values and footnotes Added Voltage Ramp data and diagrams Updated I^2C 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”
2.32	24 Mar 2023	Remove VDDS lines from Table 4
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

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

© SiTime Corporation 2018-2023. 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.