

Cascade Platform of SiT9514x ClockSoC Products

Clock Generators, Jitter Cleaners, and Network Synchronizers

GUI User Manual

Contents

1	Intro	Introduction			
2	Document applicability				
3	Casc	ade GU	II installation	5	
4	Star	ting the	GUI	10	
5	Fund	ctional o	descriptions of SiT9514x device variants	11	
6	GUI	usage	· · · · · · · · · · · · · · · · · · ·	17	
	6.1	Sectior	ns in the GUI	17	
		6.1.1	GUI Sections overview		
	6.2	Chip Co	ommunication and Interrupt		
		6.2.1	Input Clock Reference section	19	
		6.2.2	Input section	20	
		6.2.3	PLL section	21	
		6.2.4	Output section	22	
		6.2.5	Bird's Eye section	23	
7	SiT9	514x de	evice configuration tasks	24	
	7.1	TASK 1	: Select inputs	24	
	7.2	SiT951	41, SiT95145 input configuration		
		7.2.1	SiT95147, SiT95148 input configuration	27	
		7.2.2	SiT95148 clock loss configuration		
	7.3	TASK 2	: Set up PLL parameters		
	7.4	TASK 3	: Save or load UI configuration and program the SiT9514x device		
		7.4.1	Saving and loading the UI configuration file		
		7.4.2	Using the dump function		
		7.4.3	Saving efuse.NVM.py (I2C/SPI) files		
		7.4.4	Saving the Cascade SiTime GUI state		
		7.4.5	Using the load NVM function	41	



		7.4.6 Using the load NVM function	46
		TASK 4: Use Realtime to set output frequency with DCO or view clock monitor status	48
		7.5.1 Interrupts	49
		7.5.2 On-the-fly change	51
		7.5.3 Managing on-the-fly frequencies	52
		7.5.4 Using the dump fly function	53
		7.5.5 Dynamic profile on-the-fly with single output per PLL	55
		7.5.6 Description of dynamic header files	59
	7.6	Using the load fly function	63
	7.7	FlexIO	64
	7.8	Phase sync feature	67
	7.9	Input to output delay control feature	69
	7.10	SiT9514x jitter attenuator as timing source for JESD204B RF converters in 5G RRU	71
		7.10.1 JESD204B overview	71
	7.11	Cascade as clock source for JESD204B timing signals	72
	7.12	Configuring the SiT9514x for JESD204B timing signals	74
	7.13	Generating SYSREF via SYSREF_REQ	74
8	Snap	shots of specific use case scenarios	. 81
	8.1	Free running DCO	85
	8.2	Zero-delay buffer mode	88
9	Low	wander mode	. 89
10	Usag	e guidelines for jitter performance optimization	. 90
	10.1	Output placement and frequency planning	90
	10.2	CMOS output type selection	93
11	Docu	ment Information	. 95



1 Introduction

The SiT9514x ClockSoC[™] products are based on the SiTime's Cascade Platform[™] that integrates multiple clock ICs and oscillators into a single device. The SiTime ClockSoC products include clock generators, jitter cleaners, and network synchronizers that support up to four clock inputs and up to the 11 differential or the 22 single-ended clock outputs. The clock outputs can be derived from the 4 PLLs in a manner that provides high flexibility in terms of frequency planning options. These clocks are fully programmable with the I2C/SPI interface for selecting the input frequency to output frequency translations and associated jitter attenuation loop bandwidths. Using advanced design technology, SiT9514x devices provide excellent jitter performance while working reliably under ambient temperatures from -40°C to 85°C. These features make it ideally suited for communications applications (e.g., OTN, SONET/SDH, xDSL, GbE, networking, wireless infrastructure, IEEE 1588 clock steering), broadcast video with genlock, test and instrumentation applications, and high-speed data converters. Additionally, on-chip programmable non-volatile memory enables factory preprogrammed devices that power up with a known configuration.



ard

2 Document applicability

This document applies to the Cascade Platform products shown in Table 1.

This document only describes operation of the Cascade SiTime GUI version 1.30.4 with the SiTime products shown in Table 1. Operation with other versions may differ from that described in in this document.

Refer to the user documentation for your SiTime evaluation board for information about its deployment.

	Part number	Number of outputs	Device type	Supported by GUI as described in this document	Supported evaluation boa
	SiT95141	10	Clock	Yes	SiT6503EB
			Generator		
ľ	SiT95143	10	Clock	No — contact SiTime Technical	SiT6503EB
			Generator	Support to configure	
	SiT95145	10	Jitter Cleaner	Yes	SiT6503EB
	SiT95147	8	Network	Yes	SiT650 <u>2</u> EB
			Synchronizer		
	SiT95148	11	Network	Yes	SiT6503EB
			Svnchronizer		

Table 1: Applicable part numbers

NOTE: About operating parameters

For this document, the application was tested using Microsoft Windows[™] 10 Pro, version 1909. Other versions should also work in most cases.

Recommended display resolution is 1600 x 1080 or better. Other resolutions will work; however, if you operate this application using a display with lower resolution (e.g. 1366 x 768), portions of the screen may appear different, such as overlapping sections or wrapping of labels, buttons, fields, etc. Some screenshots in this document were created using lower resolution and may display somewhat different in your deployment.



3 Cascade GUI installation

Open the **setup_Cascade-vn.n.n-SiTime.exe** file and select the folder in which to install the Cascade GUI application, see Figure 1 and Figure 2.

Select Destination Location Where should Cascade be installed?	
Setup will install Cascade into the following folder	r.
To continue, dick Next. If you would like to select a differ	ent folder, click Browse.
C:\Program Files (x86)\Cascade\v1.30.4-SiTime	Browse
At least 55,9 MB of free disk space is required.	
	Next > Cancel
	Conce

Figure 1: Selection of the destination location



Figure 2: Selection of the Start Menu folder



Optionally, click the checkbox to create a desktop shortcut. Click **Next** to proceed with the installation, see Figure 3.

Additional T n additional ta	asks Isks should be	performed?			
t the addition dick Next. Create a <u>d</u> esk	al tasks you w top shortcut	ould like Set	up <mark>to</mark> perform	ı while installing (Cascade,
	Additional T n additional ta t the addition dick Next. Create a <u>desk</u>	Additional Tasks n additional tasks should be t the additional tasks you w dick Next. Create a desktop shortcut	Additional Tasks n additional tasks should be performed? t the additional tasks you would like Set dick Next. Create a desktop shortcut	Additional Tasks n additional tasks should be performed? t the additional tasks you would like Setup to perform dick Next. Create a desktop shortcut	Additional Tasks n additional tasks should be performed? t the additional tasks you would like Setup to perform while installing of dick Next. Create a desktop shortcut

Figure 3: Option to create a desktop shortcut and proceed

All SiT9514x related software is installed first, see Figure 4.



Figure 4: Ready to install



The SiTime evaluation boards use an FTDI chip solution for the USB-to-serial interface conversion. The FTDI driver is installed next, see Figure 5 and Figure 6.

Installing Please wait while Setup installs Cascade on your computer.	ð
Extracting files C:\Program Files (x86)\Cascade\v1.30.4-SiTime\tk\ttk\combobox.td	
	Cancel

Figure 5: Installation progress



Figure 6: Extracting FTDI CDM drivers

Click Extract to proceed with the FTDI driver installation, see Figure 7, Figure 8, and Figure 9.



Device Driver Installation Wizard			
	Welcome to the Device Driver Installation Wizard! This wizard helps you install the software drivers that some computers devices need in order to work.		
	< Back Next > Cancel		

Figure 7: Click Next to start the device driver installation wizard



Figure 8: Read and accept the license agreement



Device Driver Installation Wizard					
Completing the Device Driver Installation Wizard					
	The drivers were successfully installed on this computer.				
You can now connect your device to this computer. If your device to this computer. If your device to them first.					
	Driver Name	Status			
	✓ FTDI CDM Driver Packa ✓ FTDI CDM Driver Packa	Ready to use Ready to use			
< <u>B</u> ack Finish Cancel					

Figure 9: Finish the driver installation

Click **Finish** to complete setup installation Figure 10.



Figure 10: Finish the Cascade setup wizard



4 Starting the GUI

The Cascade GUI provides an easy interface to configure the selected device. Since the GUI uses the FTDI chip for the USB-to-serial I2C communication on the SiTime evaluation board, the FTDI chip related software drivers are also installed by the installation file.

When the Cascade GUI application is launched, the **Choose Variant** selection menu appears to prompt for selection of one of the product variants. Select the **SiT9514**x variant you are using and click **Select**.

It is possible to start and use multiple instances of the application simultaneously.

NOTE: Contact SiTime Technical Support to configure the SiT95143 device.

An example selection of the SiT95141 product variant is shown in Figure 11:



Figure 11: Option for SiT95141

The GUI software will launch for the selected SiT95141 device variant, see Figure 12.



Figure 12: GUI view for SiT9514

Users can move between the individual sections, configuring parameters, while related parameters in the other sections remain visible. Programmed configurations can be saved in configuration profile files containing the configuration parameters, or in sets of I2C/SPI read and write scripts for reuse, etc.



5 Functional descriptions of SiT9514x device variants

The SiT95141 is a clock generator device that offers four fractional-frequency translations from the same input. Any one of the four clock inputs map to all four PLLs. The PLL outputs are mapped to the 10 outputs, offering flexible frequency translation configurations, see Figure 13.

The SiT95145 is a jitter attenuating frequency translation device that offers four fractional translations from the same input. The four clock inputs map to all four PLLs. The PLL outputs can be mapped to a subset of the 10 outputs, offering flexible frequency translation configuration with independent control of each PLL in terms of jitter attenuation, bandwidth control, and input clock selection with redundancy, see Figure 13.





SiT95147 is a jitter attenuating and network synchronizing frequency translation device that offers four independent PLLs. The four clock inputs can map to any of the four PLLs. The PLL outputs are mapped to the eight outputs offering flexible frequency translation configurations with independent control of each PLL in terms of jitter attenuation, bandwidth control, and input clock selection with redundancy, see Figure 14.





SiT95148 is a jitter attenuating frequency translation device that offers four independent fractional PLLs. PLL outputs are mapped to the 11 outputs. This allows flexible frequency translation configuration with independent control of each PLL in terms of jitter attenuation, bandwidth control, and input clock selection with redundancy, see Figure 15.





Clock hierarchy, various frequency dividers nomenclature, and clock translation pathways available are shown in Figure 16 for the SiT95141 and SiT95145 devices.



Figure 16: SiT95141 and SiT95145 clock hierarchy



The clock hierarchy, nomenclature of the various frequency dividers, and clock translation pathways available for the SiT95147 are shown in Figure 17.



Figure 17: SiT95147 clock hierarchy

The clock hierarchy, various frequency dividers nomenclature, and clock translation pathways available on the SiT95148 device are shown in Figure 18.



Figure 18: SiT95148 clock hierarchy



6 GUI usage

6.1 Sections in the GUI

6.1.1 GUI Sections overview

The GUI layout is divided into sections, used to configure distinct sets of parameters, and unique to the specific sections, described in detail in subsequent section of this document.

- Chip Communication See the top-left section.
- Internal Clock Configuration See the top-center section.
- Input (0,1,2,3) See the left-middle section.
- PLL (A,B,C,D), Clock Switch and Lock Loss See the center section.
- Output (0T,7,6,5,4,3,2,1,OB) See the right-middle section.
- Bird's Eye view See the bottom section.

Chip Communication Internal Clock Refere	nce	📋 Phase Sync	Interrupt	
SFT ® DC Golden I	Imbedded MEMS	□ A □ C	INTRE Edit	Reset Chip Send 2 Chip
Connection O Address Mdd .	76.86344 MHz •	e 🗆 🗠 🗠	DCD from Pins	ZDB None Verity EFUSE
Input #2 Input #2 Input #2	PILA PILB PILC PILD		0T 7 6 5 3 2	1 (6)
E leput #0	Bandwitth		El Output 0T	
C <u>Getten Ovol</u> Gepped Ovol	Fast (Hz)	Normal (Hz)	Frequency (Ht)	
Differential =				0.0T 5 8.4
	() Conduction	Clock Switch	208 10	vui sysine
- Coese Drift	TOPOTONI	Hitless Switch		
Togger Edge Clear Edge Val Time Set 2000 + 5 + 4 + 2ms +	Selection Auto + F Reserve	Phase Build Out.	Deday (s)	
Cir 800 -	Thursday Birther	Phase Propagation	C. Tanka Index	
Ock Switch	Deav	Stepe	C ange cover	(1 m)
	Lessy Firmer			
		The frequency Rent	0 CUP	0.009
Ω <u>3</u>	5et Car 1094 *	Slope	୍ର ପାହା	© CLINN
PLL Configuration	4 + 2 + 7 m'oaluie		and the second	
A B C D	(1) www.minut	1.00	C Difeedal	
	Est Lock HoldDver	Input #3 Sync	Mode	-
VCBIO FledO Ealt	Input Order	Output:List	D PLL Configuration	
VOD	n		2 2	
No Field'selected	Ĩ.	. 0:		
			Do-the-Fly Change	
U Sestine Bod's Eye	Load NVM	Dump	Frequencies Dump Fly	Change fout Dyn Load Ply
	P	•		
	PILA	8		T00 III
				07
1	PLLS	1		■ 06
	*	<u> </u>		05
12 🔳	-	5		03
	2	5		02
13 🔳	×			01
	PLL D			00B
	•			





6.2 Chip Communication and Interrupt

The **Chip Communication** option allows the user to specify how to connect to the chip.

The Interrupt section is used to configure interrupt related settings.



Figure 20: Communication and interrupt options



6.2.1 Input Clock Reference section

The Input Clock Reference section is used to configure:

- Golden clock (for frequency drift monitoring)
- Embedded MEMS frequency

Internal Clock Reference				
Golden	Embedded MEMS			
Clock	76.86144 MHz 🗸			

Figure 21: Input Clock Reference section

IMPORTANT:

Embedded MEMS *must* be set to 76.86144 MHz only. Operation in the 76.8 MHz setting is *not* generally supported and should *only* be used if advised to do so by SiTime Technical Support.



6.2.2 Input section

The **Input** section is divided into similar individual tabs for **Input #(0-3)** where the following parameters can be set:

- Frequency, clock type, and clock loss/frequency drift
- FlexIO, for setting **Clock Switch** fine and coarse frequency drift monitors for the input clocks
- Input assignment (or not) to all PLLs

Input #0 Input #1 Input #2	Input # <u>3</u>	
	Gapped Clock	
Frequency (Hz)	Clock Typ Differential	e
Trigger Edge 5 ~	Clear Edge	Val Time 2ms 🗸
PLL Configuration	All	
VDDIO FlexIO		Edit
VDD		
	No FlexIO selected	

Figure 22: GUI Input section

NOTE: If no input is assigned to the PLL section, then the device's internal PLL oscillator is the primary clock source.



6.2.3 PLL section

The **PLL** section is used for setting all PLL related parameters including:

- PLL (A,B,C,D) Up to four independently configurable PLL sections. Each PLL supports up to four clock inputs with Frac-N dividers, enabling flexible input to output frequency translation configurations. PLL input clock priority settings can be changed in Page 1h, registers 49h 4Bh.
- Bandwidth
- Lock Loss
- Clock switching options (SiT95145 only)
- Input clock priorities

PLL A PLL B PLL C PLL D
Bandwidth
PLL Bandwidth (Hz)
4 kHz
✓ Free Running
IP Clock Selection
Selection Manual V Revertive
Lock Loss
Delav Timer
1.02ms V
PPM
Set Clear
4 🗸 2 🗸
🗹 DCO Mode
Output List
OT 7 6

Figure 23: GUI PLL section



6.2.4 Output section

The **Output** section is used to set the output standard type, voltage supply, and to assign outputs to the PLLs.

0T 7 6 5 4 3 2 1 0 0B
Frequency (Hz)
ZDB VDD 0T
Delay (s)
O Single Ended
ON OP
○ CLKP ○ CLKP
O CLKN O CLKN
O Differential
Mode
PLL Configuration
A B C D O O O O

Figure 24: GUI Output section

Additionally, if the DCO mode is enabled in the **Output** section, the **Realtime** section can be used to move the output frequency in the DCO mode. Once the chip is programmed, the **Realtime** section can be used for observing the status of the clock loss monitors.

In the Free running mode, the DCO is enabled by default and is available in the Realtime section.

For more information, see TASK 4: Use Realtime to set output frequency with DCO or view clock monitor status.



6.2.5 Bird's Eye section

The **Bird's Eye** section can be used to visualize the resulting configuration of Inputs to PLLs to Outputs.

Bulling	Bird's Eye (Internal) Load Load WM	Oump	On-the Ply Change Frequencies	Dump Hy	Change had then	Destry
io m	PLA PLA	E				00T
n 🔳	PLE					05
a 🖩	RLC RLC					03
13 🔳	A La	1				00 00 00B

Figure 25: GUI Bird's Eye section

Click the **Bird's Eye** button to activate the Bird's Eye section, showing the configured signal paths. Click the **Bird's Eye** button again to open the Bird's Eye section in a separate window. Close that view to completely close the Bird's Eye view.

This lower section of the Cascade SiTime GUI is also used to:

- Dump (save) and Load (open) device configuration files
- Specify On-the-Fly frequencies
- Change fout parameters
- Dump and Load On-the-Fly configuration parameters
- Select Realtime status of the device

The **Realtime** button opens the Realtime window that displays the detailed operational status of the programmed device based on the current configuration. It can be is used to monitor device operation and view the impact of the set configuration parameters. For an example, see *TASK 4: Use Realtime to set output frequency with DCO or view clock monitor status*.



7 SiT9514x device configuration tasks

An overall approach for using the Cascade SiTime GUI is described in the following tasks:

- TASK 1: Select inputs
- TASK 2: Set up PLL parameters
- TASK 3: Save or load UI configuration and program the SiT9514x
- TASK 4: Use Realtime to set output frequency with DCO or view clock monitor status

7.1 TASK 1: Select inputs

The four input clocks with frequencies **fin_ext***k* translate to the PLL input clocks **fin***k*, following division by the respective input dividers with fractional or integer frequency division ratios **DIVN1***k*, where the index $k \in \{0, 1, 2, 3\}$. Each of the PLLs are driven by one of the four divided input clocks **fin***k* as its active input clock. Each PLL sets the priority for up to three spare clocks from the remaining three input clocks, if required for switching to a redundant input, see Figure 26 and Figure 27.







Figure 27: SiT95147 and SiT95148 input clock distribution

The **DIVN1** input dividers are internally computed by the Cascade SiTime GUI software. The **Input** section is used to set up the input frequencies and clock loss status as well as directing each input to a particular **PLL**. Further, the frequency drift monitors (with respect to the inputs) are set in this section. An example of setting the pathways for the **Input #0** clock is shown in Figure 28.

Time



7.2 SiT95141, SiT95145 input configuration

When an input is selected, it is assigned by default to all the PLLs as input. It is added to the input order of the PLLs.

For example, if **Input #0** is selected, then **Input #0** goes to *all* the PLLs as input, which is also shown in the **Bird's Eye** section, see Figure 28.

Connection D	IN: Address 0.69	Embedded MEMS 36.80141 MPts •	S Time		Dellifia Lan	Reset Chip Send 2 Chip 2 ZDE Name VietNy EF1/5E
legent 40 langua 40 langua 40 la	ingust #2		PLL& PLL& PLLS PLLS		07 2 6 5 4 3 2 1	1 0 00
Frequency (HQ 504)	Oock Type Differential	•	PLI Execution (etc) 4 kine		208 C	VIIID OT
Trigger Silge 3 •	Clear Edge	Val Time 2m =	(2) Free Renning IP Could Selection Soluction (Microall -r.)	E Routine	Delay (i) [
PLL Carligution	aan 191		Delay	Tiesar (2)	0 con	0 CUP 0 CUP 0 CUP
VDDD FiedD		[_£dt_]	Set	Our (I +	 Differential Mode [
© 10094	Ins TestO selected		UI DCO Mode Ovejuš List [] 47 [] 7	01	A B D D	8 8
() toma		fird1 Ere 🕑	Load N/M	Jure 🔒	Do-the-Fly Change Frequencies Dump Fly	[Change from Dyn] [Lood Hg.]
10 1 50 MAL 11 1			PLA			00T 07 06 05 04 03
			MLD			02 01 00 00 00 00

Figure 28: SiT95141 Input #0 assigned to all PLLs



7.2.1 SiT95147, SiT95148 input configuration

When an input is selected, that input is assigned to a particular PLL (A,B,C,D) as input and is added to the **Input Order** of that PLL. To select an input, click the tab for the selected **Input #(0,1,2,3)**, and then check the box.

For example, if the box for **Input #0** is checked, it becomes the selected input for **PLL A** as shown in the **Bird's Eye** view, see Figure 29.

Cascade v1.30.4	(\$1795145									1	- 0
X Connection	Chip Com SPI O	Address (Out	Golden Clock	Reference Erobedded MENS 7626144 MHz ==	S Time		Phase Sync A C B D	Interrupt Interrupt DCOme	alt.	Feat Orp	Send 2 Ohp
hput ¶ hqui f ⊡input #S	input #2	input #2		PLE PLE PLE PLE Bandwith				07 7 8 5	4 9 1	1 0 08	
	Belden Der	5	Gapper Co.	Read (Hac)	-	Hores	4 (Hz)	Freque	ing the literature		
50M		Inpu	ut #0 is selec	ted		-	-	2218	1	11.01	System
Trippe Edge	Clear Edge	Val Time Imi 🐨	ConnerDoft	Pas Re IP Oper Selection Selection Rules	ning	Cherk Savier Hillen: Sw O Pha	h Auh se Build Out se Prepagation		beley tes [
Clock Switch	Coorte FD	Fine FD	C Fire Date See 110 1	Defey Terrer	HeldDoor Delay Sithey	L	Steps (1)	0%	up.	() (P	ø
			10(10)10	im.	dramp	0 fire	ercy fange	0.0	uite -	0.6	(K)
PLL Configurat	tion	AL FI	Selected	Input added to Inpu	t Order		Sear .	- Differential			
000	FactO	20.77	1	Tarting Tarting	wither		npud #3 Sync		More		
¥00	1111				•	Ovlpist Eist		C PL Config	e player		
O VODN		No Fie	siO selected	Input Order	4	(iii) (iii)	±7:0.4	Č.	d.	3	8
Inter.			lint's typ	(Internet) Load Load WVM			Dump	Children Py C	During Fig.	Dange had Dyr	Class No.
0 🚮 —					BILA.	E					■ 007 ■ 07
SOMAH2					File	F					06
				*							E 04
2 🗐				*	MLC						02
3 🗐				1							01

Figure 29: SiT95147 Assigning Input #0 to PLL A as input



7.2.2 SiT95148 clock loss configuration

One of four different clock loss conditions can be selected to initiate clock switchover for the PLL, as shown in Figure 30. The PLL uses the selected information from the clock loss monitors to determine whether to enter holdover or switch to another input.

Org Communication Second Clock Referen	K#	🗄 PhotoSynt	interrupt (J)	t
S an ^{a DC} Gottan 6 Dock 6	mbridded MEMS	DA DE	ByTHE Las Reset Chep Ser	HZ CHU
Convertion O Address Odd # 1	5.8044 MHz • • • • • • •	E. + . D + .	🗇 DCD from First 🔹 ZDS Nove 🔤	Ny EFUSE
Input 40 (Appet 42) (Appet 42) (2) Appet 40 (2) Golden Clock (Clock (C	PLL & [PLL & [PLL & [PLL &] Bandwath Fast (Pl2)	Normal (His)	01 2 6 3 4 3 2 1 0 18 06 0454 07 Transmit 010	
Frequency (Ho) Deck Type 25M Differential • Cases Drift Trigger Edge Clear Edge Val Time Sei 1000 • 5 • 4 • 2ms •	If fee Barrieg IP Dock Interney Selector [Acts] +] Selector [Acts] +]	Clock Switch Hitter Switch O Phane Build Out	ane venet set	4
Clock Switch Clock Learning Counter FD Free FD Chick Learning Counter FD Free FD Chick Learning Chick The material Counter FD Free FD Chick The material Counter FD Free FD Fr	Insking Headlog Data Same Same Same Same Same FPM Average Same Same	Dige:	C Single Extent C ON C SH C CAP C CAP C CAP C CAP	
AL Configuration	the type of clock loss used by the chip to	i byot ti tyre	- Differential Notes []	
voto voto voto voto	to the spare clock	00.01.04	C Richtgonter	#.: 01
() faiters	Load WM	Dump 🕞	On-the Ry Change Pressence: During Ry. Change Tool Dyn. 10	nai Ph
0 15 Mine	RLA RLA RLE RLC			007 07 06 05 04 03 02 01
3 🏢	PLLD			00

Figure 30: SiT95148 clock loss configuration



7.3 TASK 2: Set up PLL parameters

Click on a PLL and select its outputs. Then choose the PLL bandwidth, output frequency, output format (differential, LVDS mode should be selected for the SiTime evaluation board used, see Table 1). Choose the fast lock bandwidth and the regular bandwidth for the PLL. If you plan to use the DCO later, the DCO should be enabled when configuring the PLL.

Once the list of input clocks to a PLL is finalized, the input clock priority for the PLL can be changed in this section by dragging the clock selections up or down in the **Input Order** box (in the lower left area of the PLL section). The priority of the input clocks is listed in order with the highest priority clock at the top of the list and the lowest priority clock at the bottom.

For SiT95145 *only*, the input clock switching can be set to either **Manual** or **Auto** in the **Selection** dropdown menu in the **IP Clock Selection** area of the PLL section. For the **Auto** setting for the input clock switching, the revertive switching can be enabled or disabled (non-revertive) by checking the **Revertive** checkbox.

Manual clock switching is supported by changing the respective **IN_SEL1/0** pins on the SiTime SiT6503EB evaluation board, see Figure 31.



Figure 31: Manual clock switching for SiT95145



The lock loss and output frequency (Fout) settings can be set in the respective sections. Go to the **Lock Loss** sub-section (in the center of the PLL section) to select the holdover (HO) history and to enable the lock loss monitor for each PLL. Note that the holdover delay determines the time back in history that the PLL goes for averaging holdover. In other words, the PLL ignores HO delay seconds of the most recent history before the clock loss event. The HO average is the time over which the frequency is averaged to arrive at the holdover frequency, see Figure 32.





Important note about the PLL bandwidth

The PLL bandwidth for both fast lock and normal bandwidth is normally recommended to be less than $1/100^{\text{th}}$ and $1/500^{\text{th}}$ of the PLL input frequency, respectively. Since the PLL input frequency is determined by the Cascade SiTime GUI based on the setting for the input dividers (DIVN1), the PLL bandwidth should be configured based on the internal input frequency of the PLL. The input frequency for each PLL is displayed in the background **Realtime** terminal that runs along with the Cascade SiTime GUI. Additionally, a pop-up warning message appears if a PLL bandwidth is chosen such that fast lock and normal bandwidth are larger than the normally recommended bound of $1/100^{\text{th}}$ and $1/500^{\text{th}}$ of the PLL input frequency.



7.4 TASK 3: Save or load UI configuration and program the SiT9514x device

7.4.1 Saving and loading the UI configuration file

Press the **Send 2 Chip** button (top-right) to save the UI configuration file. The GUI will prompt you to save the UI configuration file in .py format to a user specified directory, see Figure 33. With the **Load** button, this file can be used to load the saved configuration later as necessary.

Internal Clock Refere Ċ Save UI Config Windows7_OS (C:) + Program Files (x86) + Cascade + Temp • +• Sem + Con Q ZDR Non Verify EFUSE Pine Organize * New folder 311. **•** 0 5 3 2 1 08 Name **Date modified** Type Size T Favorites SiT05147 F31 11.11.2020 14:00 E Desktop Pythan File 139 KE ency (Hz) 400M a Downloads Recent Places Documents VDD 08 Sys Ref OneDrive 33V . E Desktop Libraries Delay (1) Documents J Music Pictures Subversio CD OP Videos CLIO CLIOP IL Oleh Seheda File.name: 5i795147_F31 CLKN CLKN Saves the UI configuration file Save as type: Legacy State ÷ - Hide Folders Saye Cancel Made LVDS + Input Order Output List Field VDDIO Edit PLL Configuration 3 VDD 14 (2) (0) D 0 No FledO selected . O VDDIN 0n-the-Fly Change Load NVM Dump 🔒 (Bealtime Bird's Eye 🕢 Frequencies Dump Fly Change fout Dyn Load Fly 3.3V @ 200 MHz 00T PLL A 10 07 25 MH 3.3V @ 350 MH 06 PLL B 11 🗐 05 03 12 = 3.3V @ 250 MHz 02 PLL C 01 13 3 3V @ 400 MHz 00B 25 M PLL D

This is the preferred method to save the configuration profile.

Figure 33: Saving the UI configuration profile file



Figure 29 shows an example of a Cascade SiTime GUI configuration profile file.

import builtins as glob	
glob.dev_addr	= int(0x55)
## Frequency Profile	
glob.FREF_FREQ	= 114.285e6; # Frequency Algorithm
glob.FREF_FREQ_OCXO	= 150e6; # Frequency Algorithm
glob.FIN0_EXT_FREQ	= 42.898562e6; # Frequency Algorithm
glob.FIN1_EXT_FREQ	= 169.030400e6; # Frequency Algorithm
glob.FIN2_EXT_FREQ	= 14.722212e6; # Frequency Algorithm
glob.FIN3_EXT_FREQ	= 12.495564e6; # Frequency Algorithm
glob.FIN0_FREQ	= 6.128366e6; # Frequency Algorithm
glob.FIN1_FREQ	= 6.761216e6; # Frequency Algorithm
glob.FIN2_FREQ	= 7.361106e6; # Frequency Algorithm
glob.FIN3_FREQ	= 6.247782e6; # Frequency Algorithm
glob.OUT0T_FREQ	= 120e3; # Frequency Algorithm
glob.OUT1T_FREQ	= 136301.39643134; # Frequency Algorithm
glob.OUT0_FREQ	= 192364.96007515; # Frequency Algorithm
glob.OUT1_FREQ	= 100e3; # Frequency Algorithm
glob.OUT2 FREQ	= 500e3; # Frequency Algorithm
glob.OUT3_FREQ	= 339209.99219359; # Frequency Algorithm
glob.OUT4_FREQ	= 378507.39371534; # Frequency Algorithm
glob.OUT5_FREQ	= 427e3; # Frequency Algorithm
glob.OUT6 FREQ	= 333e3; # Frequency Algorithm
glob.OUT7_FREQ	= 123e3; # Frequency Algorithm
glob.OUT0B_FREQ	= 159048.719335332; # Frequency Algorithm
glob.OUT1B_FREQ	= 392e3; # Frequency Algorithm
glob.PLLA_FVCO	= 7.0277e9; # Frequency Algorithm
glob.PLLB FVCO	= 6.51792e9; # Frequency Algorithm
glob.PLLC_FVCO	= 6.55272e9; # Frequency Algorithm
glob.PLLD_FVCO	= 6.47042e9; # Frequency Algorithm
glob.XO_PPM	= 200; # Default
## Generic Page	

Figure 34: Example content of the configuration profile file



Click the **Send 2 Chip** button (top-right) to program the chip with the current configuration or a previously loaded configuration profile, see Figure 35.

	ion Internal Crock Ref	larence		Phase Sync	Internapt	(1)	+
Ø 581 ● BC	Galder	Enibedded MEMS	Time	0.4.0.5	INTRE EAR	Reset Only	Send 2 Ch
Connection O Address	dada 😐	76,86144 MHz +	9711me	0.000	E DCO from Pine	· ZDE None	Nerty Gru
of #0 loput #2 loput #2 loout #3		PLLA PLLE PLLC PLLD			07 7 0 5 3	2 1 08	
Input 40		Bandwidth			Cutgist 08		
C Goldes Clock	📋 Gapped Clock	Fast (Ht)		Normal (Hz)	Trequency (H	43 800M	
Frequency (HU	Clock Type	30.1		100			
25M	Collection •	Tree Bar	ning	Clock Switch	206	VDD 98	Sys Ref
	Coarse Drift	P Clock Selection	01080	Hitless Switch		3.34 💌	10
iggerEdge ClearEdge ValTime	Set 2000 - +	Selection Airts -	2 Reletive	Phase Build Out			
3 • 4 • 2m •		LACCORNE	14.116	Phase Propagation	Ddiy I	0	
	Ch 000 +	E Lock Loui	Dalay	Sept	C Single Ended		
lock Switch	🗇 Fire Drift	Lium - V	547ms +		(1 0N	10 OF	
Clock Loss Coaste FD Fine FD	Set 11 +	PPM	Average	171 Frequency Ramp	2.007		0.07
0 0	26	Set Clear	1.0944 *	them.			
	(m. 5)	25147	Droutet		10,000		STIM
LL Configuration			Dec mede	100 million and 100	B. Differential		
A 8	C	The second second second second	Sec.		- LALARDINA		
100 000	10	Programming successful					
121 121	8	Programming successful	Los Warster No.	(e	i	Wede LVDS +	
RI RI NO Field	R	Programming successful	Loss Warster Me	de Output List	III Pill Cardauatter	Mode LVDS +	
(R) (R) 010 Fleid0	R	Programming succesful	Low Wander My	de Output List	📰 PLL Configuration	Wede LVDS +	
IP IP IP	RealD selected	Programming succesful	Los Wander Me	e Gutput Lin 1 1/ 08	III PLL Configuration	Mede LVDS +	p
IP IP NO FlueBO * VDD FlueBO NODEN FlueBO	e RedD selected	Pregramming successful OK	Loss Wander Mo	e Gutput Lint	E PLL Certifiquation	Wede LVDS +	•
10 00 • VDD • VDD • VDD • VDD	e RedO selected	Pregramming successful	Less Warster Mo	ec Gutput Lint	PLL Certifiquenties	Made LVDS -	8 8
IP IP NO. FieldO • VDD • VDDIN D Bestime	e RedO selected		Less Warster Mo	er Output Lint	A Control of the Pick Control of the Pick Control of the Pick Change	Made LVDS - B C C C B C	0 •
IP IP NO PaulO * VDD VODIN D Beattime	RealD selected	Programming successful OK OK OK OK OK	Los Wande No	ec Output Lint	A Contraction Denthe Fly Change Tequencia Duri	Made LVDS - B C C C suffy [Change Nat 2	D B [1] [Lood Ph
IP IP NO FlastO VODEN No Spectrum	e ReiD selected	Programming successful OK OK Lovet C	Low Warster Mo	ec Output Lint I I I 08	E PLL Carifiguestics A C Denthe Fly Obarge Treasences During	Mede LVDS + I C U C U C U C U C U C U C U C U	0 911 [Load 75
IP IP NO FielD VOD VODN D Exiting	e Real D selected	Programming successful OK OK Lovet C	Los Wander Mo	er Output Lint I 1 10 08	E PLL Carifiguestion A On the Fly Change Trepuncts Duri	Mede LVDS + II C O O III C III C III C III C III C III C III C	0 0.041 0 00 0 01
IP IP NO FluidO • VDD • VDD • VDDN Destine	e Real O selected	Programming successful	Los Wander Mo	er Output Lint	PLL Cardiguestion	Mede LUDS • II C O O III C III C IIII C IIIII IIIII IIIIII IIIIII IIIIII IIIIII	0 0 0 0 0 0 0 0 0 0 0 0 0 0
R R NO FlastO VODEN N Statione	e Real O selected	Programming successful	Los Warder Mo and MML PLLA	e Output Lint	PLL Cardiguestion A On the Fly Change Tremancis Duri	Mede LUDS • I C O O UTFL Orange Nat 2 3.37 @ 25 3.37 @ 25	0 0 0 0 0 0 0 0 0 0 0 0 0 0
IP IP IP	e Real O selected		Los Wander Mo and ROM. PLLA PLLB	e Output Lint	PLL Cardguester A Oethe Fly Charge Trepanois Dur	Mede LUDS - I C C U C U C U C S S S S S S S S S S S S S	0 0 0 0 0 0 0 0 0 0 0 0 0 0
VODEN	e Real O selected	Pregramming successful	PLLS	e Output Lint	PLL Cardguester A Orethe Fly Change Trepuncis Duri	Mede UVDS • C C C C C C C C C C C C C	0 0 0 0 0 0 0 0 0 0 0 0 0 0
VODIN VODIN VODIN	e Real O selected		PLLA PLLC	e Output Lint	PLL Cardguester A Orethe Fly Overge Trepuncis Dur	Mede UVDS -	D 0.044

Figure 35: Example of successful programming of a SiT95147



7.4.2 Using the dump function

The dump function can be used to save the NVM (or I2C/SPI writes) file containing the sequence of register writes for any profile.

NOTE: Depending on the **Chip Communication** selected, **I2C** or **SPI**, one of the following NVM file types is created:

- If I2C is selected, then the NVM (I2C write) file is created.
- If **SPI** is selected, then the NVM (SPI write) file is created.

To dump the NVM (or I2C/SPI writes) file, the file must first be activated using the **Send 2 Chip** button, even if the SiTime evaluation board is *not* connected. This runs the algorithms that optimize the internal configuration based on the required inputs and outputs. After this, the **Dump** button can be used to save the list of register writes, see Figure 36.

Note, the **Send 2 Chip** process *also* provides the option of saving the UI configuration profile file, but this does not need to be saved *if* the user is only interested in the NVM files.



Figure 36: The "Dump" button can be used to save the list of register writes



When the **Dump** button is clicked, the user is prompted to save the configuration to a file. Select the NVM (I2C/SPI write) option from the drop down. Use the **Legacy State [NVM] (*.NVM)** option in this case, which allows the user to save the file to any directory location needed, see Figure 37.

· · · ·	munication Internal Clock Reference I I2C Golden Embedded	S	aves the NVM	(I2C/SPI	write) file	e	1. Send 2 Ch
• Save User Profile	Windows7_O5 (C:) Program Files (#86) Cas	scade + Temp	• • • Search Temp	P	from Pina 🔹	ZDB None	Verify EFL
	Name	Date modified Type No items roatch your search.	Siza		uency (Hz) 200M VDD 01 3,3V •		Sys Ref
OneDrive Desktop Libraries Documents Music Pictures					Delay (s)	□ o₽	
Videos Oleh Seheda	* 45.nvm				CUKP CLKN	0.0	кр кN
Save as type: Legacy	State [NVM]				tial		
Save as type: Legacy Hide Folders Hide View (State (NVM)	inipue sante	Save	Cancel	Sal Mode LVI)S +	
Save as type Legacy Hide Folders VOD VDDIN	State (NVM) Eaix No FledO selected	elipet sociel 0 3 4	Save Save	Cancel	figuration	c 0	0 0
Save as type Legacy (*) Hide Folders (*) Hide Folders (*) VDD (*) VDD (*) Mashima	State (NVM) Edit No FlesIO selected Bind's Eye ()	Load NVM	Save Save Save Save Save Save Save Save	Cancel	Isal Mode LVI ifiguration	C C ampe fout Dyn	D C
Save as type Legacy Hide Folders	State (NVM) Edit No FlesIO selected Bind's flye 💽	Load NVM	Save	Cancel	Ial Mode LVI	C C Juny fout Dyn 1.3V © 200 h	D Lond Fi
Save as type Legacy Hide Folders VDD VDDIN Realtime Hz	State (NVM) Edit No FlesIO selected Bind's five 👀	Ingent Sector	Save () Ourper con 2 0T 7 6 Dump r	Cancel A Con-the-F Frequencies	tial Mode LVI riguration B In Change Dump Fly Ch	C C C S ange fout Dyn 3.3V © 200 N	
Save as type: Legacy Hide Folders VDD Realtime Heg	State (NVM) Edit No FledO selected Bind's Syre ③	Input State	Save () Ourpue son 2 0T 7 6 Dump C	Cancel	tial Mode LVI	C C J arrupt fand Dyn J JY & 200 M	D C C C C C C C C C C C C C C C C C C C

Figure 37: Saving the NVM



Figure 38 shows an example of an NVM (I2C write) file.

mport time
Version: w1.2H.4rc4
PIF HARD REFET
ine.sleep(le-1)
2c 12cw(0x69,0xfe,0x00)
2c.12cw(0x69.0xfe,0x013
ing. sleep (16-1)
To disable reset from reg.
2c.12cw(0869,0stw,0s00)
ise.sleep(40De-6)
GENERICI bayin
2c.12cv(0x65,0xff,0x00)
2c.12cw (0x89, 0x10, 0x00)
2c.12cw(0x69,0x24,0x6e)
2c.12cw(0x60,0x25,0x25)
2c.12cw (0x49, 0x28, 0x10)
2c.12cw(0x69,0x27,0x00)
Zc_12cw(0x49,0x28,0x00)
20.1200(0869,0829,0801)
2c_12cw(0x69,0x19,0x00)
2c_12cw(0x69,0x11,0x00)
2c_12cw(0x69,0x12,0x00)
2c_12cw(0x60,0x13,0x00)
Bc.iBcw(0x65,0x14,0x07)
2c.12cw(0669,0m15,0m00)
Bc.i2cw(0x85,0x16,0x00)
2c.12cw(0x69,0#17,0#90)
2c.12cv(0x69,0x18,0x00)
lc.12dw(0x89,0x23,0x81)
2c.12cw(0x69,0x2s,0x00)
2c.12cw(0x69,0x2c,0x06)
2c.12cw(0x49,0x20,0x00)
26_12cw (0x69, 0x24, 0xc0)
2c.12cw(0x09,0x21,0x00)
SEMERIC: end

Figure 38: Example NVM (I2C write) file

Figure 39 shows an example of an NVM (SPI write).

import time
Version: v1.20.4rc4
FIF HARD RESET
time.sleep(le-3)
spi.spiw(Oxfe.0x00)
spi.spiw(Oxfe,Ox01)
time.sleep(1e-3)
spi.spiw(Oxfe,0x00)
time.sleep(400e-6)
GENERIC: begin
spi.spiw(0xff,0x00)
spi.spiw(0x10,0x0e)
spi.spiw(0x24,0x6e)
spi.spiw(0x25,0xff)
spi.spiw(0x26,0x10)
spi.spiw(0x27,0x00)
spi.spiw(0x28,0x00)
spi.spiw(0x29,0x01)
spi.spiw(0x19,0x00)
spi.spiw(0x11,0x00)
spi.spiw(0x12,0x00)
spi.spiv(0x13,0x08)
spl.spiw(0x14,0x07)
spi.spiw(0x15,0x00)
spi.spiw(0x16,0x00)
spi.spiw(0x17,0x80)
spi.spiw(Gm18,Cm00)
spi.spiw(0x23,0x81)
spi.spiw(0x2a,0x00)
spi.spiw(0x2c,0x06)
spi.spiw(0x2d,0x00)
spi.spiw(0x2e,0xc0)
spi.spiw(0x2f,0x00)
GENERIC: end

Figure 39: Example NVM (SPI write) file

The NVM file can be directly written to the chip using the **Load NVM** button, but it cannot be used to load the UI configuration profile.


7.4.3 Saving efuse.NVM.py (I2C/SPI) files

Select the NVM (I2C/SPI write) option from the drop down. Use the **EFUSE Locking (*efuse.NVM.py)** option for this case. This allows the user to save the file to any directory, see Figure 40.

6	ip Communication Internal Clock Reference	Sa	ives the Efuse N	NVM (I2C/S	Pl write)	1.
Seve User Profile	mputer Windows7_O5 (C) Program Files (db)	Cascade + Temp	• [4 ₉] Search Temp	p from Pi	ns 2DB None 4 3 2 1 0 0B	Verify EFU
Favorites Focktop Desktop D	* Mame	Date modified Type No ferme match your search.	Sire	Delay nded CLKP	V00 0T 1.3V • (4)	Sys Ref
Save as tuner	FEUSE Locking			-		
Save as type Hide Folders VDD VDDIN	EFUSE Locking 10 Iddt No FiedO selected	approvide 0 3	5ave	Cancel PLL Configurati	Mode LVDS V an B C O O	D
Save as type Hide Folders VDD VDDIN Estimate Estimate	EFUSE Locking 10 I dit No FiedO selected Dird's Eye 💽	Load	Save	Cancel PLL Configuration	Mode LVDS - en B C O Ge ge mop Fly Change fout Dy	D D D
Save as type Hide Folders VDD Evaluation Evaluation	EFUSE Locking 10 & &dit No FieldO selected Dird's Eye 💽	Input Crite 0 1 1 1 1 1 1 1 1 1 1 1 1 1	Save	PLL Configurati A On-the-Fly Chan Frequender De	Mode LVDS on B C ge ge ge 3.3V (\$ 200	D D MHz O O
Save as type Hide Folders VOD VODIN Estimate He	EFUSE Locking 10 Eddt No FledO selected Dird's Eye 🕑	I Lood III/M	Save	Cancel PLL Configuration	Mode LVOS -	n) Lond Fl
Save as type Hide Folders VDD S VDDIN Essitteme He	EFUSE Locking 10 Edit No FiedO selected Bird's Eye 💽	Input Urbe 0 1 1 Losd Dented NVM PLLA PLLB PLLC	Save	PLL Configurati A On-the-Fly Chan Feequendits Du	Mode LVDS on B C O ge ge ang Py Chonge fout Dy 3.3V \oplus 200	D D D D D D D D D D D D D D

Figure 40: Saving Efuse NVM(I2C/SPI) files



Figure 41 shows an example of an Efuse NVM (I2C write) file.

```
# Version: v1.28.4rc4
# FIF HARD REDET
 time.sleep(1e-3)
ilc.i2cw(0x69,0xfe,0x00)
i2c.i2cw(0x69,0xfe,0x01)
time.sleep(1e-3)
# To disable reset from
12c.12cw(0x60,0x60,0x00)
time.sleep(400e-6)
# GENERIC: begin
120.12cw(0x09,0xff,0x00)
12c.12cw (0x69, 0x10, 0x0e)
12c.12cw(0x69,0x24,0x6e)
13c.12cw(0x00,0x25,0xff)
12c.12cw(0x00,0x26,0x10)
12c.12cw(0x00,0x26,0x10)
12c.12cw(0x00,0x20,0x00)
12c.12cw(0x00,0x20,0x00)
12c.12cw (0x69,0x29,0x01)
12c.12cw (0x69,0x19,0x00)
ile.ilew(0x69,0x11,0x00)
12c.12cw(0x49,0x12,0x00)
12c.12cw(0x69,0x13,0x00)
ilc.ilcw(0x49,0x14,0x07)
120.12cw(0x49,0x15,0x00)
12c.12cw (0x69,0x16,0x00)
12c.12cw(0x69,0x17,0x80)
12c.12cw(0x69,0x17,0x80)
ilc.ilcw(0x69,0x23,0x01)
12c.12cw(0x69,0x2s,0x00)
 12c.12cw(0x65,0x2c,0x06)
12c.12cw(0x09,0x2d,0x00)
12c.12cw(0x69,0x2d,0x00)
 12c.12cw(0x69,0x25,0x00)
# GENERIC: end
```

Figure 41: Example Efuse NVM(I2C write) file

Figure 42 shows an example of an Efuse NVM (SPI write) file.

import time
Version: v1.20.4rc4
PIF BARD RESET
time.sleep(le-3)
spi.spiw(Oxfe,Ox00)
spi.spiw(0xfe,0x01)
time.sleep(le-3)
spi.spiw(0xfe,0x00)
time.sleep(400s-6)
GENERIC: begin
spi.spiw(0xff,0x00)
spi.spiw(0x10,0x0e)
spi.spiw(0x24,0x6e)
spi.spiw(0x25,0xff)
spi.spiw(0x26,0x10)
spi.spiw(0x27,0x00)
spi.spiw(0x28,0x00)
spi.spiw(0x29,0x01)
spl.splw(0x19,0x00)
spi.spiw(8x11,0x00)
spi.spiw(0x12,0x00)
spi.spiw(0x13,0x00)
spi.spiw(0x14,0x07)
spl.splw(0x15,0x60)
spi.spiw(0x16,0x00)
spi.spiw(0x17,0x80)
spi.spiw(0x18,0x00)
spi.spiw(0x23,0x01)
spl.splw(0x2a,0x00)
spi.spiw(0x2c,0x0e)
spi.spiw(0x2d,0x00)
spi.spiw(0x2e,0xc0)
spi.spiw(0x2f,0x00)
* GENERIC: end

Figure 42: Example Efuse NVM (SPI write) file



7.4.4 Saving the Cascade SiTime GUI state

The state of the Cascade SiTime GUI (i.e. the values of the widgets) can be saved as a .json file.



Figure 43: Saving the state of the Cascade SiTime GUI

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 44: Loading files using the Cascade SiTime GUI

SiTime



7.4.5 Using the load NVM function

When the **Load** button is pressed, an open file dialog window pops up. Navigate to select the Cascade SiTime GUI configuration file, see Figure 45.

	statistics internet lines between		The state in the				
COOP + Computer + V	Windows7_OS (C:) + Program Files (x86) + 4	Cascade 🕨 Temp	Search Temp	P	INTRB Edit	Reset Chip	Send 2 Chip
Organize 🔹 New folder	A		8⊒ • [1 0	DCO from Pins	 ZDB None 	Verify EFUSE
🙀 Favorites 🍈 Name	Date	e modified Type Size			6 5 4 3	2 1 0 0B	1
Desktop		No items match your search.			put 0T		1
Recent Plac						2001	
Documents					Frequency (Hz)	2001/4	
 OneDrive 							
E					DB	VDD 0T	Sys Ref
Elibraries						3.3V +	
Documen							
a) Music					Delay (s)		
Pictures					and a Forded		
Videos					ui interiore		
🔒 Oleh Sehed							
Computer					C CLKP	0.0	1.KP
Network					S CTRN	20	1 634
File name:			Current Version	-	C) sent		
			Open Can	cel	fferential		
-		Fast Lock HoldOver	Input #3 Sync	-	Mod		
VDDIO FlexIO	Edit	Input Order	Output List	PL	L Configuration		
VDD		0 1					
		3			S 51	100	22.5
	No FlexIO selected	3	😨 OT 🗌 7 🔲 6	1	в	с	D
O VODIN	No FledO selected	3	1 0T 🗌 7 🔲 6	0	в	c	D .0
O VDDIN	No FlexIO selected		0T 07 06	L On-	A B	¢	D
© VDDIN	No FlexID selected Bird's Eye	Load NVM	0 0T 7 6	Cn- Freque	the-Fly Change	C Change fout Dy	D C
© VDDIN	No FlexID selected Bird's Eye	Load NVM	0 0T 7 6	E On- Freque	the Fly Change	C O Y Change fout Dy	D C
VDDIN Eraltime	No FlexID selected Bird's Eye	Load NVM	0 0T 7 6	E On- Freque	the-Fly Change	C Change fout Dyn 3.3V © 200	Load Fly
VDDIN Eraltime	No FiedO selected	Load NVM	0 OT 7 6	E On-	the-Fly Change	C O Y Change fout Dyr 3.3V @ 200	D Load Fly MHz OOT
© VDDIN () Braitime 10 25 MHz	No FledO selected	Load NVM	0 OT 7 6	D On Freque	the-Fly Change	C O (Change fout Dy 3.3V @ 200	D Load Fly MHz 00T 07 06
© VDDIN () Braitime 25 MHz	No FledO selected	Load NVM PLL A	0 OT 7 6	E On Freque	the-Fly Change	C 0 y] [Change fout Dyr 3.3V @ 200	D C Load Fly MHz O O T O 7 O 6 O 5
© VDDIN () Braitime 25 MHz 11	No FledO selected Bird's Eye 💽	Load NVM PLL A PLL B	0 OT 7 6	Con- Freque	the-Fly Change	C (Change fout Dy 3.3V © 200	D C Load Fly MHz OOT OT OC OC OC OC
VDDIN Ecaltime IO S MHz I1 I2 I	No FledO selected Bird's Eye 💽	Load NVM PLLA PLL B	0 OT 0 7 0 6	Preque	the-Fly Change	C (Chainge fout Dy 3.3V © 200	D C Load Fly MHz OOT OC OC OC OC OC OC OC
VDDIN Ecaltime I S MHz I I I I I I I I I I I I I	No FledO selected Bird's Eye 💽	Load NVM PLLA PLL B PLL C	0 OT 0 7 0 6	D On- Freque	the-Fly Change nncies Dump Fl	C 0 3.3V © 200	D C Load Fly MHz OOT O7 O6 O5 O4 O3 O2
VDDIN Scattime	No FledO selected	Load NVM PLL A PLL B PLL C	0 OT 7 6	Don- Freque	the-Fly Change	C () (Change fout Dyr 3.3V © 200	D C Load Fly MHz OOT O7 O6 O5 O4 O3 O2 O1
VDDIN Scattime	No FledO selected Bird's Eye	Load NVM PLL A PLL B PLL C	OT 7 6	Don- Freque	the Fly Change	C (Change fout Dyr 3.3V © 200	D C C C C C C C C C C C C C C C C C C C

Figure 45: Using the Load function

Note, the state of the Cascade SiTime GUI (.json) file can be loaded using the **Load** button.



When a Cascade SiTime GUI configuration file is loaded, the programmed configuration is shown in the GUI.

See Figure 46 for an example showing a basic SiT95141 variant configuration. Examples for other SiT9514x variants are shown in the subsequent figures.

	Chie Co	munication	Internal Cleck Reference		D Phase Sync	keternapt	ch	+
S	575	* BC	Embedded MDMS	12 Times	DA DE	DVTHB Exit	Reset Clin	Sent 2 Chip
Connection	. 0	Address 0-69	36.861.61 MHz +	Milime	0.00	21 DCO from I	Pires 208 No	Ne Verty Brass
put #5 Japan #2	ingen All in	ent.62		PILA PILE PILE PILE		01 7 6 5 4	3 2 3 0 00	
l legist #0				Bandwidth		2 Output 01		
		Gapped Clock		1000 C		Imparts	0941 12344	
Preparaty (HØ)	Ged	Туря	FLL Randonder In	4 7			
9M		Differe	ntal 💌	19.998		206	V00.01	
							137 •	
Titlereta		Chair Lifest	Uni Time	🖺 Free flareing	1		0.001	
5 +		d •	2ma =	P Oack Selection		D46	wy (st)	
				Selection Manual +	C Reventive	C Single Ended		
1. Continuintin						🖂 0N	- C1 (P	
				E LockLose		0.047		0.00
		44		Delay	Timar			
		100		1.03we -	8	CODE		CONF
010	FedO		2.04	PPNA		· Officientia		
				Set	Clear		11000	
xco				14.14	2.4		Minde 19D1 +	
		124040	124	TH PERMIT		PLL Certificanties		
		Concentration and		in country		100	14.1	12
 V008V 				Output List			5	
				[]] # []] #	10 A	1.000		
0				Ser	- 0	🗇 On-the-Fly Durige		
9 comment			and rele 🕰	Tone Tone Level	Downe -	Frequences D	omp Hy Change first Dy	toud Py
						-	3.37 @	
-				PLLA	1			07
1.4Hz	-				•			III Of
				PLLB	E			E 05
-				1	B			III 04
10				>	E			03
				ALC .	E			02
								01
-				FLD	•			E 00
								= 01

Figure 46: Showing the programmed configuration for SiT95141

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 47: Showing the programmed configuration for SiT95145

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 48: Showing the programmed configuration for SiT95147

SiTime

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 49: Showing the programmed configuration for SiT95148

SiTime



7.4.6 Using the load NVM function

Load NVM loads the NVM (I2C sequence of writes) file.

When the Load NVM button is clicked, it prompts the user to load an NVM (I2C write) file, see Figure 50.

Chip Communication	sication Internal Clo CC Gottler Inter 0.45 *	ck Reference Tremediad MINU 76.06144 MPc • PLA BLE PLC C PLS	M Time	Phase Syne A A C A A A A A A A A A A A A A A A A	bitsmpt bitR8 Eds III DCO from Price 01 7 7 8 3 4 3 1	Chap Reset Chap ZD0 None 1 0 00	Send J Chap Verdy (FUSE
Septer Clock Golden Clock Frequency (Hi)	- Load NVM Profile	Hankaldh u + Windows7_05(C + Prògram	Film (M) + Cascade + Temp		• fg Sant Targ	P	
Trigger Lége Clevi Lége Vol T 5 • 4 • 2m Clock Swetch Cleck Leuis Coense FD Fine Cleck Leuis Coense FD Fine FL Configuration	Favorites Deutopo Devotoode Accent Plac Docurrents Docur	sins si795845.rvm	Data mudified 7; 10.11.20001423 P;	pa Soa Man Fila	2.0		be Ref
0010 Plui0 * VDD © VDDN	Se Network. **	ane]			Legacy State (NMM) Open	• Cancel	a) 0
() [salar	BodyE	• • • • • • • • • • • • • • • • • • •	PLL R	Dump	Co-the Ply Change Frequences Dury Ply	Otimpe Fout Due 9.3V @ 3001	
		-	MLB MLC				07 06 05 04 03 02 01
3 MH			PLD				00 COE

Figure 50: Loading the NVM (I2C sequence of writes) file



Once the file is loaded and after the chip is programmed, the **Realtime** section can be used, see Figure 51.

NOTE: If the SiTime evaluation board is not connected, then none of the on-the-fly functions will work. But the Realtime window can be useful to check the estimated current consumption for the selected configuration, even with the SiTime evaluation board disconnected.

Connection Dis Connection Debu	mai Clock Reference den Embedded MEMS sch 7636144 MHz +	S Time	A C Diviting East	U Send 2 Chip
Realtime Window			a (0)	SS Verify EPUISE
 M Jopet #2 Input #2 Input Golden Dock Trepancy Pti; Dati 	Input #0 [2] PLLA Calcul	ated Bandwidth TOTO -3.8953 kHz -1.07.0176 Hz	PLL 8 Calculated Bandwidth IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	
ZAME		6.25 MHz	6.25 MHz 2 Hour O	System D
5 • 4 • 2n		72 OHL Outputs	0C0 Cutputs	
Docktoss Coone PD FA	C and She	1 ppm • ≥ 07 ⊡ 1 ⊡ 8	Step 0 ppm +	2.69
Li Configuration	Input #2 [] FLL C Catcal	and Bandwidth	PUL 0 Calculated Sendworth	DXN
Click Realtime to open the		6.25 MHz	13 13 Mit	
e voon	Select a Page to	Set the Data form	Click Write to write at the entered value to	0
Dentires	view its Registers	to display and ent	er the specified register	
Type in which	1 Concumption Feylo 1 (mA) 10 Page (mA) -272.7 +	Mengulation Generic • Data Register Value	Decimal The Water	get the value of the specified register
Register to get		rus -		5779201440 02
Arte		PLLD	A REAL PROPERTY AND A REAL	3.37 0 400 MHz 00B

Figure 51: Descriptions of the Realtime section

When the **Return** button is clicked, the Realtime Window closes, and focus returns to the main window.

The **Register Manipulation** sub-section allows the user to read or write to a specified register on the selected page, see Figure 51.



7.5 TASK 4: Use Realtime to set output frequency with DCO or view clock monitor status

After the chip is programmed, the **Realtime** view can be used to change the output frequency using the DCO feature, or to check the status of all clock monitors, see Figure 52.

Cernection 0 Add	Clock - Realtime Window	26.36144 M	siTim	ie DA DA		Reset Chip Send 2 Chi Col Col Col Verily LPU
of #2 [next #3_head #2] laps lapse #3 © Golden Disck Prequency (H5)	Diff	input #0 [2]	FLLA Calculated Bandmidth 	PLL 8 Calcul 10770176 Hz	atud Bandwidth -42095 HHz -	00000000 40.7285 Hz
25.50%	Fine		6.25 MHz		6.25 MHz	Syster E
nggerlidge Oserlidge Vel 5 • 4 • 2m	- 2	topat #1. ()	72 0Hz 0C0 Output		7 GHz Cidge	Set the PPM change desired here (10 ppr
lock Switch Dock Loss Coarse FD File	on C	-	Step 3 ppm + 2 0T	□ 7 □ 6 9 ep		in this example). Use the '+' button to increase frequency
LL Configuration	2	hput 42 []	PLL C Calculated Bandwidth 	PLC 0 Calcul 1210200 -151235 Hz	ated Bandwidth -4.2301 MHz	and '-' button to decrease frequency.
010 Riel0		inert#122	6.25 MHs		6.25 MHz -151.7229 MHz	Hold Gyer
Beattime	Dvitt	CCC)	000 004pu 000 004pu 304p 0 ppm •	000 0 3 (2) 2 5 mg	6.8 GHz Out	pets 0
	Current Con VODEN (mA) Fetum 100 (mA)	0 10 +	Click Continuous Re Real Time informati Frequency Drift and	ad to log continuo on from Clock Loss Lock Loss Monitor	us , s.	Continuous Real
•		~	tus.	<u> </u>		5.97 (2007 1940 1940 1940 1940 1940 1940 1940 1940

Figure 52: Viewing the Realtime section after the chip is programmed



7.5.1 Interrupts

Click the **INTRB EDIT** button to open the **Interrupt Defect Selection** panel to enable or disable sticky notifications on the **INTRB** pin, see Figure 53.

yr HC Gate yr Albrack Medi mit Gate Gate mit Gate Gate <th>Chip Commu</th> <th>nication Internal Clock Rationer</th> <th>nce</th> <th>📰 Phase Sync</th> <th>Interrupt</th> <th>(h) 🕈</th>	Chip Commu	nication Internal Clock Rationer	nce	📰 Phase Sync	Interrupt	(h) 🕈
Contention Astinus fadi Resting Pade 1 If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the proof [shows file] If the pro	S 91 *	DC Golden g	Embedded MEMS	Time DA DC	Dittle Cas	Ranat Chips Send 2 Chip
Inter Base Righter Signed Terrer Terrer Terrer Caster Circle Cast	Contection 🗠 Add	HESA Gulfel	76.86344 MHz •		DCO from Pine	ZDB None Kerty BUS
Bidden Cardel Bidgened Ouck Heapenery (Hd) Carls Dyn Differential Carls Differential So Solido Carls Differential Solido Carls Differential Solido Carls Differential Solido Free	put FD Input F3 Input F3 Input F3 // Input F0	1	PLLA PLE PLE PLE		07 2 8 5 2 . V Output 07	1 1 00
Prepercy (M) Circle Type 21 min Otheratel inger fdys Circle Stells 3 min Circ	O Golden Clock	E Gapped Clock	Fait (Hz)	Normal 04d	Frequency (Hz)	200 14442
23.04: Differential Image: Lage Claret Sage: Val Time Image: Lage: Claret Sage: Val Time Image: Lage: Claret Sage: Val Time Image: Lage: Claret Sage: Val Time Image: Claret Sage: Val Time	Frequency (Hz)	Clock Type	43Hz	339 Hz		
Inger fage Cher Edge Vill Time 3 * 4 * 2 mm A 8 C B W10 Time VID Pres Proce Selection W10 Fine Detr Detr Liste Cher Edge Detr Detr Detr Detr Detr Detr Detr Detr Detr Liste Cher Edge Detr	25 M9 tz	Differential •			206	VDD 0Y Sys Ref
sgget føge Cher Søge Val Time 3 • 4 • 2min 3 • 4 • 2min bot Søde bot Sød		E Ceane Dvitt	Free Flanning	Cleck Switch	0	3.3V •
A B Both Sorder Fine Dott Sorder Fin	nggerädge ClearEdge ValTin	Set 2000 -	P Clock Selection Selection Auto + 12 Revent	Phase Build Out	Oelay (s)	
bold Sorbely DeckLoss Caese FD Free FD Free FD Fe FD	12 12 12 12 12 12 12	Ch 300 -	10000	Phase Propagation	100000000	
Deletion: Finite Diff. Provide <t< td=""><td>Clock Switch</td><td>17 Exclude</td><td>I Lack Loss HeldDver</td><td>Slope</td><td>🗇 Single Ended</td><td></td></t<>	Clock Switch	17 Exclude	I Lack Loss HeldDver	Slope	🗇 Single Ended	
Construin Canes PD Fire PD 31 P 10 10 P 10 10 P 10 10 P 10 10 P 10 10 10 P 10			Deley Timer Deay	Defte Saukom -	0N	0.04
IL Configuration Image: Set of the s	Clock Loss Coana FO Fine F	0 34 (12 *	1.00mm Me Dramupt Celect Select		0.006	0.000
A B C D BERT CLUNA PALA PALA PALA PALA PALA PALA PALA PA	196 R. H	0 1 -	PPM I DRIFT_CLAINS	IN PERALOL	CONTRACTOR INC.	C (10)
A B C D B DBFT_CLUBB PLLCUDI D DBFT_CLUBB PLLCUDI D DBFT_CLUBB PLLCUDI D DBFT_CLUBB PLLCUDI CLUBS_CLUBB PLLCUDI CLUBS_CLUBB PLLCUDI CLUSS_CLUBB PLLCUDI D DBFTT_CLUBB D DBFTT_CLUBB D DBFTT_CLUBB D DBFTTT D DBFTTT D DBFTTTT D DBFTTTT D DBFTTTT D DBFTTTT D DBFTTTTT D DBFTTTTT D DBFTTTTT D DBFTTTTTT D DBFTTTTTT D DBFTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	LL Configuration		Set 2 DRIFT_CLAINE	PILBLOK PT		
Image: Construction Image: Construct		c 1	🖉 DRIFT_CLKINQ	PLLC, LOL	· Differential	
OD Normal PLA,HO, PREZZ PLA,HO, PREZZ PLA,HO, PREZZ PLA,HO, PREZZ PLA <	2 2	8 8	PRIFT_CLADED	PLLD_1OL ##1 Sync		
000 FeedO Edit If CLOSS_CLININ If ALL_HO_HOEFREZE If A B C D • VDD No ReadD velocited If CLOSS_CLININ If ALL_HO_HOEFREZE If A B C D • VDDN No ReadD velocited If CLOSS_CLININ If ALL_HO_HOEFREZE If A B C D • VDDN Bird's Eye Image: CLOSS_CLININ If ALL_HOEFREZE Image: CLOSS_CLININ Image			CLOSS,CLADIG 19	RUA,HO,FREER		
VDD No Floid) velocited VDDN No Floid) velocited	0010 Field	Eda	91 CLOSS_CLKINE 19	/ RULHOJIUIZI	PEL Configuration	
No Field) velocited Image: Closs_CLODel PLD_HD_HD_FREEze Image: Closs_CLODel Image: Close Image:	VDD		🖉 CLOSS_CLIONE 🖉	RLIC HO FREEZE		
Bird's Eye Excel Could	- YDDIN	No Field whethed	🛛 CLOSS_CLAINE 👰	ATTO HO LEASE	• °	0 0
Bit D at type Lond WMM Oump II Integrity Change Fruit Dyn Lond Hit NHL 3.31 @ 200 MHL 0 PIL B 3.31 @ 200 MHL 0 PIL C 3.31 @ 200 MHL 0 RL C 3.31 @ 200 MHL 0 RL C 3.31 @ 200 MHL 0 RL C 3.31 @ 200 MHL 0	0				🗐 On-the-Fly Change	
PLA 2.31 @ 201 MHz 0 PLB 3.37 @ 201 MHz 0 PLL 0 0 0	O feature	Batt's tye	Lord NVM	Orab 🔳	Frequencies Durng F	Glanga frue Dyn Load Hy
PiLB PiLB PiLB PiLC 337 0 250 MHz 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0						3.31 @ 200 MHz - 00
PIL 8 3.37 © 230 MHz 0 PIL 8 0 0 PIL 0 3.37 © 250 MHz 0						III 07
PLE 000000000000000000000000000000000000	MHI					3.3V @ 250 MHz
		X.	PILE			III 05
		XX				= 00
	•		PLC			3.3V @ 250 MHz
840 337 0 400 MHz	1					
	15b		~	×		3.31 @ 400 MHz

Figure 53: Selecting the interrupt defects to enable sticky notifications on the INTRB pin



Click the **Clear** button to clear all sticky notifications on the chip, see Figure 54 and clear the INTRB pin (whose state should now be '1', assuming no defect exists now among the selected list).

Note: Particle Particle <t< th=""><th>Gamedian C Ad</th><th>DC Golden Disck</th><th>Ensbedded MEM</th><th>5</th><th>Time</th><th>A D C DURE</th><th>Edit Reset Chip</th><th>Seud 2 Ch</th></t<>	Gamedian C Ad	DC Golden Disck	Ensbedded MEM	5	Time	A D C DURE	Edit Reset Chip	Seud 2 Ch
Imput 12 [lock 12] Imput 10 [K] PLA PLB Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Imput 10 [K] Impu		- Realtime Window	10000000000	1. State 1.				Verify EPIZ
1 MH2 Image: fdge	of #] input #] input #2 input input #i © Golden Dock Inspuency Pits	Diff	Input #0 🗵 🕴	LA alculated Bandwidth -3,8953 MHz	-107.0178 Hz	PLL 8 Calculated Bandwidth IIIII -4 2095 MHz	-40,7295 Hz	
insper fdge Oue fdge No	Z MHz			6.25 MHz	Hold Over	5.25 MHz	Heat Over	Sys Ref 10
Deck Swinds Def Image: Swinds Sign Image: Swinds Si	nggerfidge Oserfidge Vel 5 • 4 • 2n	2	West 41.	72 GHE	Outputs	000 7.0Hz	Culputs	
Not Configuration A B B B B B B B B B B B B B B B B B B B B B B B B B B </td <td>leck Switch Dock Less Coarse FO File</td> <td>be C D</td> <td></td> <td>+ Dep 1 ppm + -</td> <td>$\mathbb{N}[\mathbf{u}] \square \mathbb{L} \square = \{$</td> <td>* Step II ppm *</td> <td></td> <td>21.69</td>	leck Switch Dock Less Coarse FO File	be C D		+ Dep 1 ppm + -	$\mathbb{N}[\mathbf{u}] \square \mathbb{L} \square = \{$	* Step II ppm *		21.69
M M	Li Configuration	2.000	input #2 [] #1	L C Viculated Bandwidth	(Alternational Action of the Alternational Actional Action	PUL D Calculated Bandwidth		DIXN.
Image: Section of the section of th	200 Field			6.25 MHz	Hold Diver	 6.25 MHz ~151.7229 MHz 	House Over	
Bastime Step 0 ppm Step 0 ppm 0 1 0 </td <td>C VCCIN</td> <td>Clock I</td> <td>1001 *1 (V)</td> <td>-6,75 GHI 000 +</td> <td>Outputs</td> <td>0C0 +</td> <td>Culputz</td> <td>0</td>	C VCCIN	Clock I	1001 *1 (V)	-6,75 GHI 000 +	Outputs	0C0 +	Culputz	0
Carrent Consumption VCD0 (mA) 13 VCD (mA) -272.7 TOTAL (mA) 250 VCD (mA) 320 VCD	Beatime	- Pint	Street 1	-) poh 0 bhu -	D.1	Step 0 ppm •	C) 1 (K 00	n Lost Pr
TOTAL (má) 200 Register Value Witte Read Ceat 000		Carriet Cono VODIN (mA) Fetuen	emption Re 10 -212.7	gister Mangulation Page Generic +	Clcik the	Clear button to nterruptS	Continuous Rest	MHC 00
		TOTAL (mA)	30	Register	Takar	_wa	Read Cean	05 00 MPU

Figure 54: Using the Clear button



7.5.2 On-the-fly change

Select the **On-the-fly Change** checkbox to enable the on-the-fly change feature allowing the user to enter frequencies.

Chip Communication Driemal Cluck Referen	nteriand MEMS		Interrupt	Least Chip Sent2 Chip
Import #12 [import #2] [import #2] [V] [import #0	PLL & PLLS PLLS PLLS		41 2 4 5 3 2	1 06
C Golden Clock El Gapper Clock Frequency (Hd) Clock Type	Fart (Hz) 4 Mz	Normal BHg 100 Hg	Frequency (Hd) 7	0 MiHz
23 Mini Differential • Counts Differential •	Free Running IP Clack Selection Selection Auto IP, Revertive	Clock Switch Helens Switch Phase Build Out	2DB V © 1 Delay (10	DDBT SysRef JV • ⊡
Cluck Solidh Eliza Dath Olick Loss Coarse FD Fine FD Int (102 - + Eliza Dath Eliza Dath (102 - + Dit (102 - +)	Interview HeldOver Datay Terrar Detay 102ms III SUms SVM Average Set Class 1845.*	Nor Unit the Eartheadth (*)	© Engle Inded	C CER
PLL Configuration	4 • 3 • 3 DC0 Mode.	El Jupiter (19	 Differential Mode 	Lubs +
VDDD ReaD Late.	Broat Order	oopatal ⊛et ⊡2 (⊡4	E PLL Configuration	6 00 0 0
Diffusion	S Lund S Lond WM	Dump 🕞	Frequencies Duma Ry	[Duege boat Dyn] Load Py
10 Ba	PILA			5.3V © 200 MHz OOT
	PLE	Select On to enable change fe	-the-Fly Change the on-the-fly ature	05 03 33V © 250 MHz 02
25 MHz	PLD			3.3V 0 400 MHz 00B

Figure 55: Enabling on-the-fly change feature



7.5.3 Managing on-the-fly frequencies

Click the **On-the-Fly** section **Frequencies** button to open the panel where you can enter and manage the on-the-fly frequencies. Use the two buttons at the top to clear one or all selected frequencies.

Near the bottom of the panel is the **New (Hz)** field where you can enter a new frequency and then click the **+** button (right of the field) to add the new frequency to the list, see Figure 56.



Figure 56: Using the On-the-Fly Frequency pop up

After the chip is programmed, the dump fly and load fly functions are enabled.

Note, if the connection to the chip is not active, you will *not* be able to operate this feature.



7.5.4 Using the dump fly function

The Dump Fly function allows the user to dump the NVM (I2C write) file, see Figure 57.

	Chip Communication	Internal Clock	Reference			Phase Sync	Interrupt	
S	SPI 9 I2C	Golden	Embedded MEMS		T	A D C	INTRB Edit	Reset Chip Send 2 Chip
Connection	Address 0x6d	0	76.86144 MHz 💌	S	Time	□ 8 □ 0	DCO from Pins	ZDB None Verify EEUSE
5	and a second		Dit a succelsor	alau al				
input #2 input #1 inp	ut #2 Input #3		PLL & PLL B PLL				01 1 6 5 5 2	1 08
M Input =0			banowioth				i ouput of	
C Golder	Clock But Choose folder to due	Ganned Clock		tan bar		c	2	
Frequen		inp set the till teas			Select a	folder		
25 MHz	Cor	mputer + Windows7	OS (C:) Frogram Files (s	86) 🖡 Cascade 🕨 🗸			Search v1.30.4-SiTone	P Sur Part
	Organize 🕶 Nev	r folder					II • (Sys Ker
	🔶 Favorites	Name	*	Date modified	Туре	Size		
Trigger Edge Clear	E Desktop	au au		10.11 2020 18-12	File folder			-
5 • 4	Downloads	images		10.11.2020 18:32	File folder			
	📃 Recent Plac	Le Include		10.11.2020 18:32	File folder			
	Documents	lib2to3		10.11.2020 18:32	File folder			
Clock Switch	 OneDrive 	🗼 numpy		19.10.2020 13:40	File folder			El OP
Clock Loss Coa	-	🗼 redist		10.11.2020 18:32	File folder			
177	E Desktop	📕 static		10.11.2020 18:32	File folder			C) CLKP
121	🕞 Libraries	📕 tcl		10.11.2020 18:32	File folder			0.000
	Documen	🍌 tk		10:11:2020 18:32	File folder			U CLAN
PLL Configuration	🚽 Music	🍶 win32com		19.10,2020 13:40	File folder			
A	Pictures							
17	Subversio							
÷.	Videos							/DS -
-	🔒 Oleh Sehedi							
VDDIO	Computer							
ODV 🖲	Network *							
		Folder						C D
C VDDIN							ielect Folder Cancel	0 0
					_		On-the-Fly Change	
U Realtime		Bird's Eye		Load NVM		Oump	Frequencies Dump Fly	Change fout Dyn Load Fly
		1.	=)					
								3.3V @ 200 MHz
				PLL	A 💽			001
				PL	P			07
25 MHZ		~	Land	-			"Dump Fly" di	umps the 🔤 👝
11 🔲		~		PLL	8		NIV/N//12C writ	a) filo
		$\langle \rangle$	<	•	5			E) 1112 ■ 05
		\geq						03
12	//			PLL	с 🚬			3.3V @ 250 MHz
/	-			- F	<u> </u>			02
13								01
25 MHz				PLL	D			3.3V @ 400 MHz
								008

Figure 57: Click the Dump Fly button to save the NVM(I2C write) file

7.5.4.1 Static profile on-the-fly with single output per PLL

When the **Dump Fly** button is clicked, it prompts the user to dump the NVM (I2C write) file, see Figure 57.

Note, the number of files created are 4*N (N- number of frequencies entered). The created file will have fixed file name.

Example: PIIX_pin_fout_otf_NVM.py

```
X - A, B, C, D
pin - Output name (Example: if OUT6 it will be displayed as 6)
fout - Frequency that are entered in the Frequency box
If the frequency entered was "125MHz", then the file created will be follows:
Single output : PllA T 12500000p0 otf NVM.py
```



7.5.4.2 Selecting static profile on-the-fly with multiple outputs with same frequency

There is no special change if multiple outputs are needed from the same PLL at the same frequency. However, the outputs need to be selected and defined appropriately, see Figure 58.

Connection Input #0 Input #1 Inpu Input #0	Chip Communication SPI © I2C O Address 0x6d ut #2 Input #3	Internal Clock Referer Golden E Clock 7	ice mbedded MEMS 686144 MHz • PLL & PLL <u>8</u> PLL <u>C</u> PLL <u>D</u> Bandwidth	Si Tim		hase Sync	Interrupt INTR8 Edit DCO from Pins 0T 7 6 5 3 2 Image: Construction of the state of the sta	Reset Chip Send 2 Chip 20B None Verify EFUSE 1 0B
Frequen 25 MHz	Clock Rem Choose folder to dump Organize + New fo	Ganned Clock On-the-Fly files uter • Windows7_OS (C:) + Program Files (186) Cas	Se tade • v1.30.4-SiT	lect a folder	▼ 4 ₇ S	earch v1.30.4-SiTone	Sys Ref
Trigger Edge Clean S • 4 Clock Switch Clock Loss Con DL Configuration A W	 Favorites Desktop Downloads Recent Plac Documents OneDrive Desktop Libraries Document Music Pictures Subversio Video Schedi 	Name dll images include ibito3 ib	Date m 10.11.2 10.11.2 10.11.2 19.10.2 10.11.2 10.11.2 10.11.2 10.11.2	pdified Typ 2018;32 File 2018;32 File	e Size folder folder folder folder folder folder folder folder folder			
VDDIO VDD VDDIN	Computer Network - Fol	deri				Sele	ect Folder Cancel	C D
Realtime		Bird's Eye 🥥		ed NVM	Our		On-the-Fly Change Frequencies Dump Fly	Shange fout Dyn Load Fly
	\geq			PLL 8 PLL C			with 20 wite)	06 05 03 3.3V ⊕ 250 MHz 02 01 3.3V ⊕ 400 MHz 02

Figure 58: Selecting static profile on-the-fly with multiple same frequency outputs

The output files will look different from the single output files as shown below.

Example: Pll<mark>X_pin1_</mark>fout_pin2_fout_...otf_NVM.py

```
X - A,B,C,D
pin1/2...- output names (Example: if OUT6 it will be displayed as 6)
fout - frequency that are entered in the Frequency box
```

If the frequency entered was 125 MHz, then the file created will be like the following example.

```
Single output: PllA T 12500000p0 otf NVM
```



7.5.5 Dynamic profile on-the-fly with single output per PLL

Two additional files will be created to support the dynamic embedded algorithm for frequency change onthe-fly, see Figure 59:

- OntheFly_current_fvco_fout.json
- OntheFly_globals.json

Name	Date modified	Туре
PIIB_5_622080000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIB_5_156250000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIB_5_15000000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIB_5_12500000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIB_5_75000000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIB_5_50000000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIA_0T_622080000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIA_0T_156250000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIA_0T_15000000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIA_0T_125000000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIA_0T_75000000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIA_0T_5000000p0_otf_nvm	4/8/2020 5:26 PM	PY File
PIIA_T_12500000p0_otf_nvm	4/6/2020 5:48 PM	PY File
PIIA_T_1500000p0_otf_nvm	4/7/2020 7:13 PM	PY File
OntheFly_globals.json	4/8/2020 5:43 PM	JSON File
OntheFly_current_fvco_fout.json	4/8/2020 5:44 PM	JSON File

Figure 59: Dynamic Profile on-the-fly with single output per PLL

To run the dynamic change function, a separate folder is provided with the following functions.

The src directory has the following files:

- dyn_change_out_fout.py
- flymode_freq_change_latest_multi.py

The main function is in the file named **dyn_change_out_fout.py**. If this python file is opened in an editor, at the end of the file, the **dyn_change_out_fout** function is called with three inputs, as follows:

- 1) DIR Path to the directory where the following two files were created:
 - OntheFly_globals.json
 - OntheFly_current_fvco_fout.json

2) Output pin name e.g. 6 (for output O6).

3) Fout frequency *desired* as a string that matches the original specified list in the GUI, described in the following subsection.

To receive the relevant frequencies, the function can be run with 0 inputs and the report with the appropriate inputs specified.



7.5.5.1 Creating a dynamic profile that uses on-the-fly frequencies

To create a dynamic profile that uses on-the-fly frequencies, first create the list of the frequencies required, and then save it (this should be done before chip programing). Then create additional files for each frequency defined in the initial list that separates files with the appropriate file names (e.g. PIIA_0T_12500000p0_otf_nvm as the file name for output 0T; connected to PLLA with frequency 125 MHz).



7.5.5.2 Description of Dynamic State Files

OntheFly_globals.json: This file contains the variables set for dynamic changes. (Note, the file is formatted for easy reading, see Figure 60).



Figure 60: OntheFly_globals.json

OntheFly_current_fvco_fout.json: This file has the current FOUTs and FVCOs, which get updated when the dynamic function is run.



After Step 1, the outputs that need to be changed would have updated frequency, in this example "5" (out_fout["5"]) that comes from PLL B being changed, and therefore is the frequency of PLL B (pll_fvco["N"]) as set in Task 1, see Figure 61.







7.5.6 Description of dynamic header files

OntheFly_globals.h: Has the variables set for dynamic changes, see Figure 62.

```
struct global_state global_state X;
        int output_i_gX = 0;
int pll_i_gX = 0;
       global_state_X.dev_addr = 109;
        global_state_X.xo_doubler_dis = 0;
 6
        global_state_X.fref_freq = 155722888.8000000;
        /* pll_param */
       global state_X,pll_param[pll_idx("A")] = parse_pll_param("010000110000000");
global_state_X.pll_param[pll_idx("B")] = parse_pll_param("0100001110000000");
global_state_X.pll_param[pll_idx("C")] = parse_pll_param("0100001110000000");
       global state X.pll param[pll idx("D")] = parse pll param("0100001110000000");
        /* pl1 bw */
       global state X,pll bw[pll idx("A")] = (struct bw) { 4000,000000, 100,000000 };
global state X,pll bw[pll idx("B")] = (struct bw) { 4000,000000, 100,000000 };
global state X,pll bw[pll idx("C")] = (struct bw) { 4000,000000, 100,000000 };
global state X,pll bw[pll idx("D")] = (struct bw) { 4000,000000, 100,000000 };
global state X,pll bw[pll idx("D")] = (struct bw) { 4000,000000, 100,000000 };
14
16
18
        /* pll_out */
19
     for (pil_i_gX = 0; pil_i_gX < NUM_PLL; pil_i_gX++) (
    for (output_i_gX = 0; output_i_gX < NUM_OUTPUT; output_i_gX++) {
        global_state_X,pil_out[pil_i_gX].connected[output_i_gX] = false;
    }
}</pre>
       5
       global_state_X.pll_out[pll_idx("A")].connected[output_idx("OT")] = true;
global_state_X.pll_out[pll_idx("B")].connected[output_idx("5")] = true;
global_state_X.pll_out[pll_idx("C")].connected[output_idx("2")] = true;
24
       global state X.pll_out[pll_idx("D")].connected[output_idx("DB")] = true;
28
        /* out pli *,
1.5
       global_state_X.out_pll[output_idx("0B")] = pll_idx("D");
       global_state_X.out_pll[output_idx("DT")] = pll_idx("A");
       global_state_x.out_pll[output_idx("2")] = pll Idx("C");
global_state_x.out_pll[output_idx("5")] = pll_idx("5");
34
       /* pll_fins */
global_state_X.pll_fins[pll_idx("A")] = "25000000/4":
36
       global_state_X.pl1_fins[pl1_idx("N")] = "25000000/4";
global_state_X.pl1_fins[pl1_idx("C")] = "25000000/4";
211
       global_state_X.pll_fins[pll_idx("D")] = "2500000074";
4.5
41
         /* foots fycos */
    42.
4.1
44
45
11
```

Figure 62: Description of dynamic header files



After dumping the static files in a folder, the **Change fout Dyn** button will be highlighted after dumping the static files in a folder.

Crep communication Internal circle reference		Phase Sync	Interrupt	(1) +
Sin III Golden En Connection © Address De6d R	ABEAR MENS		NIRSER	Reset Chips Send 2 Chips
014 #0 (knjuž #2 (knjuž #2 (knjuž #2)) 2). knjuž #0	PLA PLE FLE PLE		01 7 8 5 2 2 2 Output 01	1 00
🗇 🗴 Gablen Clock 📋 🗄 Gapped Clock	Fact (Hz)	Normal (Hz)	Frequency (Hz) 201	MH2
Frequency (Hz) Clock Type 25 1/04L Cefferential •	4.640	100 Hg	2006	Det Sale
🖾 Coarce Drift	C Free Raming	Clock Switch Hitless Switch	0 33	v • 🗉
Trigger Edge Clear Edge Val Time 5et 2000 + 3 * 4 * 2ms *	Selection Auto · Revetive	Phese Build Cut.	Delay (i)	
Ci 100 - +	E Lock Loss HeldOver	O Phase Propagation	G SingleEnded	
Cleck Switch	Delay Timer Delay	Use the Bandwidth -	09	0.00
Conclusions Courses FD Final FD SHT (2011)	Annu Annu	🖂 Frequency Ramp	0.000	6 con
PU. Configuration	5et Chur 14745 *	Steps	100	
A 8 C D	(2) DCO Mode	and a state of	Differential	
N N N N	E Feit Lock HoldÖver	🗇 logist #3 Sync	Mode	LVDS +
DOID PiedD Edit	Input Order	Output List	(2) PLL Configuration	
No FielD selected	0	27 er 🗆 7 🗆 6	A 8	6 8) 0 0
Erstine Buts Eye	Lood Internet	Dump 🖶	On-the-Ry Change Fileparson Dump Ry	Overge Post Dyn Least Fig.
				1.1V @ 200 MPH
			Change fo	out Dyn
	ALS .			05
	нис		_	3.3V © 250 MHz 02
		-		

Figure 63: Using the Change fout Dyn button



When the **Change fout Dyn** button is pressed, the **On-the-fly Dynamic Frequency** window will pop-up, see Figure 64.

Chip Communicat	ion Internal Clock Refere Golden g	nce Imbedded MENS	Phase Sync	Interrupt Buttes Late	(U) Rest Cha	1. Send 2 Ok
Connection C Address	Didd .	NUMERAL MINE · MILLINE	0.0.0.0	DCO from Pine	. ZOB None	Verify EFLS
pot #1 Input #1 Input #2 Input #3		PLA PLE PLE PLE		01] [[] [] []] [] [] [] [] []	2 1 48	0.00
C Gelden Ceck	🖾 Gapped Clock	Fest (Hz)	Nermal (Hp)	Trequency (Hill	200 MPH2	
25 MHz	Differential •	- Inc	Orek Switch	209	V00 07	Sys Hef
Tnggerlidge Oserlidge VelTime	Coarse Drift	IF Occh Selection	Htless Switch	i i i	3.54 4	1075 5
5 * 4 * 2m *	Ck 800 -	Victor HoldOver	Phase Propagation	 Single Ended 		
Oeck Switch	Fine Drift	Delay Tenar Delay 1.02ms • (2) St7ms •	Une The Bandwidth =	C ON	() or	
2 0 0	Cr [8 9	PPM Average	E Tregaricy Tamp	0 0.00		ajon
PLL Configuration	c 0	4 - On-the Ry _ COLUMN	ange ange	Differential		
90 98	10 N	E Fee S - Contraction Char	input #3 Syme	t the "Erequery	de 1905 e	
VCD Field	Select the "Outp	ut Pin" from the	2 m D7 D4	A I	c	D
© VOON	dropdown that r	needs to be changed		🖲 On-the-Fly Change		
O Beastone	Harrist C		Daving -	Frequencies Durop	Fly Change fout Dy	MHz OI
MH:					1.3V @ 250	
	$\prec\!$					
•	>	PLLC			33/@25	
					2000	01

Figure 64: Specifying the On-the-fly Dynamic Frequency

After the **OK** button is pressed, the output file will be created in the exe directory.

Example: fout_pin1_freq_pin2_freq_...otf_NVM.py

pin1/2	-		outpu	t names	(Exar	ple – if	OUTOB	it will	displayed
as OB)									
fout -	frequency	that	is se	lected	in th	e Frequen	cy chan	ige drop	odown.

If the frequency selected was 156250000/5, then the file created will be like the following:

• fout_5_156250000.0.py



The python output file would look like that in Figure 65:

```
import time
## Step 1 is to go to the appropriate FLL page
12c.12cw(0x6d,0xff,0x0b)
       ##Step 2 Force power down output clocks
i2c.i2cw(0m6d,0mc0,0m01)
        ## Step3 is to go to the Output System Page
       12c.12cw (0x6d, 0xff, 0x03)
       ## Step4 is to shart the changes on output dividers that are relevant only 12c.12cw(0x60,0x38,0x00)
       12c.12cw (0x6d, 0x39, 0x00)
       12c.12cw (0x6d, 0x3a, 0x2c)
       ###Step5 is to do small trigger change
       12c.12cw (0x6d, 0x0f, 0x00)
7.4
       12c.12cw (0x6d, 0x0f, 0x04)
       i2c.i2cw(Ow6d,Ow6f,Ow00)
time.sleep(IOe-6)
## Step 6 is to go to the appropriate FLL page and Write the NVM Copy bank for FLL
# FLB: begin
       12c.12cw(0x6d,0xff,0x0b)
12c.12cw(0x6d,0x10,0x8f)
12c.12cw(0x6d,0x11,0x6b)
       12c.12cw(0x6d,0x12,0xac)
12c.12cw(0x6d,0x13,0xcs)
12c.12cw(0x6d,0x14,0x24)
       12c.12cw (0x6d, 0x15, 0x83)
26
       12c.12cw (0x6d, 0x19, 0x6d)
       12c.12cw(0x6d,0x2d,0x5d)
12c.12cw(0x6d,0x2e,0x2c)
       12c.12cw (0x6d, 0x17, 0x2d)
       i2c.i2cw(0x6d,0x18,0x8a)
i2c.i2cw(0x6d,0x18,0x8a)
       12c.12cw (0x6d, 0x1b, 0x2c)
       12c.12cw (0x6d, 0x1c, 0xb9)
34
       12c.12cw (0x6d, 0x1d, 0x4c)
       12c.12cw(0x6d,0x1e,0x04)
12c.12cw(0x6d,0x1f,0x80)
       12c.12cw (0x6d, 0x20, 0x00)
       i2c.i2cw(0x6d,0x21,0x00)
i2c.i2cw(0x6d,0x22,0x00)
40
       ize.i2cw(0x6d,0x23,0x00)
41
       i2c.i2cw(0x6d,0x24,0xff)
       120.12cw(0x6d, 0x25, 0xff)
43
       12c.12cw (0x6d, 0x26, 0xff)
64
       12c.12cw (0x6d, 0x27, 0xff)
45
       12c.12cw (0x6d, 0x16, 0x9f)
44
     i2c.i2cw(0x6d,0x20,0x00)
```

Figure 65: On-the-fly python output file example



7.6 Using the load fly function

When the **Load Fly** button is clicked, it prompts the user to load the NVM (I2C write) file created by the on-the-fly change feature, see Figure 66.

S Cormection	Chip Communication SP(BC Address 0	Golden Clock R Clock dd	eference Environdaded ME 29,885,44 MHz	M5	SiTim	e	Internapt Petition for DCO from	n Pina 🔹 208	Nap Send 2 Ch None Verity EFU
# P] Input #1 Input	a #2 Inpact #3		PLL A . PL	15 PLC PLD			07 7 8 5	1 2 1 68	
Input #0 new Line	all Oni-the Fly NVM I	le .	-			and NIVM/12C write		and Sec	
	-						/		
COL.	amp + Camp	uter + Windows7_05(C)	+ TEMP + Out	10, Py_3/195345		iles created for	CPLEASTRIC.	P 00 MPH	
10 000	and the state of the	Mar.					a 76		
	alore - core by	1			(Jn-the-Fly Change	1.40	1.00	
AD SITUE	Feverites	Name		Date Modified	Type	20-3		- 800 UT	Sys Ref.
	Deiktop	PHC_2_156250000.p0_0	eff_rwm	1717-5850 28:31	Pytheo File	2.43		34 +	10
	Drownkoada	PIC 2,20000000 pl., o	att, nom	21.11.2020.56.21	Python File	248			
şgər Edge	Recent Flac	PBC_2_250000000p6_o	nm_th	11.11.2539 10.01	Python File	3.48			
5 -	Decuments	PHC_2_350000000y0_0	itt_nim	11.11.2520 14-21	Python File	3.69			
	OneDrive	PRC 2,400000000pd_o	at mm	11.11.2020 18-30	Pythico File	248			
	ALIEN THE	PIC 2 822080000p8 o	att_room	11.11.000010-01	Python File	218			
ck Switch	1 A A A	PID 08 500000000 e	eff mm	11.11.2930 14-31	Pathon like	10		146.00	
	Destop	P P40 05 7500000v0 v	off munt	11 11 2020 14-30	Pottom File	248		1.0	
lock Less	a Litraries	P PHD 08 12500000v0	eff men	11.11.2520 58.30	Pattern Tile	210			C (192
25	Document	2 BID OF 1500000-0	off man	11 XL (0.5) 14 (0	Pathao Dia	745		-	
	Munic:	a man on sectores of	July Internet	ALL DO WARD IN THE	- Joseph Free	145			
	Pictures			11 11 10 10 10 10					Carrie
Configure	Subversio	S Pac of 200000000	ott_mm	LI LI PLO DEGL	Pjunn rat	4.68		Load .py exte	nsion files
1	Videos	10 MID_08_2300000690	etf_nvm	11.11.2520.5630	Pythen the	1 1			
S 3	A Diate Salted	PHD_08_3500000060	_eff_mm	11.11.2000 18-31	Pythore fole			created for O	n-the-Fly
20 1	Computer	0 PND_08_40000000660_	.c0_mm	11.11.3020.26-31	Pptron File	210		Change featu	ro
9	Network *	15 PHD_08_62208000060	etf_mm	11.11.2020.34-31	Pythin File	2.63		change reatu	
		Body for) -	.000 600 1000	604	Duny 🖬	Cenur Cenur Construction Construction	hanga Damp Fly Changefi	D not.Dym
		5	\mathbb{Z}		PLL			"Load Fly" Lo	ads
	\geq	\leq	\leq		FLLC			the NVM(I2C	write) file

Figure 66: Using the Load Fly button



7.7 FlexIO

SiT9514x provides flexible input output pins to monitor the status of the chip. The monitoring options available on the main GUI include:

- Input clock status of defects
- Notify (sticky until cleared by user) of frequency drift (FD) in the input clocks
- Notify of the PLL defects

The **Edit** button in the **FlexIO** section will open a pop up to select these options. Five possible outputs can be assigned to any FlexION output, shown in Figure 67, as follows:

- Clock Monitoring Defect
- Clock Notify
- PLL Notify
- All Notify (both Clock Notify and PLL Notify)
- INTRB. INTRB



Figure 67: Using the GUI "FlexIO" widget for SiT95147

A summary of the selections made is available in the Flex IO section once Settings window is closed.



The **FlexIO Settings** panels for SiT9514x differ depending on the chip type. Figure 68 shows the **FlexIO Settings** panels for SiT95141, SiT95145, SiT95147 (panel A) and SiT95148 (panel B).

FiedO14 EnelO15 Defect/Nextly Defect/Nextly Norme Norme	FlexIC 3	
Defect/Nextly Defect/Nextly None None Ocock Monitoring Defect Input Options None # 40. Fine FD Status #0. Clock Loss Status #0. Clock Loss FD Status # 15. Fine FD Status #2. Coorse FD Status #2. Clock Loss FD Status # 2. Fine FD Status #2. Coorse FD Status #2. Clock Loss FD Status # 3. Fine FD Status #2. Coorse FD Status #2. Clock Loss FD Status # 43. Fine FD Status #2. Coorse FD Status #3. Clock Loss FD Status Clock Monitoring Nextly Input Options #3. Clock Loss FD Status #3. Clock Loss FD Status	Head B Defect (Heady Trans Defect (Heady Defe	Defectively Nore Nore PesiO 13 Defectively Nore Polock Less Status Polock Less FD Status
Otock Monitoring Defect Input Options Mic Oock Loss Status Mic Oock Loss Status Mic Oock Loss FD Status Mic Oock Loss Status Mic Oock Loss FD Status<	Head 0 B Defects/Relative States All Free FO Status 41: Free FO Status 42: Free FO Status 42: Free FO Status 42: Free FO Status 43: Comer FO Status 44: Free FO Status 45: Free FO Status 46: Free FO Status 46: Free FO Status 47: Free FO Status 47: Free FO Status 48: Free FO Status 48: Free FO Status 48: Free FO Status 48: Free FO Status 49: Free FO Status 40: Free	Pleaf013 Defect/NetRy None Holder Less Status Hi Clock Less Status Hi Clock Less Status Hi Clock Less FD Status
Obek Meintoning Defect Input Options #0: Obek Loss Status #0: Obek Loss FD St	Hedd B Defect/Actify States at Monitoring Defect Joyan Options 40, Free FD Status 41, Free FD Status 42, Free FD Status 42, Free FD Status 42, Free FD Status 43, Free FD Status 44, Free FD Status 42, Free FD Status 42, Free FD Status 43, Free FD Status 44, Free FD Status 45, Free FD Status 45, Free FD Status 46, Free FD Status 47, Free FD Status 47, Free FD Status 48, Free FD Status 48, Free FD Status 48, Free FD Status 49, Free FD Status 40, Free FD Status 41, Free FD Status 41, Free FD Status 41, Free FD Status 42, Free FD Status 43, Free FD Status 44, Free FD Status 44, Free FD Status 45, Free FD Status 45, Free FD Status 46, Free FD Status 47, Free FD Statu	Field 13 Defect/RelPy None P OverLass Status O #0 Clock Lass Status O #0 Clock Lass FD Status #1: Clock Lass Status O #0 Clock Lass FD Status O #0 Clock Lass FD Status O #0 Clock Lass FD Status
#0: Fine FD Status #1: Cook Loss FD Status #1: Cook Loss FD Status #1: Fine FD Status #2: Cook Loss FD Status #3: Cook Loss FD Status #4: Coo	Defect Notify States of Monitoring Defect Joyar Options 40, Free FD Status 41: Free FD Status 42: Free FD Status 43: Free FD Status 44: Free FD Status 45: Free FD Status 45: Free FD Status 45: Free FD Status 46: Free FD Status 46: Free FD Status 46: Free FD Status 46: Free FD Status 47: Free FD	Orfect/NetRy None Plane Plane
*1: Fine FD Status *1: Coords FD Status *1: Cook Loss FD Status	Allower de Monitoring Defect Joyan Options 40. Free FD Status 41. Free FD Status 42. Free FD Status 42. Free FD Status 42. Free FD Status 54. Comme	Mone Mone
A2 Free FD Status A2 Coarse FD Status A3 Free FD Status A4 A4 A	ok Monitoring Defect Joyar Options 40, Free FD Status 41, Free FD Status 41, Free FD Status 42, Free FD Status 42, Free FD Status 42, Free FD Status 51, Free FD Status 51, Free FD Status 51, Free FD Status	#0. Clock Less Status #0. Clock Less FD Status #1. Clock Less Status #1. Clock Less FD Status #2. Clock Less Status #1. Clock Less FD Status #3. Clock Less Status #1. Clock Less FD Status
All Fine FD Status All Cook Monitoring Notify Input Options All All All All All	40 Fine FD Status (*) 40 Cosroe FD Status 41 Fine FD Status (*) 41 Cosroe FD Status 42 Fine FD Status (*) 42 Cosroe FD Status 43 Fine FD Status (*) 42 Cosroe FD Status	#0. Clock Less Status @ #0. Clock Less FD Status @ #0. Clock Less Status @ #0. Clock Less FD Status @ #0. Clock Less Status @ #0. Clock Less FD Status @ #0. Clock Less Status @ #0. Clock Less FD Status
Clock Mentoring Nettly legal Options	42: Fine FD Status () 42: Coanse FD Status 42: Fine FD Status () 42: Coanse FD Status 43: Fine FD Status () 43: Coanse FD Status	 41: Clock Less Status 41: Dock Less FD Status 42: Clock Less Status 42: Clock Less Status
AA. New O	42: Fire FD Status 💿 42: Course FD Status	42: Clock Loss Status Co. 42: Clock Loss FD Status
	23. Eng ED Status ID 23. Course ED Status	
🐨 #0: Fine FD Notify 😨 #0: Coame FD Notify 😨 #0: Clock Loss Notify 🔍	estimate and the contract and	🔿 #3: Obek Lens Status 🕜 #3: Obek Lens FD Status
🖅 #L Fire FD Notify 📧 #L Coarse FD Notify 📧 #L Coarse FD Notify Clo	di Monitoring Notify Input Options	
※ #2 Fine FD Notify 座 #2: Coase FD Notify 座 #2: Clack Loss Notify	AL.	_ None
愛 41: Fine FD Notify 変 43: Course FD Notify 図 43: Clock Loss Notify	図 40 Fine FD Nobly 図 40 Coa	ese FD Notify 😢 49: Gock Loss Notify
	IV AL Fine FD Notify IV AL: Cos	ine FD Notity 🗵 4D Clock Loss Notity
PLI Notify PLI Options	12 42: Fine FD Notify 12 42: Cos	ine FD Notity 😥 42: Clock Loss Notity
Automatica Automatica	12 #3: Fine FD Notify 12 #3: Coa	ina FO Notify 😥 #3: Oock Low Notify
A: Holdover Freeze Notify A: Loss of Lock Netify		
I B Heldover Freeze Notify II B Loss of Lock Notify PU	Notity PLL Options	(max)
(₹) C: Holdover Freese Notify (₹) C: Loss of Lock Notify	AL	Aucose
😥 D. Holdover Presse Notify 🖉 D. Loss of Lock Notify	[2] A. Holdover Freeze Notify	[2] Ar Loss of Lack Notify
	12 B. Holdover Freeze Notity	[2] B: Loss of Lock Notity
	12 C: Holdover Freeze Natify	[2] C: Leue of Lock Netify
Α	2 D: Holdover Frezza Notify	B D: Loss of Lock Notify

Figure 68: FlexIO settings differentiation depending on the device series type A – FlexIO section for SiT95141, SiT95145, SiT95147; B – for SiT95148



For example, for SiT95148, lets select FlexIO 3 for PLL A lock loss signal monitoring, the summary of the selections made is available in the Flex IO section, see Figure 69.



Figure 69: Example using the FlexIO settings for SiT95148



7.8 Phase sync feature

In this feature, the outputs from different PLLs will maintain the same relative phase difference, even in holdover when all input clocks are lost. This feature is provided by a PLL arrangement where PLLA determines the PLL dynamics and PLLB,C,D (one or more of B,C,D) are used as subordinate PLLs that work on an internal XO reference that is derived from the output of PLLA. Using this feature is recommended only for cases where the output phases are expected to stay in sync across PLLs even with loss of the input clock. For all other cases, using this is not recommended.

Figure 70 shows the PLL phase sync section in the GUI. Once the **Phase Sync** check box is clicked, you can choose the PLLs to be synced. PLLA is always set to sync because it is the internal refere PLL (even if it has no explicit output).



Figure 70: Example of phase sync settings



NOTE: PLLs which are enabled for phase sync will be in free run mode and hence the PLL block in the GUI is greyed out for those which have phase sync enabled.

The algorithm will generate an internal frequency that allows the PLLs to be in sync.

If the outputs are selected such that this synchronization is not possible, the GUI will report the appropriate output and the internal frequency used.

The user may change the output frequency to allow the multiples of the LCM of output frequency and the internal frequency (OCXO) to be within the frequency band [6.7 GHz, 8.4 GHz]. An example error message is shown below.



Figure 71: Example of 85 MHz, 125 MHz fall (error message)

In the example shown in Figure 71, PLL B output (85 MHz, 125 MHz) is 2125 MHz, the multiples of which (6375 MHz, 8500 MHz) fall beyond the band of interest. Changing 85 MHz to 75 MHz will fix this problem.



7.9 Input to output delay control feature

SiT9514x supports a unique transient performance feature for the output clock delay with respect to the input clock. There exist two different operation modes:

• Default mode of operation:

The output always starts with a fixed phase relationship to the input rising edge across multiple power ups of the chip, see Figure 72.

- The start of the output clock with respect to the rising edge of the input is the same across multiple power ups with a total uncertainty of < ±175 ps.
- The delay parameter shown above, is consistent across multiple power cycles, where: Delay= A Fixed delay ± 175 ps
- Input #3 Sync Mode of operation:

The output always starts with a fixed phase relationship to the rising edge of an independent clock on Input #3 across multiple power ups of the chip, see Figure 73.

- The start of the output clock with respect to the rising edge of Input #3 is the same across multiple power ups of the device with a total uncertainty of < ±175 ps.
 - Input #3 is an independent clock not related to the PLL input.
- The delay parameter shown above, is consistent across multiple power cycles, where: Delay = Fixed delay ± 175 ps



Figure 72: Input to output delay across multiple power ups in default mode

Clock Generators, Jitter Cleaners, and Network Synchronizers







The Input #3 (IN3) SYNC is a per PLL feature and can be enabled for each PLL selectively in the GUI, see Figure 74. This can be a very useful feature, where the Input #3 clock can be used as an independent SYNC.

	Chip Cere	munication	Internal Clack Re	America			I Phine Sync	stempt			
8	321	* ec	Golden Clock	Evolved dout MEME	GTin	ne	1 A 11 C	INTR8 EA	*	Benet Chip	Send 3 Chip
Connection	.0	Address [Dies		76.86144 MHE =		iie -	2.4 2.8	RCO Test	Dist	· ZDE None	Verity STUTE
Nout #\$ Input I Input #0	41 input 42	input 🖓		FULA PLIS PLIS PLIS Bendwidth				07 7 8 5 ⊡ DistperΩT	4 2 2 1	8 18 08	
	Collection Class	9	illapped Clock	Fast (Hz)		Normal	0HB	Freques	ey 940 125M		
16	maning (Hal		Churk Type	<u> </u>							
			Differential	Free B	uning	Oeck Switch		206	3.00		lys ker
			Course Beft	# On A Selection		Hitles Switzk				-	100
Trappe Litys	Overlage	Val Tirra	Set 1000	Selection Auto	10 Realized	O Phase B	IN3 Sy	nc Mode			
1.1	8.3	Jun -	0 10 -	S technol Delay Timer	HishiDun Datay	- O Pour P	Soge	C CN		- 60	
Click Suitch			TO PROPERTY.	T.02ma	SElver -		/	c. cia	6 E	10.0	90 C
Ookins 2	Cystic 80	FrieFD	fet 11	2004 Set Char	Alonage 2.094c	🗆 frequenc	y Ramp linge	- ini	ú.	0	iti
			10.11	1 - 1 - 1 -	101 800 Made		K	· Differential			
PLConfigue	R.			- Ention	HaldOver	81w	nut #3 Sync		Made 13	vDS v	
	225.027				D Puelock	Output	Ger C	PLL Configurat	-		
VDDHO VDDHV VDDHV	Fiel0	No Fie	iO selected	Fdit Ingut Driller	PIED (m)	- i i	07.01	*	4	č	D
								On-the-Fly Char	6		
Bietine			Bird's Eye ()	Window Lovel Loved WVM			Curre	Tressences 1	David Ratio	Doorge Hout Dare	Lastra

Figure 74: Example showing the selection Input #3 SYNC for PLLA of SiT95148



7.10 SiT9514x jitter attenuator as timing source for JESD204B RF converters in 5G RRU

SiT9514x family of jitter attenuators offer a highly integrated clocking solution for JESD204B compliant interfaces used in radar, servo loop control and multi-channel multi-carrier applications like 5G RRU and phased antenna array MIMO. All devices in the SiT9514x family meet the stringent timing requirements for JESD204B Subclass 0 and Subclass 1.

7.10.1 JESD204B overview

JESD204B is a JEDEC standard which defines a high-speed serial interface link between data converters and logic devices. A block diagram of a JESD204B link showing the data link and timing signals is shown in Figure 75.



Figure 75: Block diagram of JESD204B interface between ADC (Transmitter) and FPGA (Receiver)

To achieve deterministic latency, each transmitter and receiver in the data link must be clocked by timing references with fixed phase relationships. The SiT9514x serves as the JESD204B clock source for providing these timing references. The SiT9514x distributes both a device clock (**Dev_Clock**) and a source synchronous system reference (**SYSREF**) signal to each device in the link.



7.11 Cascade as clock source for JESD204B timing signals

The SiT9514x can be configured to support the following JESD204B timing signals:

- Device Clock
- SYSREF
- SYSREF Request

A typical clock tree block diagram of the SiT9514x with JESD204 compliant converters and logic devices used in an eCPRI clocked 5G RRU is shown in Figure 76 below.



Figure 76: eCPRI clocked 5G RRU clock tree designed around SiT95148 and JESD204 compliant RF FE

In the above 5G RRU application, SiT9514x synthesizes multiple copies of the device clock (**DEV_CLK**) and a divided down phase locked **SYSREF** from one of the two master clock references: **eCPRI** recovered clock or the local **TCXO**. The **SYSREF_REQ** is the JESD204 *Request to Generate* a **SYSREF** trigger signal from the FPGA to a FLEXIO input of SiT95148, which in turn gates the **SYSREF** clock out. To understand the timing relationship of the **SYSREF** trigger (**SYSREF_REQ**) signal to the **SYSREF** clock output, let's review the SiT95148 architecture block diagram shown in Figure 77.


Figure 77: SiT95148 architecture block diagram correspondent for eCPRI clocked 5G RRU clock tree

As shown in the example block diagram Figure 77, SYSREF_REQ is driven into SiT95148 FLEXIO13 configured as an input. In this configuration, SYSREF_REQ trigger serves as a gating signal for the PLLB output divider, DIVO5. PLLB is configured to generate SYSREF as a divided-down phase-locked copy of Dev_CLK. The SYSREF_REQ is used as a *gating* signal to the internally generated SYSREF clock – SYREF is *gated* into the output driver when SYSREF_REQ is high and *gated off* when SYSREF_REQ is low, thereby driving OUT5 to a logical high. This *gating* of the internal SYSREF to the output is timed on the positive edge of the SYSREF clock.

Si Time



7.12 Configuring the SiT9514x for JESD204B timing signals

Given that most systems are designed around more than two converter devices, select either **PLL B** or **PLL C** to synthesize the required number of device clocks and one SYSREF from the system master clock. In a 5G RRU design, the master clock source is typically an eCPRI or 10 GbE recovered clock. Configure Cascade Platform SiT9514x products so that the selected PLL is in zero-delay buffer (ZDB) mode. Feed the output of the SYSREF fan-out buffer into **Input #3** configured for ZDB mode. This will ensure a repeatable and zero phase delay between the **DEV_CLK** and **SYSREF** pairs.

The following sections describe the procedure to configure SYSREF generation using two types of stimulus: hardware trigger signal: **SYSREF_REQ** on **FLEXIO13** or by writing once each to multiple registers.

7.13 Generating SYSREF via SYSREF_REQ

To generate **SYSREF** from a trigger signal like **SYSRE_REQ** on **FLEXIO13**:

Identify the PLL and output on which SYSREF will be driven out.
 (Figure 78 shows an example of the initial configuration of SiT95148 for JESD204B.)



Figure 78: Example of the initial configuration of SiT95148 for JESD204B



2) Go to Page 0: reg 0xFF = 0x00 or go to the Generic page in the Realtime Window, see Figure 79.



Figure 79: Selecting the Generic page in the Realtime Window's Register Manipulation section of SiT95148 for JESD204B



 Update the value of register 0x19 shown in Table 2 (as set in the previous example, where PLLB: register 0x19 = 00100010_b =0x22, see Figure 80).



Figure 80: Writing the value to Register 0x19 in the Realtime section of SiT95148 for JESD204B

4) Do a small trigger update. An example I2C script is shown below:

```
i2c.i2cw(0x69,0x0f,0x00)
i2c.i2cw(0x69,0x0f,0x04)
i2c.i2cw(0x69,0x0f,0x00)
```



5) Go to PLL page: **0xFF = 0x0B**.

(For the parameters shown in the example above for PLLB, see Figure 81)



Figure 81: Selection of the PLL B page in the Realtime section of SiT95148 for JESD204B



- 6) Update bit s[5:0] of register **0x09** as per one hot active encoding (see Figure 82) of the **SYSREF** output. This means that SYSREF can be connected to the one of the PLL outputs and to decode it in register 0x09, we need to write appropriate value for PLL B.
 - For example, if PLLB Output 5 is the SYSREF output, write 0x08 (001000b) to register 0x09, see Figure 82.



Figure 82: SiT95148 overall clock hierarchy



Reg 0x19, Bit#	Function
0	Set high for PLLA generating SYSREF from trigger on FlexIO13
1	Set high for PLLB generating SYSREF from trigger on FlexIO13
2	Set high for PLLC generating SYSREF from trigger on FlexIO13
3	Set high for PLLD generating SYSREF from trigger on FlexIO13
4	Keep this bit low = 0
5	Keep this bit high = 1
6, 7	Do not change the values, default is 0,0.

Table 2: Register settings to enable SYSREF generation from trigger on FLEXIO13

Generating SYSREF via registers

SYSREF generation can also be controlled by toggling bit 1 of register **0x05** in the respective PLL. The following steps outline the procedure to gate **SYSREF** on a specific output using register writes:

- 1) Follow steps 1 to 6 as described in the previous section.
- 2) Register **0x05**, bit **1 [1]** is the **SYSREF** trigger bit (SiTime recommends reading the current value of the **0x05** register and then using the **current value** to change bit 1), see Figure 83.



Figure 83: Reading Page PLL B Register 0x05 *before* changing bit[1] of the SiT95148 for JESD204B



a. Enable SYSREF on the selected output, set bit 1 high, see Figure 84.



Figure 84: Write bit[1] high to the Page PLL B Register 0x05 of the SiT95148 to enable SYSREF

- PLE P0.8 Spat 4912 e tant CHER CHER 1000 000 107 I Des Chiefe 255 644 020 -133.7229 W 000 -111 PL29 Main Inc Lear -0.0121010 0.16 10.5 132 134 24 1 3071.1 A Test Chierre FLLE PLLD 1.6 1000 1200 22 HOULD PLL B Page Register 0x05 Lighton timite Value CA DA DA Write button 11 711.8 NDD (rest) 41.01 С TOTAL PART -117.3
- b. Disable SYREF on the selected output, set bit 1 low (see Figure 85)

Figure 85: Write bit[1] low to the Page PLL B Register 0x05 of the SiT95148 to disable SYREF



8 Snapshots of specific use case scenarios



Figure 86: Free run scenario 1

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 87: Free run scenario 2

SiTime

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 88: Lock scenario 1

SiTime

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 89: Lock scenario 2



8.1 Free running DCO

After programming the chip, click the **Realtime** button to use the DCO mode, see Figure 90.

Chip Communication Internal Cle	ck Reference	🖾 Phase Sync	bransapt	(1)	1
Security Of Address Dod	Embedded MEMS	Time	INTER Late	Reset Chip	Send 2 Chip
Constant Constants and	15.800 MPQ •	DI DI	OCO from Pro	 ZDB None 	Verity EPUSE
nput #2[Input #2] Input #2 Deput #1	PLLA PLLS PLLC PLLD Bandwidth		01 7 4 5 3 2	1 06	
C Solim Chub	() Feet (Hz)	Normal (Hi)	frequency (Hz)	DM	
Frequency DA1 Clock Type					
(294) University		Clock Switch	ZDE	00.08	Systlat
🗆 Cana Grift	If Clark Salectus	Hitless Switch	1 B S		100
Tragger Litige Clear Edge Val Time Set (2000 -	Selection Auto - C Revent	C Phase Build Out	Delay (ii)		
Or 000 -		Phase Propagation			
Class Santo	E Lock Land HutdDam	Star	Single Ended		
	Delay Terms Prov	-	00	0.06	
CleckLoss Coarse (D Fire (D Her _ av _)	Automatical Sector	Contractory Rame	(C) (Q(P))	103	CLARK
G(1)	1204 City 1204	e in indexed and	C CLAIN	0.1	0000
PLL Configuration	8 - 2 -				
A. 8 C. 5	DCD N	tule	Differential		
0 0 0	Ferflock HollOver	🖾 Input #3 Sync	Made	LVDS +	
vteno Nedo	Edd Brout Order	CurputList	PLL Configuration		
• 100		2 at 12 12 6	4 1	E	p
No FielD volated		1999 - A. 1996	16. é	ŝ.	٠
Realtime hutton			C On-the Fly Change		
(Cantine Button		sound 🔳	Temenow Dump Po	Thange front Ify	n Load Ply
	PLLA			2.5/ @ 125	MH: 001
0 🔤					07
					06
				3,3V @ 150	MHE 05
2	1				03
	RIC			134 8 75	MH2 02
					III ou
3 💼					01

Figure 90: Free Running DCO Mode configuration

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 91: Hitless Switch (Phase Build Out setting)

Si Time

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 92: Clock Switch (Phase Propagation and Slope settings)



8.2 Zero-delay buffer mode

A zero-delay buffer (ZDB) is available in the SiT95141 clock generator and SiT95145/7/8 clock jitter attenuators for use in applications that require minimum delay between the selected input and output.

The ZDB mode is available and can be configured for any of the PLLs. This provides the option to close the feedback loop of the PLL on the PCB and thereby, bypassing the internal feedback dividers, and cancelling the delays introduced by the internal dividers and clock distribution pathways. The **Input #3** pins are used as the external feedback and any of the outputs from the PLL which is being set up in ZDB mode should be routed to the **Input #3** differential inputs. SiTime recommends using **Input #0** as the input clock when using **Input #3** as the external feedback clock in ZDB mode. The terminations used for **Input #3** would depend on the driver type chosen. The preferred option is to use an LVDS or LVDS boost output AC-coupled into a differential 100 Ω termination at the **Input #3** input side, see Figure 93.







9 Low wander mode

Low wander mode provides the best jitter cleaning by employing a dual loop PLL, see Figure 94. Refer to the datasheet and application note for a complete description.

Chip Communication	Internal Clack Reference			El Phase Synt	Internapt INTER Falls	Ċ	1
Corrector O Address De	Gect Te.	BEC44 MHz ·	SiTime	0+ 00	E Marine See	Teut Chip	Send 2 Chip
htput #2 htput #2 htput #2 htput #2		PLA PLE PLC PLL			47 7 4 5 4	3 2 1 0 1	0 00
<u>Golden Clock</u>	Cleck Type	Fait (Hz) 4000		Normal (Hg)	Prequency (H	g 125M	-
125M ToggerEdge CherEdge VolTime 5 • 4 • 3me •	Citizential •	[7] Free Farmi (F Clark Indection Selection [Solo] *] 22 Look Look	ng Reventive HeidDown Delay	Clock Switch Hetter: Switch Press: Build Out Press: Prepagation Reps	ZOB W Delay O	V00 2 1.3V •	Sys Hef
Clock Switch Clock Loss Coese FD Fire FD T D D PL Configuration	54 (32 +) 04 (11 +)	Trial The Char (A (1)	SATHE + Average (1896; +)	E Trayancy Tamp	C 04 6 0.00 6 0.00	0 0 0 0	cutt cutt
8 A	c p	Taittochuk	ICver [2] Low Wander Med	🖹 legat 43 Sync k	 Differential N 	tede LVDS +	
V000 Real0	Edd .	Input Order	t Low Wand	Output Lie ler Mode	PLL Configuration	6	D.
(forthere	Bird's Eye 🕘	E Loost E Loo	d NNM.	Dump 🔒	On-the-Ry Change Trequencies Down	n Fly Overgefreet D 3.W @10	
10 125 MPE	$\overline{\langle}$		PLLB			33V @ 12 12 33V @ 20	07 5 MHz 06 5 MHz 05 0 04 0 MHz 03
12	\geq	\triangleleft	PLLC			337 0 12	02
125 MHz			PLLO				00B

Figure 94: Selecting low wander mode



10 Usage guidelines for jitter performance optimization

10.1 Output placement and frequency planning

The output placement and frequency planning should be such that outputs with a frequency difference in the range 12 kHz to 20 MHz should not be placed next to each other and should be spaced as far apart as possible to minimize output-to-output coupling leading to in-band spurs.

Chip Communication Internal Clock Pa	ference Embedded MEMS	-	Phase Sync	Internapt INTER East	U	1
Connection © Address Dd9	78.88144 M94z +	Time	00 00	DCO trum Pins	O ZDR None	Verity Artura
na eg (hout eg (hout eg (hout eg)) hout eg) Gaster Churk	PLA PLE ALC PLD			67 7 6 5 4 2 Output 8	3 2 1 3	08
Pressence (Ht) Clark Type	Fast (Hz)		Normal (Hz)	Frequency (H	a sim	
L254 Otherminit regger Edge Ouer Edge S = A Date Index Ouer Edge Date Index Ouer Edge	Free Sum B Clock Selection Selection Asity • If LockLoss Delay Taney J.02ms • If Selection • Selection	Neng Revertive HeldOver Delay S47m • Average LDHs • Øf DCO Mode	Clock Switch Hilless Switch Phase Reception Scope 10 park •• Frequency Ramp Singe	200 Delay (r Single Initial C Cutor C Cutor C Cutor C Cutor	yobas 13v •	Synthet E
	FastLock He	AdDrer	E Trans #3 Sync		Accie LVDS +	
vool vool vool vool vool vool		Vectore	10 m 10 r 10 e	A O	с 1 0	*1
D faatura	9 5 100	and NOMA	Dump	Tinsueminis Dem	r fly] [Dange front Dy	(south
SWR		PLEA			33V © 10 33V © 156,25	MHE 00 07 MHE 06
		PLLB			136,0 1.W @ 136,25 3.W @ 175	MHE 03
		PLLC				01
	1	FLLD			1.37 @ 33	MH6 00

Figure 95: Example where outputs O3 and O2 should be spaced apart

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 96: Optimal profile (outputs re-arranged to minimize spurs)

SiTime



Additionally, for configurations having adjacent LVCMOS outputs, avoid up to the third-harmonic of the adjacent LVCMOS output, to land within the integration frequency band of the jitter sensitive output frequency, see Figure 97.

For Outputs O5/O3 (156.25 MHz), a crosstalk spur location may occur due to adjacent placement of the 50 MHz LVCMOS (50 MHz) outputs.

Chip Communication	Internal Clock Reference			📰 Phase Sync	betweenpt	(I)	t
Connection O Address Dr	Solden Embedi Clock 16.8014	ded MEMS H MHz: 🔻	SiTime	DA DA	PUTTO Edit	Reset Chip	Send 2 Chip
Traper Litye Char Litye Vel Time	Class Type Class Type Class Type Class Class Class Class Class Class Class Class	Fact (Hz) andwidth Fact (Hz) colo Fine Factors IF Cleach Selection	· · · ·	Normal (Hz) 100 Dick Sotth Hitles Sotth	COltain Pee	© 209 Nore <u>1</u> 2 <u>1 0 0 18</u> 1 DM VOD J LJV •	Verify ET-USE
S - - 246 + Clock look Clock look - - - Clock look Clock look - - - Clock look Clock look - - - RL Configuration - - - - A - - - -	Cristian FreeDott Set 182 Cristian Cri	Selection Auto - II P LockLoss - I Dday Timer LA2ms - II Set Close 4 - 2 - II FatLockHaldO	Reventive holdOver Dolay 343mm = Average 1.094s = DCD Mode Ner	Phase Ruled Out Phase Prophysion Stops If you's Friquency Ramp Reps Phase Rule RS Synce	Datay (d) Single Ended C ON O CLOP O CLOP O CLOP O CLOP O CLOP Differential M	D DF	2.09 2.09
0000 FieldD • 100 © 10094	Edds	Deput Onder	Fuse Lock	Outputtin	PLL Configuration	c •	0
Dantone.	Berl's Eye 🕘	Losd Cost	NMM.	Oump 🔂	Conthe-Fly Change	(Thy] [Change foot Dy	LeadThy
25 MHz			PLLA			3.3V @ 100 1.3V @ 10 3.3V @ 196.25 3.3V @ 196.25	MHE 001 MHE 07 MHE 06 MHE 05
2			PLE PLC			33V @ 50 23V @ 196.25 3.3V @ 10	MHE 04 MHE 03 MHE 03 MHE 02 01
a 🔳			FLL D			3.3V @ 33	MHE 00

156.25 MHz – 3*50 MHz = 6.25 MHz

Figure 97: Third harmonic of the adjacent LVCMOS output



10.2 CMOS output type selection

When setting the **Single Ended** CMOS output format, it is recommended to always select complimentary outputs to minimize single-ended CMOS output-to-differential-output coupling.

For complimentary CMOS format, the output type (**Output OT**) selection should be with **ON** and **OP** selected, **ON=CLKN**, and **OP=CLKP**.



Figure 98: Example of an *incorrect* profile for complimentary CMOS format

Clock Generators, Jitter Cleaners, and Network Synchronizers



Figure 99: Optimal profile for complimentary CMOS format

SiTime[®]



11 Document Information

Table 3: Revision history

Version	Release Date	Change Summary
1.0	31-Jan-2019	Original doc
1.01	11-Jun-2019	Corrected block diagram in section 3
1.02	30-Mar-2020	Changed according to GUI rev.1.28.4rc4
1.03	10-Nov-2020	Added Low Wander Mode description Updated rev table date format
1.04	16-Feb-2021	Extensive editorial changes throughout.

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

© SITime Corporation, February 2021. 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.