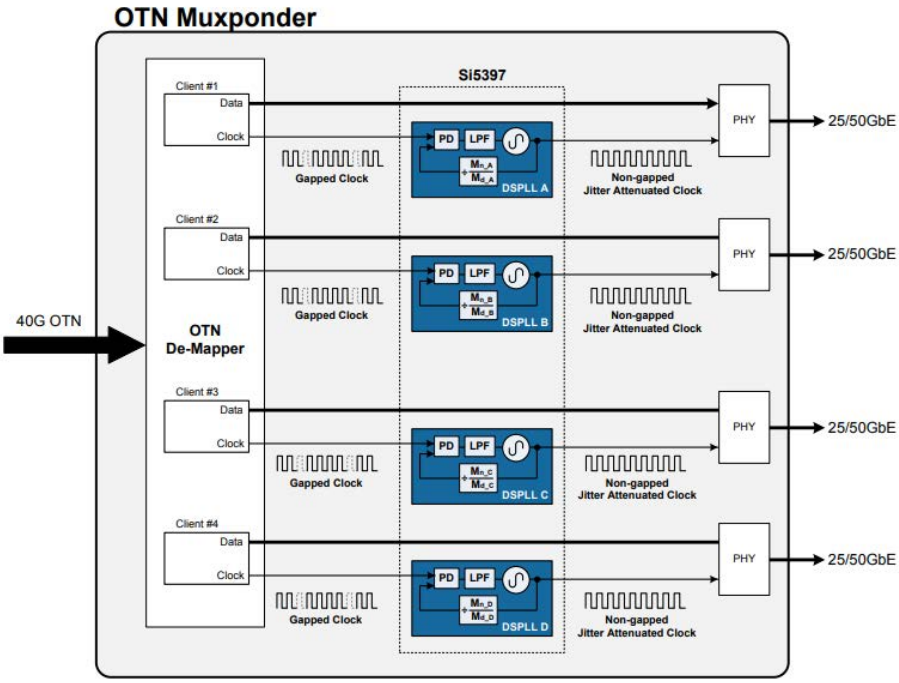
AN1178: Frequency-On-the-Fly for Silicon Labs Jitter Attenuators and Clock Generators

A key feature of select Silicon Labs' Jitter Attenuators and clock generators is the ability to reconfigure certain outputs/DSPLLs without disturbing other outputs/DSPLLs. For example, some OTN Muxponder cards (like the one shown below) need to be able to reconfigure different clients to different protocols with different data rates while keeping the rest of the clients uninterrupted.



This type of operation is called frequency-on-the-fly (FOTF). This application note describes how to use ClockBuilder Pro (CBPro) to automate and simplify that process.

KEY POINTS

- Update output frequency and other parameters on the fly
- No impact on other PLLs/MultiSynths
- Supported by easy to use tool kit
- This application note applies to
  - Single-DSPLL devices Si5395/94/92/91/80/45/44H/44/42H/42/41/40
  - Multi-DSPLL devices Si5397/96/84/83/48/47/46

1. Frequency-On-The-Fly (FOTF) Overview

The term “Frequency-on-the-fly” (FOTF) has different meanings depending on whether you are using a clock generator/single-PLL jitter attenuator or a multi-PLL jitter attenuator.

Clock generators and single-PLL jitter cleaners have only one DSPLL but independent MultiSynth dividers. Outputs on one MultiSynth can be reconfigured without affecting outputs on other MultiSynths.

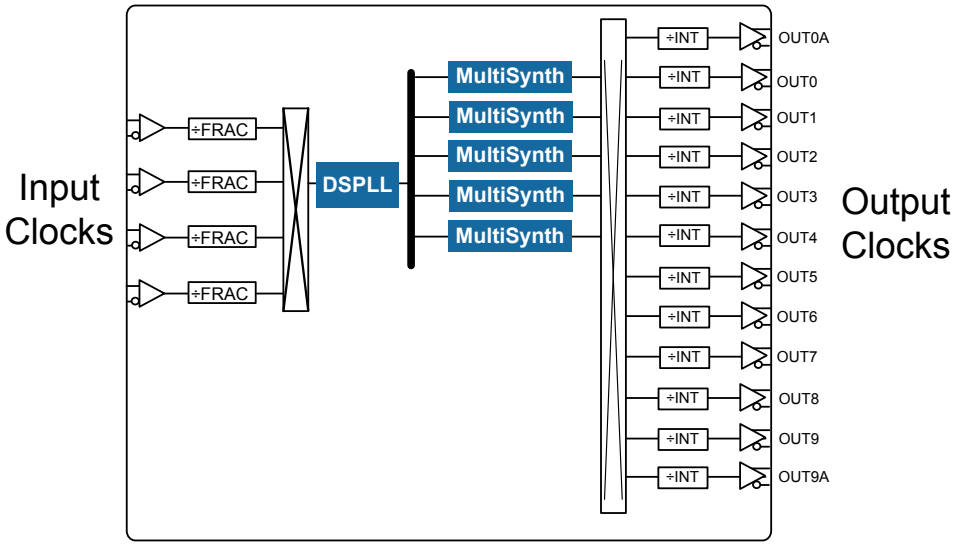


Figure 1.1. Single-PLL Jitter Cleaners

Table 1.1. FOTF Support for Single-DSPLL Topology

Product	Product Category	FOTF Features
Si5391/41/40	Clock generators	Output frequency on one MultiSynth can be reconfigured without affecting clocks on the other MultiSynths.
Si5395/94/92/80/45/44/42	Single-PLL Jitter Cleaner	
Si5344H/42H	Coherent Optical Clock	

For multiple-PLL jitter cleaners, each PLL has a completely isolated timing path. Both the input and output (and some other parameters) of each PLL can be changed without affecting the operation of the other PLLs.

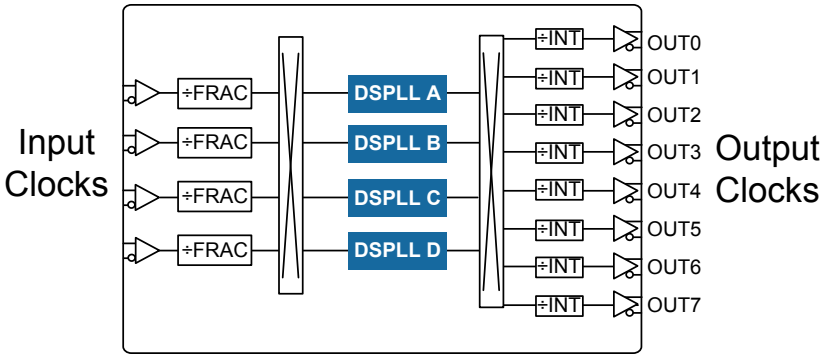


Figure 1.2. Multiple-PLL Jitter Cleaners

Table 1.2. FOTF Support for Multiple-DSPLL Topology

Product	Product Category	FOTF Features
Si5397/96/47/46	Multi-PLL Jitter Cleaner	The below features of each PLL can be changed without affecting the clocks on the other PLLs. • Input frequency • Output frequency • PLL bandwidth (normal, fastlock, holdover exit) • LOL threshold • OOF threshold
Si5384/83/48	Network Synchronizer Clock	

In the past, users who needed to configure frequency-on-the-fly would create multiple ClockBuilder Pro projects, export register files, and write the difference between different register files to the device in order to switch between plans. This process was very manual and error prone. Users also tended to miss important update registers and pre-/postambles and ended up achieving undesirable results.

To help customers configure FOTF easily and reliably, Silicon Labs has created a tool to automate the process. The CBProFOTF1.exe tool is one of several Command Line Interface (CLI) tools bundled with ClockBuilder Pro (CBPro). It is used to generate the register files that customers need to setup the device for FOTF and switch between plans.

To use the CBPro tools, ensure you have the latest version of CBPro installed, available from https://www.silabs.com/products/development-tools/software/clockbuilder-pro-software. The CLI tools are bundled in the CBPro installer and are automatically installed. To run the CBProFOTF1 CLI from a DOS prompt, you need to ensure the CLI folder is in your Windows PATH. This is normally done as part of the CBPro installation process. For installation instructions and tips, refer to the CBPro CLI User's Guide:

C:\Program Files (x86)\Silicon Laboratories\ClockBuilder Pro\Docs\CBPro CLI User Guide.pdf

Check the version of CLI tool you are using with the "--version" function. Make sure you are using the latest FOTF tool.

```
C:\>CBProFOTF1 --version
CBProFOTF1 2.29 [2018-11-04]
```

Figure 1.3. CBProFOTF1.exe Tool

The version number listed is the same as the CBPro version.

2. Workflow

The CBPro FOTF tool takes an existing base project file and performs the CLI calculations needed to generate the new register script.

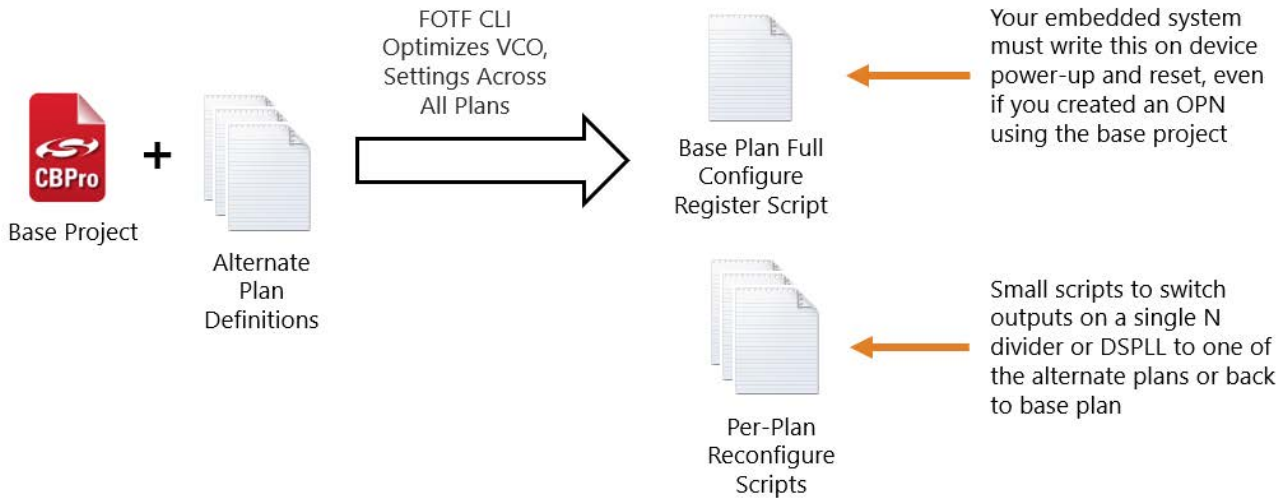


Figure 2.1. Frequency on the Fly Workflow

Some customers preferred to calculate the divider values themselves and write to the device while other customers used the output MUX to switch the output frequency among several multisynths to achieve their own versions of "FOTF". This application note does not cover such customized cases but rather focuses on how to use the CLI FOTF automated tool to obtain readily available register files that customers can use to write to the device.

2.1 Single-PLL FOTF

As mentioned previously, only the output frequencies can be changed on single-PLL devices for FOTF operation. For example, let's say you have a default frequency Plan #0 that gets downloaded to Si5345 after power up. Now say for a certain application you would like to change frequencies of OUT4–OUT9 (all from N2 divider) from 148.5 MHz to 167.33 MHz, then to 156.25 MHz, while also keeping other output frequencies undisturbed, as shown below.

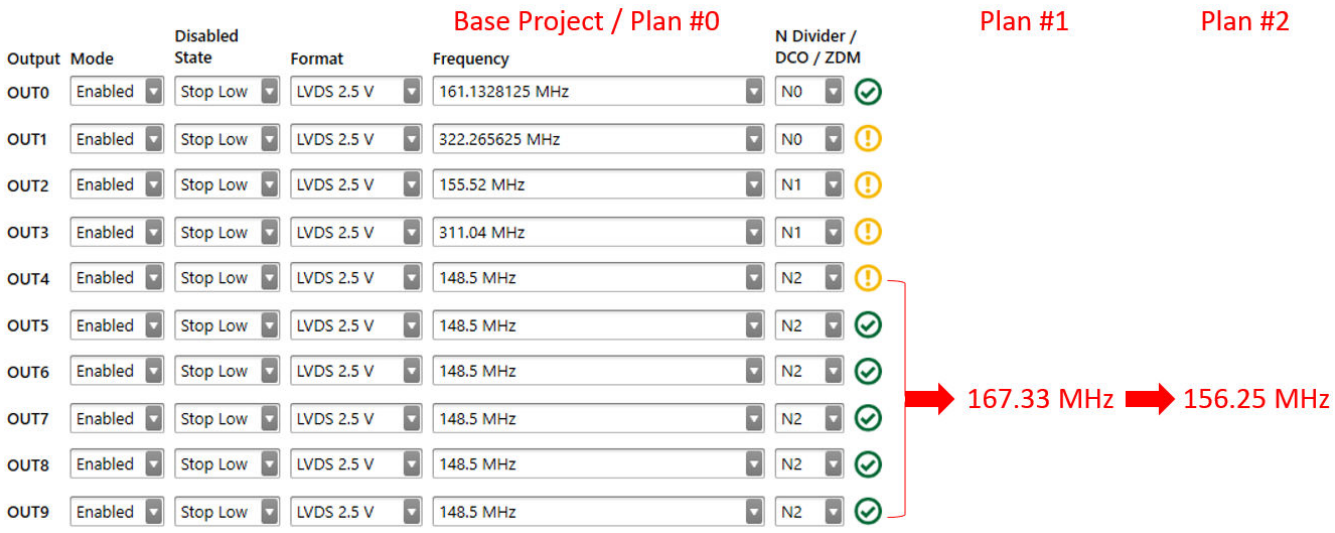


Figure 2.2. Single FOTF Diagram

To perform this task, you will need to modify the files shown in the figure below to reflect your configuration change.

Sample files can be found at C:\Program Files (x86)\Silicon Laboratories\ClockBuilder Pro\CLI\Samples\FOTF-For-Single-PLL-Device. You must copy the sample folder(s) to another folder on your hard drive. The folders will be read-only by default.

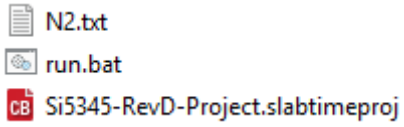


Figure 2.3. Single-PLL FOTF Files

Use the following steps to modify the files:

1. Define base configuration in CBPro and save the project (this is also referred to as Plan #0).
2. Define the changes you want to make in N2.txt.
3. Modify the batch script run.bat as needed and run from DOS prompt.

When run.bat is executed, it will create various register scripts and reports in the Output folder.

Each of these steps are described in detail in the following sections.

Base CBPro Project Plan

Define your base/default configuration in a CBPro project file.

Nx.txt

In this example, we are modifying the N2 divider. You will use this file to enter all the dynamic changes that you need.

Every N divider needs its own Nx.txt. Note that in the N2 Plan file that follows, output frequency is the only thing that is allowed to change on a single-DSPLL device.

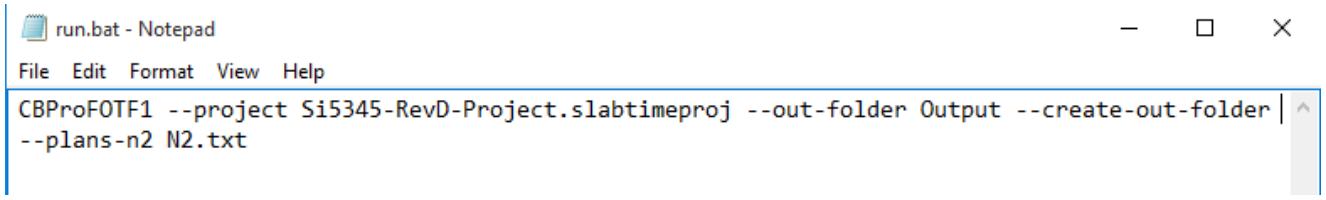
```
# N2 Plans

Plan,Item,Value

# Plan #1
1,OUT4,155.52*255/237M
1,OUT5,155.52*255/237M
1,OUT6,155.52*255/237M
1,OUT7,155.52*255/237M
1,OUT8,155.52*255/237M
1,OUT9,155.52*255/237M

# Plan #2
2,OUT4,156.25M
2,OUT5,156.25M
2,OUT6,156.25M
2,OUT7,156.25M
2,OUT8,156.25M
2,OUT9,156.25M
```

Notice that the plan number starts at #1 because the base plan is your plan #0.

Notice that Plan #1 changes OUT4–OUT9 to 167.33 MHz (equivalent of 155.52*255/237M), and Plan #2 changes OUT4–OUT9 to 156.25 MHz.

Note: For more specific questions about the syntax, refer to CLI User Guide.

Run.bat



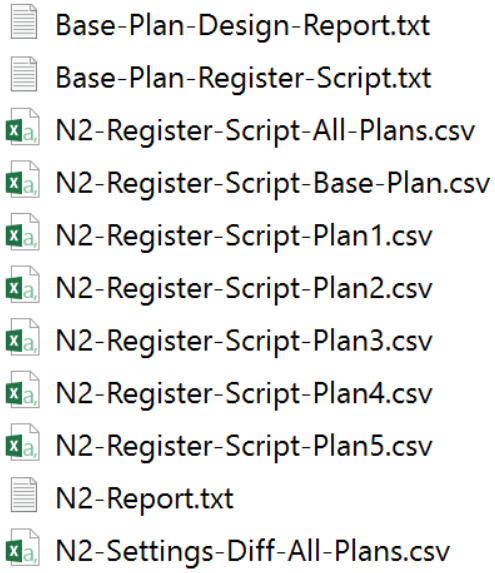
To modify and run the batch script, follow the syntax:

1. --project: put in the name of your CBPro project file.
2. --out-folder: put in the name of your output folder.
3. --create-out-folder: this will create a new folder if it does not already exist.
4. --plans-n2 N2.txt: define plans for the N2 MultiSynth. If doing FOTF on other MultiSynths, you must include these as using the same --plans-nx syntax..

You can add a PAUSE at the end of the batch file in order to see the error message:

```
C:\FOTF-For-Single-PLL-Device>CBProFOTF1 --project Si5345-RevD-Project.slabtimeproj --out-folder Output --create-out-folder
 --plans-n0 N0.txt --plans-n1 N1.txt --plans-n2 N2.txt
CBProFOTF1: N0 plan file 'N0.txt' does not exist
CBProFOTF1: exiting due to error

C:\FOTF-For-Single-PLL-Device>PAUSE
Press any key to continue . . .
```

If the script runs successfully, you end up with all the following files in the output folder:



Base-Plan-Register-Script.txt

Always write this file to the device first to get to base plan (Plan #0). Note that this register file might be different than the one exported from your CBPro base project alone because the FOTF algorithm takes into account all frequency changes/combinations and comes up with a common VCO frequency. Do not use the register script exported from your CBPro base project.

Nx-Register-Script-Planx.csv

Write this register file to switch plan. In our example, the N2-Register-Script-Plan1.csv and N2-Register-Script-Plan2.csv register scripts are created for Plan #1 and Plan #2.

Nx-Register-Script-Base-Plan.csv

Write this register file to switch back to the base plan (Plan #0) settings for Nx only. The difference between this file and the Base-Plan-Register-Script.txt is that this file only touches Nx and none of the other N dividers, whereas the Base-Plan-Register-Script.txt configures everything within the device.

Base-Plan-Design-Report.txt and Nx-Report.txt

Includes plan report files that have all the details on internal divider values and VCO frequencies. Do not write these files to the device.

Nx-Setting-Diff-All-Plans.csv

Plan differences are in one spreadsheet and provide information for debugging and review. Do not write this file to the device.

2.2 Multi-PLL FOTF

Multi-PLL FOTF allows customers to change many more parameters than just the output frequency. For example, a customer might like to change both the input frequency and the output frequency. At the same time, the customer needs to update bandwidth and LOL/OOF thresholds.

Sample files can be found at C:\Program Files (x86)\Silicon Laboratories\ClockBuilder Pro\CLI\Samples\FOTF-For-Multi-PLL-Device.

You must copy the sample folder(s) to another folder on your hard drive; they will be read-only by default.

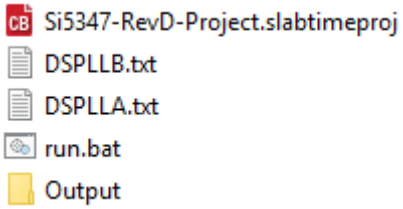


Figure 2.4. Multi-PLL FOTF Files

You will need to modify these files to reflect your configuration change. Use the following steps to modify the files:

1. Define base configuration in CBPro and save the project.
2. Define the changes you want to make in DSPLLx.txt.
    a. Input frequency
    b. Output frequency
    c. PLL Bandwidth
    d. LOL limits
    e. OOF limits

    **LOS is not supported due to its complexity*
3. Modify the batch script – run.bat as needed and run.

When run.bat is executed, it will create various register scripts and reports in the Output folder.

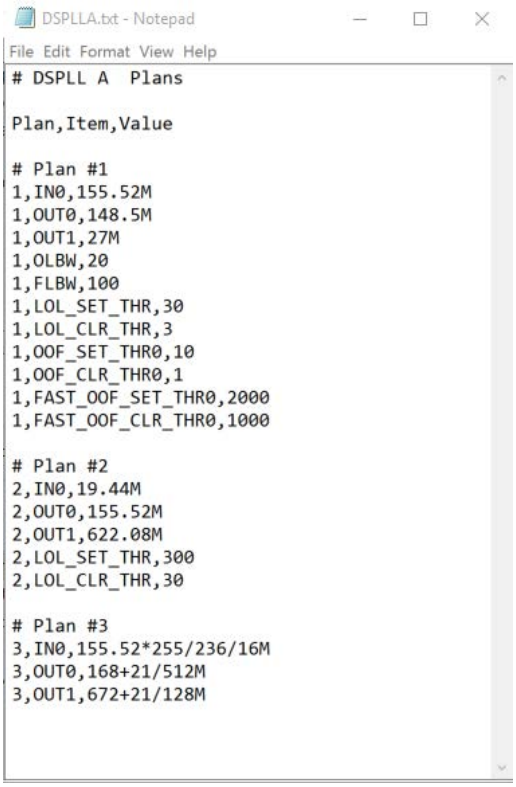
Each of these steps are described in detail in the following sections.

Base CBPro Project Plan

You need to define your base/default configuration in a CBPro project file.

DSPLLx.txt

You will use this file to enter all the dynamic changes that you need. The following figure lists all the changes that Plan #1 included: IN0, OUT0, OUT1, bandwidth, and LOL and OOF thresholds.



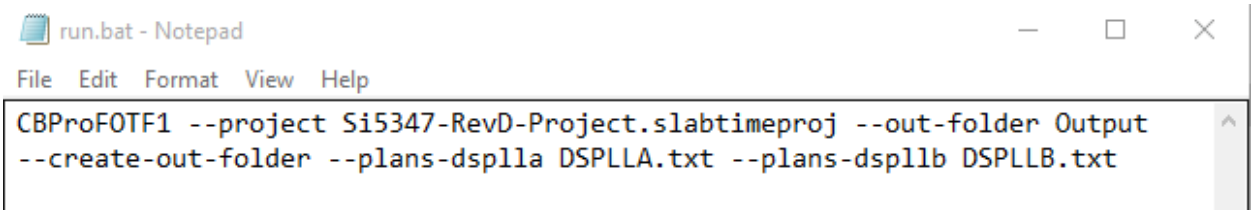
Note: DSPLLB.txt is not shown for simplicity.

Each DSPLL requires one dedicated text file. Within the file you can define multiple plans with different frequencies, such as bandwidth settings. If you don't specify a particular parameter, it will be inherited from the base project.

Notice that the plan number starts at #1 because the base plan is your plan #0.

Note: For more specific questions about the syntax, refer to CLI User Guide.

Run.bat



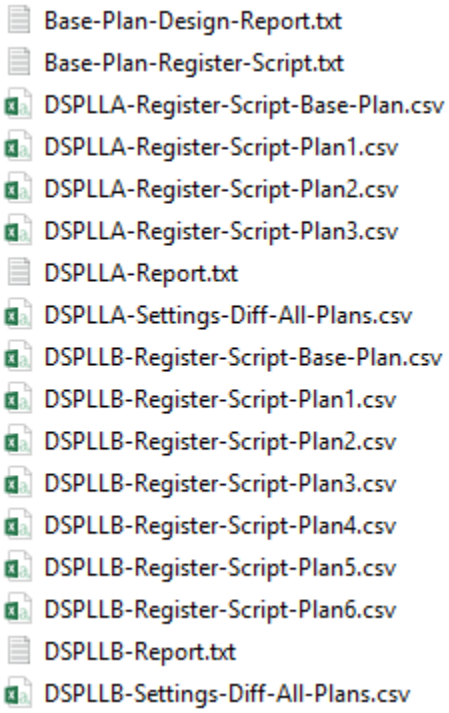
To modify and run the batch script, follow the syntax:

1. --project: put in the name of your CBPro project file.
2. --out-folder: put in the name of your output folder.
3. --create-out-folder: this will create a new folder if it does not already exist.
4. --plans-dsplla DSPLLA.txt --plans-dspllb DSPLLB.txt: define plans for DSPLL A and B. Use the --plans-dspllx syntax to define plans for all PLLs you plan to do FOTF on.

You can add a PAUSE at the end of the batch file in order to see the error message:

```
C:\FOTF-For-Multi-PLL-Device>CBProFOTF1.exe --project Si5347-RevD-Base.slabtimeproj --out-folder Output --create-out-fol
der --plans-dsplla DSPLLA.txt
Loading project Si5347-RevD-Base.slabtimeproj ...
CBProFOTF1: a fatal error occured while running the program
CBProFOTF1: there are errors in your DSPLLA plan file:
line 6: 1,IN0,8k: DSPLL A shares clock inputs with another DSPLL; IN0 can therefore not be modified
line 8: 1,OOF_SET_THR0,150: OOF_SET_THR0 cannot be changed because IN0 is assigned to more than one PLL (A,B)
line 11: 2,IN0,2M: DSPLL A shares clock inputs with another DSPLL; IN0 can therefore not be modified

C:\FOTF-For-Multi-PLL-Device>PAUSE
Press any key to continue . . .
```

If the script runs successfully, you end up with all the following files in the output folder:

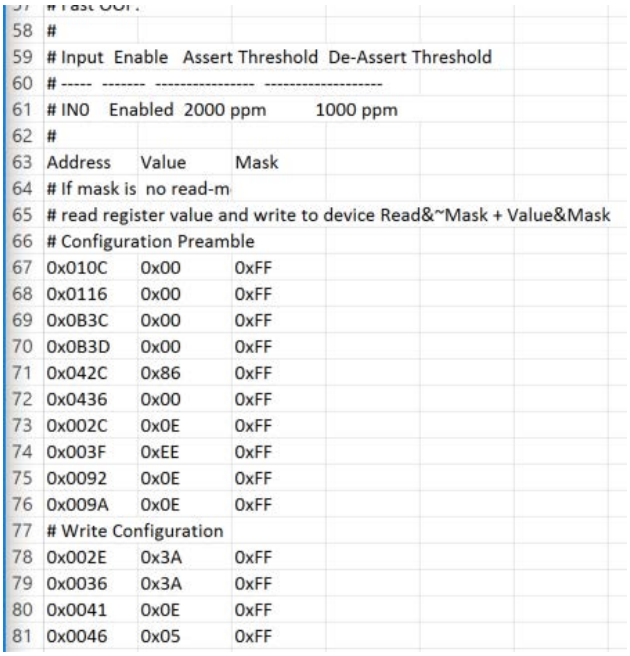


Base-Plan-Register-Script.txt

You must always write this file to the device first to get to base plan (Plan#0). Note that this register file might be different than the one exported from your CBPro base project alone, because the FOTF algorithm takes into account all frequency changes/combinations and comes up with a common VCO frequency. Do not use the register script exported from your CBPro base project.

DSPLLx-Register-Script-Planx.csv

Write this register file to switch plan:



58	#		
59	# Input	Enable	Assert Threshold De-Assert Threshold
60	# -----	-------	---------- ----------
61	# IN0	Enabled	2000 ppm 1000 ppm
62	#		
63	Address	Value	Mask
64	# If mask is	no read-m	
65	# read register value and write to device Read&~Mask + Value&Mask		
66	# Configuration Preamble		
67	0x010C	0x00	0xFF
68	0x0116	0x00	0xFF
69	0x0B3C	0x00	0xFF
70	0x0B3D	0x00	0xFF
71	0x042C	0x86	0xFF
72	0x0436	0x00	0xFF
73	0x002C	0x0E	0xFF
74	0x003F	0xEE	0xFF
75	0x0092	0x0E	0xFF
76	0x009A	0x0E	0xFF
77	# Write Configuration		
78	0x002E	0x3A	0xFF
79	0x0036	0x3A	0xFF
80	0x0041	0x0E	0xFF
81	0x0046	0x05	0xFF

The first and second columns are Address and Value. The third column is Mask. If mask is 0xFF, no read-modify-write is required. Otherwise, read register value first and then modify and write.

DSPLLx-Register-Script-Base-Plan.csv

Write this register file to switch back to the base plan (plan #0) settings for DSPLLx only. The difference between this file and the Base-Plan-Register-Script.txt is that this file only touches DSPLLx and none of the other DSPLLs, whereas the Base-Plan-Register-Script.txt configures everything within the device.

Base-Plan-Design-Report.txt and DSPLLx-Report.txt

Plan report files that have all the details on internal divider values and VCO frequencies. Do not write these files to the device.

DSPLLx-Setting-Diff-All-Plans.csv

Plan differences are in one spreadsheet and provide information for debugging and review. Do not write this file to the device.

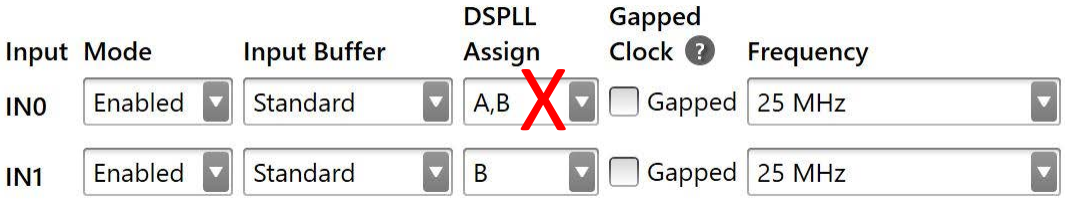
3. FAQs

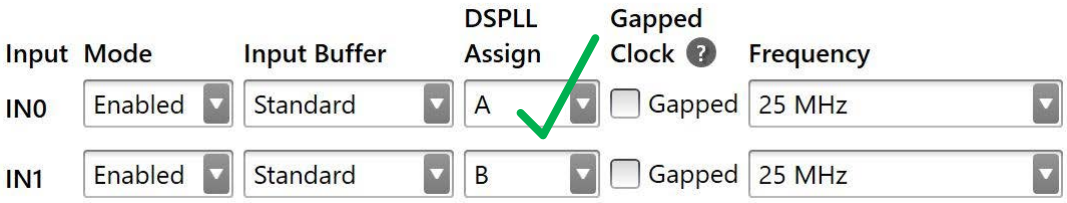
Can I do FOTF when one input is shared between multiple DSPLLs?

No. You will see an error message like the following:

```
CBProFOTF1: there are errors in your DSPLLA plan file:
line 6: 1,IN0,8k: DSPLL A shares clock inputs with another DSPLL; IN0 can therefore not be modified
```

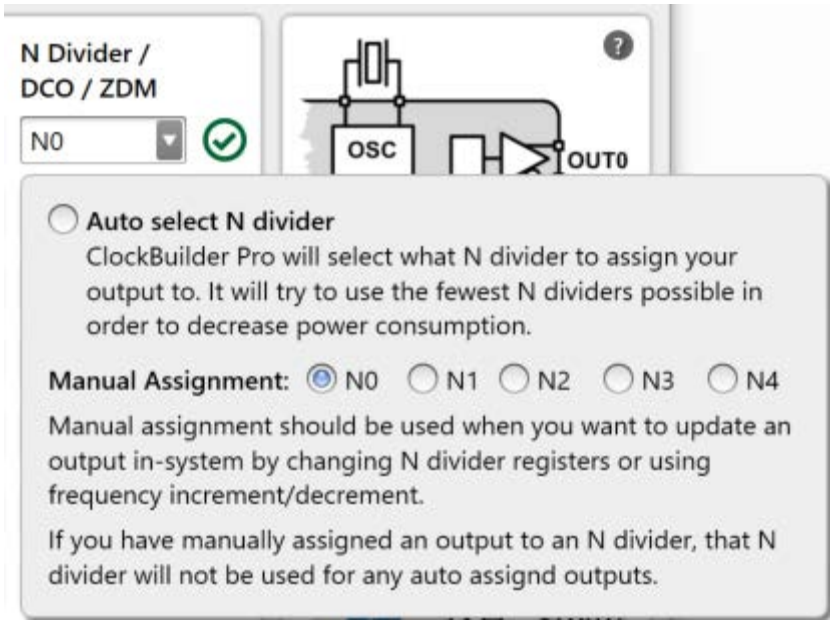
To solve the issue, you have to modify your base project to assign a dedicated input to each DSPLL.





What does the error message "All outputs must be manually assigned to an N divider to do FOTF" mean?

To do FOTF on single-PLL devices, you have to manually assign the N divider to each output in the base project.



What is the difference between FOTF and DCO?

A digitally controlled oscillator (DCO) is suitable for small frequency changes only. However, the output frequency change is glitchless.

FOTF covers not only big frequency changes, but also bandwidth and alarm threshold changes. The output clock of the targeted PLL is disabled temporarily during reconfiguration.

Do I need to write global pre- and postambles (0x0B24 and 0x0B25) when switching between plans?

No. The global pre- and postambles will disturb the operation of all the PLLs. The FOTF output files have all necessary pre- and postambles and resets included already.

I have an OPN (Orderable Part Number) that comes with my base plan preloaded. Do I need to write "Base-Plan-Register-Script.txt"?

Yes. You have to write the "Base-Plan-Register-Script.txt" because the FOTF tool takes all the frequency plans, not only the base plan, into account and calculates the optimal VCO frequency and output divider values. These values might be different from the ones in the NVM based on base plan only.

Can I update DSPLLB on the fly on Si5348?

No. DSPLLB on Si5348 provides the frequency reference for other PLLs, so you cannot update PLLB without affecting other PLLs. However, DSPLLA, C, and D are completely separate.

Can I update DSPLLs that are in free-run mode on Si5397/96?

Yes. The FOTF tool supports output frequency updates on DSPLLs that are free-running.

Can I calculate the N divider value change myself and write to my single-PLL device?

Yes. This is similar to the DCO function. Refer to application note, "AN858: DCO Applications with Jitter Attenuators Si5395/94/92/45/44/42 and AN959: DCO Applications with the Si5341/40 .

4. Conclusions

The CLI FOTF tool is a powerful tool that allows customers to change certain outputs/PLLs without affecting others. In this application note, we walked through examples of how to to use the tool for both a single-PLL device and a multi-PLL device.

4.1 Additional Resources

- CBPro CLI User's Guide – CBProFOTF1 help: From the DOS prompt, type **CBProFOTF1 --help** to view the FOTF user's guide. This guide is also available in the installer and includes overviews of other CLIs that may be helpful, including tools to write FOTF register files to your system board using the CBPro Field Programmer.
- CBPro Tools and Support for In-System Programming – Si534x/8x/9x Frequency-On-The-Fly Example Walkthroughs

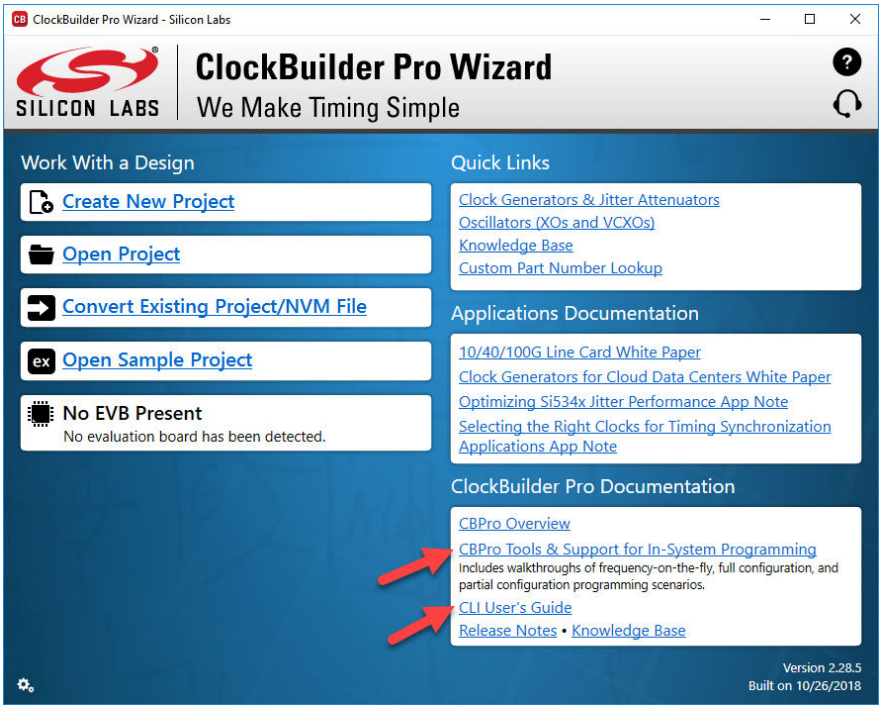


Figure 4.1. ClockBuilderPro Main Screen

ClockBuilder Pro

One-click access to Timing tools, documentation, software, source code libraries & more. Available for Windows and iOS (CBGo only).

www.silabs.com/CBPro

Timing Portfolio
www.silabs.com/timing

SW/HW
www.silabs.com/CBPro

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com