

Description

The SiT95147 is a single-chip MEMS clock jitter attenuator and network synchronizer 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 8 differential or 16 LVCMOS low jitter clock outputs. It 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 SiT95147 integrates SiTime's MEMS resonator and completely eliminates the external discrete crystal reference required by traditional clock generators. This integrated MEMS approach simplifies system design and improves system reliability by eliminating the Clock Generator's dependence on quartz and associated quartz issues:

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

The SiT95147 comes with the companion TimeMaster™ software to simplify clock tree design. The device can be shipped with a user-specified, factory pre-programmed default startup configuration. The device configuration can be re-programmed by the user in one-time-programmable (OTP) memory during system manufacturing or in-circuit via I²C/SPI for additional BOM flexibility.

Features

- Quad Fractional-N PLLs with integrated VCO and loop filter, 125 fs typical RMS integrated phase jitter
- Wide output frequency support
 - Differential outputs from 8 kHz to 2.1 GHz
 - LVCMOS outputs from 8 kHz to 250 MHz
 - Supports 1 Hz 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 VDD Supply
 - LVPECL, CML, HCSL, LVDS or LVCMOS
 - 1.8V, 2.5V or 3.3V
- Flexible input to output frequency translation with jitter attenuation, 4 inputs, 8 outputs
- Programmable jitter attenuation bandwidth for each PLL: 1 mHz to 4 kHz
- Synchronized, holdover or free run operation modes
- Hitless clock inputs with auto or manual switching
- Locks to gapped clock inputs to support OTN
- Programmable Output Delay Control
- DCO mode via I²C or SPI with 0.005 ppb resolution
- 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
- Synchronous Ethernet Clock Synchronization, ITU-T G.8262
- 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

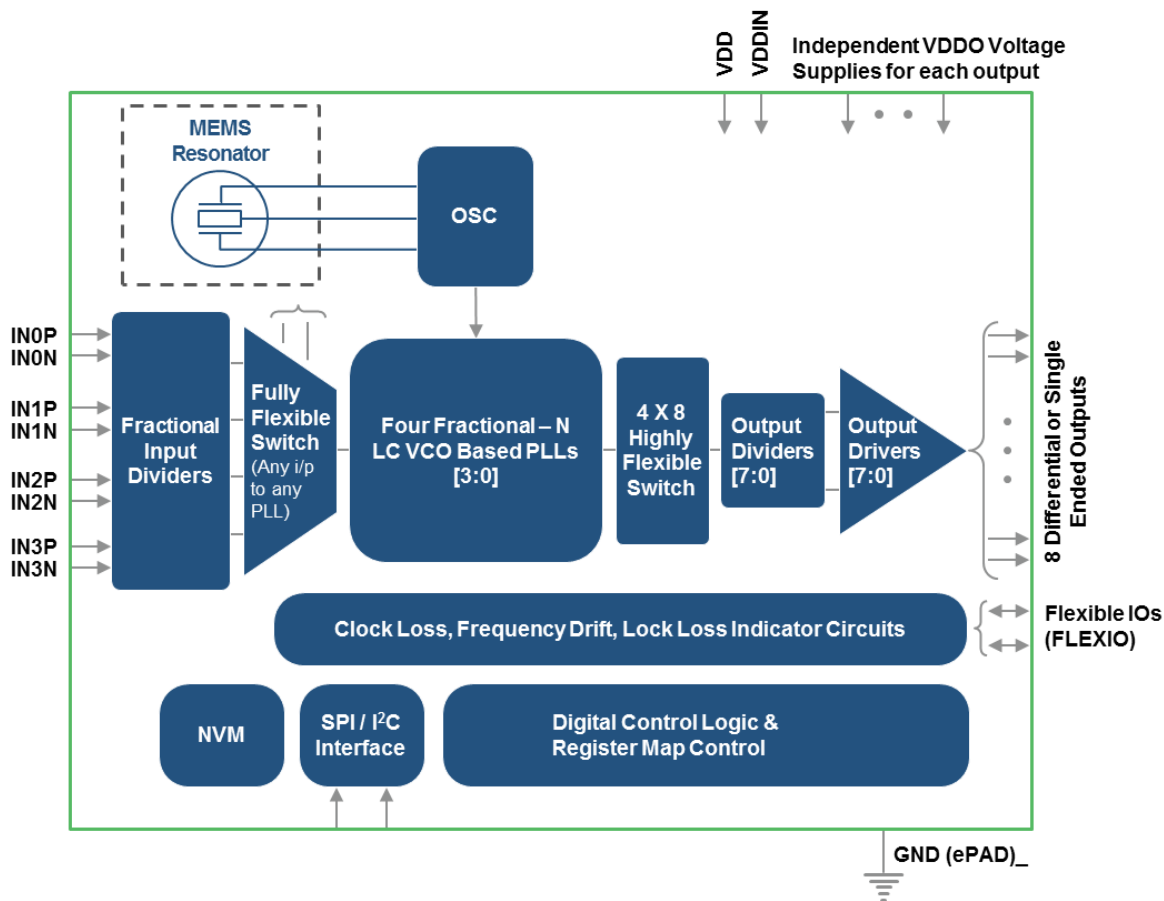


Figure 1. SiT95147 Block Diagram

Table of Contents

Description.....	1
Features.....	1
Applications	1
Electrical Characteristics.....	4
Functional Description	16
Frequency Configuration.....	17
PLL Bandwidth.....	19
Fast Lock	19
Operation Modes	19
Device Start-Up and Initialization	21
Free run mode.....	21
Synchronized mode.....	21
Holdover mode.....	21
DCO Mode operation.....	21
PLL MEMS Oscillator Reference	21
Inputs.....	21
Automatic Input Selection.....	21
Manual Input Selection.....	21
Hitless Input Switching	21
Ramped Input Switching	21
Input Clocks	22
Clock Monitoring.....	25
PLL Lock Loss Defect Monitoring	25
Outputs	25
Differential Output Termination Information	26
Output Delay Control	29
Zero Delay Mode	29
Programmable Interface (PIF) Top Level View	30
Serial Programming Interface Description	32
I ² C protocol.....	32
SPI Protocol	34
SPI Timing Details.....	35
SPI Single byte write	36
SPI Single byte read.....	36
Alarm Registers' Information.....	38
Status and Notify	38
Examples for Live Status Read Back.....	40
Examples of Sticky Bit Clearing	41
Device Initialization for a non-programmed device	42
Monitoring the Status for Master and Slave Pages.....	52
Programming the Primary E-Fuse	54
Configuration Bits to Force Power-up of Digital Slave Subsystems	54
E-Fuse Lock Configuration Bits	55
E-Fuse Write Configuration Bits	55
Configuration Bit to Remove Manual Wake Up for Primary E-Fuse	56
Pseudo Code: Programming the Primary E-Fuse	56
Programming the Secondary E-Fuse.....	60
Configuration Bit to Escape to PROGRAM_CMD State in GENERIC_SYS	60
Configuration Bit to Change the E-Fuse pointer.....	60
Configuration Bit to Enable Manual Wake Up for Secondary E-Fuse	60
Pseudo Code: Programming the Secondary E-Fuse.....	61
Register Map Details.....	65
Package Information.....	91
Ordering Information.....	92

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 2](#) for operating temperature.

Table 1. Absolute Maximum Ratings^[1,2]

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		
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 ² C Bus input voltage	V _{IN2C}	-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		+50	°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
ESD (charged device model)	ESD _{CDM}			500	V	JESD22C-101
ESD (machine model)	ESD _{MM}			200	V	JESD22A-115
Latchup	LU			100	mA	JEDEC JESD78D

Notes:

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

Table 2. Operating Temperature

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Ambient temperature	T _A	-40	–	+85	°C	
Junction temperature	T _J			+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 3](#) shows the DC electrical characteristics.

Table 3. DC Electrical Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Supply voltage, Analog Input	V _{DDIN}	2.97	3.3	3.63	V	3.3-V range: ±10%
Supply voltage, PLL	V _{DD}	2.25	-	3.63	V	2.5 to 3.3V range: ±10%
Supply Voltage, Output Drivers	V _{VDDOX}	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 Current, VDDIN	I _{DDIN} ^[3]		18		mA	All Four Inputs assumed to be enabled
Supply Current, VDD	I _{DD}		320	384	mA	All Four PLLs and All 8 Outputs enabled (Maximum current mode)
			170	204	mA	Two PLLs and 4 Outputs enabled
Power supply current, VDDOX	I _{VDDOX} ^[4,5,6,7]		40	50	mA	LVPECL, output pair terminated 50 Ω to V _{TT} (V _{DD} – 2V)
			28	35	mA	LVPECL2, output pair terminated 50 Ω to V _{TT} (V _{DD} – 2V) without common mode current.
Power supply current, VDDOX	I _{VDDOX} ^[4,5,6,7]		20	25	mA	CML, output pair terminated 50 Ω to V _{DD}
Power supply current, VDDOX	I _{VDDOX} ^[4,5,6,7]		27	34	mA	HCSL, output pair with HCSL termination
Power supply current, VDDOX	I _{VDDOX} ^[4,5,6,7]		16	20	mA	LVDS, output pair terminated 100 Ω
Power supply current, VDDOX	I _{VDDOX} ^[4,5,6,7]		15	19	mA	LVC MOS, 250 MHz, 2.5V 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')

Notes:

- VDD and VDDIN are independent supplies that are expected to be at the same voltage level.
- LVPECL and LVDS Boost standards are supported for VDDO = {2.5V, 3.3V}. LVPECL2, HCSL, CML and LVDS standards are supported for VDDO = {1.8V, 2.5V, 3.3V}.
- LVPECL mode provides 6mA of common mode current on each output. LVPECL2 mode does not provide this common mode current.
- A 50 Ω Termination resistor with a DC bias of VDDO – 2V for LVPECL standards is supported for VDDOX = {2.5V, 3.3V}.
- I_{VDDOX} 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.

Table 4 and Table 5 show input clock characteristics and output clock characteristics.

Table 4. Input Clock Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Differential Input Frequency	$F_{IN,DIFF}$	0.008		2100	MHz	Differential clock input
Differential Amplitude Peak ^[8]	$V_{IN,DIFF}$	100			mV	$f_{IN} < 400$ MHz
		225			mV	400 MHz < f_{IN} < 750 MHz
		350			mV	750 MHz < f_{IN} < 2100 MHz
Slew Rate ^[9,10]	SR	400			V/us	
Differential Input Duty Cycle	$DC_{IN,DIFF}$	40		60	%	Measured at crossover point
Differential Input Capacitance	C_{IN}		0.3		pF	
Differential Input Resistance	R_{IN}		15			AC Coupled SE
			10			Differential
Single Ended Input Frequency	$F_{IN,SE}$	0.008		250	MHz	Single ended clock input
LVC MOS Input High Voltage – DC Coupled ^[10]	$V_{IH,SE}$	0.8			V	
LVC MOS Input Low Voltage – DC Coupled ^[10]	$V_{IL,SE}$			0.4	V	
Slew Rate ^[9,10]	SR	400			V/us	
Duty Cycle	DC	40		60	%	
Minimum Pulse Width	PW	1.6			ns	
Input Resistance	R_{IN}		30		k Ω	

Notes:

8. AC Coupled input assumed with series capacitance for differential inputs
9. Resistor termination for differential input followed by series capacitors for each of true and complement differential input connecting to the device pins.
10. LVC MOS single ended is direct coupled on the true input. Connect complement input to ground with a 100 nF capacitor.

Table 5. Serial Input Clock Characteristics

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

Note:

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

Table 6 and Table 7 show output clock characteristics.

Table 6. Output Clock Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Differential Output Frequency	$F_{OUT,DIFF}^{[12]}$	1		2100M	Hz	LVPECL, CML, LVDS outputs
Differential Output Frequency	$F_{OUT,DIFFH}^{[12]}$	1		700M	Hz	HCSL outputs
Single Ended Output Frequency	$F_{OUT,SE}^{[12]}$	1		250M	Hz	LVC MOS outputs
PLL loop bandwidth	F_{BW}	0.001		4000	Hz	Programmable
Jitter Peaking	J_{PEAK}			0.1	dB	Meets SONET Jitter Peaking requirements in closed loop
Time delay before the Historical average for output Frequency is considered	$H_{DELAY}^{[13,14]}$	0.035	0.5	35	s	Programmable in register map
Length of time for which the Average of the frequency is considered	$H_{AVG}^{[13,14]}$	0.07	1	70	s	Programmable in register map
Power Supply to I ² C or SPI interface ready	T_{START}			10	ms	No I ² C transaction valid till 10 ms after all power supplies are ramped to 90% of final value.
With Speed-Up mode enabled	$T_{LOCK}^{[15]}$		300		ms	Speed-up mode is programmable. This is a Typical number. Actual wake up time depends on fast lock and normal BW settings.
DCO Mode Frequency Step Resolution	$F_{RES,DCO}^{[16]}$		0.005		ppb	Frequency Increment or Decrement resolution. This is controlled by accessing the control registers.
Resolution for output delay	$T_{RES}^{[17]}$		35		ps	Programmable per output clock with this resolution for a total delay of ± 7.5 ns
Maximum Phase Hit	$T_{MAX}^{[18]}$	-50		50	ps	Default Hitless Switching Mode (no phase propagation)
Uncertainty in Input to Output Delay	ΔT_{DELAY}	-175		175	ps	Maximum variation in the static delay from input to output clock between repeated power ups of the chip
Pull Range	ω_P		500		ppm	
POR to Serial Interface Ready	T_{RDY}			15	ms	
One free run PLL clock on fuse locked part	$T_{START,Special}$			10	ms	
Power Supply Rejection Ratio ^[19,20]	PSRR _{VDD}		-96		dBc	$F_{OUT} = 156.25$ MHz, $F_{SPUR} = 100$ kHz, BW= 100 Hz PSRR on PLL Supply, VDD = 3.3V
			-90			$F_{OUT} = 156.25$ MHz, $F_{SPUR} = 100$ kHz, BW= 100 Hz PSRR on PLL Supply, VDD = 2.5V
			-75			$F_{OUT} = 156.25$ MHz, $F_{SPUR} = 100$ kHz, BW= 100 Hz PSRR on PLL Supply, VDD = 1.8V
	PSRR _{VDDIN}		-100		dBc	$F_{OUT} = 156.25$ MHz, $F_{SPUR} = 100$ kHz, BW= 100 Hz PSRR on PLL Supply, VDDIN = 3.3V
			-100			$F_{OUT} = 156.25$ MHz, $F_{SPUR} = 100$ kHz, BW= 100 Hz PSRR on PLL Supply, VDDIN = 2.5V
			-100			$F_{OUT} = 156.25$ MHz, $F_{SPUR} = 100$ kHz, BW= 100 Hz PSRR on PLL Supply, VDDIN = 1.8V
PSRR _{VDDO}		-80		dBc	$F_{OUT} = 156.25$ MHz, $F_{SPUR} = 100$ kHz, BW= 100 Hz PSRR on PLL Supply, VDDO = 3.3V	
Output-Output Crosstalk ^[21,22,23]	XTALK		-75		dBc	156.25 MHz and 155.52 MHz on adjacent outputs

Notes:

12. 1 Hz Output available only on output OUT0 (OUT0P, OUT0N).
13. Hitless Switching enables PLL to switch between input clocks when the current clock is lost.
 - a. Clock Loss can be defined as 2 / 4 / 8 / 16 consecutive missing pulses.
 - b. Priority list for the input clocks can be set in the register map independently for each PLL.
 - c. Output is truly hitless (no phase transient and 0 ppb relative error in frequency) for exactly same frequency input clocks that are switched.
 - d. Hitless switching support is both revertive and non-revertive.
 - e. Revertive / Non-revertive Support: Assume Clock Input 0 is lost and switch is made to Clock Input 1. Then, PLL reverts to Clock Input 0 when it becomes valid again in Revertive mode. It does not switch back to Clock Input 0 even when it becomes valid again in the non-Revertive mode.
14. PLL enters holdover mode when the active input clock and all spare clocks in the clock priority list for hitless switching are lost.
 - a. Clock Loss can be defined as 2 / 4 / 8 / 16 consecutive missing pulses.
 - b. Programmable Clock Loss settings ensure Gapped Clocks can be supported by choosing higher number of missing pulses as the trigger for clock being invalid.
 - c. Entering holdover mode is supported with the frequency frozen at a historical average determined from the Hdelay and Havg settings.

15. For low PLL Loop Bandwidths, wake up time can be very large unless the speed up feature is used. The speed up feature provides the user options to use a completely independent loop bandwidth for the wake up transitioning to the regular bandwidth after frequency and phase are locked.
 - a. Fast Lock Bandwidth needs to be less than 100 times smaller than the input clock frequency (divided input at PLL phase detector) for stable and bounded (in time) lock trajectory of the PLL.
16. 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.
17. 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.
18. This test is for 2 inputs at 8M that are switched to get a 622.08M output.
19. The PSRR 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.
20. Output PSRR measured with LVDS standard which (along with the LVDS boost) are the recommended standards for AC Coupled terminations.
21. Measured across adjacent outputs- All adjacent outputs are covered and the typical value for the worst case output to output coupling is reported.
22. The adjacent output pairs are chosen at 155.52 MHz and 156.25 MHz frequencies.
23. This cross talk between outputs is mainly package dependent therefore terminated outputs are used for reporting these numbers ensuring that there is signal current in the bond wires.

Table 7. Output Serial and Status Pin Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Condition
All VDDIO						
Output Voltage	V_{OH}	$V_{DDIO}^{[24]}$ x 0.75	–	–	V	$I_{OH} = -2 \text{ mA}$
	V_{OL}	–	–	$V_{DDIO}^{[24]}$ x 0.25	V	$I_{OL} = 2 \text{ mA}$
All VDDS						
Output Voltage	V_{OH}	V_{DDS} x 0.75	–	–	V	$I_{OH} = -2 \text{ mA}$
	V_{OL}	–	–	V_{DDS} x 0.25	V	$I_{OL} = 2 \text{ mA}$

Note:

24. 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 shows the fault monitoring indicators.

Table 8. Fault Monitoring Indicators

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Clock Loss Indicator Thresholds	$CL_X^{[25,28]}$	2	4	16	Pulses	Clock Loss Indicators can be set on any of the four inputs. Loss of 2 / 4 / 8 / 16 consecutive pulses can be used to indicate a clock loss. Programmable in the register map
Fine Frequency Drift Indicator Thresholds: Step Size	$FD_X^{[26,27,28]}$		± 2		ppm	Frequency drift threshold is programmable in the range with the step size resolution specified. Frequency drift hysteresis is programmable in the range with the step size resolution specified.
Fine Frequency Drift Indicator Thresholds: Hysteresis Range		± 2		± 500	ppm	
Fine Frequency Drift Indicator Thresholds: Range		± 2		± 500	ppm	
Coarse Frequency Drift Indicator Thresholds		± 100		± 1600	ppm	Coarse Drift Indicators programmable from {Up to ± 1600 ppm in steps of ± 100 ppm}
Lock Loss Indicator Threshold	LL	± 0.2		± 4000	ppm	Lock Loss Indicator threshold is programmable in the range specified from the following choices for setting and clearing LL: { $\pm 0.2, \pm 0.4$ } ppm, { $\pm 2, \pm 4$ } ppm, { $\pm 20, \pm 40$ } ppm, { $\pm 200, \pm 400$ } ppm, { $\pm 2000, \pm 4000$ } ppm

Notes:

25. Clock Loss Indicators are used for:
 - a. Hitless Switching Triggers
 - b. Update in Status Registers in the register map
26. Coarse and Fine Frequency Drift indicators can be concurrently enabled. This enables the user to detect fast drifting frequencies since detecting fine drifts will take longer measurements.
27. 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.
28. Clock Loss can be combined with either of the frequency drift monitors (coarse and fine) to trigger the hitless switching event in the PLLs. The trigger for a hitless switching event in the PLL can therefore be either the Clock Loss event or either of Clock Loss or Frequency Drift.

Table 9 shows output RMS jitter in frequency translation modes.

Table 9. 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} ^[29,30]		120		fs rms	F _{OUT} = 622.08 MHz, F _{IN} = 10 MHz, PLL BW = 100 Hz, Single PLL Profile

Note:

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

Figure 2 shows the representative phase noise measurement at various speed levels.

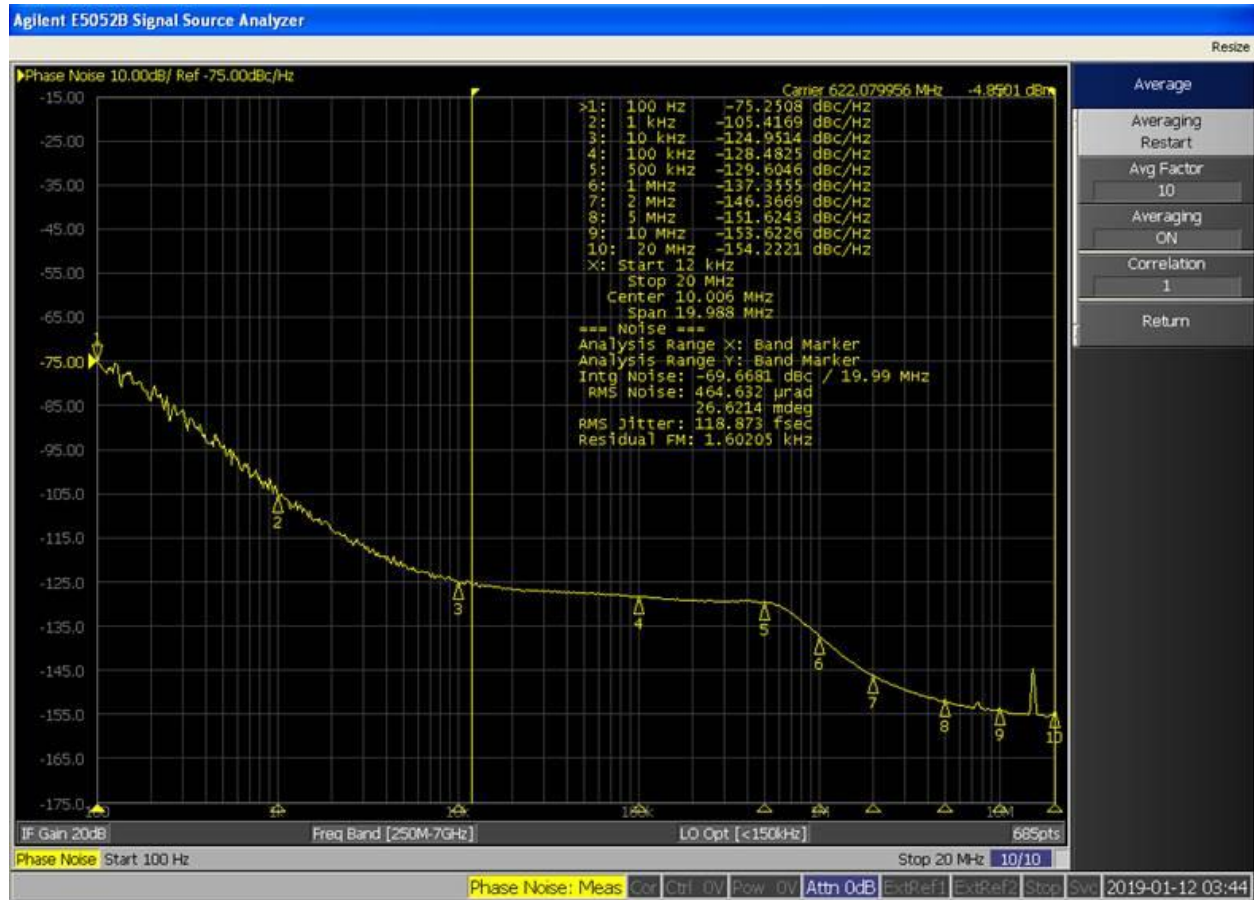


Figure 2. Representative Phase Noise Measurement: F_{OUT}=622.08MHz, F_{IN}=10MHz, BW=100Hz

Table 10 shows Low Frequency Phase Noise and Table 11 shows electrical specifications for output clocks.

Table 10. Low Frequency Phase Noise

Parameter	Symbol	Conditions	Typ.	Unit	Condition
Low Frequency Phase Noise	PN ^[31]	Offset Frequency = 10 Hz	-91	dBc/Hz	F _{OUT} = 122.88 MHz, F _{in} = 10 MHz, PLL BW = 100 Hz
		Offset Frequency = 100 Hz	-113		
		Offset Frequency = 1 kHz	-130		
		Offset Frequency = 10 kHz	-138		

Note:

- 31. Does not include noise from the input clock to the PLL or from the reference. This is additive phase noise from the chip.

Table 11. Electrical Specifications for Output Clocks

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
DC Electrical Specifications – LVCMOS output						
Output High Voltage	VOH	VDDO – 0.3		–	V	4 mA load, VDD = 3.3V
Output High Voltage	VOH	VDDO – 0.4		–	V	4 mA load, VDD = 1.8V and 2.5V
Output High Voltage	VOH	VDDO – 0.2			V	100 μ A Load
Output Low Voltage	VOL	VDDO – 0.3		0.3	V	4 mA load
Output Low Voltage	VOL	VDDO – 0.4		0.2	V	100 μ A Load
DC Electrical Specifications – LVDS Outputs (VDDO = 1.8-V, 2.5-V or 3.3-V range)						
Output Common-Mode Voltage	VOCM	1.125	1.2	1.375	V	VDDO = 2.5V or 3.3V range
Change in VOCM between complementary output states	Δ VOCM			50	mV	
Output Leakage Current	IOZ	-20		20	μ A	Output Off, VOUT = 0.75V to 1.75V
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 – 2.0V)
Output Low Voltage	VOL	VDDO – 2.0		VDDO – 1.55		Rterm = 50 Ω to VTT(VDDO – 2.0V), 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 Output High Voltage	VOHDIFF	150			mV	Measurement taken from differential waveform
Differential Output Low Voltage	VOLDIFF			-150	mV	Measurement taken from differential waveform
Absolute Crossing point voltage	VCROSS	250		600	mV	Measurement taken from single ended waveform
Variation of VCROSS over all rising clock edges	VCROSS DELTA			140	mV	Measurement taken from single ended waveform
DC Electrical Specifications – CML Outputs (VDDO = 1.8-V, 2.5-V or 3.3-V range)						
Output High Voltage	VOH	VDDO – 0.085	VDDO – 0.01	VDDO	V	Rterm = 50 Ω to VDDO
Output Low Voltage	VOL	VDDO – 0.6	VDDO – 0.4	VDDO – 0.32	V	Rterm = 50 Ω to VDDO
AC Electrical Specifications LVCMOS Output Load: 10pF < 100MHz, 7.5pF < 150MHz, 5pF > 150MHz > 200MHz						
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 = 1.8V, 20-80%, Highest Drive setting
Rise/Fall time	tRFCMOS			1.5	ns	VDDO = 2.5V, 20-80%, Highest Drive setting
Rise/Fall time	tRFCMOS			1.2	ns	VDDO=3.3V, 20-80%, Highest Drive setting
Cycle to Cycle Jitter	tCCJ			50	ps	pk, Measured at 1/2 VDDO over 10k cycles, fOUT = 156.25 MHz
Period Jitter	tPJ			100	ps	pk-pk, Measured at 1/2VDDO over 10k cycle, fOUT = 156.25 MHz
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 12 below shows the I²C bus timing specifications.

Table 12. 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	

Figure 3 shows the I²C timing waveforms.

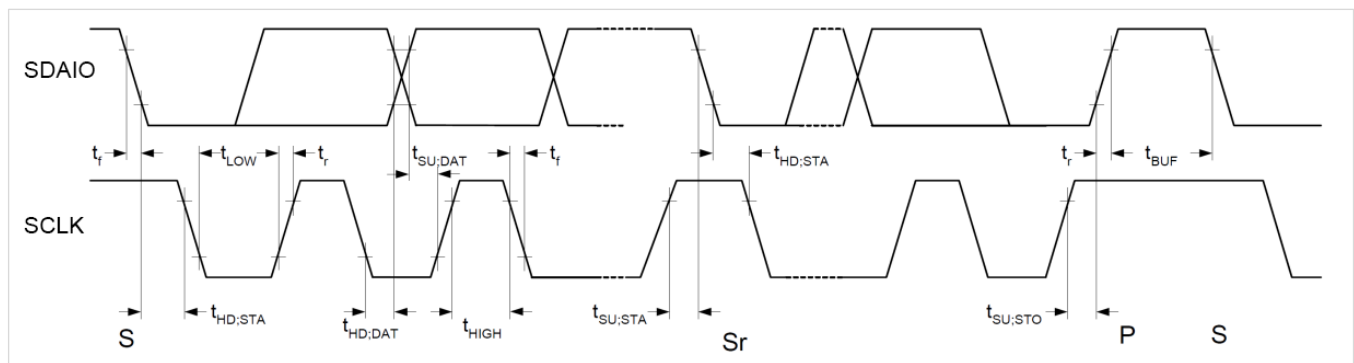


Figure 3. I²C Timing Waveforms

Table 13 below shows SPI bus timing specifications.

Table 13. SPI Bus Timing Specifications^[32]

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:

32. These parameters are guaranteed by design and are not tested.

Figure 4 shows the SPI timing waveforms.

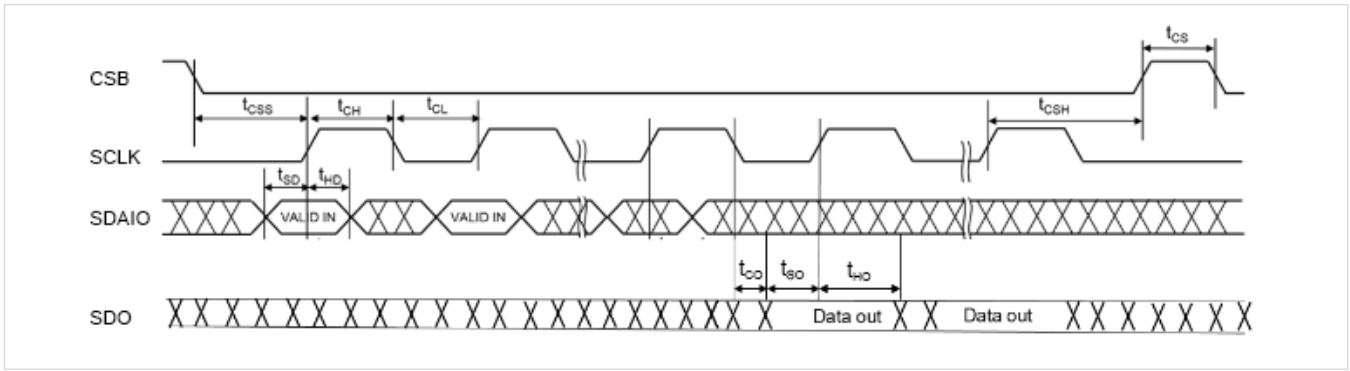


Figure 4. SPI Timing Waveforms

Figure 5 shows the top view of the SiT95147.

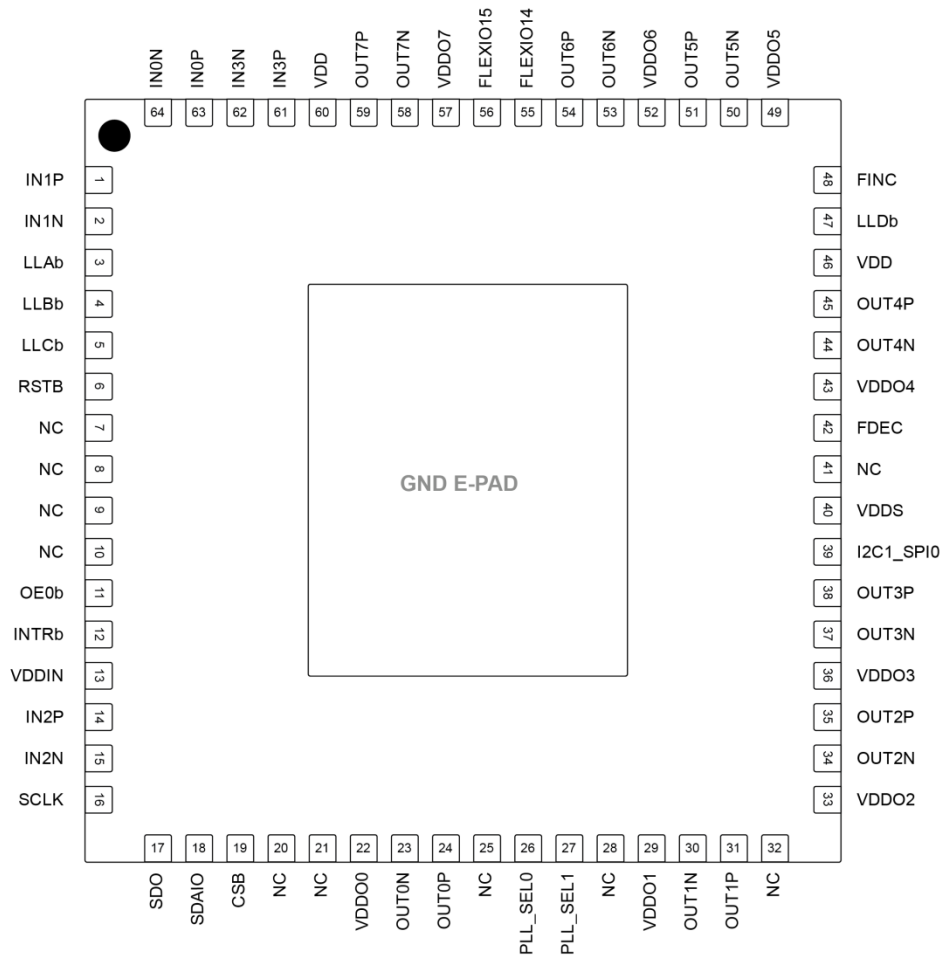


Figure 5. SiT95147 Top View

Table 14 shows the detailed pin description.

Table 14. Detailed Pin Description

Pin Name	I/O Type	SiT95147 Pin #	Function	Comments
IN0P	Input	63	True input for IN0 differential pair. Input for LVC MOS IN0 input. Need series external capacitor for differential input.	IN0 / IN1 / IN2 / IN3 inputs can be used for output clock synchronization. An active clock and three spare clocks are chosen for each PLL. There is complete flexibility in the choice of the Active and 4 Spares for each of the 4 PLLs.
IN0N	Input	64	Complement input for IN0 differential pair. Ground with capacitor for LVC MOS IN0 input. Need series external capacitor for differential input.	
IN1P	Input	1	True input for IN1 differential pair. Input for LVC MOS IN1 input. Need series external capacitor for differential input.	
IN1N	Input	2	Complement input for IN1 differential pair. Ground with capacitor for LVC MOS IN1 input. Need series external capacitor for differential input.	
IN2P	Input	14	True input for IN2 differential pair. Input for LVC MOS IN2 input. Need series external capacitor for differential input.	
IN2N	Input	15	Complement input for IN2 differential pair. Ground with capacitor for LVC MOS IN2 input. Need series external capacitor for differential input.	
IN3P	Input	61	True input for IN3 differential pair. Input for LVC MOS IN3 input. Need series external capacitor for differential input.	
IN3N	Input	62	Complement input for IN3 differential pair. Ground with capacitor for LVC MOS IN3 input. Need series external capacitor for differential input.	
GND	Power	E-PAD	Electrical and Package Ground	
OUT0P	Output	24	Output 0 True Output or Output 0 LVC MOS	LVPECL, LVDS, HCSL, CML and LVC MOS support.
OUT0N	Output	23	Output 0 Complement Output or Output 0 LVC MOS	
OUT1P	Output	31	Output 1 True Output or Output 1 LVC MOS	
OUT1N	Output	30	Output 1 Complement Output or Output 1 LVC MOS	
OUT2P	Output	35	Output 2 True Output or Output 2 LVC MOS	
OUT2N	Output	34	Output 2 Complement Output or Output 2 LVC MOS.	
OUT3P	Output	38	Output 3 True Output or Output 3 LVC MOS	
OUT3N	Output	37	Output 3 Complement Output or Output 3 LVC MOS	
OUT4P	Output	45	Output 4 True Output or Output 4 LVC MOS	
OUT4N	Output	44	Output 4 Complement Output or Output 4 LVC MOS	
OUT5P	Output	51	Output 5 True Output or Output 5 LVC MOS	
OUT5N	Output	50	Output 5 Complement Output or Output 5 LVC MOS	
OUT6P	Output	54	Output 6 True Output or Output 6 LVC MOS	
OUT6N	Output	53	Output 6 Complement Output or Output 6 LVC MOS	

Table 14. Detailed Pin Description (continued)

Pin Name	I/O Type	SiT95147 Pin #	Function	Comments
OUT7P	Output	59	Output 7 True Output or Output 7 LVC MOS	
OUT7N	Output	58	Output 7 Complement Output or Output 7 LVC MOS	
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.
VDDS	Power	40	Supply for determining VIH/VIL for FDEC	
LLAb	Output	3	Loss of Lock Indicator. Indicate when PLL A is out of lock (low) or locked (high). Unconnected when not in use.	
LLBb	Output	4	Loss of Lock Indicator. Indicate when PLL B is out of lock (low) or locked (high). Unconnected when not in use.	
LLCb	Output	5	Loss of Lock Indicator. Indicate when PLL C is out of lock (low) or locked (high). Unconnected when not in use.	
RSTB	Input	6	Active low reset internally pulled up to VDDD; Pull Up Resistor to VDDD of fixed value (200 K Ω)	Active low signal performs a complete reset of the part
OE0B	Input	11	Used to disable (when 1) all the output clocks	
INTRb	Output	12	Active low indicator of programmable sticky notifies	
SDAIO	Input / Output	18	I ² C Serial Interface Data (SDA) / SPI Input data (SDI)	
SCLK	Input	16	I ² C Serial Interface Clock or SPI Clock Input	
CSB	Input	19	Chip Select for the SPI Interface	
SDO	Output	17	Serial Data Output (SPI Interface)	
FDEC	Input	42	DCO decrement	
I2C1_SPI0	Input	39	Choose between SPI(0) and I ² C(1) interface being used	
LLDb	Output	47	Loss of Lock Indicator. Indicate when PLL D is out of lock (low) or locked (high). Unconnected when not in use.	
PLL_SEL0	Input	26	PLL selection for Pin based DCO	
PLL_SEL1	Input	27		
FINC	Input	48	DCO increment	
FLEXIO14	Input/Output	55	Flexible Input / Output 14	
FLEXIO15	Input/Output	56	Flexible Input / Output 15	
VDDO0	Power	22	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	43	Output Power Supply for Bank 4 outputs	
VDDO5	Power	49	Output Power Supply for Bank 5 outputs	
VDDO6	Power	52	Output Power Supply for Bank 6 outputs	
VDDO7	Power	57	Output Power Supply for Bank 7 outputs	
NC	No Connect	7, 8, 9, 10, 20, 21, 25, 28, 32, 41	No connect. This pin is not connected to the die.	

Functional Description

SiT95147 is a jitter attenuating and network synchronizing frequency translation device that offers four independent fractional PLLs. The output high frequency voltage controlled oscillator (VCO) associated with each PLL is mapped to the 8 outputs in a very flexible fashion. This offers a very flexible frequency translation arrangement with independent control of each PLL in terms of jitter attenuation, bandwidth control and input clock selection with redundancy.

Figure 6 shows a high level architecture diagram of SiT95147.

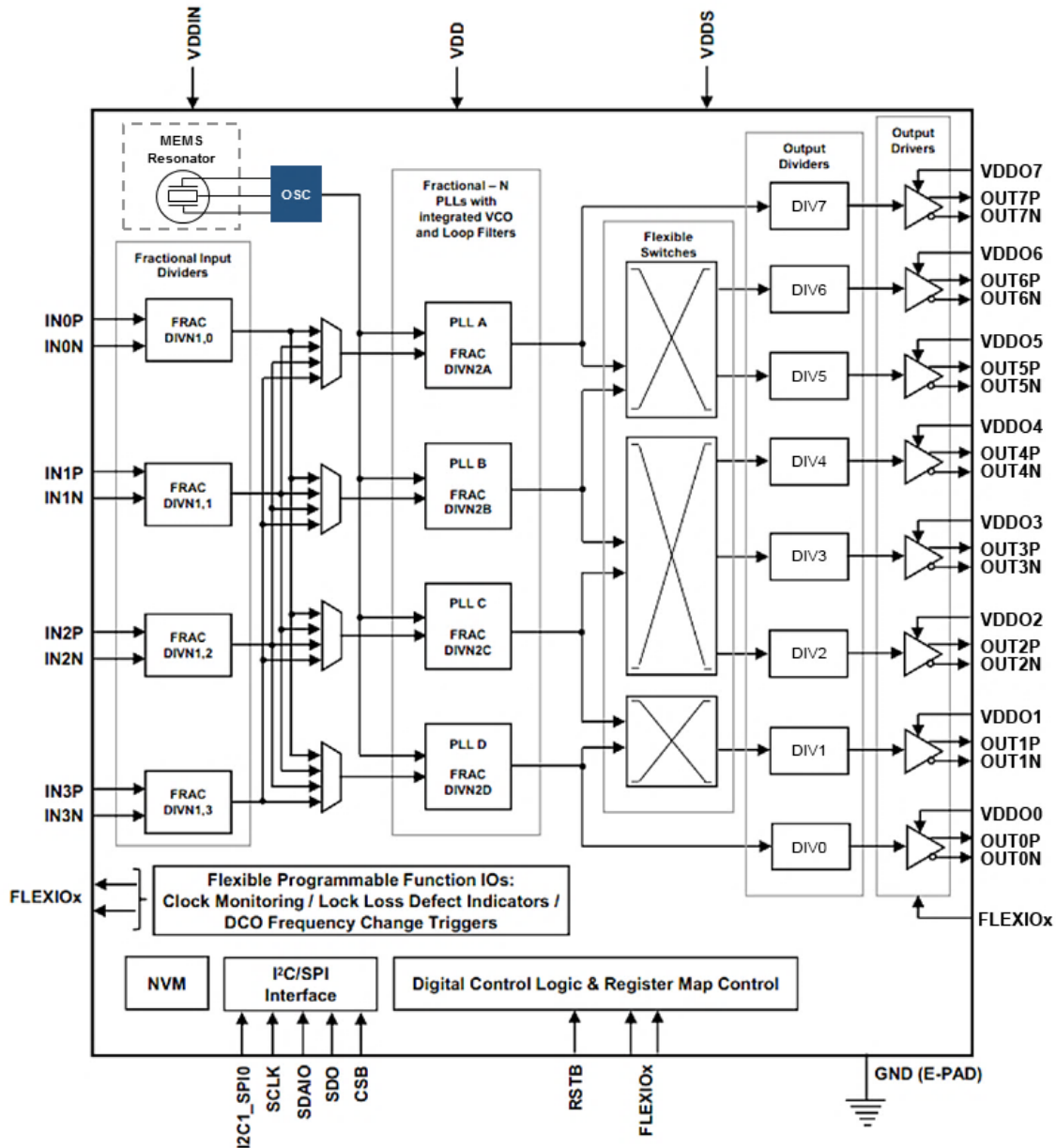


Figure 6. Overall Architecture

The overall programmable interface (PIF) structure is a register map that is divided into several pages according to function. Each controller (master and slaves) has an associated unique Page number. Each Page has an independent 8 bit addressable PIF memory. In all the pages, the last address, FF, holds the current page

number and is reserved for changing the page. The current page to be communicated with can be set by writing the page number in hexadecimal form {0, 1, 2, 3, A, B, C, D} in the address FF on any page. Table 15 shows a summary of the PIF contents residing on each page.

Table 15. PIF Description

Page	What?	Summary of Contents
0	Master	All Generic Information related to the chip Chip Configuration details Control for the master sequencer FSM MEMS Oscillator Reference Related Information Fuse Pointer for each of the remaining pages
1	ClkMon Slave	Clock Loss related function Frequency Drift related function
2	Input Slave	Input 3 / 2 / 1 / 0 related information (Input type, DIVN1 divider configuration)
3	Output Slave	Flexible Outputs 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 as well as the clock translation pathways available on the chip are shown in Figure 7.

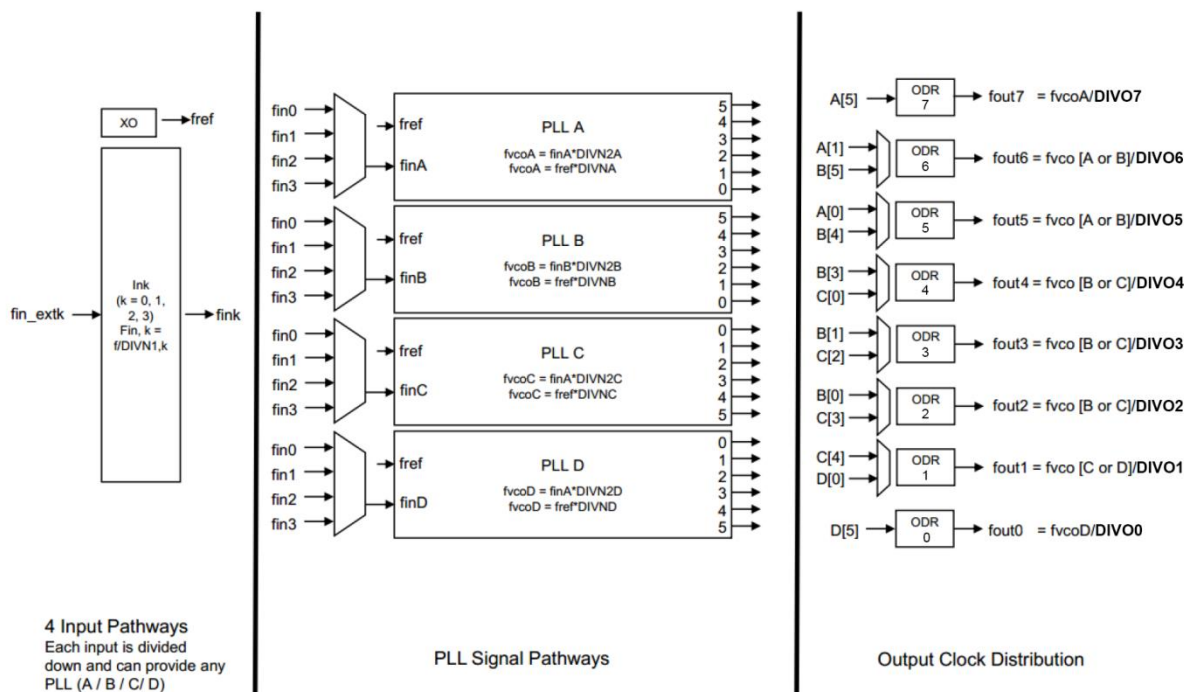


Figure 7. Overall Hierarchy of Clocks

The four input clocks with frequency fin_extk translate to PLL input clocks fin_k following division by the respective input dividers with fractional or integer frequency division ratios $DIVN1k$ where the index $k \in \{0, 1, 2, 3\}$. See Figure 8. Each of the PLLs chooses one of the four

divided input clocks fin_k as its active input clock and sets the priority for up to three spare clocks from the remaining three input clocks if required for hitless switching to a redundant input.

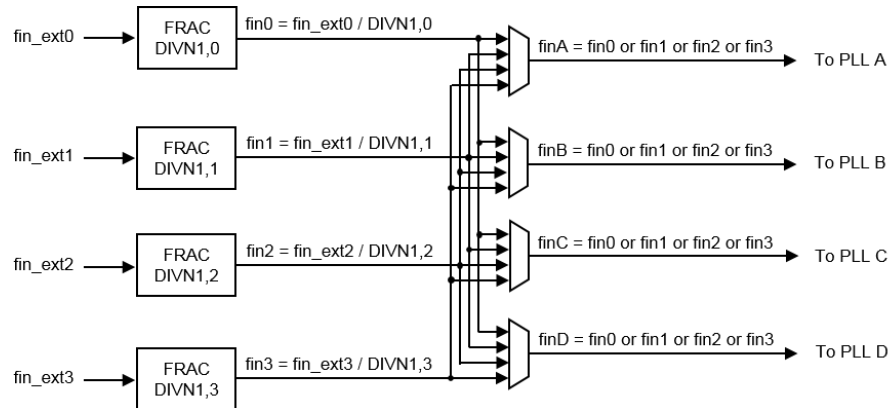


Figure 8. Input Clock Distribution

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

Each of the eight Output Drivers (ODR_j , $j \in \{0:7\}$) chooses an appropriate VCO frequency and divides it using their respective integer divider $DIVO_j$ to get the output frequency f_{outj} . See Figure 9.

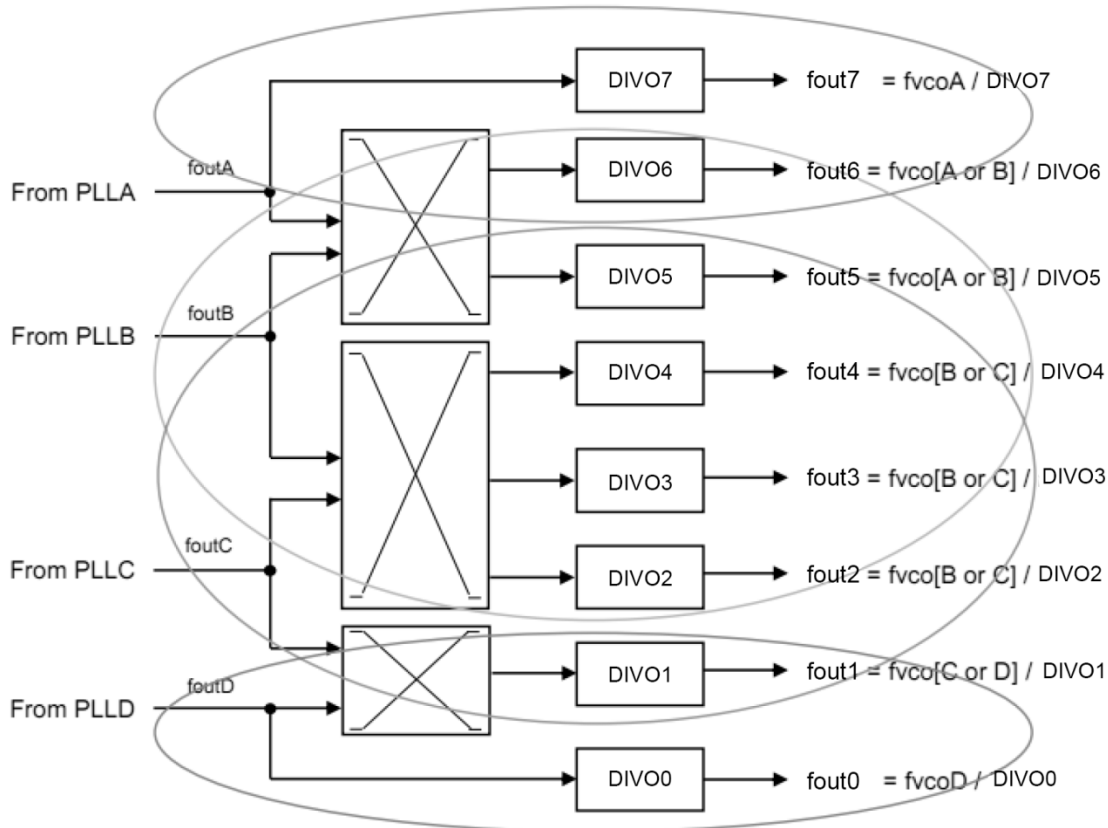


Figure 9. Output Clock Distribution

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

PLL Bandwidth

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

Fast Lock

An independent choice for a fast bandwidth is also available that can be used for speeding up the initial lock. After the PLL lock is achieved and the system is in the synchronized mode, the bandwidth is automatically transitioned to the steady state jitter attenuation

bandwidth. This feature avoids the abnormally large wake up times that may be needed for very low PLL bandwidths. For stability considerations of the PLL, the fast lock bandwidth or regular bandwidth for the PLL should be no larger than $1/100^{\text{th}}$ of the input frequency at the input of the PLL (post the DIVN1k dividers).

Operation Modes

There are three distinct modes of operation of the PLL: free run mode, synchronized mode and holdover mode. The frequency of the high frequency VCO in the PLL is determined by the specific mode of operation. The state diagram with the modes of operation is showing on the [Figure 10](#).

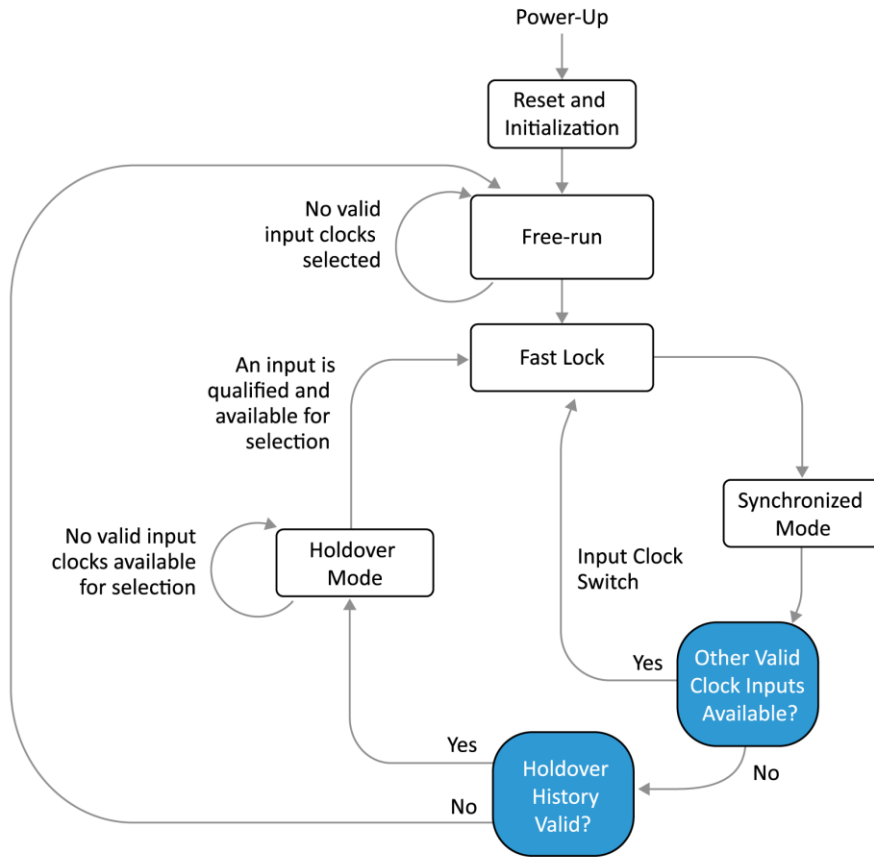


Figure 10. Operation Modes

Device Start-Up and Initialization

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

Free run mode

The PLL in the free run mode can be described as a MEMS based oscillator where the output frequency is determined by the relation $f_{VCOx} = DIVN_x \times f_{ref}$. This is the mode of operation before the loop is locked to the selected input clock or the mode of operation for the case none of the input clocks is available.

Synchronized mode

After locking to the chosen input clock, the PLL enters the synchronized mode of operation where the output is now locked to the input frequency with the relation $f_{VCOx} = DIVN_2 \times f_{inx}$. The PLL Loop that synchronizes (locks) the output to the input clock has a programmable loop bandwidth between 1 mHz to 4 kHz and is not affected by static or dynamic drifts in the crystal oscillator based fref frequency. In synchronized mode, the PLL is able to lock to a Gapped Input clock with some edges missing producing a smooth output clock without any gaps with the requested frequency translation from input to output. Frequency translation ratios in this case should be specified with respect to the average input frequency of the gapped clock rather than the faster instantaneous frequency.

Holdover mode

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

DCO Mode operation

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

PLL MEMS Oscillator Reference

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

Inputs

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

Automatic Input Selection

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

Manual Input Selection

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

Hitless Input Switching

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

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

Ramped Input Switching

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

Input Clocks

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

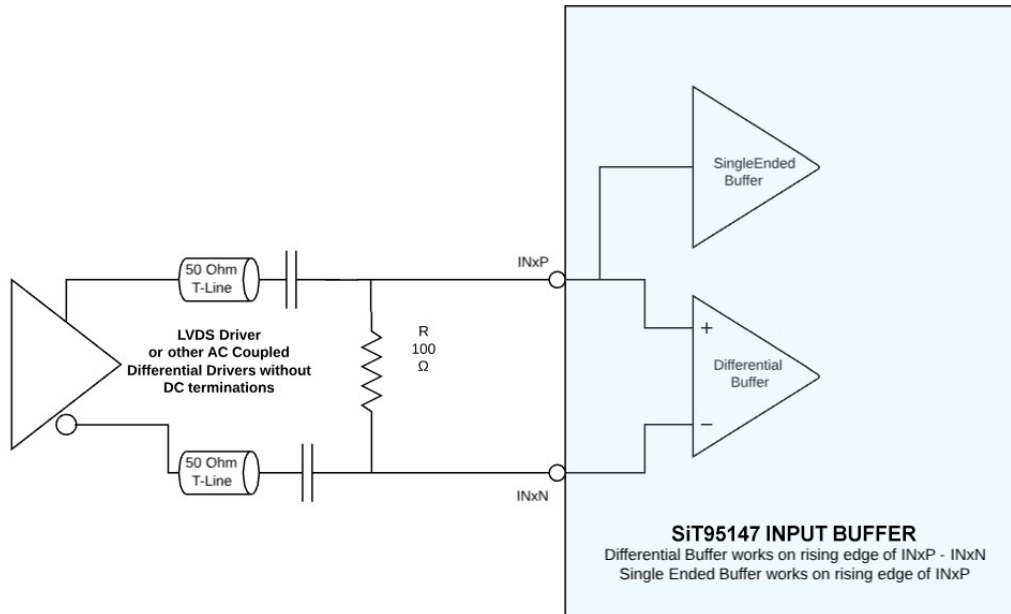


Figure 11. AC Coupled Differential LVDS Input or Other AC Coupled Driver without DC Terminations – Uses Differential Buffer Pathway

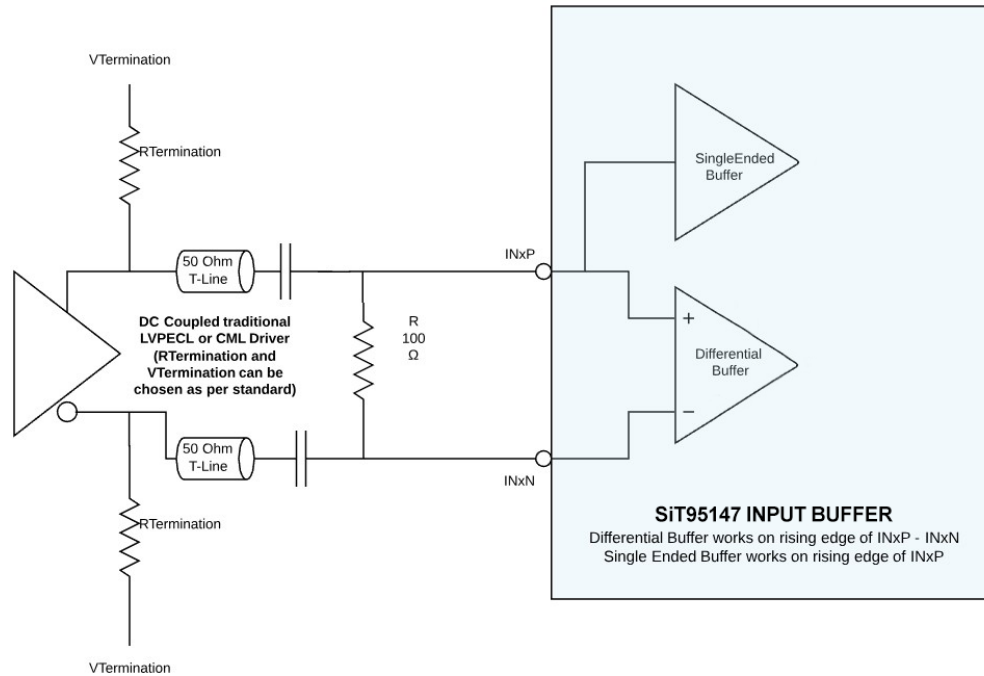


Figure 12. AC Coupled Differential LVPECL or CML (Resistor and Voltage termination is as per the standard) – Uses Differential Buffer Pathway

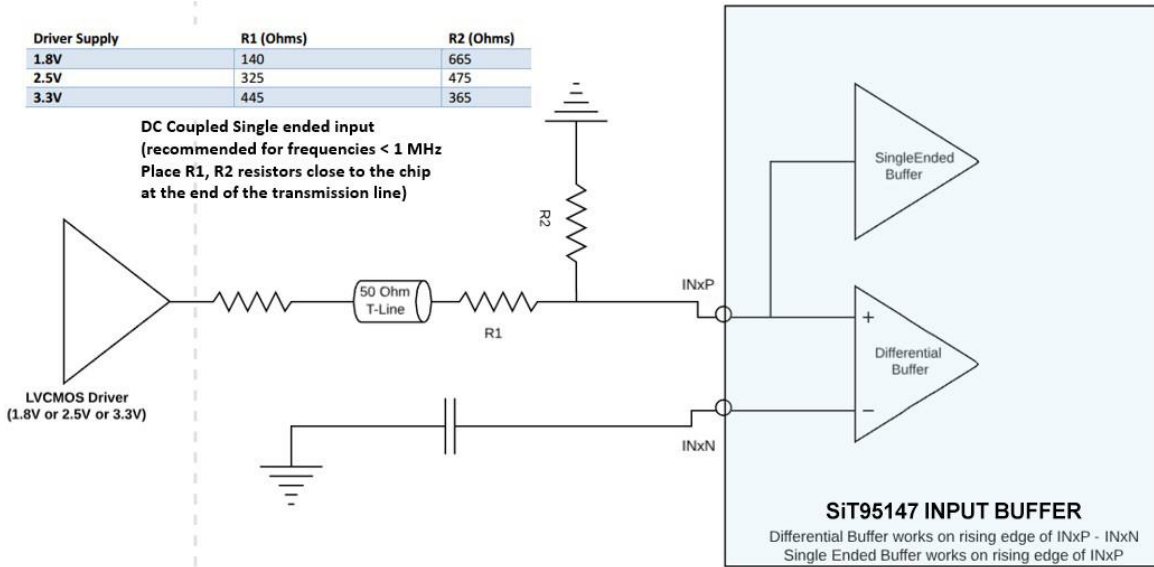


Figure 13. DC Coupled Single Ended Driver – Uses Single Ended Buffer Pathway^[33]

Note:

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

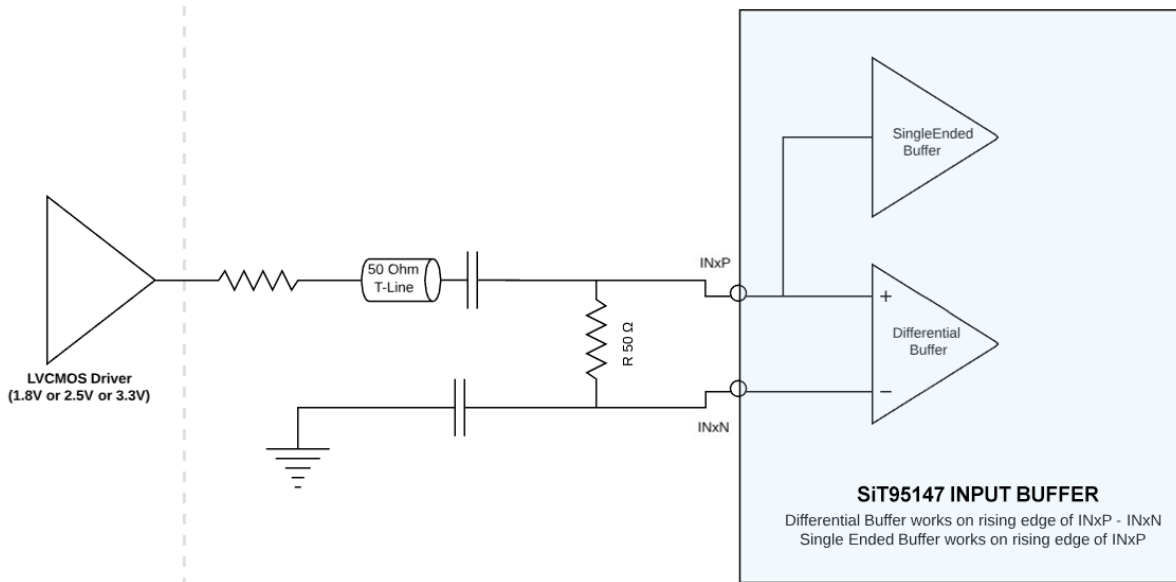


Figure 14. AC Coupled Single Ended Driver with 50 Ω Termination on receiver (chip) side

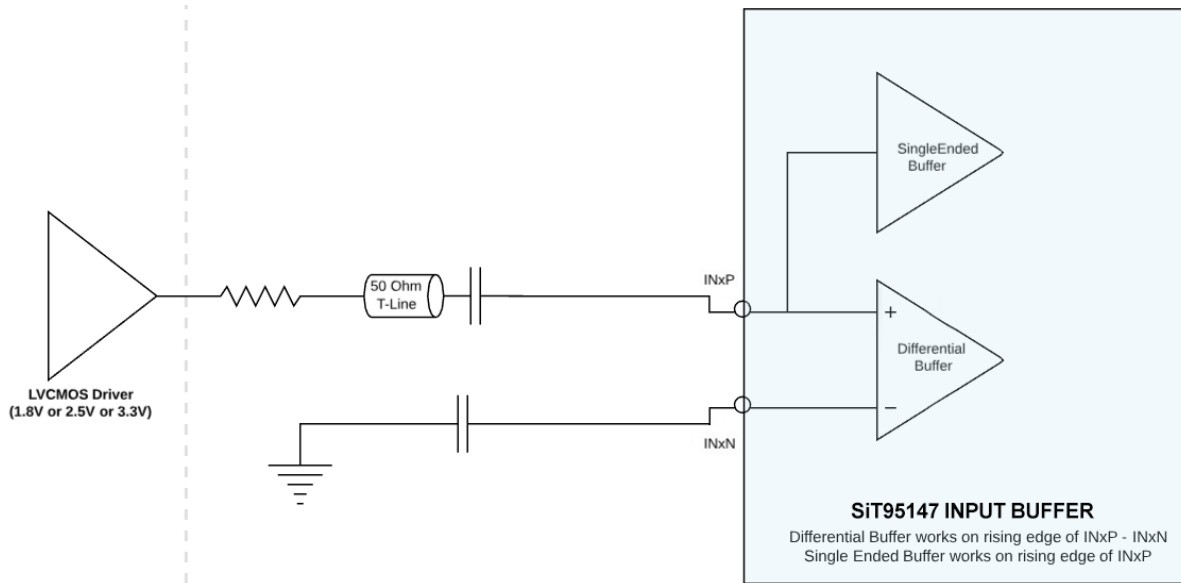


Figure 15. AC Coupled Single Ended LVC MOS input without 50 Ω Termination

Clock Monitoring

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

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

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

PLL Lock Loss Defect Monitoring

PLL Lock Loss is another fault monitor whose specifications are available in. 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. Similar to the faults monitored by the Clock Monitor Slave, this defect can be tracked with status, notify and on the FLEXIOs.

Outputs

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

Differential Output Termination Information

Figure 16 to Figure 19 show LVPECL termination.

LVPECL:

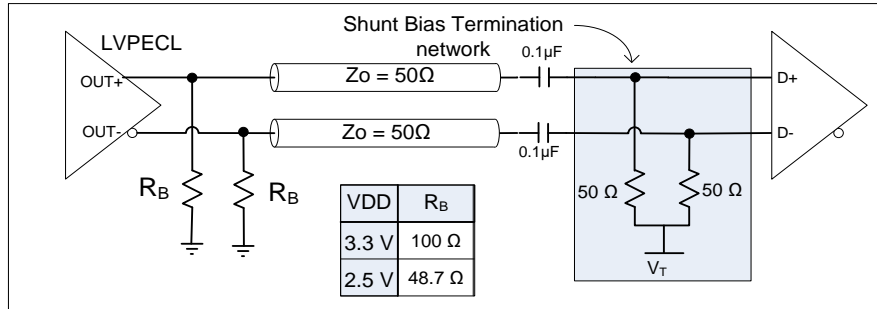


Figure 16. LVPECL with AC-coupled Termination

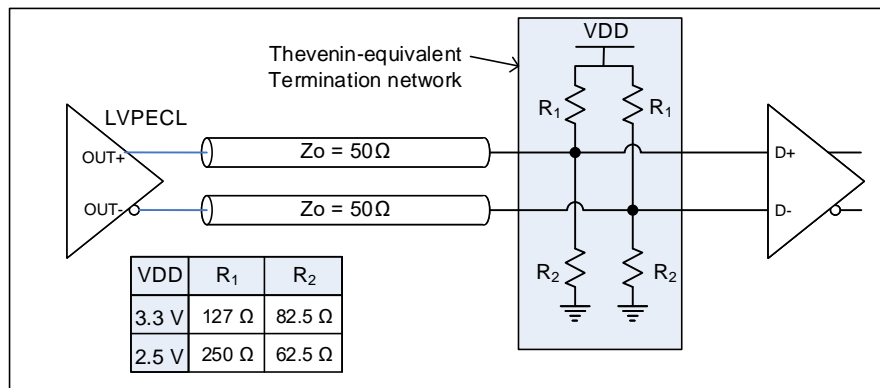


Figure 17. LVPECL DC-coupled Load Termination with Thevenin Equivalent Network

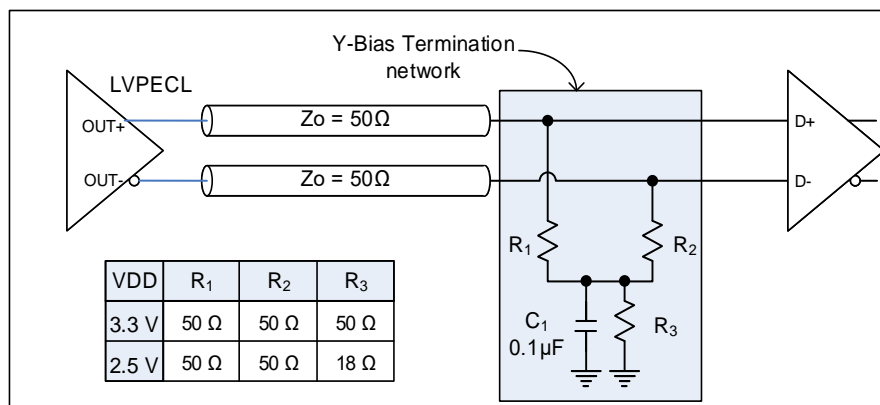


Figure 18. LVPECL with Y-Bias Termination

Differential Output Termination Information (continued)

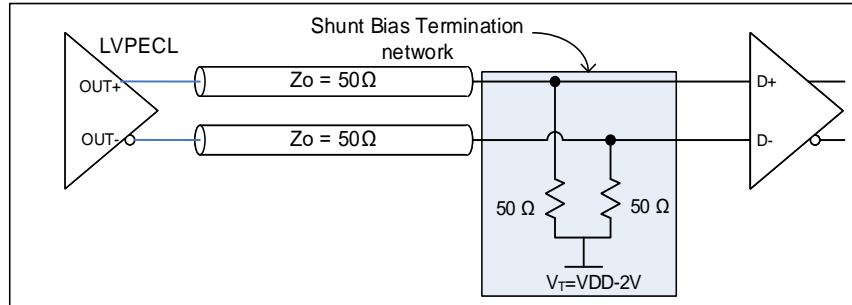


Figure 19. LVPECL with DC-coupled Parallel Shunt Load Termination

Figure 20 to Figure 22 show LVDS termination.

LVDS:

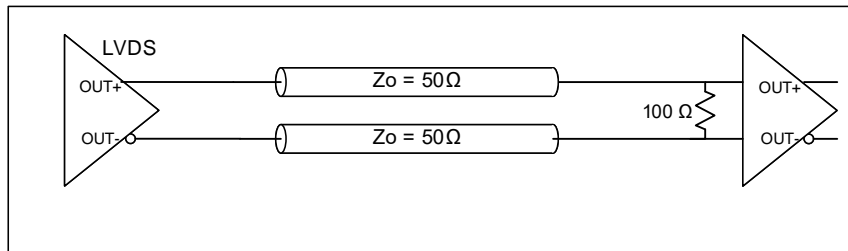


Figure 20. LVDS Single DC Termination at the Load

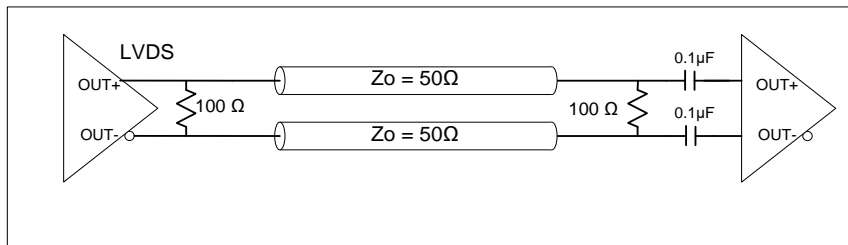


Figure 21. LVDS double AC Termination with Capacitor Close to the Load

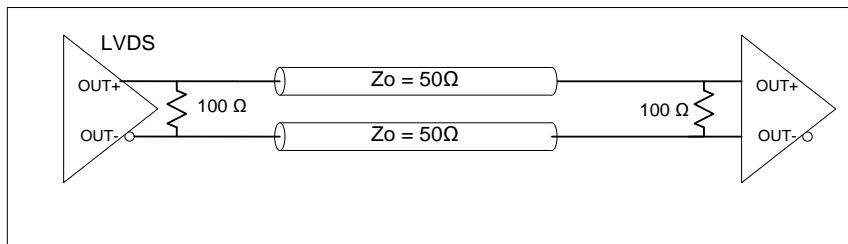


Figure 22. LVDS Double DC Termination

Differential Output Termination Information (continued)

Figure 23 and Figure 24 show HCSL and CML termination, respectively. Figure 25 shows LVCMOS termination.

HCSL:

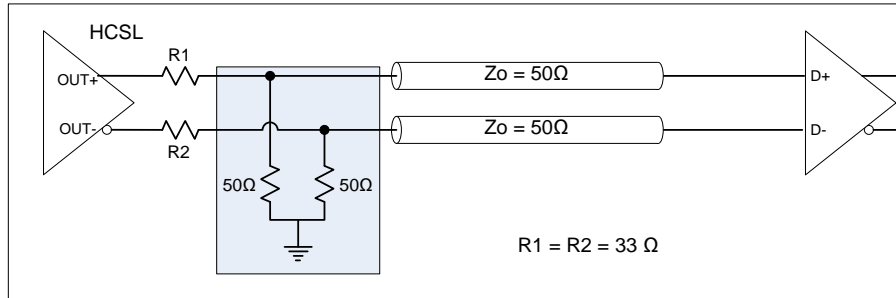


Figure 23. HCSL Interface Termination

CML:

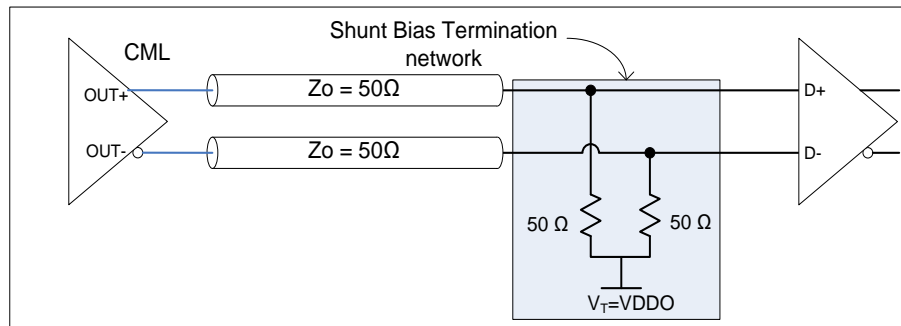


Figure 24. CML Interface Termination

LVCMOS:

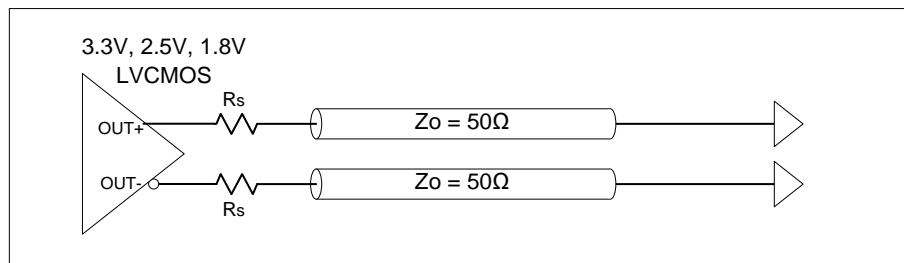


Figure 25. LVCMOS output Termination

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.

Output Delay Control

By default, all output clocks are phase-aligned. A delay path associated with each of Fractional Input Dividers (see Figure 26) is available for applications that need a specific output skew configuration. This is useful for PCB trace length mismatch compensation. The resolution of the phase adjustment is 35 ps definable in a range of ± 7.5 ns. Phase adjustments are register-configurable. All phase delay values are restored to their default values after power-up, hard reset, or a reset using the RSTB pin. Phase delay default values can be written to NVM, allowing a custom phase offset configuration at power-up or after power-on reset, or after a hardware reset using the RSTB pin.

Zero Delay Mode

Zero Delay Mode is used 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 26. This ensures minimum delay between the input and output. It can be used for one of the four PLLs at any time. The input pins must be terminated and ac-coupled when Zero Delay Mode is used. A differential external feedback path connection is necessary for best performance.

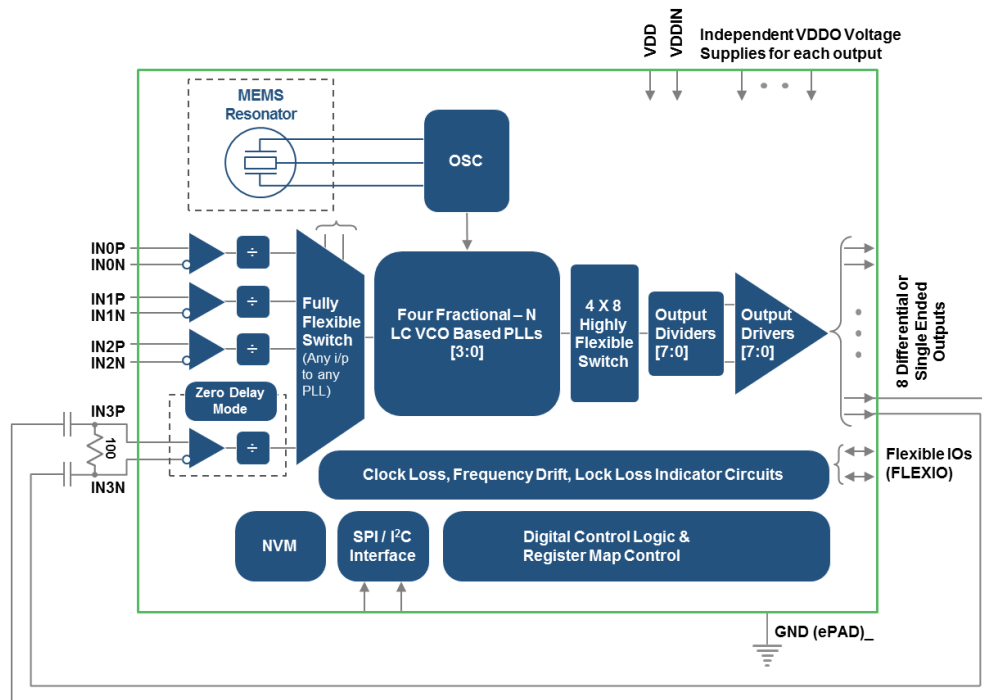


Figure 26. SiT95147 Zero Delay Mode Setup

Programmable Interface (PIF) Top Level View

Table 16. PIF Overview (Top Level Summary of the Programmable Interface)^[34]

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

Table 16. PIF Overview (Top Level Summary of the Programmable Interface)^[34] (continued)

Page Number	Function	Comments	Fuse repeated twice
Ah	PLL A	<p><u>2Fh[7:6] : Customer - Lock Pattern for the Fuse</u></p> <p>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</p>	YES

Note:

34. All Addresses in Hexa Decimal.

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 Kb/s 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 operation.

The default Slave address is 11010A1A0 where A1 and A0 are controlled by pins on the device. The device also supports variable Slave addresses which can be provided via the efuse. Therein too, the LSbs of A1 and A0 are controlled via the pins on the device. This thus allows four choices of Slave addresses for any system where in the first 5 bits of the slave address can be the same.

Figure 27 shows the read block and Figure 28 shows the write block.

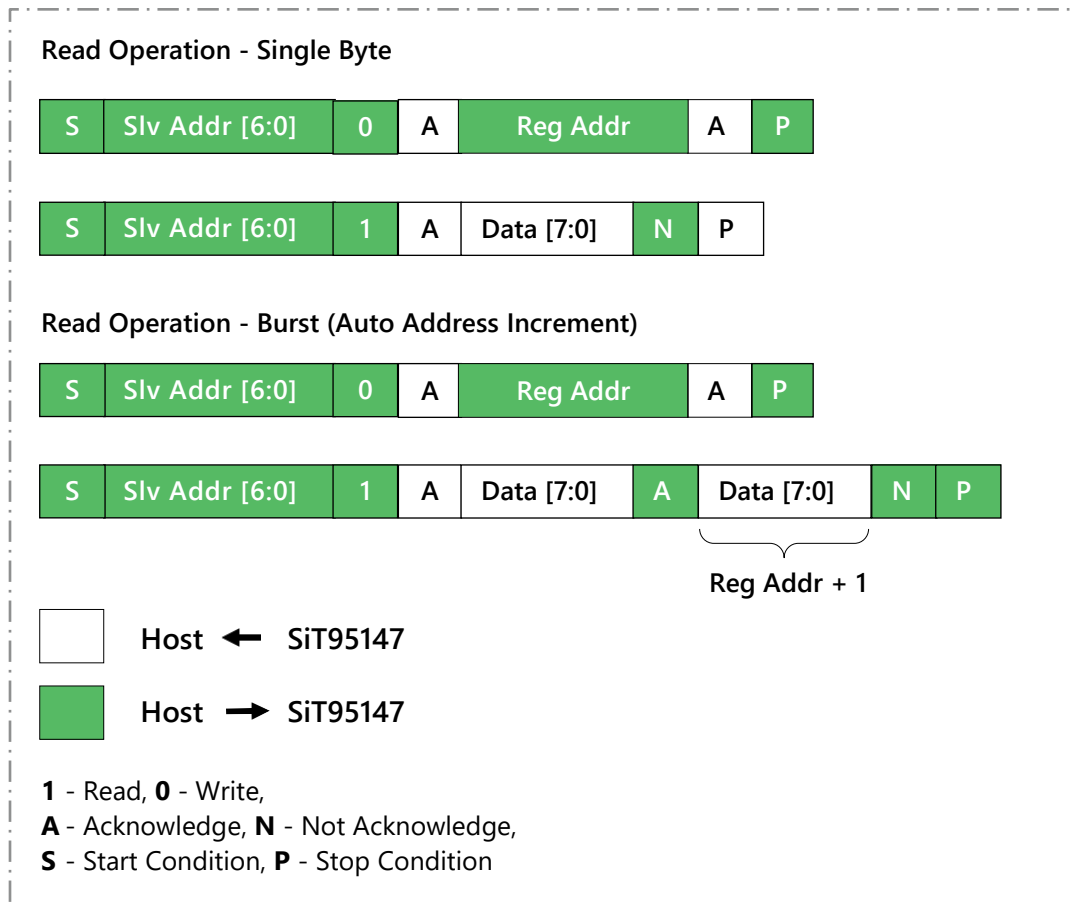


Figure 27. Read Block

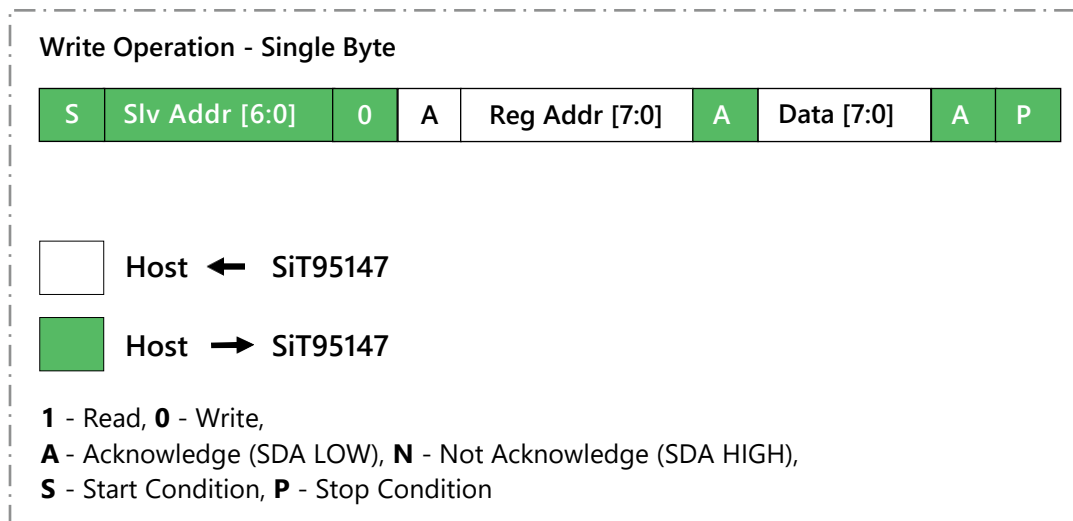


Figure 28. Write Block

■ 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_SPIO pin is Low, a Low on the CSB pin activates the SPI access.

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 SiT95147, 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 SiT95147 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 SiT95147 captures the same on the rising edge of the SPI clock. There is no loopback provided here.
6. In SiT95147 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 SiT95147 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 SiT95147 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 SiT95147 does not support burst address write or read operations.

SPI Timing Details

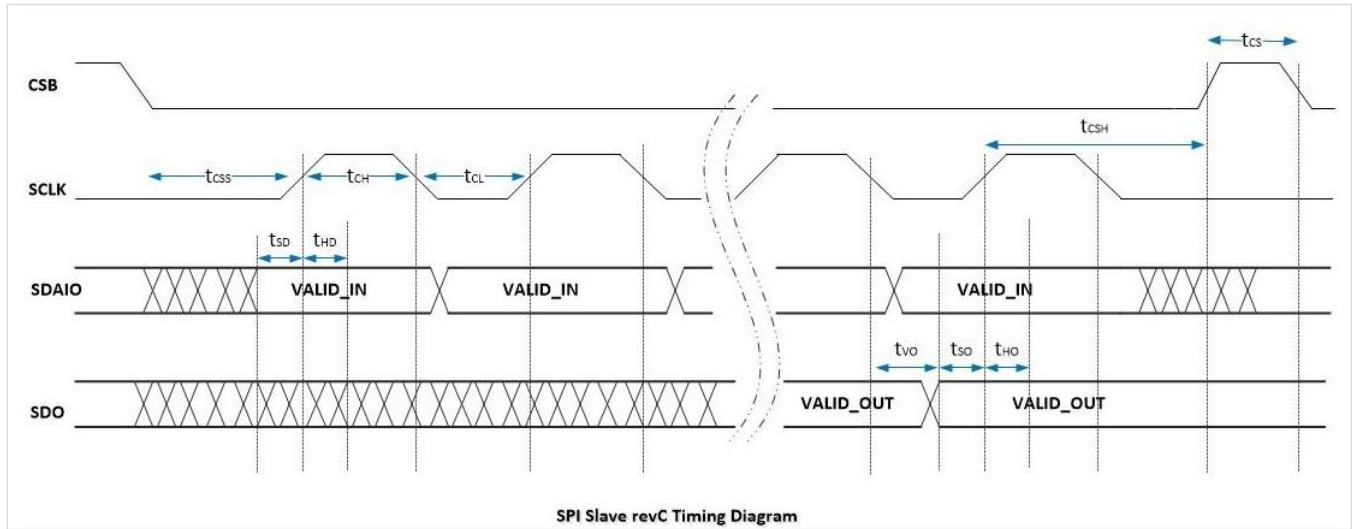


Figure 29. SPI Timing Diagram

Table 17. 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_{so}	Output valid			10	ns
t_{so}	Output valid			10	ns

SPI Single byte write

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

SPI Single byte read

- The master initiates the transaction by issuing a start condition of pulling `csn_i` to active low
- The master assembles the serial data on the falling edge of the SPI clock so the SPI slave can capture the same on the rising edge of the SPI clock
- The first 7 bits are don't care with the 8th bit being set to 0 to indicate a read operation
- The next 8 bits (second byte) are used for the register map address
- The next 8 bits (third byte) are used for the register map read data that is supplied by the pif slave block
- The 16th rising edge of the SPI clock is used to capture the last address bit. The SPI slave then assembles the address, enable and `wr_rdn` to the PIF slave block. The slave block then uses the address when enable is high to provide the read back data via a multiplexer. This operation has to be completed within half a SPI clock since the SPI slave has to assemble the first read back data bit on the falling edge of the SPI clock so the SPI slave can capture the same on the next rising edge. After 7 additional clocks, all 8 serial read back data bits are sent out from the SPI slave
- The `csn` is then de-activated (by going high) by the master
- For the next read operation, `CSN` is held high for at least a duration of two SPI clocks following which the entire operation can start again.

Figure 30 shows the SPI Read Operation and Figure 31 shows the SPI Write Operation.

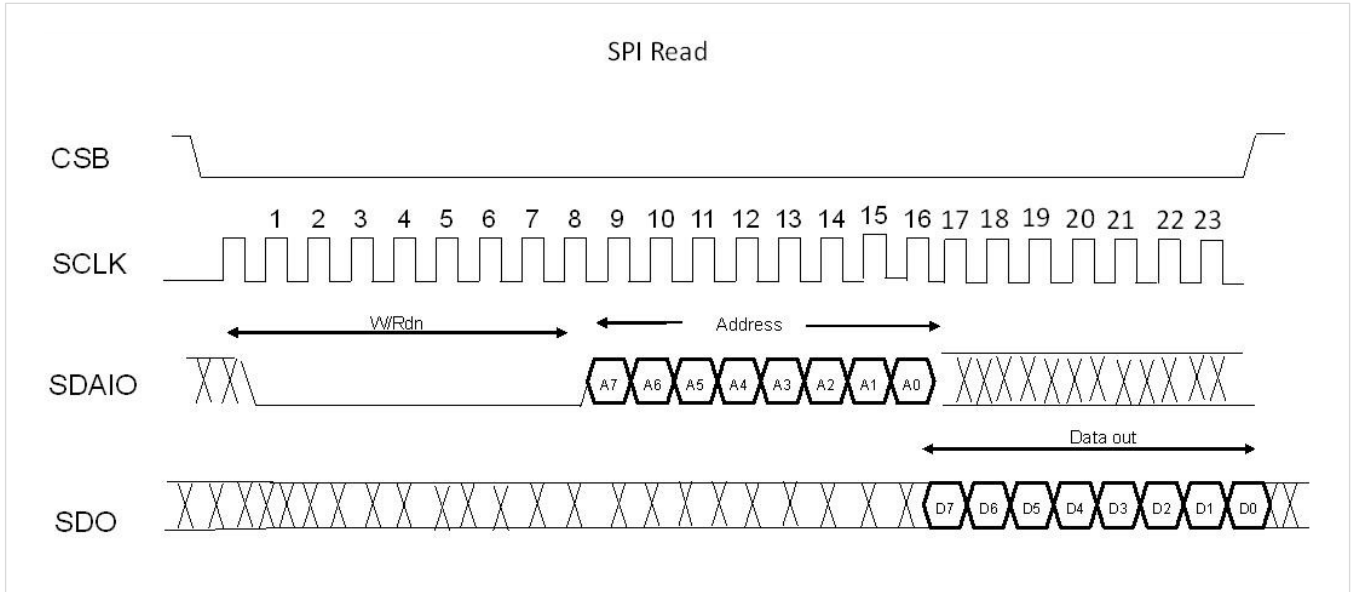


Figure 30. SPI Read Operation

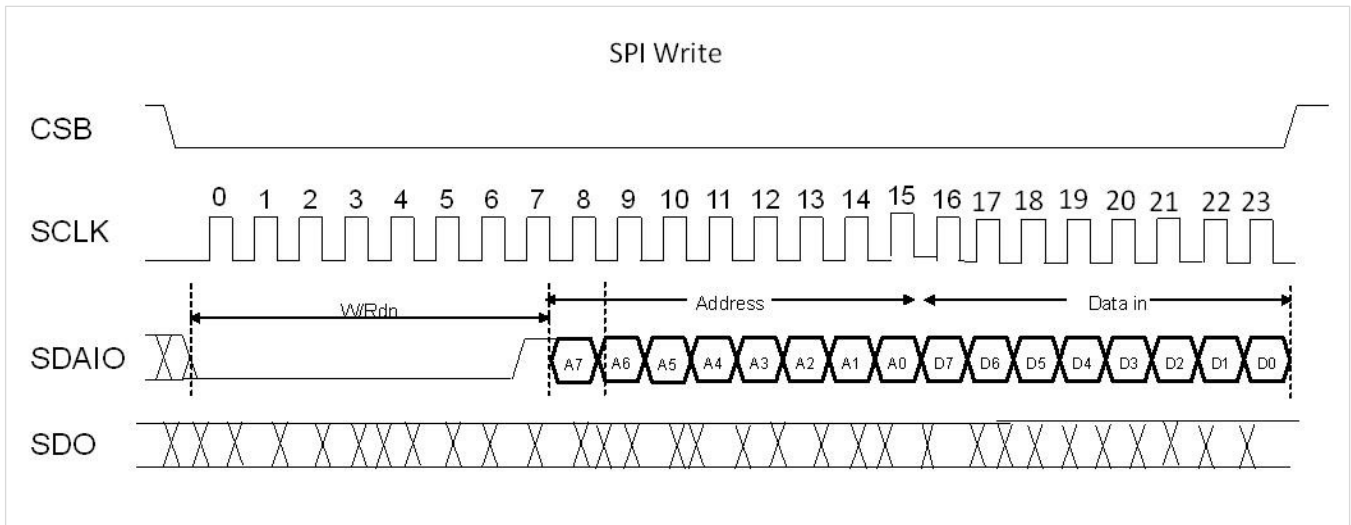


Figure 31. SPI Write Operation

Alarm Registers' Information

Status and Notify

SiT95147 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 registers – Register 0x02 is the status register, 0x03 is the notify and 0x04 is the masking register for the notify.

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 register 0x02 to see the current status at any time. Sometimes, we may get a very short pulse for the status and 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 status registers. This is a sticky bit and stays at 1 till the user writes a 1 to that specific bit in register 0x03 to clear it. The notify sticky bit operates only if the masking register in 0x04 is set to 1. If the masking register bit is set to 0, notify will not be asserted even when status toggles. The default value for the mask register is 0xFF so all the notify signals are enabled. Once the user writes a 1 to clear the

notify bit, the notify bit can again go high on the next rising edge of the status.

The default value of 0x03 and 0x07 is 0xFF – user has to write 0xFF to both these registers to clear them at the beginning.

A NOR operation of all the notify signals creates the INTRB (Interrupt bar) signal.

In some of the register pages, there is a second set of status/notify/mask in the address of 0x06, 0x07 and 0x08 respectively. The operation is exactly the same as the registers 0x02, 0x03 and 0x04 respectively.

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 18. Alarm Register Listing

S.NO	Name of Signal	Description	Page Number	Register Address	Bit Number
1	xo_clkloss_dynamic_status	Xo clock dynamic status	00	0x02	2
2	xo_clkloss_dynamic_ntfy	Xo clock dynamic Notify (write 0x04 with 0x02 to enable)	00	0x03	2
3	pll_a_lol_dyn_status	pll lol dynamic status	00	0x06	0
4	pll_b_lol_dyn_status	pll lol dynamic status	00	0x06	1
5	pll_c_lol_dyn_status	pll lol dynamic status	00	0x06	2
6	pll_d_lol_dyn_status	pll lol dynamic status	00	0x06	3
7	pll_a_ho_frz_status	pll ho freeze dynamic status	00	0x06	4
8	pll_b_ho_frz_status	pll ho freeze dynamic status	00	0x06	5
9	pll_c_ho_frz_status	pll ho freeze dynamic status	00	0x06	6
10	pll_d_ho_frz_status	pll ho freeze dynamic status	00	0x06	7
11	pll_a_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x01 to enable)	00	0x07	0
12	pll_b_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x02 to enable)	00	0x07	1
13	pll_c_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x04 to enable)	00	0x07	2
14	pll_d_lol_dyn_ntfy	pll lol dynamic Notify (write 0x08 with 0x08 to enable)	00	0x07	3
15	pll_a_ho_frz_ntfy	pll ho freeze dynamic Notify (write 0x08 with 0x10 to enable)	00	0x07	4
16	pll_b_ho_frz_ntfy	pll ho freeze dynamic Notify (write 0x08 with 0x20 to enable)	00	0x07	5
17	pll_c_ho_frz_ntfy	pll ho freeze dynamic Notify (write 0x08 with 0x40 to enable)	00	0x07	6
18	pll_d_ho_frz_ntfy	pll ho freeze dynamic Notify (write 0x08 with 0x80 to enable)	00	0x07	7
19	in0_clock_loss_dyn_status	Clock Loss Dynamic Status	01	0x02	0
20	in1_clock_loss_dyn_status	Clock Loss Dynamic Status	01	0x02	1
21	in2_clock_loss_dyn_status	Clock Loss Dynamic Status	01	0x02	2
22	in3_clock_loss_dyn_status	Clock Loss Dynamic Status	01	0x02	3
23	in0_clock_loss_fd_dyn_status	Clock Loss with Fd_dynamic Status	01	0x02	4
24	in1_clock_loss_fd_dyn_status	Clock Loss with Fd_dynamic Status	01	0x02	5
25	in2_clock_loss_fd_dyn_status	Clock Loss with Fd_dynamic Status	01	0x02	6
26	in3_clock_loss_fd_dyn_status	Clock Loss with Fd_dynamic Status	01	0x02	7

Table 18. Alarm Register Listing (continued)

S.NO	Name of Signal	Description	Page Number	Register Address	Bit Number
27	in0_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x01 to enable)	01	0x03	0
28	ln1_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x02 to enable)	01	0x03	1
29	ln2_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x04 to enable)	01	0x03	2
30	ln3_clock_loss_ntfy	Clock Loss Dynamic Notify (write 0x04 with 0x08 to enable)	01	0x03	3
31	in0_clock_loss_ntfy	Clock Loss with Fd_dynamic Notify (write 0x04 with 0x10 to enable)	01	0x03	4
32	ln1_clock_loss_ntfy	Clock Loss with Fd_dynamic Notify (write 0x04 with 0x20 to enable)	01	0x03	5
33	ln2_clock_loss_ntfy	Clock Loss with Fd_dynamic Notify (write 0x04 with 0x40 to enable)	01	0x03	6
34	ln3_clock_loss_ntfy	Clock Loss with Fd_dynamic Notify (write 0x04 with 0x80 to enable)	01	0x03	7
35	in0_fd_fine_dyn_status	Frequency drift dynamic status	01	0x06	0
36	ln1_fd_fine_dyn_status	Frequency drift dynamic status	01	0x06	1
37	ln2_fd_fine_dyn_status	Frequency drift dynamic status	01	0x06	2
38	ln3_fd_fine_dyn_status	Frequency drift dynamic status	01	0x06	3
39	in0_fd_coarse_dyn_status	Frequency drift dynamic status	01	0x06	4
40	ln1_fd_coarse_dyn_status	Frequency drift dynamic status	01	0x06	5
41	ln2_fd_coarse_dyn_status	Frequency drift dynamic status	01	0x06	6
42	ln3_fd_coarse_dyn_status	Frequency drift dynamic status	01	0x06	7
43	in0_fd_fine_ntfy	Frequency drift dynamic notify (write 0x08 with 0x01 to enable)	01	0x07	0
44	ln1_fd_fine_ntfy	Frequency drift dynamic notify (write 0x08 with 0x02 to enable)	01	0x07	1
45	ln2_fd_fine_ntfy	Frequency drift dynamic notify (write 0x08 with 0x04 to enable)	01	0x07	2
46	ln3_fd_fine_ntfy	Frequency drift dynamic notify (write 0x08 with 0x08 to enable)	01	0x07	3
47	in0_fd_coarse_ntfy	Frequency drift dynamic notify (write 0x08 with 0x10 to enable)	01	0x07	4
48	ln1_fd_coarse_ntfy	Frequency drift dynamic notify (write 0x08 with 0x20 to enable)	01	0x07	5
49	ln2_fd_coarse_ntfy	Frequency drift dynamic notify (write 0x08 with 0x40 to enable)	01	0x07	6
50	ln3_fd_coarse_ntfy	Frequency drift dynamic notify (write 0x08 with 0x80 to enable)	01	0x07	7
51	fast_lock_dynamic_status_plla	PLL Fast lock dynamic status	0a	0x06	1
52	fast_lock_dynamic_status_pllb	PLL Fast lock dynamic status	0b	0x06	1
53	fast_lock_dynamic_status_pllc	PLL Fast lock dynamic status	0c	0x06	1
54	fast_lock_dynamic_status_plld	PLL Fast lock dynamic status	0d	0x06	1
55	fast_lock_dynamic_ntfy_plla	PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable)	0a	0x07	1
56	fast_lock_dynamic_ntfy_pllb	PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable)	0b	0x07	1
57	fast_lock_dynamic_ntfy_pllc	PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable)	0c	0x07	1
58	fast_lock_dynamic_ntfy_plld	PLL Fast lock dynamic Notify (write 0x08 with 0x02 to enable)	0d	0x07	1
59	ho_valid_dynamic_status_plla	PLL Ho Valid dynamic status	0a	0x06	2
60	ho_valid_dynamic_status_pllb	PLL Ho Valid dynamic status	0b	0x06	2
61	ho_valid_dynamic_status_pllc	PLL Ho Valid dynamic status	0c	0x06	2
62	ho_valid_dynamic_status_plld	PLL Ho Valid dynamic status	0d	0x06	2
63	ho_valid_dynamic_ntfy_plla	PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable)	0a	0x07	2
64	ho_valid_dynamic_ntfy_pllb	PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable)	0b	0x07	2
65	ho_valid_dynamic_ntfy_pllc	PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable)	0c	0x07	2
66	ho_valid_dynamic_ntfy_plld	PLL Ho Valid dynamic Notify (write 0x08 with 0x04 to enable)	0d	0x07	2

Examples for Live Status Read Back

Some examples are presented based on the [Table 18](#) 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)

Table 19. Dynamic registers to read the various alarm registers in the RealTime page

<p>1. Input Clocks CL and FD Related Real Time live status read back</p>	<p>wr_cmd(0xff, 0x01) # Program the CLKMON_SYS page number</p>	<p># Clock Loss dynamic status</p> <p>clock_loss_dyn_status = rd_cmd(0x02) & 0xff</p> <p>(clock_loss_dyn_status >> 0) & 0x01 // IN0 Status for CL, Read bit position [0]</p> <p>(clock_loss_dyn_status >> 1) & 0x01 // IN1 Status for CL, Read bit position [1]</p> <p>(clock_loss_dyn_status >> 2) & 0x01 // IN2 Status for CL, Read bit position [2]</p> <p>(clock_loss_dyn_status >> 3) & 0x01 // IN3 Status for CL, Read bit position [3]</p> <p># Frequency Drift dynamic status</p> <p>fd_fine_dyn_status = rd_cmd(0x06) & 0x0f</p> <p>fine = (fd_fine_dyn_status >> 0) & 0x01 // IN0 Status for Fine FD, Read bit position [0]</p> <p>fine = (fd_fine_dyn_status >> 1) & 0x01 // IN1 Status for Fine FD, Read bit position [1]</p> <p>fine = (fd_fine_dyn_status >> 2) & 0x01 // IN2 Status for Fine FD, Read bit position [2]</p> <p>fine = (fd_fine_dyn_status >> 3) & 0x01 // IN3 Status for Fine FD, Read bit position [3]</p> <p>fd_coarse_dyn_status = rd_cmd(0x06) >> 4</p> <p>coarse = (fd_coarse_dyn_status >> 0) & 0x01 // IN0 Status for Coarse FD, Read bit position [4]</p> <p>coarse = (fd_coarse_dyn_status >> 1) & 0x01 // IN1 Status for Coarse FD, Read bit position [5]</p> <p>coarse = (fd_coarse_dyn_status >> 2) & 0x01 // IN2 Status for Coarse FD, Read bit position [6]</p> <p>coarse = (fd_coarse_dyn_status >> 3) & 0x01 // IN3 Status for Coarse FD, Read bit position [7]</p>
<p>2. PLL Related Real Time live status read back</p>	<p>wr_cmd(0xff, 0x00) # Program the GENERIC_SYS page number, Page 0</p> <p>pll_lol_ho_freeze_dyn_status = rd_cmd(0x06) & 0xff</p>	<p># PLL Lock Loss dynamic status</p> <p>(pll_lol_ho_freeze_dyn_status >> 0) & 0x01 // PLLA Status for LL, Read bit position [0]</p> <p>(pll_lol_ho_freeze_dyn_status >> 1) & 0x01 // PLLB Status for LL, Read bit position [1]</p> <p>(pll_lol_ho_freeze_dyn_status >> 2) & 0x01 // PLLC Status for LL, Read bit position [2]</p> <p>(pll_lol_ho_freeze_dyn_status >> 3) & 0x01 // PLLD Status for LL, Read bit position [3]</p> <p># Holdover Status</p> <p>(pll_lol_ho_freeze_dyn_status >> (0 + 4)) & 0x01 // PLLA Status for HO, Read bit position [4]</p> <p>(pll_lol_ho_freeze_dyn_status >> (1 + 4)) & 0x01 // PLLA Status for HO, Read bit position [5]</p> <p>(pll_lol_ho_freeze_dyn_status >> (2 + 4)) & 0x01 // PLLA Status for HO, Read bit position [6]</p> <p>(pll_lol_ho_freeze_dyn_status >> (3 + 4)) & 0x01 // PLLA Status for HO, Read bit position [7]</p>
<p>3. XO clock loss Related Real Time live status read back</p>	<p>wr_cmd(0xff, 0x00) # Program the GENERIC_SYS page number, Page 0</p>	<p># CLOS_X1X2, XO Clock Loss</p> <p>clos_x1x2 = rd_cmd(0x02) & 0x04 // XO CL Status, Read bit position [2]</p>

Examples of Sticky Bit Clearing

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

In the pseudo code presented below,

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

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

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

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

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

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

Device Initialization for a Non-Programmed Device

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

The SiT95147 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 “Save NVM” 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.

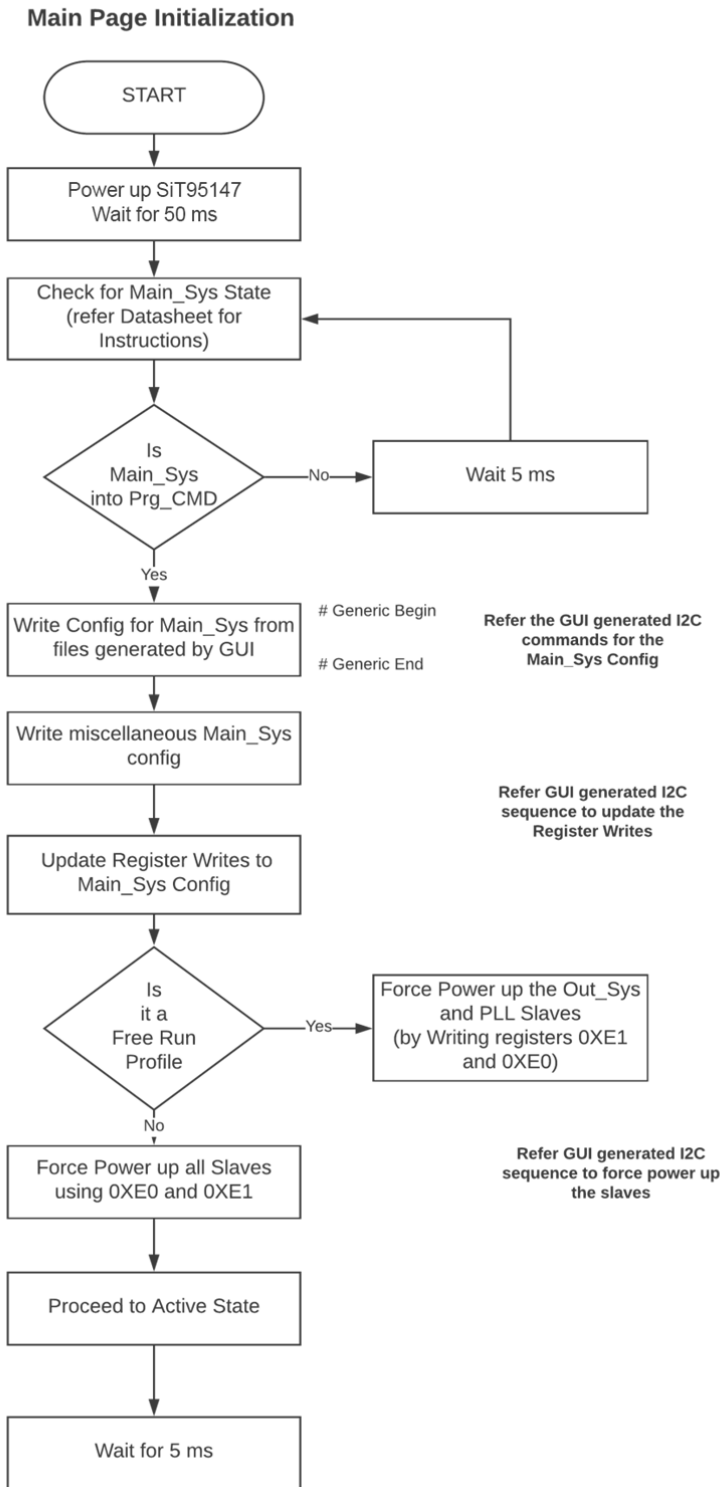


Figure 32. Step 1: Initialize the Main Page – Page 0

We 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.

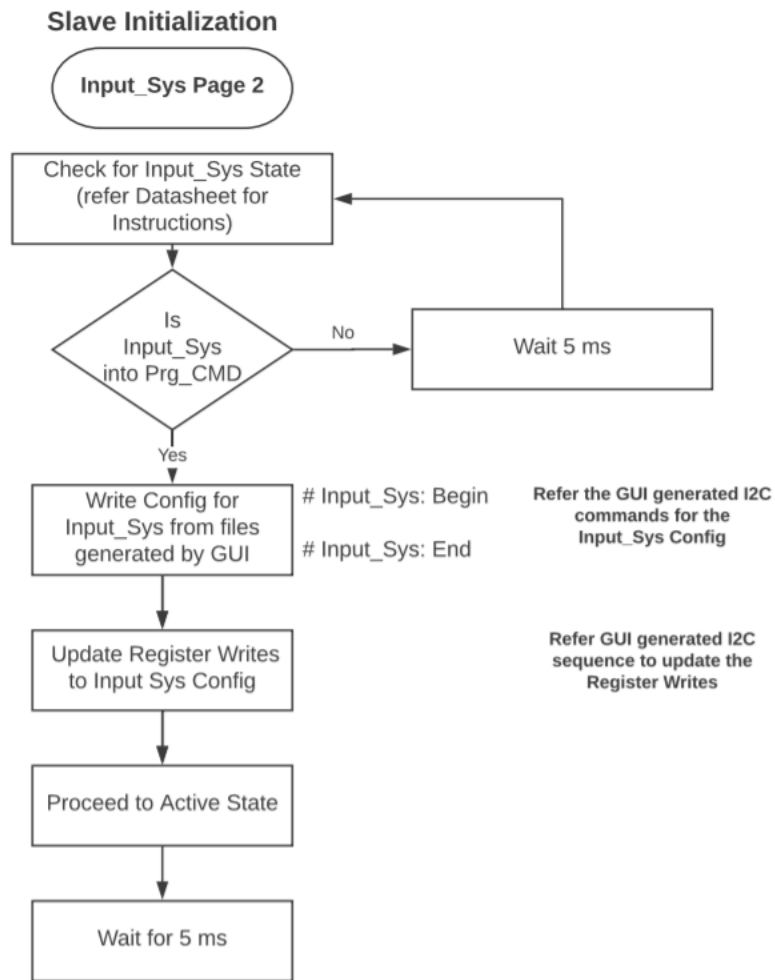


Figure 33. STEP 2: Initialize the Input System Page – Page 2

We 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.

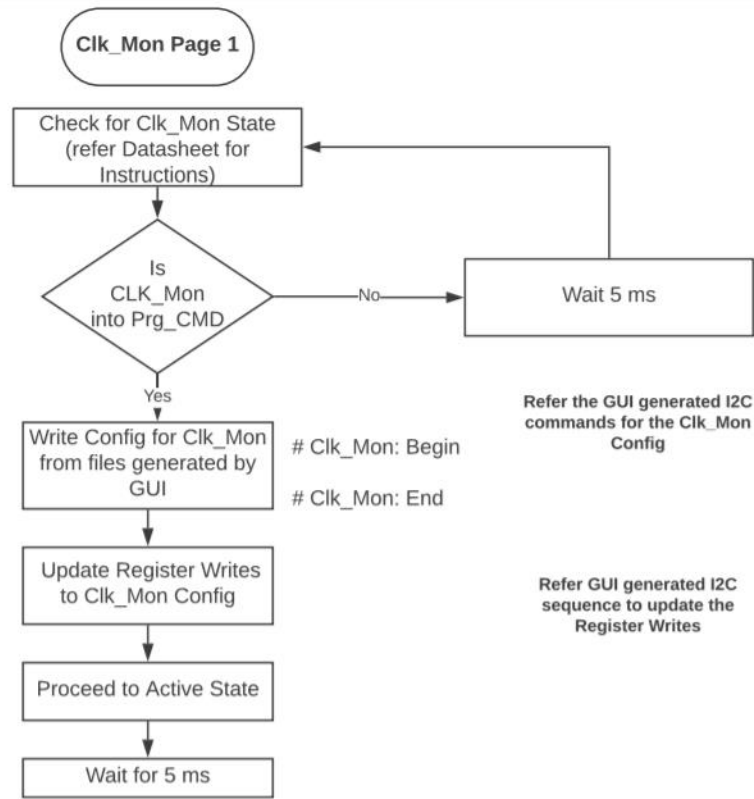


Figure 34. STEP 3: Initialize the Clock Monitor System Page – Page 1

We 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.

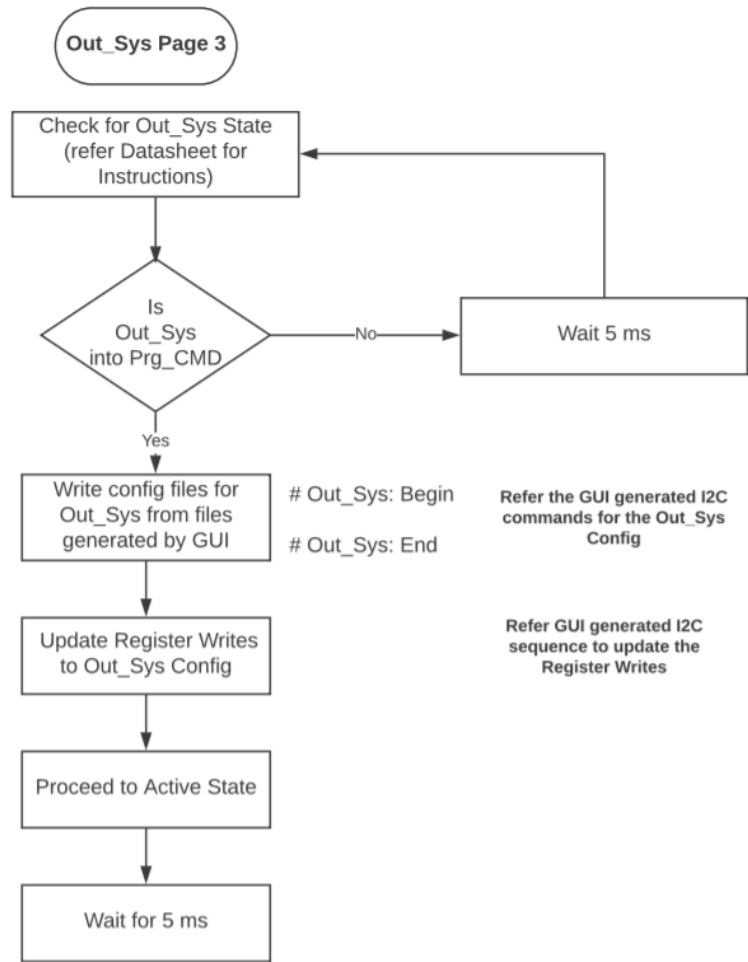


Figure 35. STEP 4: Initialize the Output System Page – Page 3

We 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.

PLL PAGE

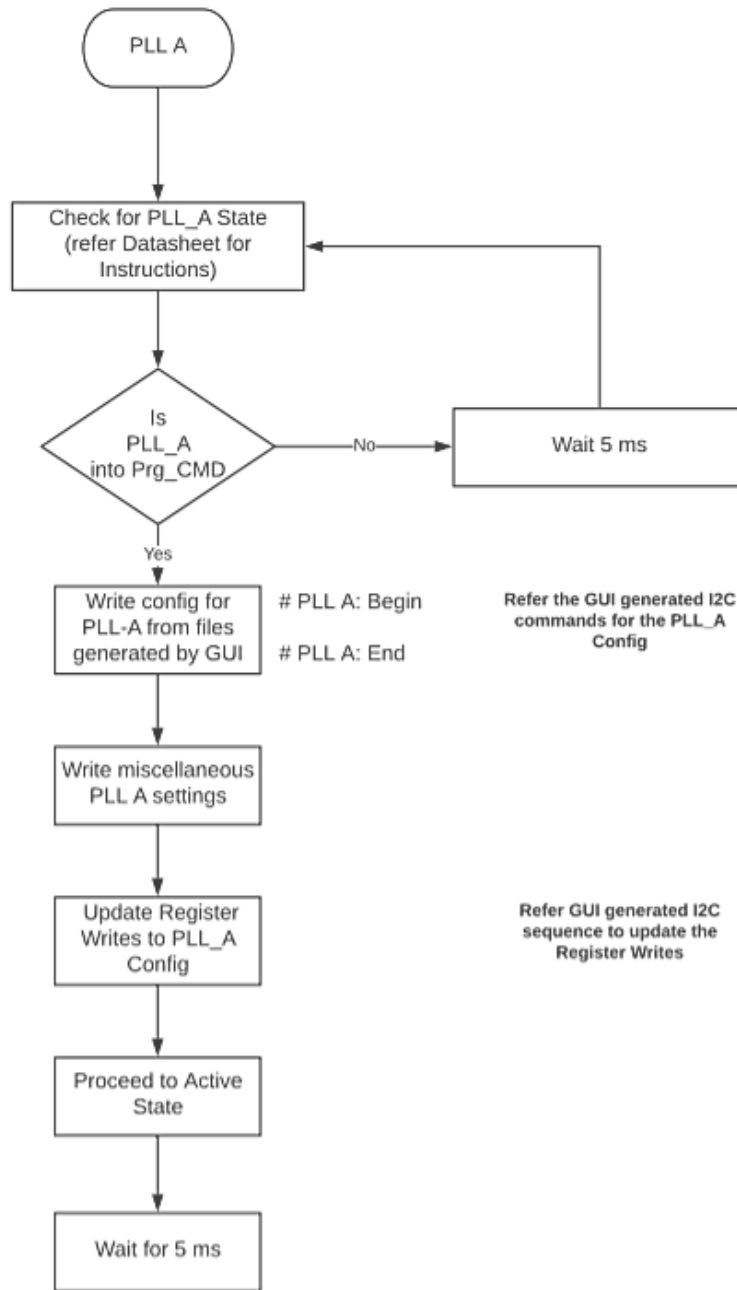


Figure 36. STEP 5: Initialize the PLL A System Page – Page A

We 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.

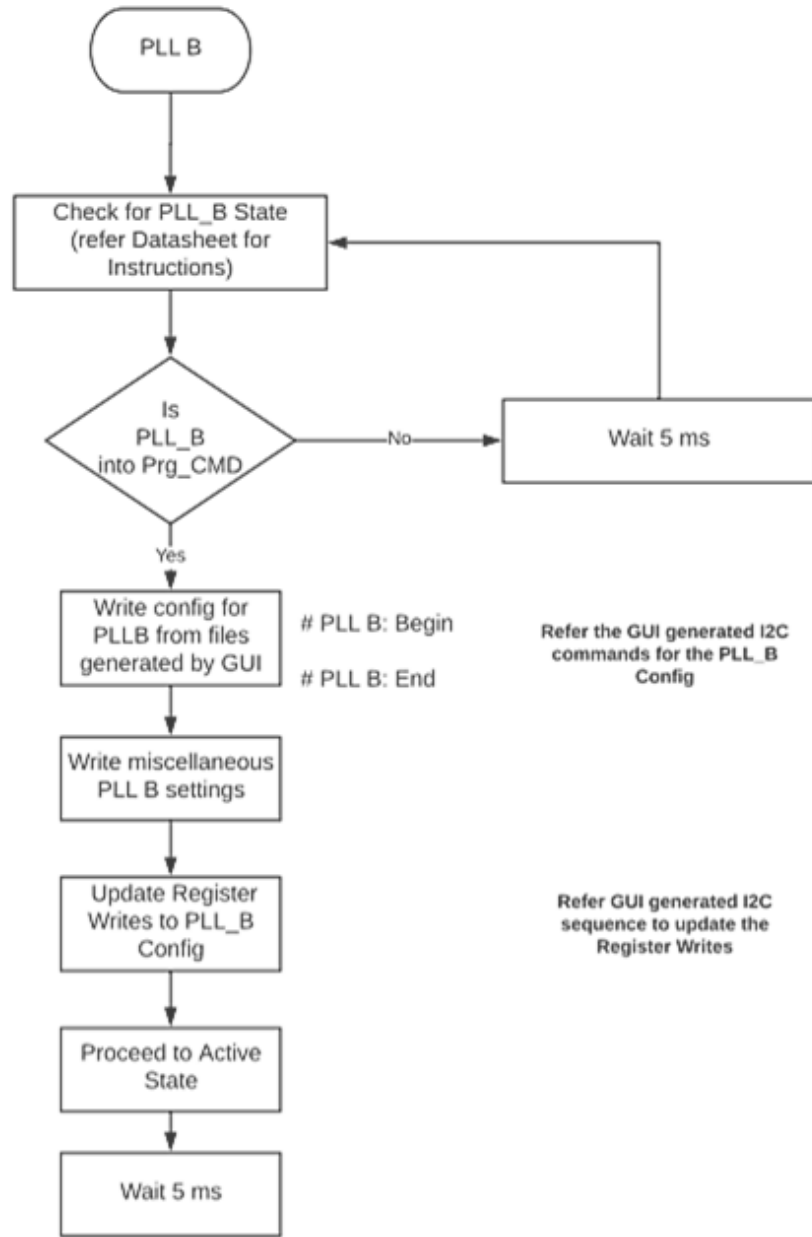


Figure 37. STEP 6: Initialize the PLL B System Page – Page B

We 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.

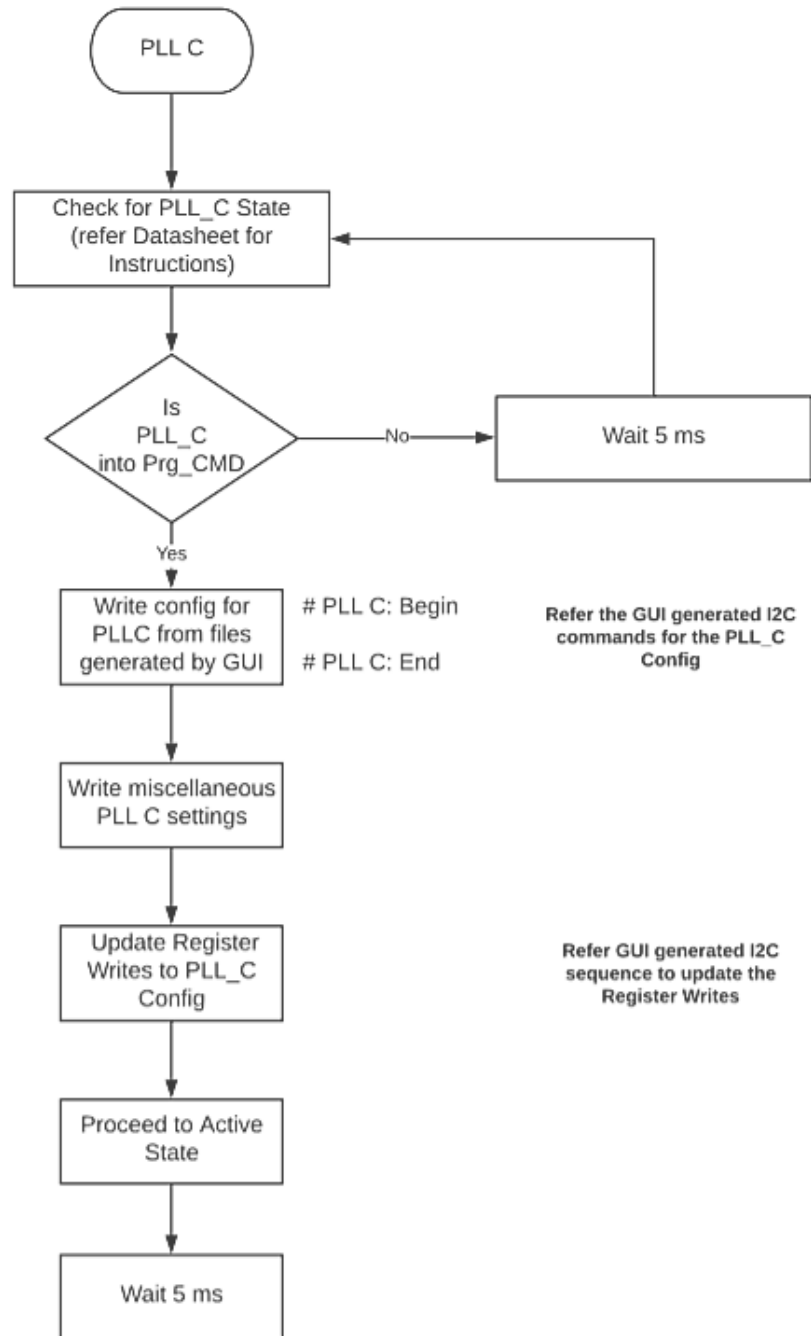


Figure 38. STEP 7: Initialize the PLL C System Page – Page C

We 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.

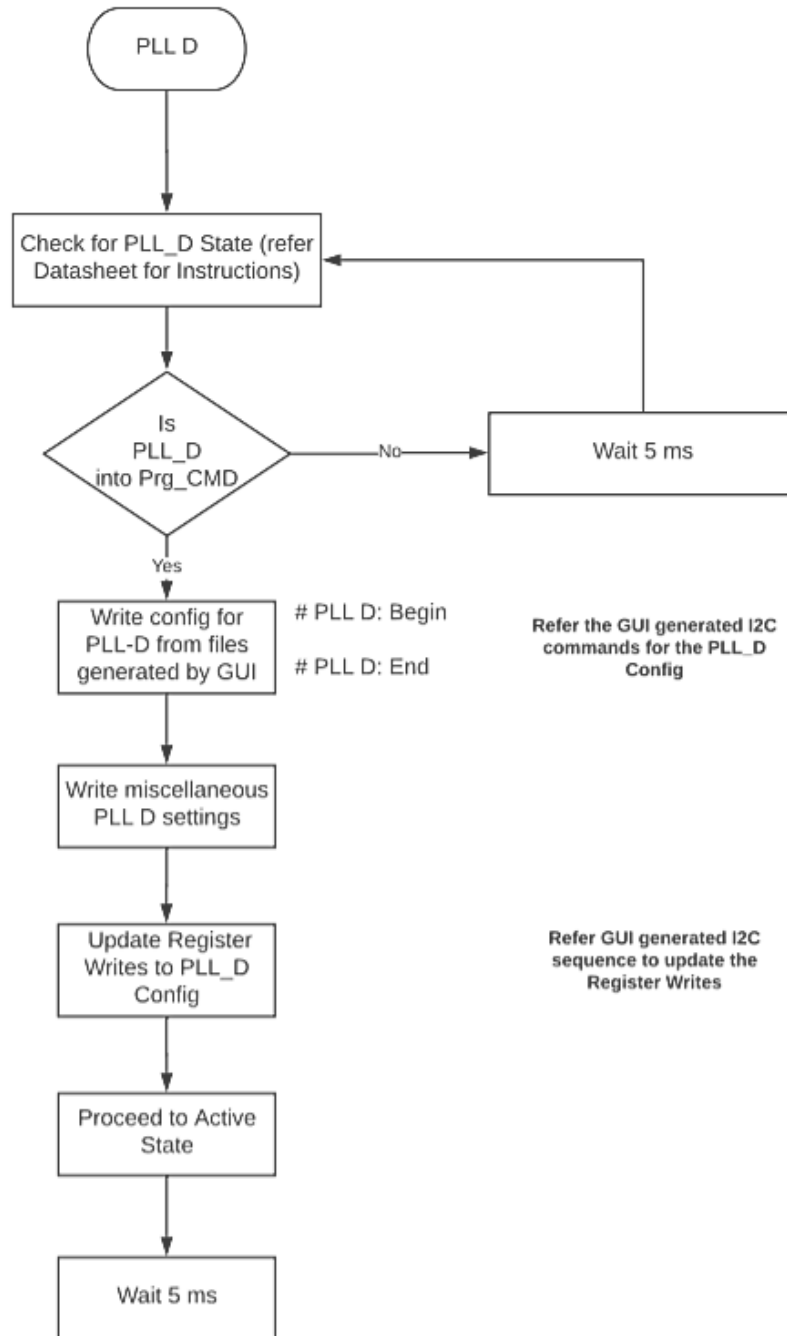


Figure 39. STEP 8: Initialize the PLL D System Page – Page D

We 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.

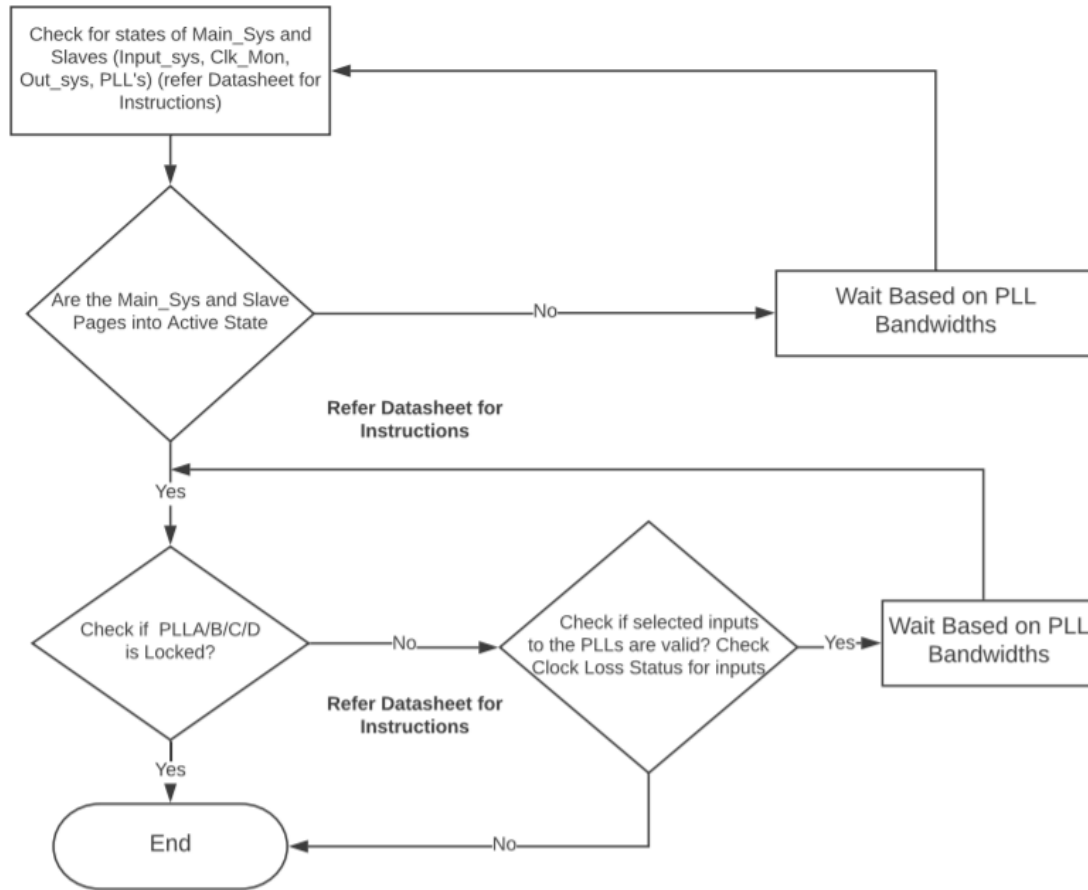


Figure 40. Active State Confirmation

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 SiT95147 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 “Save NVM” 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 0 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 0 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 0 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 Page 0 to 0x20.
- 3) Read the register 0xD0.

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

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

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

Monitoring the Loss of Lock Status for PLL

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

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

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

read data == 0 (PLL is locked)

Monitoring the Hold Over Status for PLLs

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

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

If read data == 1

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

read data == 0

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

- 3) Write 0xB3 register to 0x0D.

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

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

If read data == 1

PLL is in Holdover State

read data == 0

PLL is not in Holdover State

Programming the Primary E-Fuse

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

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

Configuration Bits to Force Power-up of Digital Slave Subsystems

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

This section describes GENERIC_SYS page configuration required in register 0xe0 and 0xe1 to enable respective slave subsystems [INPUT_SYS, CLKMON_SYS, OUTPUT_SYS, PLLA_SYS, PLLB_SYS, PLLC_SYS, PLLD_SYS] as described in [Table 20](#).

Table 20. Configuration Bits to Force Power-up of Digital Slave Subsystems

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

E-Fuse Lock Configuration Bits

This section describes the location of the two bits to lock the respective slave subsystem E-Fuse by writing into the register 0x2F in all pages as described in [Table 21](#).

Table 21. 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 22](#).

Table 22. 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 23](#).

Table 23. 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_position, 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 24](#).

Table 24. 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 25](#).

Table 25. 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 26](#).

Table 26. 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_position, 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 27. PAGE 0: Generic Master 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
VERSION_ID	01h	7:0	R	C0h (SiT95145/7) C1h (SiT95148)	Overall SiTime Platform Revision
STATUS_1_GENERIC	02h	7	R	00h	SPARE
		6	R	00h	SPARE
		5	R	00h	SPARE
		4	R	00h	SPARE
		3	R	00h	SPARE
		2	R	00h	Dynamic status for xoclk_loss,
		1	R	00h	SPARE
		0	R	00h	SPARE
NOTIFY_1_GENERIC	03h	7	R/W	1h	SPARE
		6	R/W	1h	SPARE
		5	R/W	1h	SPARE
		4	R/W	1h	SPARE
		3	R/W	1h	SPARE
		2	R/W	1h	Sticky/Notify status for _xoclk_loss,
		1	R/W	1h	SPARE
		0	R/W	1h	SPARE
MASKb_1_GENERIC	04h	7	R/W	1h	SPARE
		6	R/W	1h	SPARE
		5	R/W	1h	SPARE
		4	R/W	1h	SPARE
		3	R/W	1h	SPARE
		2	R/W	1h	Mask bit for NOTIFY_1_GENERIC (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[2].
		1	R/W	1h	SPARE
		0	R/W	1h	SPARE
Directives_GENERIC	05h	3:2	R/W	00h	0 : DCO_PLLA 1 : DCO_PLLB 2 : DCO_PLLC 3 : DCO_PLLD
		1	R/W	00h	dco increment commonly used for all plls if 0xe7[7] is set as '1'
		0	R/W	00h	dco decrement commonly used for all plls if 0xe7[7] is set as '1'
STATUS_2_GENERIC	06h	7	R	00h	Dynamic status for plld_ho_freeze
		6	R	00h	Dynamic status for pll_c_ho_freeze
		5	R	00h	Dynamic status for pllb_ho_freeze
		4	R	00h	Dynamic status for plla_ho_freeze
		3	R	00h	Dynamic status for plld_loss_of_lock
		2	R	00h	Dynamic status for pll_c_loss_of_lock
		1	R	00h	Dynamic status for pllb_loss_of_lock

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		0	R	00h	Dynamic status for pll_loss_of_lock
NOTIFY_2_GENERIC	07h	7	R/W	1h	Sticky/Notify status for plld_ho_freeze
		6	R/W	1h	Sticky/Notify status for pll_c_ho_freeze
		5	R/W	1h	Sticky/Notify status for pllb_ho_freeze
		4	R/W	1h	Sticky/Notify status for plla_ho_freeze
		3	R/W	1h	Sticky/Notify status for plld_loss_of_lock
		2	R/W	1h	Sticky/Notify status for pll_c_loss_of_lock
		1	R/W	1h	Sticky/Notify status for 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
NVMPLEN_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)}
		3:0	R/W	0h	Bits {3,2,1,0} correspond to enable of PLL {D,C,B,A}: Active Low.
NVMFLEXIO8_GENERIC	18h	7	R/W	0h	Select the direction of FLEXIO14. '0' – Input, '1' – Output

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		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/ + Freq drift coarse for clkin 0/1/2/3 + Freq drift coarse for clkin 0/1/2/3 + Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b010 : Ored_all_pll_ntfy (Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b011 : Ored_all_clkmon_ntfy ((clock loss for clkin0/1/2/3/ + Freq drift coarse for clkin 0/1/2/3 + Freq drift coarse for clkin 0/1/2/3) 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/ + Freq drift coarse for clkin 0/1/2/3 + Freq drift coarse for clkin 0/1/2/3 + Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b010 : Ored_all_pll_ntfy (Loss of lock for PLL A/B/C/D+HO freeze for PLL A/B/C/D) 3'b011 : Ored_all_clkmon_ntfy ((clock loss for clkin0/1/2/3/ + Freq drift coarse for clkin 0/1/2/3 + Freq drift coarse for clkin 0/1/2/3) 3'b100 : Dynamic_Clock_Monitoring_Status 3'b101 : Internal debug 3'b110 : Internal debug 3'b111 : Internal debug
NVMSPPARE1_GENERIC	19h	7	R/W	0h	if set '1', Enables vdd padding functionality(0x23[7]) for padding selection. Keeping at 0 enables pad rail switch, select 1 to ensure no switch and keep rail to left
		6	R/W	0h	To map the external clock in select and allow the each pll independetly to force into holdover mode. Each pll ctrl - refer 0x2B[3:0]
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.
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

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		3:2	R/W	0h	To program delay between enabling PLLA and other PLLs in fuse locked modes. 2'b00 - 4ms 2'b01 - 40 ms 2'b10 - 400 ms 2'b11 - 4s
		1	R/W	0h	if set as '1' then set PLLX_OEB as '1' if loss of lock status is asserted for respective PLL
		0	R/W	0h	if set as '1' then set PLLX_OEB as '1' if there is XO clock loss
INTR_MASK_1_CONFIG	27h	7	R/W	0h	If set '1', Interrupt will not be generated for DRIFT in freq for clk_in3
		6	R/W	0h	If set '1', Interrupt will not be generated for DRIFT in freq for clk_in2
		5	R/W	0h	If set '1', Interrupt will not be generated for DRIFT in freq for clk_in1
		4	R/W	0h	If set '1', Interrupt will not be generated for DRIFT in freq for clk_in0
		3	R/W	0h	If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in3
		2	R/W	0h	If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in2
		1	R/W	0h	If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in1
		0	R/W	0h	If set '1', Interrupt will not be generated for CLOCK LOSS for clk_in0
INTR_MASK_2_CONFIG	28h	7	R/W	0h	If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLA
		6	R/W	0h	If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLB
		5	R/W	0h	If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLC
		4	R/W	0h	If set '1', Interrupt will not be generated for LOSS OF LOCK for PLLD
		3	R/W	0h	If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLA
		2	R/W	0h	If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLB
		1	R/W	0h	If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLC
		0	R/W	0h	If set '1', Interrupt will not be generated for HOLDOVER FREEZE for PLLD
CHIP_FLEXIO_CONFIG	29h	7	R/W	0h	Reserved
		6	R/W	0h	
		5	R/W	0h	
		4	R/W	0h	
		3	R/W	0h	
		2:0	R/W	0h	010: SiT95147 110: SiT95148 011: SiT95146 001: SiT95145 100: SiT95144 101: SiT95142
I2C_GENERIC	2Ah	7	R/W	0h	Enable the new I2C Address (if changed from the default 0x69)
		6:0	R/W	0h	I2C Address (when changed from the default 0x6D)
XO2_GENERIC	2Dh	7:6	R/W	0h	OT3 GM programmability: Frequency selectivity configuration settings to support different frequency range of 3rd OT crystals

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		5:0	R/W	0h	Crystal frequency trim settings
XO3_GENERIC	2Eh	7	R/W	0h	Crystal Pathway Settings: Use defaults from GUI based on crystal type and pathway used.
		6	R/W	0h	
		5:3	R/W	0h	
		2:0	R/W	0h	
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	Crystal Pathway Settings: Use defaults from GUI based on crystal type and pathway used.
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 28. PAGE 1: Clock Monitor System Related registers:
Registers from 11h to 4Fh are equivalent NVMCopy Registers for this Page

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
STATUS_CLKMON	02h	7	R	00h	Dynamic status for clk_in3_loss_OR_with_FD
		6	R	00h	Dynamic status for clk_in2_loss_OR_with_FD
		5	R	00h	Dynamic status for clk_in1_loss_OR_with_FD
		4	R	00h	Dynamic status for clk_in0_loss_OR_with_FD
		3	R	00h	Dynamic status for clk_in3_loss
		2	R	00h	Dynamic status for clk_in2_loss
		1	R	00h	Dynamic status for clk_in1_loss
		0	R	00h	Dynamic status for clk_in0_loss
NOTIFY_CLKMON	03h	7	R/W	1h	Sticky/Notify status for clk_in3_loss_OR_with_FD
		6	R/W	1h	Sticky/Notify status for clk_in2_loss_OR_with_FD
		5	R/W	1h	Sticky/Notify status for clk_in1_loss_OR_with_FD
		4	R/W	1h	Sticky/Notify status for clk_in0_loss_OR_with_FD
		3	R/W	1h	Sticky/Notify status for clk_in3_loss
		2	R/W	1h	Sticky/Notify status for clk_in2_loss
		1	R/W	1h	Sticky/Notify status for clk_in1_loss
		0	R/W	1h	Sticky/Notify status for clk_in0_loss
MASKb_CLKMON	04h	7	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[7]
		6	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[6]
		5	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[5]
		4	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[4]
		3	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[3]
		2	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[2]
		1	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[1]
		0	R/W	1h	Mask bit for NOTIFY_CLKMON (03h) If programmed as '0': Mask sticky/Notify bit generation for 03h[0]
STATUS_FD_COARSE_CLKMON	06h	7	R	00h	Dynamic status for clk3_freq_coarse_drifted
		6	R	00h	Dynamic status for clk2_freq_coarse_drifted
		5	R	00h	Dynamic status for clk1_freq_coarse_drifted
		4	R	00h	Dynamic status for clk0_freq_coarse_drifted
		3	R	00h	Dynamic status for clk3_freq_fine_drifted
		2	R	00h	Dynamic status for clk2_freq_fine_drifted
		1	R	00h	Dynamic status for clk1_freq_fine_drifted

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		0	R	00h	Dynamic status for clk0_freq_fine_drifted
NOTIFY_FDCOARSE_CLKMON	07h	7	R/W	1h	Sticky/Notify status for clk3_freq_coarse_drifted
		6	R/W	1h	Sticky/Notify status for clk2_freq_coarse_drifted
		5	R/W	1h	Sticky/Notify status for clk1_freq_coarse_drifted
		4	R/W	1h	Sticky/Notify status for clk0_freq_coarse_drifted
		3	R/W	1h	Sticky/Notify status for clk3_freq_fine_drifted
		2	R/W	1h	Sticky/Notify status for clk2_freq_fine_drifted
		1	R/W	1h	Sticky/Notify status for clk1_freq_fine_drifted
		0	R/W	1h	Sticky/Notify status for clk0_freq_fine_drifted
MASKb_FDCOARSE_CLKMON	08h	7	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[7]
		6	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[6]
		5	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[5]
		4	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[4]
		3	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[3]
		2	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[2]
		1	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[1]
		0	R/W	1h	Mask bit for NOTIFY_FDCOARSE_CLKMON (07h) If programmed as '0': Mask sticky/Notify bit generation for 07h[0]
FD32_STATUS_COARSE_CLKMON	0Dh	7:0	R/W	0h	Reserved
FD10_STATUS_COARSE_CLKMON	0Eh	7:0	R/W	0h	Reserved
PRG_Directives_CLKMON	0Fh	7:3	R/W	0h	PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active
		2	R/W	0h	Spare
		1	R/W	0h	Escape to the PRG_CMD state from ACTIVE state
CL_REG2_CLKMON	11h	7:6	R/W	0h	IN3_VAL_TIME: timer setting for deassertion of clock loss for FD3 Values {0,1,2,3} correspond to {2ms, 100ms, 200ms, 1sec}
		5:4	R/W	0h	IN2_VAL_TIME: timer setting for deassertion of clock loss for FD2 Values {0,1,2,3} correspond to {2ms, 100ms, 200ms, 1sec}

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		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 GUI calculates the correct CL register threshold trigger value for IN2
CL2_SET_THR1_CLKMON	16h	7:0	R/W	0h	
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 GUI calculates the correct CL register threshold trigger value for IN1
CL1_SET_THR1_CLKMON	19h	7:0	R/W	0h	
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 GUI calculates the correct CL register threshold trigger value for IN0
CL0_SET_THR1_CLKMON	1Ch	7:0	R/W	0h	
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 GUI calculates the correct CL register threshold clear value for IN3
CL3_CLR_THR1_CLKMON	1Fh	7:0	R/W	0h	
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 GUI calculates the correct CL register threshold clear value for IN2
CL2_CLR_THR1_CLKMON	22h	7:0	R/W	0h	
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 GUI calculates the correct CL register threshold clear value for IN1
CL1_CLR_THR1_CLKMON	25h	7:0	R/W	0h	
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 GUI calculates the correct CL register threshold clear value for IN0
CL0_CLR_THR1_CLKMON	28h	7:0	R/W	0h	
CL0_CLR_THR0_CLKMON	29h	7:0	R/W	0h	
FD_REG1_CLKMON	2Ah	7:4	R/W	0h	FD0COARSE Enable Bit Value of position {3,2,1,0} correspond to Inputs {3,2,1,0}
		3:0	R/W	0h	FD0FINE Enable Bit Value of position {3,2,1,0} correspond to Inputs {3,2,1,0}
FD_REG2_CLKMON	2Bh	7	R/W	0h	SPARE
		6:3	R/W	0h	XO divider configuration for FD monitors. GUI configures these dividers

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		2:0	R/W	0h	Golden reference clock selection for frequency drift monitors. Value of 4 corresponds to the XO Reference. Value of {3,2,1,0} correspond to Inputs {3,2,1,0}
FD_REG3_CLKMON	2Ch	7:4	R/W	0h	FD3 divider configuration for FD monitors. GUI configures these dividers
		3:0	R/W	0h	FD2 divider configuration for FD monitors. GUI configures these dividers
FD_REG4_CLKMON	2Dh	7:4	R/W	0h	FD1 divider configuration for FD monitors. GUI configures these dividers
		3:0	R/W	0h	FD0 divider configuration for FD monitors. GUI configures these dividers
FXOBYFIN3_LOG2_BAND	2Eh	7:4	R/W	0h	SPARE
		3:0	R/W	0h	IN3 FD monitor configuration bits calculated by GUI
FXOBYFIN2_LOG2_BAND	2Fh	7:6	R/W	0h	PATTERN : {01/10} => Efuse Locked, {00/11} => Efuse NOT Locked
		5:4	R/W	0h	SPARE
		3:0	R/W	0h	IN2 FD monitor configuration bits calculated by GUI
FXOBYFIN1_FIN0_LOG2_BAND	30h	7:4	R/W	0h	IN1 FD monitor configuration bits calculated by GUI
		3:0	R/W	0h	IN0 FD monitor configuration bits calculated by GUI
PLLA_ACT_SPARE_SEL	31h	7:6	R/W	0h	PLLA ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		5:4	R/W	0h	PLLA SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		3:2	R/W	0h	PLLA SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		1:0	R/W	0h	PLLA SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
FD3_CLR_THRFINE_CLKMON	32h	7:0	R/W	0h	IN3 FD clear threshold is calculated as 2X of FD3_CLR_THRFINE_CLKMON
FD2_CLR_THRFINE_CLKMON	33h	7:0	R/W	0h	IN2 FD clear threshold is calculated as 2X of FD2_CLR_THRFINE_CLKMON
FD1_CLR_THRFINE_CLKMON	34h	7:0	R/W	0h	IN1 FD clear threshold is calculated as 2X of FD1_CLR_THRFINE_CLKMON
FD0_CLR_THRFINE_CLKMON	35h	7:0	R/W	0h	IN0 FD clear threshold is calculated as 2X of FD0_CLR_THRFINE_CLKMON
FD32_CLR_THRCOARSE_CLKMON	36h	7:4	R/W	0h	IN3 FD clear threshold is calculated as 100*(FD3_CLR_THRCOARSE_CLKMON +1)
		3:0	R/W	0h	IN2 FD clear threshold is calculated as 100*(FD2_CLR_THRCOARSE_CLKMON +1)
FD10_CLR_THRCOARSE_CLKMON	37h	7:4	R/W	0h	IN1 FD clear threshold is calculated as 100*(FD1_CLR_THRCOARSE_CLKMON +1)
		3:0	R/W	0h	IN0 FD clear threshold is calculated as 100*(FD0_CLR_THRCOARSE_CLKMON +1)
FD32_SET_THRCOARSE_CLKMON	38h	7:4	R/W	0h	IN3 FD set threshold is calculated as 100*(SET FD3_SET_THRCOARSE_CLKMON+1)
		3:0	R/W	0h	IN2 FD set threshold is calculated as 100*(FD2_SET_THRCOARSE_CLKMON +1)

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
FD10_SET_THRCOARSE_CLKMON	39h	7:4	R/W	0h	IN1 FD set threshold is calculated as $100 * (\text{SET_TH_FD1_SET_THRCOARSE_CLKMON} + 1)$
		3:0	R/W	0h	IN0 FD set threshold is calculated as $100 * (\text{FD0_SET_THRCOARSE_CLKMON} + 1)$
FD3_FGBYFM2_CLKMON	3Ah	7:0	R/W	0h	IN3 FD monitor configuration bits calculated by GUI
FD3_FGBYFM1_CLKMON	3Bh	7:0	R/W	0h	IN3 FD monitor configuration bits calculated by GUI
FD3_FGBYFM0_CLKMON	3Ch	7:0	R/W	0h	IN3 FD monitor configuration bits calculated by GUI
FD2_FGBYFM2_CLKMON	3Dh	7:0	R/W	0h	IN2 FD monitor configuration bits calculated by GUI
FD2_FGBYFM1_CLKMON	3Eh	7:0	R/W	0h	IN2 FD monitor configuration bits calculated by GUI
FD2_FGBYFM0_CLKMON	3Fh	7:0	R/W	0h	IN2 FD monitor configuration bits calculated by GUI
FD1_FGBYFM2_CLKMON	40h	7:0	R/W	0h	IN1 FD monitor configuration bits calculated by GUI
FD1_FGBYFM1_CLKMON	41h	7:0	R/W	0h	IN1 FD monitor configuration bits calculated by GUI
FD1_FGBYFM0_CLKMON	42h	7:0	R/W	0h	IN1 FD monitor configuration bits calculated by GUI
FD0_FGBYFM2_CLKMON	43h	7:0	R/W	0h	IN0 FD monitor configuration bits calculated by GUI
FD0_FGBYFM1_CLKMON	44h	7:0	R/W	0h	IN0 FD monitor configuration bits calculated by GUI
FD0_FGBYFM0_CLKMON	45h	7:0	R/W	0h	IN0 FD monitor configuration bits calculated by GUI
CLKX_HITLESS_SW_SOURCE	46h	7:6	R/W	0h	0 : CL3 1 : CL3 + FD3 COARSE 2: CL3 + FD3 FINE 3 : CL3 + FD3 COARSE + FD3 FINE
		5:4	R/W	0h	0 : CL2 1 : CL2 + FD2 COARSE 2: CL2 + FD2 FINE 3 : CL2 + FD2 COARSE + FD2 FINE
		3:2	R/W	0h	0 : CL1 1 : CL1 + FD1 COARSE 2: CL1 + FD1 FINE 3 : CL1 + FD1 COARSE + FD1 FINE
		1:0	R/W	0h	0 : CL0 1 : CL0 + FD0 COARSE 2: CL0 + FD0 FINE 3 : CL0 + FD0 COARSE + FD0 FINE
PLL_B_ACT_SPARE_SEL	49h	7:6	R/W	0h	PLL_B ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		5:4	R/W	0h	PLL_B SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		3:2	R/W	0h	PLL_B SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		1:0	R/W	0h	PLL_B SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
PLL_C_ACT_SPARE_SEL	4Ah	7:6	R/W	0h	PLL_C ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		5:4	R/W	0h	PLL_C SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		3:2	R/W	0h	PLL SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		1:0	R/W	0h	PLL SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
PLLD_ACT_SPARE_SEL	4Bh	7:6	R/W	0h	PLLD ACTIVE Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		5:4	R/W	0h	PLLD SPARE0 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		3:2	R/W	0h	PLLD SPARE1 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
		1:0	R/W	0h	PLLD SPARE2 Clock Selection 0 : CLKIN0, 1 : CLKIN1, 2 : CLKIN2, 3 : CLKIN3
FD3_SET_THRFINE_CLKMON	4Ch	7:0	R/W	0h	IN3 FD set threshold is calculated as 2X of FD3_SET_THRFINE_CLKMON
FD2_SET_THRFINE_CLKMON	4Dh	7:0	R/W	0h	IN2 FD set threshold is calculated as 2X of FD2_SET_THRFINE_CLKMON
FD1_SET_THRFINE_CLKMON	4Eh	7:0	R/W	0h	IN1 FD set threshold is calculated as 2X of FD1_SET_THRFINE_CLKMON
FD0_SET_THRFINE_CLKMON	4Fh	7:0	R/W	0h	IN0 FD set threshold is calculated as 2X of FD0_SET_THRFINE_CLKMON_THR
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 29. 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	Enable the Frequency Ramp FD Monitoring
		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
		7:6	R/W	0h	IN0 DIVN1 Divider Integer Value
		5	R/W	0h	IN0 Direct Bypass for DIVN1
CLKIN0_CFG1_INPAGE	1Ah	4	R/W	0h	IN0 Single Ended Enable, Default is Differential
		7:0	R/W	0h	IN0 FD Ramp monitor configuration bits calculated by GUI
		1Bh	7:0	R/W	
1Ch	7:0	R/W			
CLKIN0_CFG2_INPAGE	1Bh	7:0	R/W	0h	IN0 FD Ramp monitor configuration bits calculated by GUI
CLKIN0_CFG3_INPAGE	1Ch	7:0	R/W	0h	
CLKIN0_CFG4_INPAGE	1Dh	7:0	R/W	0h	
CLKIN0_OUTSEL_INPAGE	1Eh	7:4	R/W	0h	XO divider configuration for FD Ramp monitors. GUI configures these dividers
		3:0	R/W	0h	IN0 divider configuration for FD Ramp monitors. GUI configures these dividers
CLKIN0_RAMPDF_MEAS_COUNT	1Fh	7	R/W	0h	SPARE
		6:5	R/W	0h	SPARE
		4:0	R/W	0h	Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI.
CLKIN1_DIVN1_INT1_INPAGE	20h	7:0	R/W	0h	IN1 DIVN1 Divider Integer Value
CLKIN1_DIVN1_INT2_INPAGE	21h	7	R/W	0h	IN1 DIVN1 Integer Mode of Division
		6	R/W	0h	SPARE
		5:0	R/W	0h	IN1 DIVN1 Divider Integer Value
CLKIN1_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

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
CLKIN1_DIVN1_FRACN3_IN PAGE	24h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Numerator
CLKIN1_DIVN1_FRACN4_IN PAGE	25h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Numerator
CLKIN1_DIVN1_FRACD1_IN PAGE	26h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Denominator
CLKIN1_DIVN1_FRACD2_IN PAGE	27h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Denominator
CLKIN1_DIVN1_FRACD3_IN PAGE	28h	7:0	R/W	0h	IN1 DIVN1 Divider Fractional Value Denominator
CLKIN1_DIVN1_FRACD4_IN PAGE	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_CFG2_INPAGE	2Bh	7:0	R/W	0h	IN1 FD Ramp monitor configuration bits calculated by GUI
CLKIN1_CFG3_INPAGE	2Ch	7:0	R/W	0h	
CLKIN1_CFG4_INPAGE	2Dh	7:0	R/W	0h	
		3:0	R/W	0h	
CLKIN1_RAMPFD_MEAS_COUNT	2Fh	7:6	R/W	0h	PATTERN :{'01'/'10'} =] Efuse Locked , {'00'/'11'} =] Efuse NOT Locked
		4:0	R/W	0h	Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI.
CLKIN2_DIVN1_INT1_INPAGE	30h	7:0	R/W	0h	IN2 DIVN1 Divider Integer Value
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_IN PAGE	32h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACN2_IN PAGE	33h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACN3_IN PAGE	34h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACN4_IN PAGE	35h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Numerator
CLKIN2_DIVN1_FRACD1_IN PAGE	36h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Denominator
CLKIN2_DIVN1_FRACD2_IN PAGE	37h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Denominator
CLKIN2_DIVN1_FRACD3_IN PAGE	38h	7:0	R/W	0h	IN2 DIVN1 Divider Fractional Value Denominator
CLKIN2_DIVN1_FRACD4_IN PAGE	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
CLKIN2_CFG2_INPAGE	3Bh	7:0	R/W	0h	IN2 FD Ramp monitor configuration bits calculated by GUI
CLKIN2_CFG3_INPAGE	3Ch	7:0	R/W	0h	
CLKIN2_CFG4_INPAGE	3Dh	7:0	R/W	0h	
CLKIN2_OUTSEL_INPAGE	3Eh	7:4	R/W	0h	Internal FD Ramp related settings
		3:0	R/W	0h	IN2 divider configuration for FD Ramp monitors. GUI configures these dividers

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		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_CFG2_INPAGE	4Bh	7:0	R/W	0h	IN3 FD Ramp monitor configuration bits calculated by GUI
CLKIN3_CFG3_INPAGE	4Ch	7:0	R/W	0h	
CLKIN3_CFG4_INPAGE	4Dh	7:0	R/W	0h	
CLKIN3_OUTSEL_INPAGE	4Eh	7:6	R/W	0h	SPARE
		5:4	R/W	0h	SPARE
		3:0	R/W	0h	IN3 divider configuration for FD Ramp monitors. GUI configures these dividers
CLKIN3_RAMPDF_MEAS_COUNT	4Fh	4:0	R/W	0h	Exponent that determines the ideal measurement count for Ramp: Calculated by the GUI.
PAGE_NUMBER	FFh	7:0	R/W	0h	On all pages register FF is a READ / WRITE register used to change the page number

Table 30. PAGE 3: Output System and Output Dividers Related Registers.

Registers from 18h to 67h are equivalent NVMCopy Registers for this Page

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
PRG_Directives_OUTPAGE	0Fh	7:3	R/W	0h	PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active
		2	R/W	0h	Spare
		1	R/W	0h	Escape to the PRG_CMD state from ACTIVE state
		2	R/W	0h	
		1	R/W	0h	
		0	R/W	0h	
DIVO1_DIV2_OUTPAGE	18h				
		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
		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 HCSSL 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
		2	R/W	0h	{0, 0, 1, 0} Internal Pull Up, Differential Output
		1	R/W	0h	{0, 0, 0, 1} Internal Pull Dn, Differential Output
		0	R/W	0h	{0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN
DIVO2_DIV2_OUTPAGE	20h				
		3:0	R/W	0h	DIVO Divider for this output: Bits [19:16]

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
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.
		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
		2	R/W	0h	{0, 0, 1, 0} Internal Pull Up, Differential Output
		1	R/W	0h	{0, 0, 0, 1} Internal Pull Dn, Differential Output
		0	R/W	0h	{0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN
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, 2, 1, 0 times 30 ps based on this 2 bit code

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		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 HCSSL 0b000: LVDS (Can be AC Coupled or DC Coupled) 0b100: Boosted LVDS (Use for LVPECL like swings with AC Coupled loads) 0b001: DC Coupled LVPECL (with Common mode current) 0b101: DC Coupled LVPECL2 (without Common mode current)
DIVO3_MISC0_OUTPAGE	2Fh	7:6	R/W	0h	PATTERN : {01/'10'} =] Efuse Written, {00/'11'} =] Efuse NOT Written
		4	R/W	0h	PRG_DELAY[8:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 4,3, 2, 1, 0 times 30 ps based on this 3 bit code
		3	R/W	0h	{0, 0, 0, 0} External Termination, Differential Output
		2	R/W	0h	{0, 0, 1, 0} Internal Pull Up, Differential Output
		1	R/W	0h	{0, 0, 0, 1} Internal Pull Dn, Differential Output
		0	R/W	0h	{0, 1, 0, 0} CMOS On OutP, Nothing on OutN
		0	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				
		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
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
		4:3	R/W	0h	VDD_DEF : VDD {1.8(00), 2.5(01), 3.3(10)}. VDD Definition for this particular output.

Reg 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
		2	R/W	0h	{0, 0, 1, 0} Internal Pull Up, Differential Output
		1	R/W	0h	{0, 0, 0, 1} Internal Pull Dn, Differential Output
		0	R/W	0h	{0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN
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 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
		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

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		3	R/W	0h	{0, 0, 0, 0} External Termination, Differential Output
		2	R/W	0h	{0, 0, 1, 0} Internal Pull Up, Differential Output
		1	R/W	0h	{0, 0, 0, 1} Internal Pull Dn, Differential Output
		0	R/W	0h	{0, 1, 0, 0} CMOS On OutP, Nothing on OutN
DIVO7_DIV2_OUTPAGE	48h				
		3:0	R/W	0h	DIVO Divider for this output: Bits [19:16]
DIVO7_DIV1_OUTPAGE	49h	7:0	R/W	0h	DIVO Divider for this output: Bits [15:8]
DIVO7_DIV0_OUTPAGE	4Ah	7:0	R/W	0h	DIVO Divider for this output: Bits [7:0]
DIVO7_PROG1_OUTPAGE	4Bh	7:6	R/W	0h	CMOS Driver Phase Selection phase_sel<1> phase_sel<0> ODR_P ODR_N 0 0 CLKP CLKN 0 1 CLKP CLKP 1 0 CLKN CLKN 1 1 CLKN CLKP
		2:0	R/W	0h	Single Ended Driver Programming of strength: Use 0b111= 0d7
DIVO7_PROG0_OUTPAGE	4Ch	7:0	R/W	0h	Programmable Output Delay PRG_DELAY[5:0] is the coarse delay on the clock output. It determines relative delay on this clock programmable from 0 to 63 VCO clock delays based on this number PRG_DELAY[7:6] is the fine delay on the clock output. It determines relative delay on this clock programmable in 3, 2, 1, 0 times 30 ps based on this 2 bit code
		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
		2	R/W	0h	{0, 0, 1, 0} Internal Pull Up, Differential Output
		1	R/W	0h	{0, 0, 0, 1} Internal Pull Dn, Differential Output
		0	R/W	0h	{0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN
DIVO0T_DIV2_OUTPAGE	50h				
		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]

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
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
		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
		3	R/W	0h	{0, 0, 0, 0} External Termination, Differential Output
		2	R/W	0h	{0, 0, 1, 0} Internal Pull Up, Differential Output
		1	R/W	0h	{0, 0, 0, 1} Internal Pull Dn, Differential Output
		0	R/W	0h	{0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN
DIVO1T_MISC0_OUTPAGE	5Fh	7:6	R/W	0h	DIVO Divider for 0B output: Bits [27:26]
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

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		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	DIVO Divider for 0B output: Bits [27:26]
		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 {0, 1, 0, 0} CMOS On OutP, Nothing on OutN {1, 0, 0, 0} Nothing on OutP, CMOS on OutN {1, 1, 0, 0} CMOS on OutP, CMOS on OutN
		2	R/W	0h	
		1	R/W	0h	
		0	R/W	0h	
		2	R/W	0h	
		1	R/W	0h	
0	R/W	0h			
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 A: PLL A Related Registers (Similar for Pages B, C, D).
The Registers from 10h to 2Fh are equivalent NVMCopy Registers for this Page

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

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description	
NOTIFY_1_PLLA	07h	7	R/W	1h	SPARE	
		6	R/W	1h	SPARE	
		5	R/W	1h	SPARE	
		4	R/W	1h	SPARE	
		3	R/W	1h	Sticky/Notify status for cycle slip detection	
		2	R/W	1h	Sticky bit indicating holdover window is valid	
		1	R/W	1h	Sticky/Notify status for fast lock mode	
		0	R/W	1h	Sticky/Notify status for holdover	
MASKb_1_PLLA	08h	7	R/W	1h	SPARE	
		6	R/W	1h	SPARE	
		5	R/W	1h	SPARE	
		4	R/W	1h	SPARE	
		3	R/W	1h	Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[3]	
		2	R/W	1h	Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[2]	
		1	R/W	1h	Mask bit for NOTIFY_1_PLLA FASTLOCK (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[1]	
		0	R/W	1h	Mask bit for NOTIFY_1_PLLA (07h)If programmed as '0': Mask sticky/Notify bit generation for 07h[0]	
PRG_Directives_PLLA	0Fh	7:3	R/W	0h	PRG_CMD Directives: 5b1_1000: PROGRAM_EFUSE 5b0_1100: READ_EFUSE 5b0_0110: Copy NVM Copy to Settings 5b1_1011: Proceed to Active	
		2	R/W	0h	Spare	
		1	R/W	0h	Escape to the PRG_CMD state from ACTIVE state	
PPATH_PLLA	10h	7:5	R/W	0h	DLPF Settings from the GUI	
		4:0	R/W	0h		
IPATH1_PLLA	11h	7:3	R/W	0h		
		2:0	R/W	0h		
IPATH2_PLLA	12h	7:5	R/W	0h		
		4:0	R/W	0h		
FASTLOCK_PPATH_PLLA	13h	7:5	R/W	0h		
		4:0	R/W	0h		
FASTLOCK_IPATH1_PLLA	14h	7:3	R/W	0h		
		2:0	R/W	0h		
FASTLOCK_IPATH2_PLLA	15h	7:5	R/W	0h		
		4:0	R/W	0h		
CYCLESIP_MISC_CTRL_PLLA	16h	7	R/W	0h		CP gain configuration settings provided by GUI
		6	R/W	0h		
		5	R/W	0h	PLL loop filter configuration provided by GUI	
		4	R/W	0h	PLL loop filter configuration provided by GUI	
		3	R/W	0h	ZDB related setting provided by GUI	

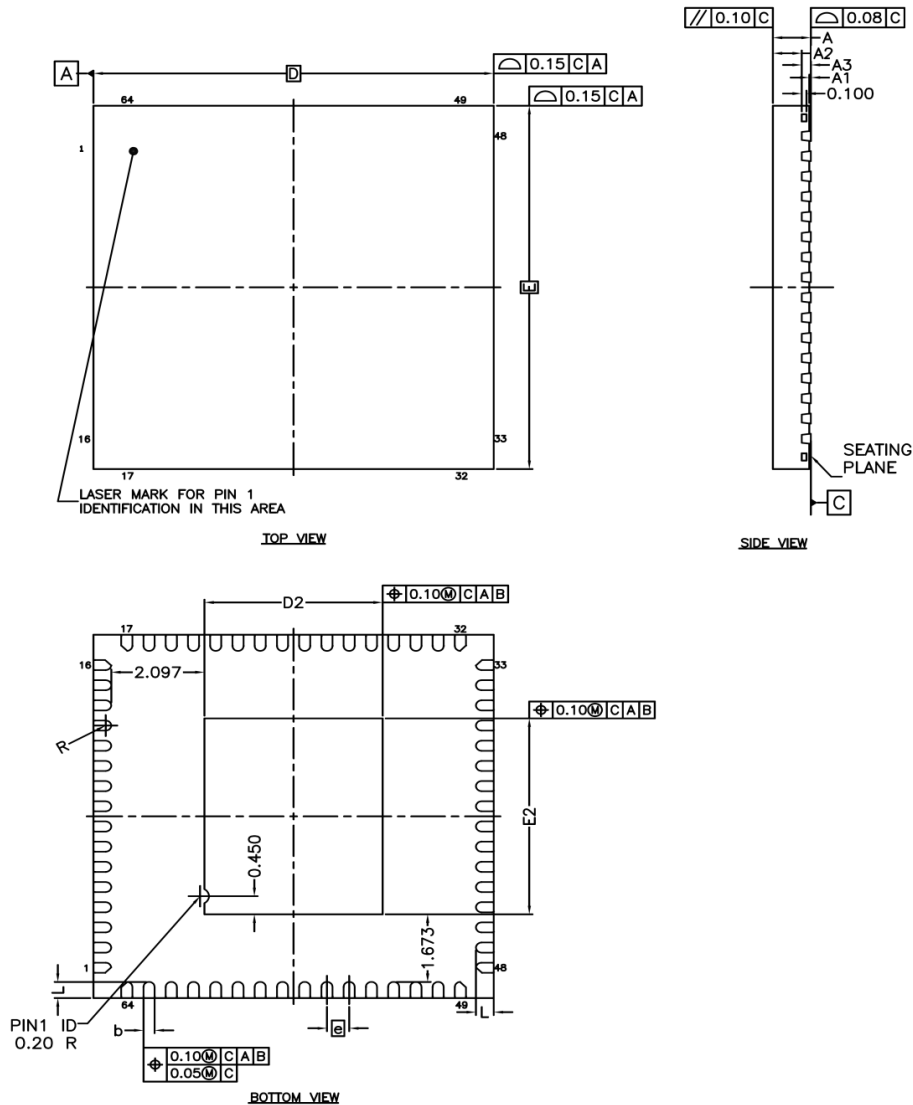
Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		2:1		0h	Cycle Slip detector threshold settings computed by GUI
		0	R/W	0h	Enable for cycle slip detector for DLPF
DIVNINT_PLLA	17h	7	R/W	0h	Master Disable for PLLA in sync mode and all associated functions
		6:0	R/W	0h	DIVN Integer part computed by GUI
MISCXO_PLLA	18h	7	R/W	0h	Enable ADC Dither
		6	R/W	0h	Offset Enable in the PLL Charge Pump
		5	R/W	0h	Select ACTIVE clock manually from manual input select pins when in manual active select mode
		4	R/W	0h	DIVN Integer part computed by GUI
		3:1	R/W	0h	Charge Pump Binning for VCO KV
		0	R/W	0h	0: Fractional Mode; 1: Integer Mode
DLPF_PHI_CORRECTION_PLLA	19h	7:6	R/W	0h	Phase averager configuration settings
		5:2	R/W	0h	Phase measurement filter BW setting (3 is spare)
		1	R/W	0h	Phase Correction Setting: Use default from GUI
DIVNFRAC2_PLLA	1Ah	7:0	R/W	0h	DIVN Fractional part computed by GUI
DIVNFRAC3_PLLA	1Bh	7:0	R/W	0h	DIVN Fractional part computed by GUI
DIVNFRAC4_PLLA	1Ch	7:0	R/W	0h	DIVN Fractional part computed by GUI
DIVN2_INT1_PLLA	1Dh	7:0	R/W	0h	DIVN2 Integer part computed by GUI
DIVN2_INT2_PLLA	1Eh	7:0	R/W	0h	DIVN2 Integer part computed by GUI
DIVN2_INT3_PLLA	1Fh	7	R/W	0h	Enable Revertive switching for input clock switching
		6	R/W	0h	Force manual selection of ACTIVE clock
		5	R/W	0h	Force Integer mode for the DIVN2 DSM
		4:0	R/W	0h	DIVN2 Integer part computed by GUI
DIVN2_FRN1_PLLA	20h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRN2_PLLA	21h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRN3_PLLA	22h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRN4_PLLA	23h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRD1_PLLA	24h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRD2_PLLA	25h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRD3_PLLA	26h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
DIVN2_FRD4_PLLA	27h	7:0	R/W	0h	DIVN2 Fractional part computed by GUI
OUTPUT_EN_PLLA	28h	7:6	R/W	0h	PSEQ_SYNC_DELAY 0: 500us 1: 8.17ms 2: 131.07ms 3: 2.09sec
		5:0	R/W	0h	One Hot Output Enable for Outputs 5:0
LL_REG1_PLLA	29h	7:6	R/W	0h	Programmable Phase Propagation Mode slope for the PLL 00: Use the Bandwidth 01: 10 usec/sec 10: 40 usec/sec 11: 160 usec/sec

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
		5	R/W	0h	Wait for Input Clock in power up in the PLL wake-up sequence
		4	R/W	0h	Use Fast Lock BW for exit from holdover to latch on fast to the new clock
		3:1	R/W	0h	Loss of Lock Delay = $(2^{(26-2*LLDELAYTIMER)})/4M$. Wait for this delay time before announcing LL de-assertion
		0	R/W	0h	LL Clear Threshold {LL_CLR_VALUE_PLLA[1], 0x2A[0]} 2'b00 : 0.2 PPM 2'b01 : 0.4 PPM 2'b10 : 2 PPM 2'b11 : 200 PPM
LL_REG2_PLLA	2Ah	7:5	R/W	0h	LL_SET_VALUE_PLLA[7:4]
		4	R/W	0h	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
		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	Holdover Tdelay settings for the PLL computed by the GUI
		4	R/W	0h	Enable Zero Delay Buffer mode with feedback clock routed from the PCB on in3
		3:1	R/W	0h	Holdover Average settings for the PLL computed by the GUI
		0	R/W	0h	Enable output clock sync on an independent input coming from IN3
		7	R/W	0h	Cycle Slip Detector related default
		6	R/W	0h	1: Enable Phase Propagation during input clock switch 0 : Enable Phase Build Out Mode where the phase difference between input clocks is absorbed by the PLL

Reg Name	Register Number	Bit Range	Access Type	Default Value	Description
HOLDOVER2_PLLA	2C	5	R/W	1h	Dither configuration for DIVN2 DSM
		4	R/W	0h	Enable revert to spare input clock during clock switching
		3:1	R/W	0h	DLPF related constant from the GUI
		0	R/W	0h	Internal voltage programming: Use default from GUI Profile
DECIMATION_RATIO_PLLA	2Dh	7	R/W	0h	Enable fast lock mode based on loss of lock status
		6:3	R/W	0h	Internal Rate Change factors in DLPF computed by GUI
		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	Program the frequency ramp slope from the following- 00: 0.2 ppm/s, 01: 2 ppm/s, 10: 20 ppm/s, 11: 200 ppm/s
		3	R/W	0h	Enable the frequency ramp feature
		2:0	R/W	0h	Internal Frequency Divider : Computed by the GUI
DCO_FRAC1_PLLA	31h	7:0	R/W	0h	DCO fractional control code
DCO_FRAC2_PLLA	32h	7:0	R/W	0h	DCO fractional control code
DCO_FRAC3_PLLA	33h	7:0	R/W	0h	DCO fractional control code
DCO_FRAC4_PLLA	34h	7:0	R/W	0h	DCO fractional control code
DCO_FUNCTION_PLLA	35h	7:4	R/W	0h	Reserved
		3	R/W	0h	DCO Increment in Frequency from registers
		2	R/W	0h	DCO Decrement in Frequency from registers
		1	R/W	0h	Enable DCO free run mode
		0	R/W	0h	DCO Mask for this PLL
DCO_BUMP2_PLLA	36h	7:0	R/W	0h	DCO integer control code
DCO_BUMP3_PLLA	37h	7:6	R/W	0h	DCO integer control code
		5	R/W	0h	Enable DCO sync mode
		4	R/W	0h	SPARE
		3:0	R/W	0h	SPARE
PAGE_NUMBER	FFh	7:0	R/W	0h	On all pages register FF is a READ / WRITE register used to change the page number

Package Information

Figure 41 shows the package dimensions.



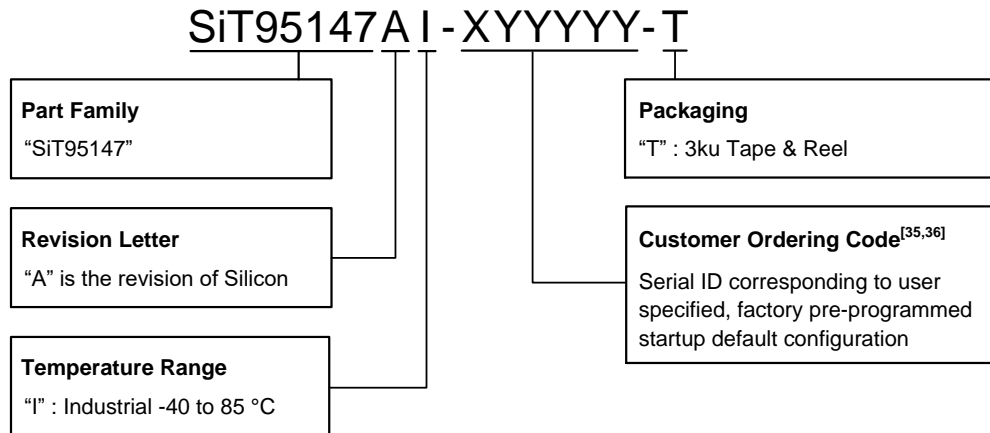
SYMBOL	DIMENSIONS		
	MIN.	NOM.	MAX.
A	—	—	0.90
A1	0.00	—	0.05
A2	—	0.65	0.70
A3	—	0.203 REF	—
b	0.18	0.25	0.30
D	—	9.00 BSC	—
D2	3.90	4.00	4.10
E	—	9.00 BSC	—
E2	4.75	4.85	4.95
L	0.30	0.40	0.50
e	—	0.50 BSC	—
R	0.09	—	—

NOTES

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIE THICKNESS ALLOWABLE IS 0.305 mm MAXIMUM (0.12 INCHES MAXIMUM)
3. DIMENSIONING & TOLERANCES CONFORM TO ASME Y14.5M, -1994.
4. THE PIN #1 IDENTIFIER MUST BE PLACED ON THE TOP SURFACE OF THE PACKAGE BY USING INDENTATION MARK OR OTHER FEATURE OF PACKAGE BODY.
5. EXACT SHAPE AND SIZE OF THIS FEATURE IS OPTIONAL.
6. PACKAGE WARPAGE MAX 0.08 mm.
7. APPLIED FOR EXPOSED PAD AND TERMINALS. EXCLUDE EMBEDDING PART OF EXPOSED PAD FROM MEASURING.
8. APPLIED ONLY TO TERMINALS.
9. JEDEC SPECIFICATION NO. REF: N.A.

Figure 41. Package Dimensions

Ordering Information



Notes:

- 35. X = "A" customer device, "B to "Z" reserved
- 36. Y = 0..9, A...Z for custom serial ID

Table 32. Revision History

Revisions	Release Date	Change Summary
0.1	11/26/2018	Preliminary release
0.2	02/15/2019	Updated Figure 6 Overall Architecture diagram Updated SPI Operations Figures
0.3	05/16/2019	Added Register Map sections
0.31	08/31/2019	Fixed typos

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

© SiTime Corporation 2018-2019. 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.