Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Si5383/84 Reference Manual

Overview

This Reference Manual is intended to provide system, PCB design, signal integrity, and software engineers the necessary technical information to successfully use the Si5383/84 devices in end applications. The official device specifications can be found in the Si5383/84 data sheet.

The Si5383/84 combines the industry's smallest footprint and lowest power network synchronizer clock with unmatched frequency synthesis flexibility and ultra-low jitter. The three independent DSPLLs are individually configurable as a SyncE PLL or a general-purpose PLL for processor/FPGA clocking, and support digitally controlled oscillator (DCO) mode for IEEE 1588 (PTP) clock steering applications. In addition, locking to a 1 pps input frequency is available on DSPLL D. The DCO mode provides precise timing adjustment to 1 part per trillion (ppt). The Si5383/84 can also be used in legacy SETS systems needing Stratum 3/3E compliance. The unique design of the Si5383/84 allows the device to accept a TCXO/OCXO with any frequency, and the reference clock jitter does not degrade output performance. The Si5383/84 is configurable via a serial interface with in-circuit programmable non-volatile memory so it always powers up into a known configuration. Programming the Si5383/84 is easy with ClockBuilder Pro software. Factory pre-programmed devices are also available.

RELATED DOCUMENTS

- Si5383/84 Data Sheet
- Si5383/84 Device Errata
- Si5383-EVB User Guide
- Si5383-EVB Schematics, BOM & Layout
- IBIS models
- To download support files, go to: 16. Accessing Design and Support Collateral

Table of Contents

1. Scope

This Reference Manual is intended to provide system, PCB design, signal integrity, and software engineers the necessary technical information to successfully use the Si5383/84 device in end applications. The official device specifications can be found in the Si5383/84 data sheet.

1.1 Related Documents

- Si5383/84 Data Sheet
- Si5383/84-EVB User Guide
- SiOCXO1-EVB User Guide

2. Overview

The Si5383/84 is a high performance, jitter attenuating clock multiplier with capabilities to address Telecom Boundary Clock (T-BC), Synchronous Ethernet (SyncE), IEEE-1588 (PTP) slave clock synchronization, and Stratum 3/3E network synchronization applications. The Si5383/84 is well suited for both traditional and packet-based network timing solutions. The Si5383/84 contains three independent DSPLLs allowing for flexible single-chip timing architecture solutions. The Si5383 contains a single DSPLL D that can be configured for 1PPS applications to lock to a 1 Hz input, requiring no additional external circuitry. Each DSPLL contains a digitally controlled oscillator (DCO) for precise timing for IEEE 1588 (PTP) clock steering applications. The Si5383/84 requires both a crystal and a reference input. The TCXO/OCXO reference input determines the frequency accuracy in Free Run and stability in Holdover, while the crystal determines the output jitter performance. The TCXO/OCXO input supports all standard frequencies. Each DSPLL has access to IN0, IN1, and IN2, which are the three main inputs for synchronizing the DSPLLs. DSPLL D has access to two additional CMOS only inputs, IN3 and IN4. Each DSPLL can provide low jitter clocks on any of the device outputs. Based on 4th generation DSPLL technology, these devices provide any-frequency generation. Each DSPLL supports independent free-run and holdover modes of operation, and except for 1PPS inputs, offers automatic and hitless input clock switching. The Si5383/84 is programmable via a serial I2C interface with in-circuit programmable non-volatile memory so that it always powers up with a known configuration. Programming the Si5383/84 is made easy with Silicon Labs' ClockBuilder Pro software available at www.silabs.com/CBPro. Factory preprogrammed devices are available.

2.1 Work Flow Using ClockBuilder Pro and the Register Map

The purpose of this reference manual is to describe all the functions and features of the devices in the product family with register map details on how to implement them. Customers should use the ClockBuilder Pro software to provide the initial configuration for the device. Although the register map is documented, all the details of the algorithms to implement a valid frequency plan are fairly complex and are beyond the scope of this document. Real-time changes to the frequency plan and other operating settings are supported by the devices. However, describing all the possible changes is not a primary purpose of this document. Refer to Applications Notes and Knowledge Base article links within the ClockBuilder Pro GUI for information on how to implement the most common, real-time frequency plan changes.

The primary purpose of the software is to enable use of the device without an in-depth understanding of its complexities. The software abstracts the details from the user to allow focus on the high level input and output configuration, making it intuitive to understand and configure for the end application. The software walks the user through each step, with explanations about each configuration step in the process to explain the different options available. The software will restrict the user from entering an invalid combination of selections. The final configuration settings can be saved and written to an EVB, and a custom part number can be created for customers who prefer to order a factory preprogrammed device. The final register maps can be exported to text files, and comparisons can be done by viewing the settings in the register map described in this document.

The Si5383 offers three DSPLLs - A,C,D - and the Si5384 offers DSPLLD exclusively. The Reference Manual includes registers for all DSPLL's however DSPLLA and DSPLLC do not apply to the Si5384. The reference to "Standard Input Mode" applies to input frequencies between 8 kHz and 750 MHz whereas any reference to "1PPS Mode" applies to a 1 Hz input frequency.

2.2 Product Family

The table below lists a comparison of the various Si5383/84 family members.

Table 2.1. Product Selection Guide

Part Number	Max Frequency	Package Type	RoHS/Lead-Free	Temperature Range
Si5383A-Dxxxxx-GM	718.5 MHz	56-Lead 8x8 LGA	Yes	-40 to 85 °C
S5383B-Dxxxxx-GM	350 MHz	56-Lead 8x8 LGA	Yes	-40 to 85 °C
Si5384A-Dxxxxx-GM	718.5 MHz	56-Lead 8x8 LGA	Yes	-40 to 85 °C
Si5384B-Dxxxxx-GM	350 MHz	56-Lead 8x8 LGA	Yes	-40 to 85 ºC
Si5383-EVB		Evaluation Board		
SiOCXO1-EVB		OCXO Reference Clock Evaluation Board for Si5383-EVB (optional)		

3. Functional Description

The Si5383/84 takes advantage of Silicon Labs' fourth-generation DSPLL technology to offer the industry's most integrated and flexible jitter attenuating clock generator solution. The Si5383 offers three DSPLLs and the Si5384 offers one DSPLL. Each of the DSPLLs operate independently from each other and are controlled through a common serial interface. DSPLLs (A, C, and D) all have access to any of the three inputs (IN0 to IN2), after having been divided down by the input P dividers, which are either fractional or integer. DSPLL D has access to two additional CMOS inputs (IN3 and IN4). Clock selection can be either manual or automatic except for 1PPS inputs which must be controlled by manual clock selection. Any of the output clocks (OUT0 to OUT6) can be configured to connect to any of the DSPLLs using a flexible crosspoint connection, however 1PPS outputs can only be supplied by OUT5. Both a Crystal and a Reference must be installed for the device to operate.

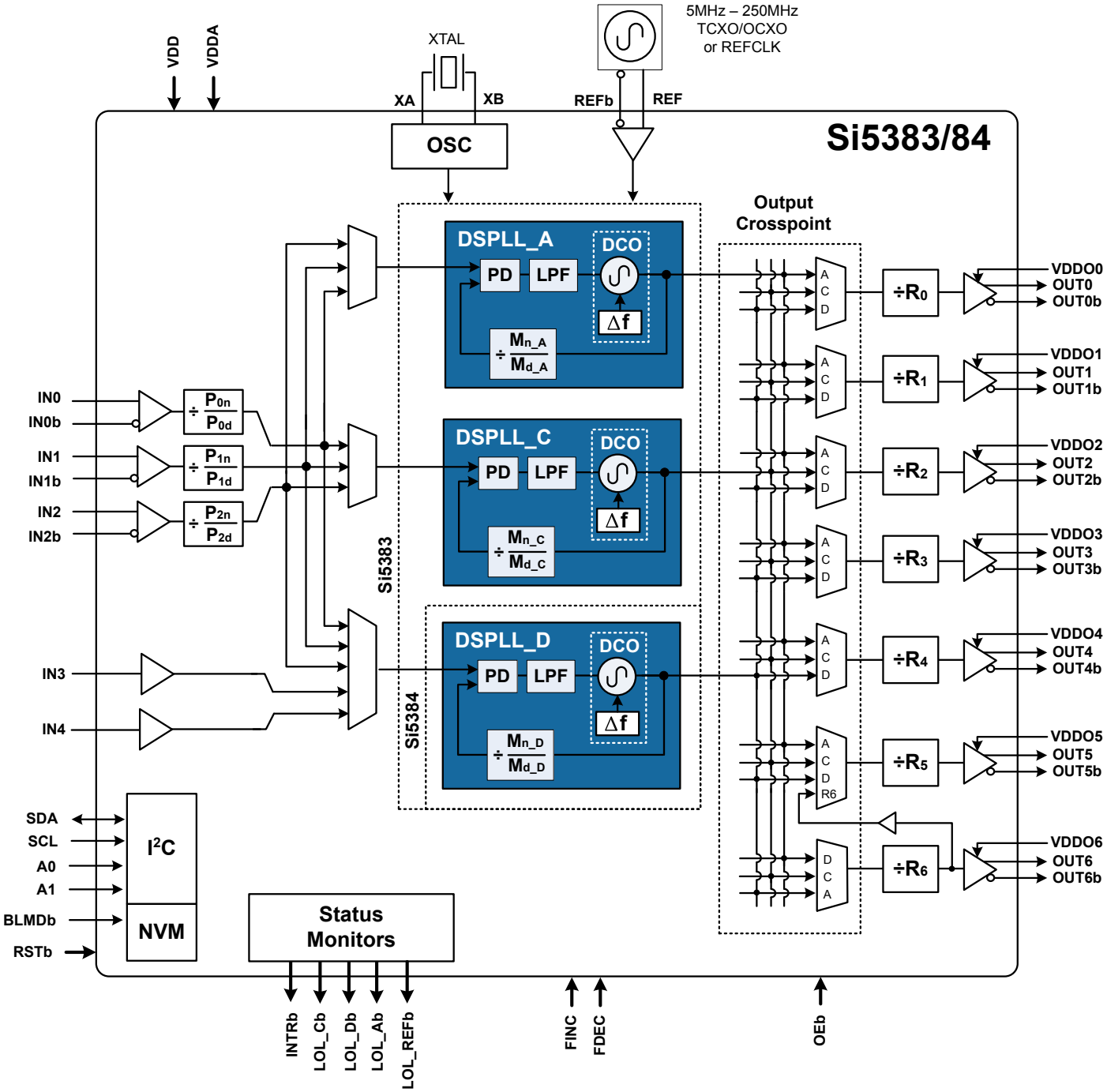


Figure 3.1. Block Diagram

3.1 DSPLL

The DSPLL is responsible for input frequency translation, jitter attenuation and wander filtering. Fractional input dividers (Pn/Pxd) allow the DSPLL to perform hitless switching between input clocks (IN0, IN1, IN2) when in standard input mode. Input switching is controlled manually in 1PPS mode and manuallly or automatically in standard input mode using an internal state machine. Automatic switching applies to any 4 inputs when in non 1PPS mode. The reference input determines the frequency accuracy while in free-run and stability while in holdover modes. The external crystal completes the internal oscillator circuit (OSC) which is used by the DSPLL for intrinsic low-jitter performance. A crosspoint switch connects any of the DSPLLs to any of the outputs. An additional integer divisor (R) determines the final output frequency.

The frequency configuration of the DSPLL is programmable through the I^2C serial interface and can also be stored in non-volatile memory. The combination of fractional input dividers (Pn/Pd), fractional frequency multiplication (Mn/Md), and integer output division (Rn) allows the generation of virtually any output frequency on any of the outputs. All divider values for a specific frequency plan are easily determined by using the ClockBuilder Pro software.

Because a jitter reference is required for all applications, either a crystal or an external clock source needs to be connected to the XAXB pins. See Chapter 10. Recommended Crystals and External Oscillators and Chapter 11. Crystal and Device Circuit Layout Recommendations for more information.

3.2 DSPLL Loop Bandwidth

The DSPLL loop bandwidth determines the amount of input clock wander and jitter attenuation. When in standard input mode register configurable DSPLL loop bandwidth settings from 1 mHz up to 4 kHz are available for selection for each of the DSPLLs and for the reference DSPLL (DSPLL B). Note that after changing the bandwidth parameters, the appropriate BW_UPDATE_PLLx bit (DSPLL A = 0x0414, REF B = 0x0514, DSPLL C = 0x0614, DSPLL D = 0x0715) must be set high to latch the new values into operation. SOFT_RST_PLLx will not update the BW registers so that BW_UPDATE_PLLx should typically be asserted when SOFT_RST_PLLx is asserted. Note each of these update bits will latch both loop and fastlock bandwidths.

When in 1PPS mode the loop BW selections are 1 mHz and 10 mHz. Since the loop bandwidth is controlled digitally, each of the DSPLLs will always remain stable with less than 0.1 dB of peaking regardless of the loop bandwidth selection. All changes to PPS BW settings should take place while PPS mode is disabled, register 0x5320[0].

The higher the PLL bandwidth is set relative to the phase detector frequency (Fpfd), the more chance that Fpfd will cause a spur in the Phase Noise plot of the output clock and increase the output jitter. To guarantee the best phase noise/jitter, it is recommended that the normal PLL bandwidth be kept less than Fpfd/160 although ratios of Fpfd/100 will typically work fine.

Table 3.1. DSPLL Loop Bandwidth Registers

Setting Name	Hex Address [Bit Field]	Function
	Si5383/84	
BW_PLLA[1]	0408[7:0] - 040D[7:0]	This group of registers determine the loop bandwidth for DSPLL A, C, D and B (reference). They are all independently selectable in the range from 1 mHz up to 4 kHz. Register values determined by ClockBuilderPro.
BW_PLLC[1]	0608[7:0] - 060D[7:0]	
BW_PLLD, (Standard Input Mode)	0709[7:0] - 070E[7:0]	
BW_PLLB	0508[7:0] - 070E[7:0]	
NL_NF, NL_NI, (BW, 1PPS Mode)	0x53D1[3:0], 0x53D2[4:0]	DSPLL D has 2 loop BW settings in 1PPS mode; 1 mHz and 10 mHz.

Note:
1. Si5383 only.

3.2.1 Fastlock

Selecting a low DSPLL loop bandwidth (e.g. 0.1 Hz) will generally lengthen the lock acquisition time. The fastlock feature allows setting a temporary Fastlock Loop Bandwidth that is used during the lock acquisition process. Higher fastlock loop bandwidth settings will enable the DSPLLs to lock faster. Fastlock Loop Bandwidth settings in the range from 100 Hz up to 4 kHz are available. Once lock acquisition has completed, the DSPLL's loop bandwidth will automatically revert to the DSPLL Loop Bandwidth setting. The fastlock feature can be enabled or disabled independently for each of the DSPLLs. If enabled, when LOL is asserted, Fastlock is enabled. When LOL is not asserted, Fastlock is disabled. Note that after changing the bandwidth parameters, the appropriate BW_UPDATE_PLLx bit (0x0414,0x0514, 0x0614, 0x0715) must be set high to latch the new values into operation. Each of these update bits will latch both loop and fastlock bandwidths. For 1PPS input applications, a Smartlock feaure is incorporated instead of fastlock.

Table 3.2. Fastlock Registers

Setting Name	Hex Address [Bit Field]	Function
	Si5383/84	
FASTLOCK_AUTO_EN_PLLA[1]	042B[0]	Auto Fastlock Enable/Disable. Manual Fastlock must be 0 for this bit to have effect. 0: Disable Auto Fastlock 1: Enable Auto Fastlock (default)
FASTLOCK_AUTO_EN_PLLC[1]	062B[0]	
FASTLOCK_AUTO_EN_PLLD	072C[0]	
FASTLOCK_AUTO_EN_PLLB	052B[0]	
FAST_BW_PLLA[1]	040E[7:0] - 0413[7:0]	Fastlock bandwidth is selectable in the range of 100 Hz up to 4 kHz. Register values determined using ClockBuilder Pro.
FAST_BW_PLLC[1]	060E[7:0] - 0613[7:0]	
FAST_BW_PLLD	070F[7:0] - 0714[7:0]	
FAST_BW_PLLB	050E[7:0] - 0513[7:0]	The reference fastlock bandwidth is selectable in the range of 1mHz to 4kHz
FASTLOCK_EXTEND_EN_PLL(A,B,C,D)	0x00E5[4:7]	Enables FASTLOCK_EXTEND
FASTLOCK_EXTEND_PLLA[1]	[0x00E9[4:0] 0x00E8[7:0] 0x00E7[7:0] 0x00E6[7:0]]	Set by CBPro to minimize phase transients when switching the PLL bandwidth
FASTLOCK_EXTEND_PLLB	[0x00ED[4:0] 0x00EC[7:0] 0x00EB[7:0] 0x00EA[7:0]]	
FASTLOCK_EXTEND_PLLC[1]	[0x00F1[4:0] 0x00F0[7:0] 0x00EF[7:0] 0x00EE[7:0]]	
FASTLOCK_EXTEND_PLLD	[0x00F5[4:0] 0x00F4[7:0] 0x00F3[7:0] 0x00F2[7:0]]	
FASTLOCK_EXTEND_SCL_PLLA[1]	0x0294[3:0]	Set by CBPro
FASTLOCK_EXTEND_SCL_PLLB	0x0294[7:4]	
FASTLOCK_EXTEND_SCL_PLLC[1]	0x0295[3:0]	
FASTLOCK_EXTEND_SCL_PLLD	0x0295[7:4]	
HOLDEXIT_BW_SEL0	0x059B[6]	Set by CBPro
HOLDEXIT_BW_SEL1	0x052C[4]	Set by CBPro
LOL_SLW_VALWIN_SELX_PLL(A,B,C,D)	0x0296[3:0]	Set by CBPro,
FASTLOCK_DLY_ONSW_PLLA[1]	0x02A6[19:0]	Set by CBPro
FASTLOCK_DLY_ONSW_PLLB	0x02A9[19:0]	
FASTLOCK_DLY_ONSW_PLLC[1]	0x02AC[19:0]	
FASTLOCK_DLY_ONSW_PLLD	0x02AF[19:0]	

Setting Name	Hex Address [Bit Field]	Function
	Si5383/84	
FASTLOCK_DLY_ON-LOL_EN_PLL(A,B,C,D)	0x0299[3:0]	Set by CBPro
FASTLOCK_DLY_ONLOLA[1]	0x029A[19:0]	Set by CBPro
FASTLOCK_DLY_ONLOLB	0x029D[19:0]	
FASTLOCK_DLY_ONLOLC[1]	0x02A0[19:0]	
FASTLOCK_DLY_ONLOLD	0x02A3[19:0]	

Note:
1. Si5383 only.

3.2.2 Smartlock Feature

When operating in 1PPS input mode, the Si5383/84 offers the Smartlock feature to achieve fast locking to 1PPS inputs. The Smartlock feature locks to 1PPS inputs in two phases. During the first phase, large adjustments are made to eliminate the majority of the frequency and phase error. During the second phase, finer adjustments are made until the PLL is locked. Once the PLL is locked, the DSPLLs loop bandwidth will automatically revert to the DSPLL loop bandwidth setting.

Table 3.3. Smartlock Registers

Setting Name	Hex Address [Bit Field]	Function
INIT_ACQ_TYPE	0x5320[4]	Initial Acquisition Lock B Disable - When set, acquisition doesn't wait for DSPLLB to lock before proceeding. Must be set before PPS_EN.
DCO_SCALE	0x5358[2:0]	DCO Scaling Factor -Used to keep DCO tuning range relatively constant over all frequency plans. Must be set before PPS_EN.
PP_CW_LMT	0x535C[7:0] - 0x535F[7:0]	Phase Pull Control Word Limit - - Maximum CW value to ensure DCO doesn't exceed maximum operating frequency (Fvco/10). Must be set before PPS_EN.
PD_ADJ	0x5360[7:0]-0x5363[7:0]	Phase Detector Adjustment - a 2's complement number used to modify DSPLLD outputs phase when in 1PPS mode. This is used to statically zero out or adjust the phase, such as compensating for fixed system delays. Must be set before PPS_EN.
PD_CW_2_ADJ	0x5364[7:0]-0x5367[7:0]	Phase Detector Control Word to Adjustment Conversion - Factor to convert the DCO Control Word to a PD adjustment. Must be set before PPS_EN.
SL_PER_2_ADJ	0x5368[7:0]-0x536B[7:0]	SmartLock Period to Adjustment Conversion - Factor to convert measured period difference to a DCO adjustment. Must be set before PPS_EN.
SL_PE_2_ADJ	0x536C[7:0]-0x536F[7:0]	SmartLock Phase Error to Adjustment Conversion - Factor to convert measured phase error to a DCO adjustment that achieves zero phase error in one second. Must be set before PPS_EN.
SLA_FA_CNT	0x5371[7:0]	SmartLock Frequency Average Count - The number of cycles to average the frequency difference before attempting frequency pull. Must be set before PPS_EN.
SLA_FP_NCYC	0x5372[2:0]	SmartLock Frequency Pull Cycles - The number of cycles (2^N) to complete frequency pull. Must be set before PPS_EN.
SLA_FP_VAL_CNT	0x5373[7:0]	SmartLock Frequency Pull Cycles - The number of consecutive cycles with frequency error below the threshold to complete frequency pull. Must be set before PPS_EN.

Setting Name	Hex Address [Bit Field]	Function
SLA_FE_THR	0x5374[6:0]-0x5377[7:0]	SmartLock Frequency Error Threshold - Threshold specified as the maximum difference in period between the reference and feedback clocks. Must be set before PPS_EN.
SLA_PPn_NCYC	0x5378[4:0]-0x537F[4:0]	SmartLock Phase Pull Cycles - The number of cycles (2^N) to complete phase pull. Must be set before PPS_EN.
SLA_PE_THR	0x5380[5:0]-0x5383[7:0]	SmartLock Frequency Error Threshold - Threshold specified as the maximum difference in phase between the reference and feedback clocks. Must be set before PPS_EN.
SLA_RL1_NF	0x5384[3:0]	SmartLock RapidLock NF, sets the loop BW. Must be set before PPS_EN.
SLA_RL1_NI	0x5385[4:0]	SmartLock RapidLock NI, sets the loop BW. Must be set before PPS_EN.
SLA_RL1_CNT	0x5386[7:0] - 0x5387[7:0]	SmartLock RapidLock Count - The number of cycles to remain in the RapidLock state. Must be set before PPS_EN.
SLA_RL2_NF	0x5388[3:0]	SmartLock RapidLock NF, sets the loop BW. Must be set before PPS_EN.
SLA_RL2_NI	0x5389[4:0]	SmartLock RapidLock, sets the loop BW. Must be set before PPS_EN.
SLA_RL2_CNT	0x538A[7:0] - 0x538B[7:0]	SmartLock RapidLock Count - The number of cycles to remain in the RapidLock state. Must be set before PPS_EN.
SLA_RL3_NF	0x538C[3:0]	SmartLock RapidLock NF, sets the loop BW. Must be set before PPS_EN.
SLA_RL3_NI	0x538D[4:0]	SmartLock RapidLock NI, sets the loop BW. Must be set before PPS_EN.
SLA_RL3_CNT	0x538E[7:0] - 0x538F[7:0]	SmartLock RapidLock Count - The number of cycles to remain in the RapidLock state. Must be set before PPS_EN.
SLA_RL4_NF	0x5390[3:0]	Smartlock RapidLock NF, sets the loop BW. Must be set before PPS_EN
SLA_RL4_NI	0x5391[4:0]	SmartLock RapidLock NI, sets the loop BW. Must be set before PPS_EN
SLA_RL4_CNT	0x5392[7:0] - 0x5393[7:0]	SmartLock RapidLock Count - he number of cycles to remain in RapidLock state. Must be set before PPS_EN.
SLB_FA_CNT	0x53A1[7:0]	SmartLock Frequency Average Count - The number of cycles to average the frequency difference before attempting frequency pull. Must be set before PPS_EN.
SLB_FP_NCYC	0x53A2[2:0]	SmartLock Frequency Pull Cycles - The number of cycles (2^N) to complete frequency pull. Must be set before PPS_EN.

Setting Name	Hex Address [Bit Field]	Function
SLB_FP_VAL_CNT	0x53A3[7:0]	SmartLock Frequency Validation Count. - The number of consecutive cycles with frequency error below the threshold to complete frequency pull. Must be set before PPS_EN.
SLB_FE_THR	0x53A4[6:0]-0x53A7[7:0]	SmartLock Frequency Error Threshold. Threshold specified as the maximum difference in period between the reference and feedback clocks. Must be set before PPS_EN.
SLB_PPn_NCYC	0x53A8[4:0]-0x53AF[4:0]	SmartLock Phase Pull Cycles - The number of cycles (2^N) to complete phase pull. Must be set before PPS_EN.
SLB_PE_THR	0x53B0[5:0]-0x53B3[7:0]	SmartLock Phase Error Threshold - Threshold specified as the maximum difference in phase between the reference and feedback clocks. Must be set before PPS_EN.
SLB_RL1_NF	0x53B4[3:0]	SmartLock RapidLock NF, sets the loop BW. Must be set before PPS_EN.
SLB_RL1_NI	0x53B5[4:0]	SmartLock RapidLock NI, sets the loop BW. Must be set before PPS_EN.
SLB_RL1_CNT	0x53B6[7:0] - 0x53B7[7:0]	SmartLock RapidLock Count - The number of cycles to remain in the RapidLock state. Must be set before PPS_EN.
SLB_RL2_NF	0x53B8[3:0]	SmartLock RapidLock NF, sets the loop BW. Must be set before PPS_EN.
SLB_RL2_NI	0x53B9[4:0]	SmartLock RapidLock NI, sets the loop BW. Must be set before PPS_EN.
SLB_RL2_CNT	0x53BA[7:0] - 0x53BB[7:0]	SmartLock RapidLock Count - The number of cycles to remain in the RapidLock state. Must be set before PPS_EN.
SLB_RL3_NF	0x53BC[3:0]	SmartLock RapidLock NF, sets the loop BW. Must be set before PPS_EN.
SLB_RL3_NI	0x53BD[4:0]	SmartLock RapidLock NI, sets the loop BW. Must be set before PPS_EN.
SLB_RL3_CNT	0x53BE[7:0] - 0x53BF[7:0]	SmartLock RapidLock Count - The number of cycles to remain in the RapidLock state. Must be set before PPS_EN.
SLB_RL4_NF	0x53C0[3:0]	SmartLock Rapid LockNF, sets the loop BW. Must be set before PPS_EN
SLB_RL4_NI	0x53C1[4:0]	SmartLock Rapid LockNF, sets the loop BW. Must be set before PPS_EN
SLB_RL4_CNT	0x53C2[7:0]-0x53C3[7:0]	SmartLock Rapid Count - The number of cycles to remain in RapidLock state. Must be set beore PPS_EN.
NL_NF	0x53D1[3:0]	NormalLock NF - Feed forward coefficient used in NormalLock mode. Must be set before PPS_EN.

Setting Name	Hex Address [Bit Field]	Function
NL_NI	0x53D2[4:0]	NormalLock NI - Integrating coefficient used in NormalLock mode. Must be set before PPS_EN.
HO_EXIT_EN	0x53E0[0]]	Holdover Acquisition Type - When enabled, HOLDOVER will automatically exit and attempt reacquisition when a valid input is detected. When disabled, FORCE_HOLD will be set on entry into holdover and must be manually cleared to exit the holdover state. Must be set before PPS_EN.
FORCE_HOLD	0x53E0[1]	Holdover Force - When asserted, the PPS loop will transition from the FREERUN or LOCKED states to the HOLDOVER state. It will remain in this state until the force is removed.
HO_ACQ_TYPE	0x53E0[5:4]	Holldover Acquisition Type - Determines the acquisition mode when holdover is exited. Must be set before PPS_EN.
HOLD_HIST_LEN	0x53E1[2:0]	Holdover History Length - Specifies the holdover window size. Larger windows provide more averaging. Must be set before PPS_EN.
HOLD_HIST_DELAY	0x53E2[4:0]	Holdover History Lengthy - Specifies the holdover delay time. Delay value allows ignoring corrupt frequency data before the input clock failure. Must be set before PPS_EN.

3.3 Dividers Overview

The frequency configuration for each of the DSPLLs is programmable through the I2C interface and can also be stored in non-volatile memory. The combination of fractional input dividers (Pn/Pd), fractional frequency multiplication (Mn/Md), and integer output division (Rn) allows each of the DSPLLs (A,C,D) to lock to any input frequency and generate virtually any output frequency. All divider values for a specific frequency plan are easily determined using the ClockBuilder Pro utility.

There are five main divider classes within the Si5383/84. See Chapter 3. Functional Description for a block diagram that shows them. Additionally, the DCO step word is used to scale the nominal output frequency in DCO mode. See Chapter 7. Digitally Controlled Oscillator (DCO) Mode for more information and block diagrams on DCO mode.

- PXAXB: XAXB Crystal / Reference input divider (0x0206)
  - XAXB Divide clock by 1, 2, 4, or 8 to obtain an internal clock <= 54 MHz
- P0-P3: Input clock wide range dividers (0x0208-0x022F)
  - Integer or Fractional divide values
  - Min. value is 1, Max. value is 2^{24} (Fractional-P divisors must be > 5)
  - 48-bit numerator, 32-bit denominator
  - Practical P divider range of (Fin/2 MHz) < P < (Fin/8 kHz)
  - Each P divider has a separate update bit for the new divider value to take effect
- MA-MD: DSPLL feedback dividers (0x0415-0x041F, 0x0515-0x051F, 0x0615-0x061F, 0x0716-0x0720))
  - Integer or Fractional divide values
  - Min. value is 1, Max. value is 2^{24}
  - 56-bit numerator, 32-bit denominator
  - Practical M divider range of (Fdco/2 MHz) < M < (Fdco/8 kHz)
  - Each M divider has a separate update bit for the new divider value to take effect
  - Soft reset will also update M divider values
- FSTEPW: DSPLL DCO step words (0x0423-0x0429, 0x0623-0x0629, 0x0724-0x072A)
  - Positive Integers, where FINC/FDEC select direction
  - Min. value is 0, Max. value is 2^{24}
  - 56-bit step size, relative to 32-bit M numerator
- R0-R6: Output dividers (0x0250-0x026A)
  - Even integer divide values: 2, 4, 6, etc.
  - Min. value is 2, Max. value is 2^{24}
  - 24-bit word where Value = 2 x (Word + 1), for example Word=3 gives an R value of 8

4. Modes of Operation

Once initialization is complete, each of the DSPLLs operates independently in one of four modes: Free-run mode, Lock Acquisition Mode, Locked Mode, or Holdover Mode. A state diagram showing the modes of operation is shown in the figure below. The following sections describe each of these modes in greater detail.

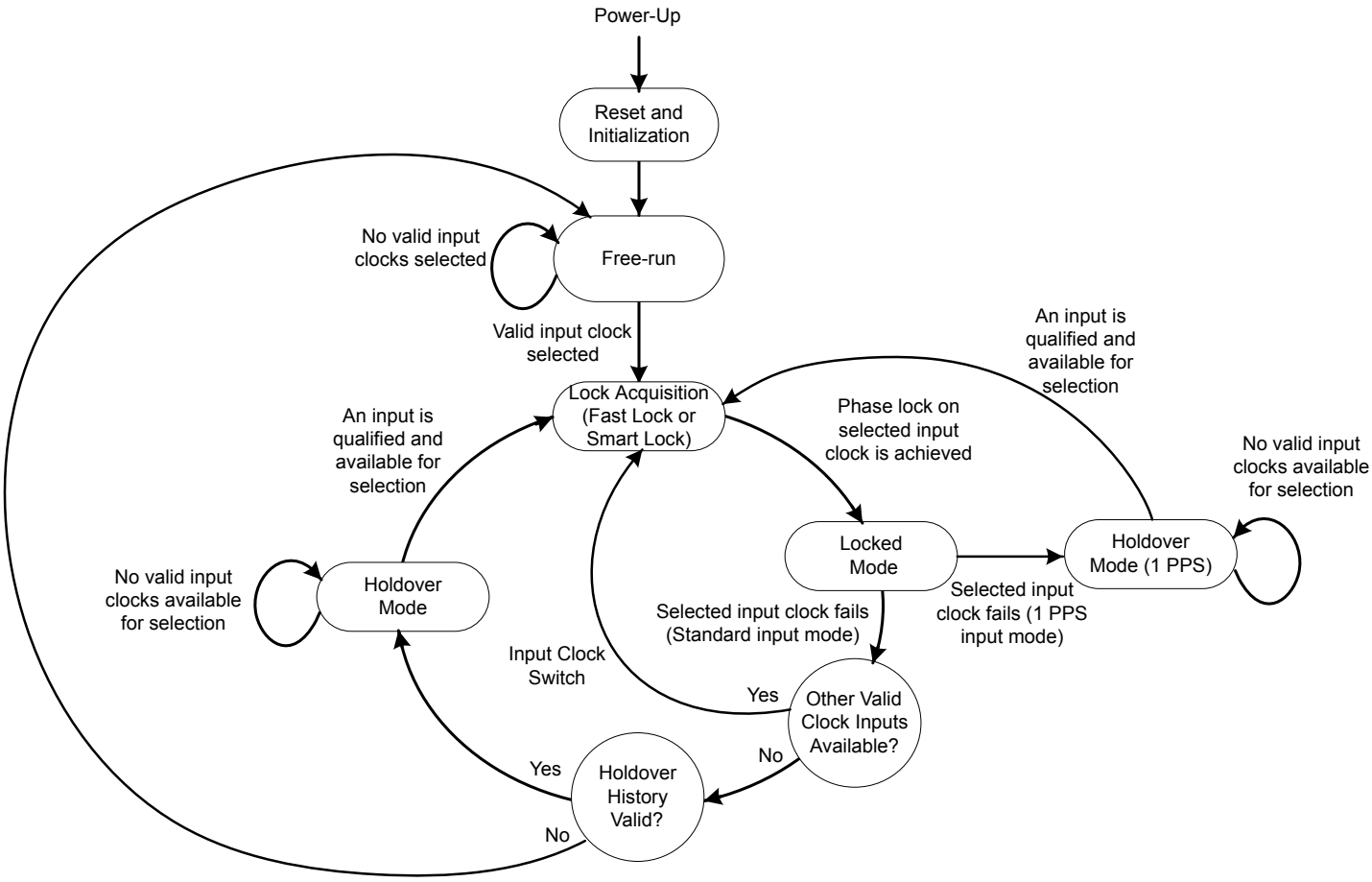


Figure 4.1. Modes of Operation

4.1 Reset and Initialization

Once power is applied, the device begins an initialization period where it downloads default register values and configuration data from NVM and performs other initialization tasks. Communicating with the device through the serial interface is possible once this initialization period is complete. No clocks will be generated until the initialization is complete. It is recommended that the device be held in reset on power-up by asserting the RSTb pin. RSTb should be released once all supplies have reached operational levels. Note, RSTb also functions as an open-drain output and drives low during POR. External devices must be configured as open-drain to avoid contention.

There are two types of resets available. A hard reset is functionally similar to a device power-up. All registers will be restored to the values stored in NVM, and all circuits will be restored to their initial state including the serial interface. A hard reset is initiated using the RSTb pin or by asserting the hard reset bit. A soft reset bypasses the NVM download. It is simply used to initiate register configuration changes. A hard reset affects all DSPLLs, while a soft reset can affect all or each DSPLL individually.

Table 4.1. Reset Control Registers

Setting Name	Hex Address [Bit Field]	Function
	Si5383/84	
HARD_RST	5303[0]	Performs the same function as power cycling the device. All registers will be restored to their default values.
SOFT_RST_ALL	001C[0]	Resets the device without re-downloading the register configuration from NVM.
SOFT_RST_PLLA[1]	001C[1]	Performs a soft reset on DSPLL A only.
SOFT_RST_PLLB	001C[2]	Performs a soft reset on DSPLL B, affecting all PLLs.
SOFT_RST_PLLC[1]	001C[3]	Performs a soft reset on DSPLL C only.
SOFT_RST_PLLD	001C[4]	Performs a soft reset on DSPLL D only.

Note:
1. Si5383 only.

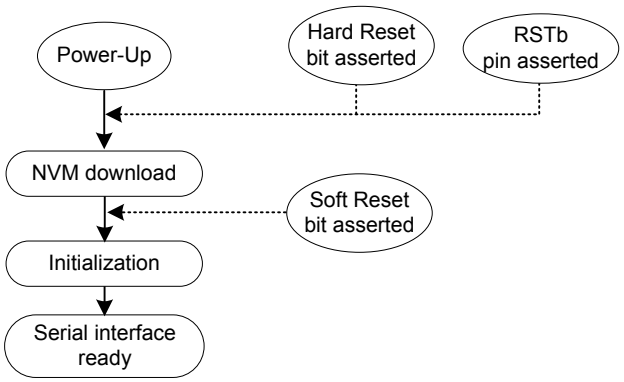


Figure 4.2. Initialization from Hard Reset and Soft Reset

The Si5383/84 is fully configurable using the serial interface (I^2C). At power-up the device downloads its default register values from internal non-volatile memory (NVM). Application specific default configurations can be written into flash memory allowing the device to generate specific clock frequencies at power-up. Writing default values to the flash memory is in-circuit programmable with normal operating power supply voltages applied to its VDD (1.8 V) and VDDA (3.3 V) pins. VDDOx supply is not required to write the flash memory.

4.2 Changing Registers while Device in Operation

ClockBuilder Pro generates all necessary control register writes for the entire device, including the ones described below. This is the case for both “Export” generated files as well as when using the GUI. This is sufficient to cover most applications. However, in some applications it is desirable to modify only certain sections of the device while maintaining unaffected clocks on the remaining outputs. If this is the case, please contact Silicon Labs Technical Support for further information: http://www.silabs.com/support/Pages/default.aspx .

If certain registers are changed while the device is in operation, it is possible for the PLL to become unresponsive (i.e. lose lock indefinitely). The following are the affected registers:

Table 4.2. Registers Affecting PLL Lock Status

Control	Register(s)
XAXB_FREQ_OFFSET	0x0202 – 0x0205
PXAXB	0x0206[1:0]
MXAXB_NUM	0x0235 – 0x023A
MXAXB_DEN	0x023B – 0x023E

The issue can easily be avoided by using the following preamble and post-amble write sequence below when one of these registers is modified or large frequency steps are made. ClockBuilder Pro software adds these writes to the output file by default when Exporting Register Files. Preamble and post-amble writes should be included when writing upon initilization, power-up, hard reset and RSTb.

1. To start, write the preamble by updating the following Write sequences:

Table 4.3. Preamble Sequence

Register	Value
0x0B24	0xC0
0x0B25	0x04
0x0540	0x01

2. Disable 1PPS mode, if used, by writing a 0 to register 5320[0].
3. Wait 300 ms.
4. Then modify all desired control registers.
5. Write 0x01 to Register 0x001C (SOFT_RST_ALL) to perform a Soft Reset once modifications are complete.
6. Write the post-amble by updating the following Write sequences:

Table 4.4. Postamble Sequence

Register	Value
0x0540	0x00
0x0B24	0xC3
0x0B25	0x06

7. Enable 1PPS mode, if used, by writing a 1 to register 5320[1]

Note, however, that this procedure affects all DSPLLs and outputs on the device. For assistance in changing only certain portions of the device without affecting the other outputs while the device is operating, please contact Silicon Labs technical support using the link on the last page of this document.

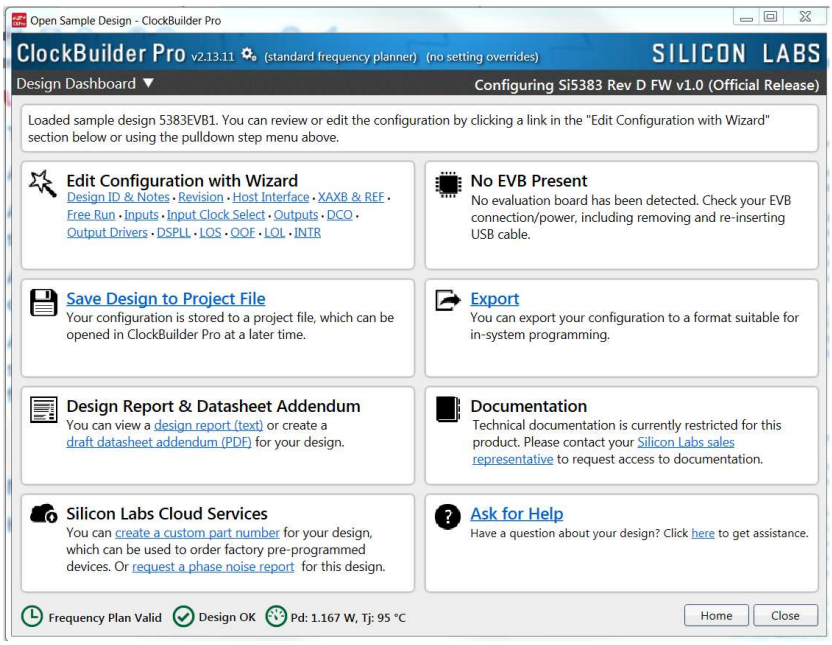
4.3 Flash Update/Programming

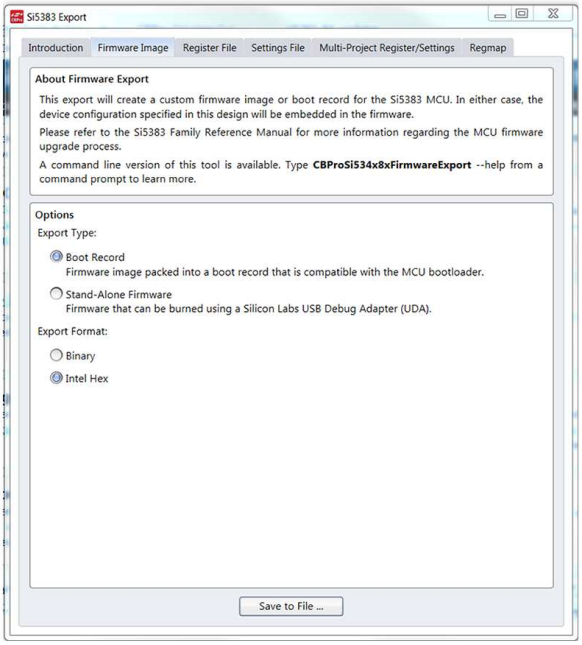
CBPro software is used to generate desired project files which include input/output frequency selection, output logic formats, loop BW values and a variety of associated PLL controls and alarm settings. The CBPro project file and Si5383 EVB can be tested to verify the Si5383/84 works as intended in the application.

Once complete, there are a variety of files which can be saved using the Export command and includes the Firmware Image, Register File, Settings File, Multi-Project Register Settings and RegMap. The Firmware Image should be saved in Boot Record or Stand Alone mode, the most common method would be Boot Record, and in either Intel hex or Binary format. From this, the hex or binary file is loaded into the Si5383/slave to update a frequency plan and firmware update, by using an I2C master. The I2C master is either SiLabs supported - by using the Si5383 EVB or Field Programmer - or by the users I2C master.

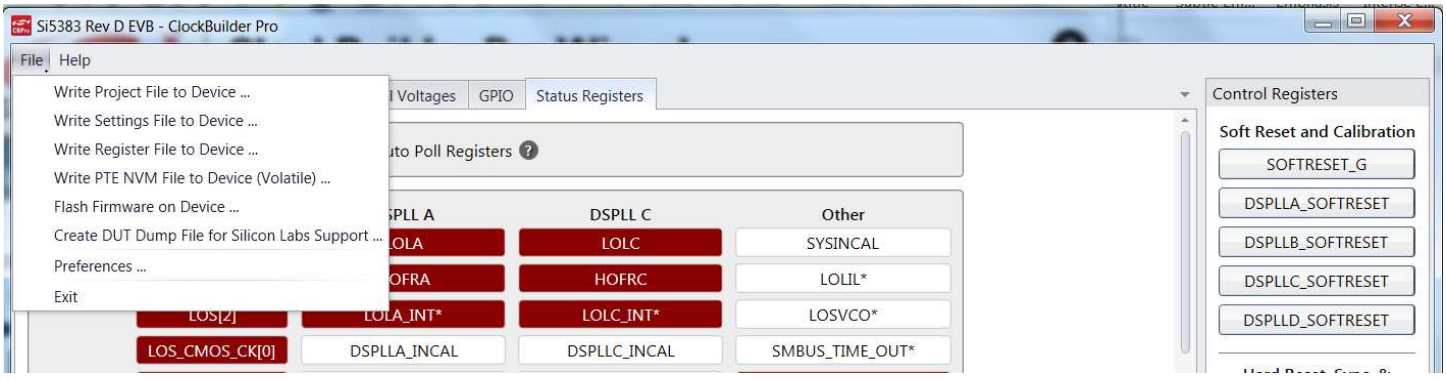
4.3.1 Upgrading Flash Firmware Image using Silabs Tools

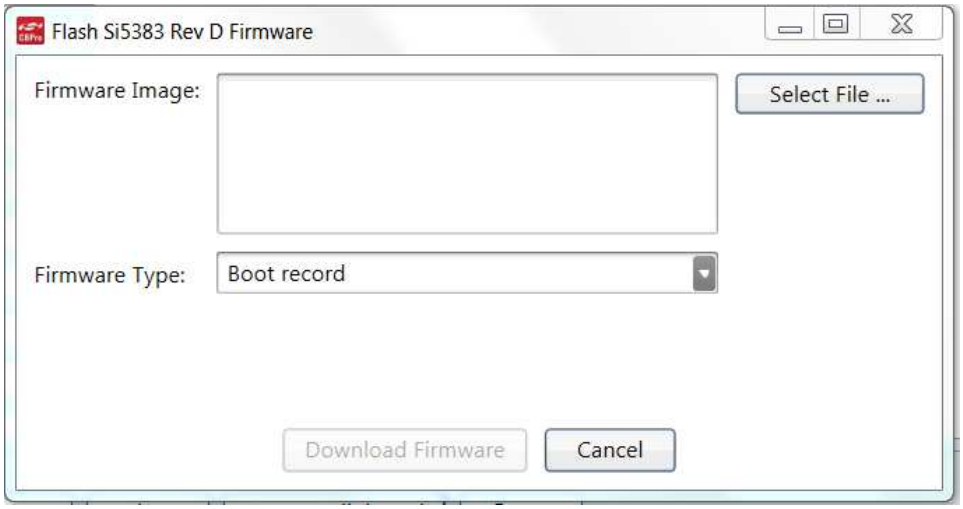
A Firmware Image can be written to an Si5383, on an Si5383 EVB, or by using a CBPROG-Dongle and Si538x4x-56SKT-DK / socket board hardware *(ClockBuilder Pro Field Programmer)*with CBPro software or CLI commands. Once a frequency plan has been built and verified, the required Firmware Image should be saved by selecting “Export” then “Firmware Image.” Select “Boot Record” or “Stand-alone,” and then either “Intel Hex” or “Binary” format. Additionally, Register File, Settings File, RegMap and MultiProject Register Settings, if applicable, can be saved in the Si5383 Export GUIs, as well as saving the Project File and design report (text) in the ClockBuilderPro GUI .





Once the Firmware Image has been saved it can be downloaded to devices by either using CBPro or CLI commands. If using CBPro, open the EVB GUI and select “File “and “Flash Firmware on Device” and locate where the file is located, plus whether it’s a bootloader or a standalone file, then select “Download Firmware”. The Si5383 will now be loaded with a new Firmware Image.





CBpro includes a series of command line interface tools, CLI, which are thoroughly detailed in “CBPro Tools & Support for In-System Programming” and “CBPro CLI User’s Guide”. These are normally downloaded along with CBPro and should be located in `C:\Program Files (x86)\Silicon Laboratories\ClockBuilder Pro\Docs`. A CLI can also be used to flash the Firmware Image, which includes the project plan, and firmware revision updates if applicable.

The CLI tool used to flash the Firmware Image is the “CBProSi534x8xFirmwareExport”. The command line can be located by typing “command prompt” in the window's search tool, and a base directory can be selected by typing `cd C:\` . Once a frequency plan has been built and verified, the required Firmware Image should be saved by selecting the “Export”, “Firmware Image,” and then, in most cases, “Boot Record” – which includes either Intel Hex or Binary format. Additionally, Register File, Settings File, RegMap, and Multi-Project Register Settings, if applicable, can be saved for bookkeeping.

The Firmware Image is then downloaded using a command write, as an example `CBProSi53488xFirmwareDownload.exe --bootrecord-file` *`si5383file.hex`* (this is the hex or bin file saved). Below is an example of a successful download.

```
C:\>CBProSi534x8xFirmwareDownload.exe --bootrecord-file si5383_April11-Boot-Reco
rd.hex
Firmware file is si5383_April11-Boot-Record.hex
Parsing firmware file ...
Trying to send device to bootloader mode ...
Success
Flashing firmware ...
0% complete
5% complete
10% complete
15% complete
20% complete
25% complete
30% complete
35% complete
40% complete
45% complete
50% complete
55% complete
60% complete
65% complete
70% complete
75% complete
80% complete
85% complete
90% complete
95% complete
100% complete
Verifying flash ...
Firmware flash complete
Trying to connect to device
Detected Si5383 in program mode
FIRMWARE_TYPE is 0
FIRMWARE_MAJOR_REV is 1
FIRMWARE_MINOR_REV is 0
FIRMWARE_BUILD is 19
DESIGN_IDx is 5383EVB1
```

In this example, the hex file was copied into the `C:\ directory`, otherwise a pathname needs to be identified.

The full Firmware Export command is:

```
CBProSi534x8xFirmwareExport --bootrecord (or standalone)-file --format bin (or hex)
--project pathname --outfile pathname

The help command is: "CBProSi534x8xFirmwareExport -help"

Additional information on CBProSi534x8xFirmwareExport command options:

--format bin|hex = the file format: binary or Intel Hex.

--outfile pathname = the file to save the firmware to. If this file already exists, it will
be overwritten. You must specify this.

--project pathname = the CBPro project file. The configuration present in this design will
be embedded in the firmware.

--type bootrecord|standalone = the type of firmware image to create: boot record that can be used with
bootloader or stand-alone firmware image.

--version - print this program's version number and exit.
```

The CBProDONGLE and Si538x4x-56SKT-DK socket board can also be used to flash a Firmware Image. See the CBPro-DONGLE-UG for more information. *(ClockBuilder Pro Field Programmer).* The CBPro-DONGLE can be used to flash a Firmware Image on a users PCB design as well, if similar pin connections and accommodations are made to the Si5383/84 layout, as those on the Si538x4x-56SKT-DK. The CBProDONGLE, socket board, plus Si5383/4 devices also make it easier to try new DSPLLn modifications, such as design verification or prototyping.

4.3.2 Upgrading Firmware Image Using I²C Master Routine

Updating Firmware Image is a four-step process:

1. Create and verify a Project Plan, saves Firmware Image plans, hex and or bin files etc as previously described.
2. Put the device in bootloader mode, either by write commands or hardware settings (as described in 4.3.2.1 Place in Bootloader Mode).
3. Take the `.bin` or `.hex` file and break up the file into separate boot records, which are a list of arrays. Each boot record starts with a frame start byte (0x24) and then has a data length number, which will allow the boot records to be separated. This is explained in 4.3.2.1 Place in Bootloader Mode.
4. Write in each boot record separately over I²C .

   Note: Either set up ack polling or put in delays of 20 ms for each boot record. The 3rd from last boot record is the CRC check which may take up to 6 seconds to complete. This is explained in 4.3.2.1 Place in Bootloader Mode. After each boot record has been accepted by the bootloader a reply will be sent back. The bootloader reply response codes are described in 4.3.5 Bootloader Reply Response Codes. Details for step 2 -4 follows.

4.3.2.1 Place in Bootloader Mode

The first step to updating the Firmware Image is to put the Si5383/84 in bootloader mode. This is a two-step write process to register 0x05:

- 0x05, 0x57
- 0x05, 0xBA

This two-key sequence is used to avoid putting the Si5383/84 in bootloader mode erroneously upon other writes. After writing the register sequence, the Si5383/84 should be in bootloader mode.

The second method (the hardware method) for putting the device in bootloader mode is to do the following:

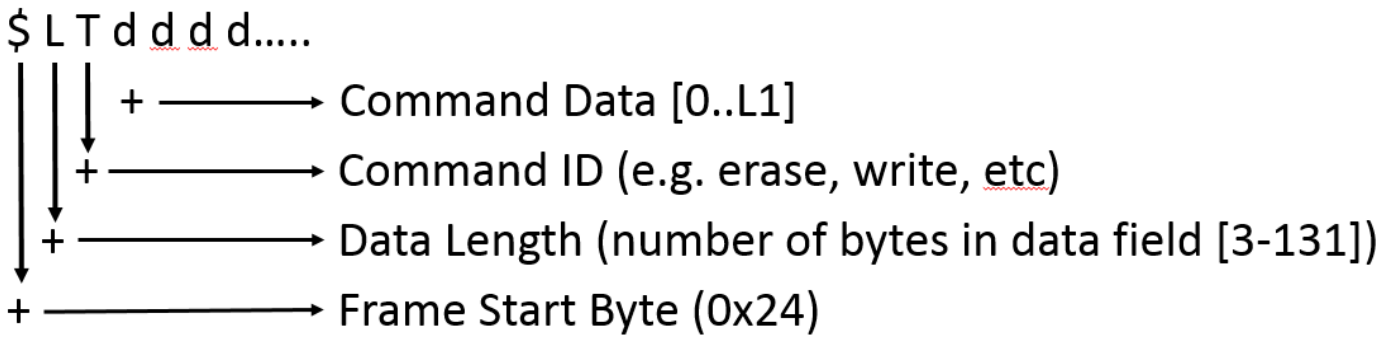
1. Set the BLMDb pin low and the RSTB pin low for greater than 15 µs.
2. Release the RSTb pin followed by the BLMDb pin.

4.3.3 Data Sequence to Write File

The hex file provided from CBPro should be parsed out into separate “records” that will be sent one record at a time from the I2C master to the Si5383/84. This format is similar to that used by the Intel Hex file. The most important thing is to make sure the correct I2C Address is used.

The I2C bootloader commands will be formatted into a binary record format. This simple and consistent format, which is inspired by hex records, is designed to make parsing easy. By including a frame start byte and an explicit length field, a file parser or communication transport code can be written without knowledge of the underlying commands. Also, the format allows command parameters to be added at a future date without impacting backward compatibility.

The following diagram shows the format for the binary boot record.



The communication protocol will work as follows:

1. The I2C host will send one complete record at a time to the Si5383/84 and then will wait for acknowledgment from the Si5383/84.
2. The Si5383/84 will process the record and then produce a one byte response that indicates if the command was successful or failed.

The starting indicator of the record is hex 0x24. This is the record start byte and must be the first byte sent into the IIC Write transaction each time. The next byte indicates how many bytes are to follow.

0x24- Start Indication

Example: 24/07/34/00/00/F4/FF/21/52

0x24: Start Indicator

0x07: Number of Bytes to Follow

34/00/00/F4/FF/21/52: data

4.3.4 ACK Polling or Delay Cycles

The Si5383/84 erase and verify commands can take several milliseconds to complete. The Si5383/84 will be unable to respond while these commands are processing. As a part of the expected protocol, the I2C master should query the Si8384/84 after each write sequence to ensure that the last write sequence was serviced without error. To let the I2C master know that the slave is busy processing the last command the Si5383/84 bootloader has an ACK polling mechanism in place. (This is the same mechanism I2C EEPROM devices use during flash write cycles.) The Si5383/84 bootloader will NAK its slave address while it is processing. The I2C master can poll the Si5383/84 bootloader by attempting to do a master read transfer. If the Si5383/84 slave address is NAK’d the I2C master knows the Si5383/84 bootloader is still busy with the last command. The master should continue to retry the transfer until the Si5383/84 slave address is ACK’d and the read transfer is completed. Then the master can proceed by writing the next boot record.

Otherwise a delay can be used between the write and read cycles. All write cycles typically take 20ms, with the exception of the 3rd last record command which may take up to 6 seconds to complete. This command is the verify sequence, which will check the entire program and compute overall success or failure with a CRC check.

4.3.5 Bootloader Reply Response Codes

The following are responses provided by the Si5383;

1. 0x40: Acknowledged.
2. 0x41: Data Range Error. This error response would indicate that the bootloader sees the targeted address range cannot be written by the bootloader.
3. 0x43: CRC Error if the CRC does not match the expected.

4.4 DSPLL Modes of Operation

4.4.1 Free Run Mode

Once power is applied to the Si5383/84 and initialization is complete, all three DSPLLs will automatically enter freerun mode, generating the frequencies determined by the NVM. The frequency accuracy and stability of the generated output clocks in freerun mode is entirely dependent on the reference clock (REF/REFb), while the external crystal at the XA/XB pins determines the jitter performance of the output clocks. For example, if the reference frequency is ±10 ppm, then all the output clocks will be generated at their configured frequency ±10ppm in freerun mode. Any drift of the reference frequency will be tracked at the output clock frequencies in this mode.

4.4.2 Lock Acquisition Mode

Each of the DSPLLs independently monitors its configured inputs for a valid clock. If at least one valid clock is available for synchronization, a DSPLL will automatically start the lock acquisition process.If the fast lock feature is enabled for inputs $\geq$ 8 kHz, a DSPLL will acquire lock using the Fastlock Loop Bandwidth setting and then transition to the DSPLL Loop Bandwidth setting when lock acquisition is complete. If the input frequency is configured for 1 PPS, the Smartlock mode is used. During lock acquisition the outputs will generate a clock that follows the VCO frequency change as it pulls-in to the input clock frequency.

4.4.3 Locked Mode

Once locked, a DSPLL will generate output clocks that are both frequency and phase locked to their selected input clocks. At this point any XTAL or Ref/Refb frequency drift will not affect the output frequency. Each DSPLL has its own LOL pin and status bit to indicate when lock is achieved.

4.4.4 Holdover Mode

Any of the DSPLLs will automatically enter holdover mode when the selected input clock becomes invalid and no other valid input clocks are available for selection. Each DSPLL calculates a historical average of the input frequency while in locked mode to minimize the initial frequency offset when entering the holdover mode. The averaging circuit for each DSPLL stores several seconds of historical frequency data while locked to a valid clock input. The final averaged holdover frequency value is calculated from a programmable window within the stored historical frequency data. Both the window size and the delay are programmable as shown in the figure below and should be modified to match the application requirements. The window size determines the amount of holdover frequency averaging. The delay value allows ignoring frequency data that may be corrupt just before the input clock failure. Each DSPLL computes its own holdover frequency average to maintain complete holdover independence between DSPLLs.

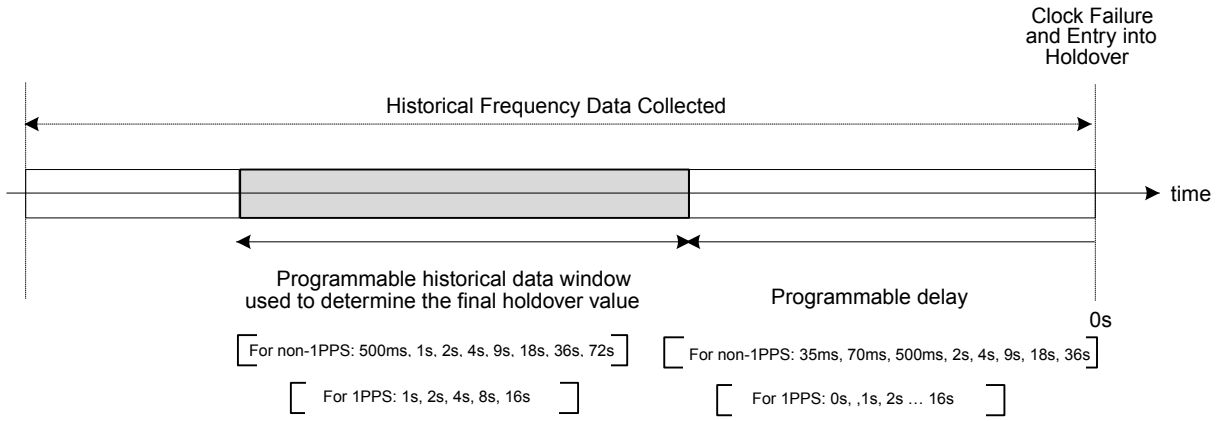


Figure 4.3. Programmable Holdover Window

When entering holdover, a DSPLL will pull its output clock frequency to the calculated averaged holdover frequency. While in holdover, the output frequency drift is entirely dependent on the external reference clock connected to the REF/REFb pins. If the clock input becomes valid, a DSPLL will automatically exit the holdover mode and re-acquire lock to the new input clock. This process involves pulling the output clock frequencies to achieve frequency and phase lock with the input clock. This pull-in process is glitchless. Note that because DSPLL B will always have a TCXO/OCXO as its clock input, under normal operation DSPLLB will never enter holdover. The holdover register bits below are listed for completeness.

The recommended mode of exit from holdover for non 1PPS applications is a ramp in frequency. Just before the exit begins, the frequency difference between the output frequency while in holdover and the desired, new output frequency is measured. It is quite possible (even likely) that the new output clock frequency will not be the same as the holdover output frequency because the new input clock frequency might have changed and the holdover history circuit may have changed the holdover output frequency. The ramp logic calculates the difference in frequency between the holdover frequency and the new, desired output frequency. Using the user selected ramp rate, the correct ramp time is calculated. The output ramp rate is then applied for the correct amount of time so that when the ramp ends, the output frequency will be the desired new frequency. Using the ramp, the transition between the two frequencies is smooth and linear. The ramp rate can be selected to be very slow (0.2 ppm/sec), very fast (40,000 ppm/sec) or any of ~40 values that are in between. The loop BW values do not limit or affect the ramp rate selections (and vice versa). CBPro defaults to ramped exit from holdover. Ramped exit from holdover is also used for ramped input clock switching for non 1PPS applications.

Table 4.5. DSPLL Holdover Control and Status Registers, Standard Input Mode

Setting Name	Hex Address [Bit Field]	Function
	Si5383/84	
Holdover Status		
HOLD_PLL(D,C,A)	000E[7:4]	Holdover status indicator. Indicates when a DSPLL is in holdover or free-run mode and is not synchronized to the input reference. The DSPLL goes into holdover only when the historical frequency data is valid, otherwise the DSPLL will be in free-run mode.