[Si5351 alone outputs quadrature signals of 3MHz or less](https://1-tj--lab-org.translate.goog/2020/08/27/si5351%e5%8d%98%e4%bd%93%e3%81%a73mhz%e4%bb%a5%e4%b8%8b%e3%81%ae%e7%9b%b4%e4%ba%a4%e4%bf%a1%e5%8f%b7%e3%82%92%e5%87%ba%e5%8a%9b%e3%81%99%e3%82%8b/?_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp&_x_tr_enc=1)

[August 27, 2020](https://1-tj--lab-org.translate.goog/2020/08/27/si5351%e5%8d%98%e4%bd%93%e3%81%a73mhz%e4%bb%a5%e4%b8%8b%e3%81%ae%e7%9b%b4%e4%ba%a4%e4%bf%a1%e5%8f%b7%e3%82%92%e5%87%ba%e5%8a%9b%e3%81%99%e3%82%8b/?_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp&_x_tr_enc=1)[uebo](https://1-tj--lab-org.translate.goog/author/tjlabuebo/?_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp&_x_tr_enc=1)

The Si5351A can output orthogonal signals using the Multisynth delay parameter, but because the parameter setting range is a maximum of 127, the range in which the orthogonal signal can be output is limited to approximately 3 MHz or higher. [(Orthogonal signal with Si5351)](https://1-tj--lab-org.translate.goog/2017/03/14/si5351%e3%81%a7%e7%9b%b4%e4%ba%a4%e4%bf%a1%e5%8f%b7/?_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp&_x_tr_enc=1)　Therefore, we decided to use a different method to obtain orthogonal signals below 3 MHz.

The configuration of the Si5351 is as shown in the figure, and it is common to fix the Mutisynth division ratios M0 and M1 at certain values ​​and control the frequency by changing the PLL feedback division ratio N.

A diagram of a computer program

Description automatically generated

The key point here is that M0 and M1 can also have fractional division ratios, so it is possible to finely control the frequency by using M0 and M1. However, I don't think smooth frequency control can be achieved by controlling the value of M, so I will use control of M only as a means to obtain a phase difference of π/2, and control the frequency by controlling the division ratio N of the PLL.

To obtain the desired phase difference by controlling M, M0 and M1 can be changed with a time lag.  
For example, set M0, M1 = m' and reset the PLL (this will align the phase to 0). If you then set  
M0 = m (leaving m<m', M1 = m' as is), then fI>fQ will occur, and  
a frequency difference of fd = fI – fQ will be generated.  
As a result, the phase of fI will lead relative to fQ over time.  
If the elapsed time is Td, the phase lead θd can be expressed by the following equation.

eq1

Therefore, Td for which θd = π/2 is



Therefore, if you set M1 = m after this time has elapsed, you will obtain a signal with a phase difference of π/2. This operation only needs to be done once. After that, if there is no change in m and the frequency is changed by the PLL, the phase difference will be maintained.

As an example, the procedure for obtaining a 2.0MHz orthogonal signal is shown below.  
The conditions are m = 300, fd = 4Hz, and the Multisynth mode is fractional.

1. Set N = 600MHz/25MHz = 24 so that fvco = 300\*2.0MHz = 600MHz
2. Set M0, M1 = 600MHz / ( 2MHz-4Hz ) = 300.0006  so that the output frequency f I, fQ = 2.0MHz – 4Hz.
3. PLL Reset
4. Set M0 = 300
5. Set M1 = 300 after Td = 1/(4fd) = 62.5ms has elapsed

This will generate a 2MHz orthogonal signal. From now on, if you want to change the frequency, for example to 1.99MHz, just change N and set N = 1.99MHz\*300 / 25MHz = 23.88. The  
important thing here is not to reset the PLL.  
fd can be any value, but a smaller value may result in better phase difference accuracy. However, Td will become larger, resulting in longer waiting times.

Below is an example of operation

Output frequency = 1.18MHz  


Output frequency = 588kHz  
A screen shot of a computer

Description automatically generated

It may not be a guaranteed behavior, but so far it doesn't seem to be doing anything strange.