Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 001-52133

Spec Title: AN52133 - FREQUENCY MARGINING USING FLEXO(TM) AND ITS APPLICATIONS

Replaced by: None

AN52133

Frequency Margining Using FleXO™ and Its Applications

Author: Amitava Banerjee

Associated Project: No

Associated Part Family: CY2XFnn/CY2X0xx

Software Version: CyClockWizard 1.0

Related Application Note: AN62914

Cypress FleXO™ devices provide a frequency margining feature that lets you change output frequency. This capability is useful in all stages of system design for troubleshooting, design optimization, and testing. AN52133 serves as a guide to this feature and includes examples of how it can be used.

1 Introduction

Most clock generators available to designers use a crystal oscillator to provide a fixed frequency clock output with little or no programmability. Cypress's FleXO clock generators include a frequency margining capability that allows you to change the preprogrammed output frequency with ease, either by using the frequency select (FS) pins on the device or through an I^2C serial interface. Frequency can be changed by as little as 100 ppm and as much as 100 percent. Thus, on a 100-MHz output frequency device, the output frequency can be changed from 100.01 MHz to 200 MHz.

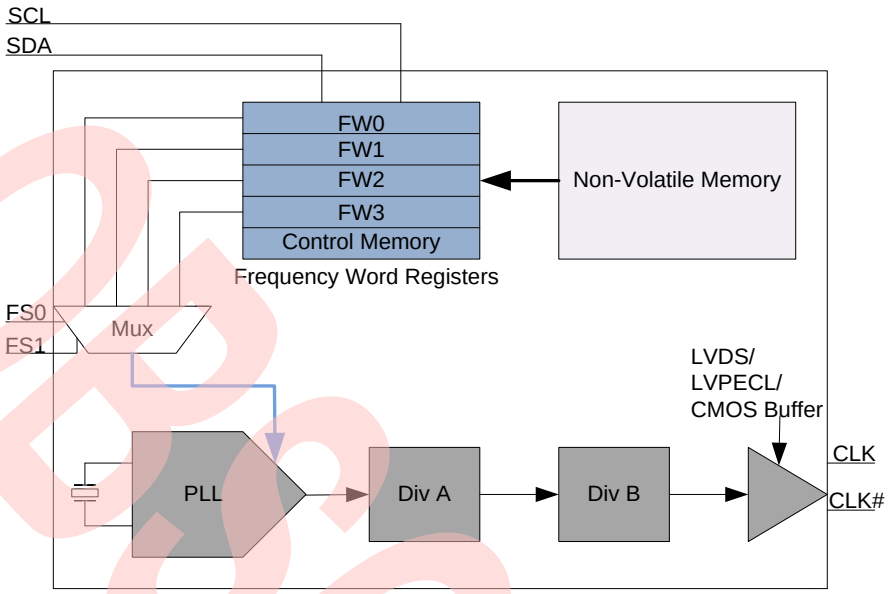
2 FleXO – A Flexible Crystal Oscillator: Architecture and Programming

The FleXO device includes an ultra-low-noise PLL with an on-chip crystal reference input, as shown in Figure 1. The output of the PLL is fed into a series of dividers. A programmable frequency word selects divisors and PLL parameters to control the output frequency.

Four registers are provided for the storage of frequency word arrays. One of four words can be selected through frequency select pins or an I^2C interface. The selected word decides the output frequency. An OTP nonvolatile memory is provided for the permanent storage of frequency word arrays. On power up, the contents of this nonvolatile memory are loaded into frequency word registers.

I^2C-supported devices also provide the ability to change frequency select words at run time. This feature can be used to change frequency once the device is already programmed. However, these changes are not stored in nonvolatile memory and remain in effect only until the device is powered down.

Figure 1. FleXO Architecture



For initial configuration and programming, refer to the application note and video AN62914 – Programming FleXO Low Noise Clock Generators.

The FleXO family currently includes five devices that support frequency margining. Three support I^2C, and two support pin-based frequency selection. Table 1 lists the FleXO devices that are capable of frequency margining.

Table 1. FleXO Devices with Frequency Margining Support

Part Number	Function	Output Standard
CY2XF23 CY2X013/ CY2X0137-Type2	Crystal oscillator with frequency margining, I^2C controlled	LVDS
CY2XF24 CY2X014/ CY2X0147-Type2	Crystal oscillator with frequency margining, I^2C controlled	LVPECL
CY2XF32	Crystal oscillator with frequency margining, pin select	CMOS
CY2XF33 CY2X013/ CY2X0137-Type3	Crystal oscillator with frequency margining, pin select	LVDS
CY2XF34 CY2X014/ CY2X0147-Type3	Crystal oscillator with frequency margining, pin select	LVPECL

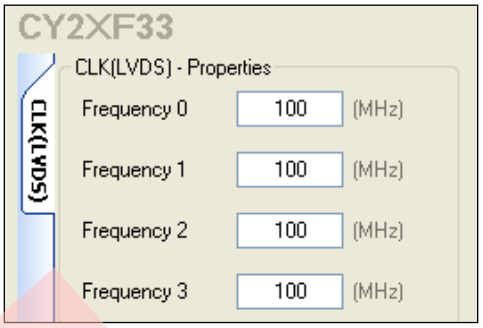
2.1 Frequency Margining in FleXO devices

To use the frequency margining feature available in FleXO devices, follow these steps:

1. Generate the base configuration JEDEC file for the FleXO device using CyClockWizard 1.0 (see the AN62914 video).
2. For devices with pin-based frequency selection (CY2XF32/33/34), enable the frequency select pins (FS0, FS1), as desired.
3. The clock **Properties** tab allows you to enter up to four frequency settings. Enter the desired frequencies, as shown in Figure 2.
4. Program the device and mount it on the application board.

Note: Contact sales or local Cypress FAEs for information on factory-programmed devices.

Figure 2. Frequency Selection Screen



2.2 Frequency Selection Using FS Pins

On devices with pin-based frequency select, the output frequency is selected based on the state of the enabled FS pins, as described in Table 2.

Table 2. Output Frequency Select

FS1	FS0	Output Frequency
0	0	Frequency 0
0	1	Frequency 1
1	0	Frequency 2
1	1	Frequency 3

2.3 Frequency Margining Using I²C

On devices with an I²C interface, frequency select is achieved by setting the two least significant bits of the “Select Byte” register, with address 40h, to the desired value, as shown in Figure 3.

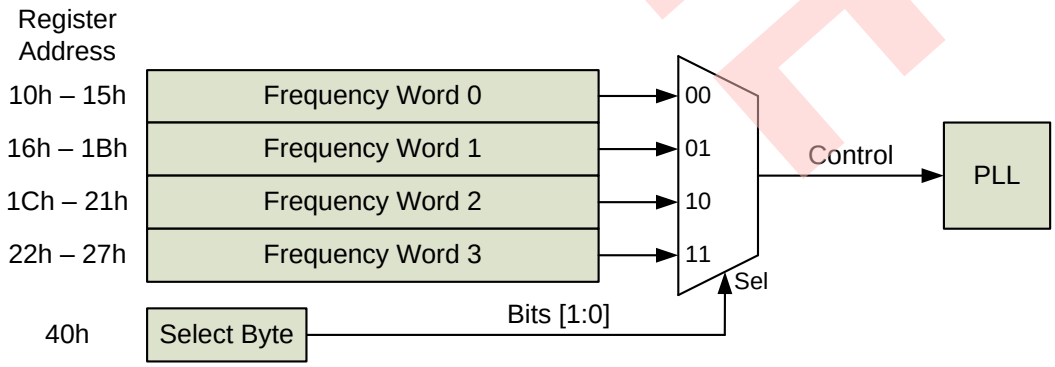
I²C devices also enable you to change the output frequency to values that are not defined during device programming. This is done by writing the desired frequency data to “Frequency Word” registers at the addresses shown in Figure 3. Each register is set to a 6-byte value that selects a specific output frequency.

To calculate the frequency select word corresponding to a specific frequency, do the following:

1. Generate the configuration JEDEC file (see the AN62914 video).
2. Obtain the frequency select word corresponding to the desired frequency from the JEDEC file.

For detailed information, refer to Appendix A: Frequency Margining Using I2C (CY2XF23/4).

Figure 3. Address of Frequency Select Words



3 Application Examples

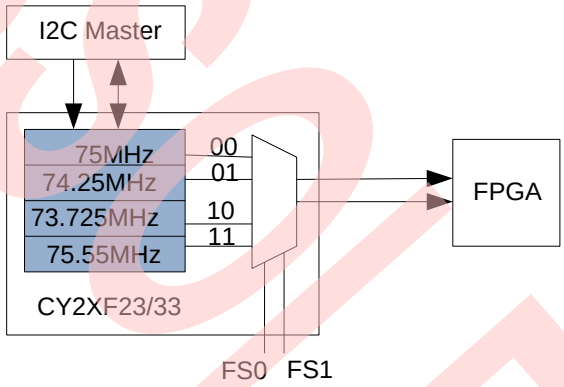
Frequency margining can be used in multiple application and design scenarios. This section provides common application examples for illustration.

3.1 Example 1: FPGA Logic Design Validation

System development using FPGA involves many challenges, particularly when it comes to achieving the optimum system speed. In a typical development scenario, designers are faced with the prospect of specification changes, leading to logic updates. In many cases, these updates affect timing and create a need to tune the operating clock frequency.

The FleXO frequency margining feature provides you with an easy method to change the system clock frequency. Consider, for example, an FPGA-based design that is set to operate at 75 MHz. In the event of a design change, timing violations can surface, and the clock may need tuning. With a FleXO device, you can program up to four frequencies (see Figure 4) and switch between them by applying the relevant logic signals on the FS pins. In this example, you can choose any of the four preprogrammed frequencies: 75 MHz, 74.25 MHz, 73.725 MHz, 75.55 MHz.

Figure 4. Frequency Margining for FPGA Logic Design



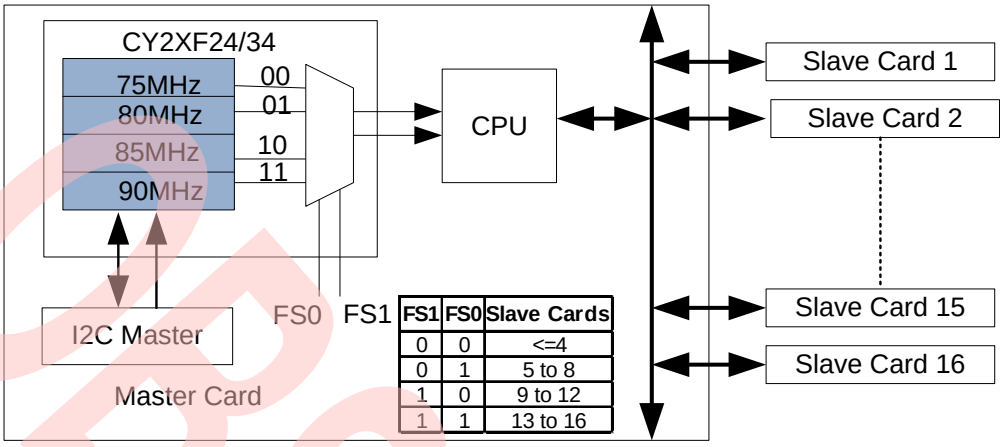
FleXO devices with an I^2C interface provide even greater flexibility by allowing you to set output frequencies that are different from the four values set during programming.

3.2 Example 2: Dynamic System Clock Adjustment

In a number of data processing applications, it is desirable to change the CPU clock frequency based on the operating load. Consider a data logging system, for example, as illustrated in Figure 5. The central module is connected to multiple slave units, from which it processes data. The processing load on the central unit varies, based on the number of slave cards connected to the system and/or the number of active slave cards in the system. In such a scenario, modulating the operating clock frequency of the CPU on the central unit will enhance system efficiency.

FleXO devices provide a simple method to modify system clock frequency on the fly. Based on processing load, the CPU can change the state of the FS pins on the FleXO device to select a different operating frequency.

Figure 5. Frequency Margining for Flexible System



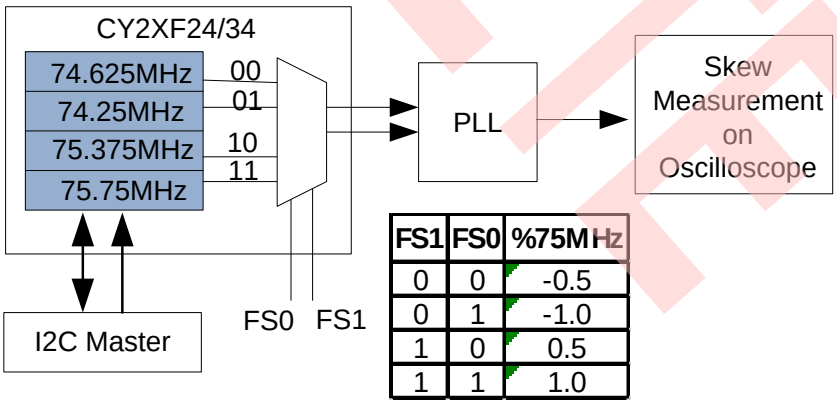
This feature can also be used to modulate system operating frequency efficiently and control the resulting system temperature. In performance-critical applications, it is desirable to operate at the highest possible clock speed. However, while improving performance, doing so also causes heat build-up. Coupled with a temperature monitor, the FleXO device can be used to reduce clock speeds when temperatures exceed the maximum thresholds. System speed can be increased again once the system temperature decreases to a safe operating level.

3.3 Example 3: System Characterization

Consider an example in which a component engineer wants to second-source an IC on a board and needs to ensure a sufficient timing margin in the system. The FleXO device can serve as a clocking solution in such situations, as it can vary the system frequency to extremes and help verify the IC functionality with such varied timing.

Another use of the FleXO device is characterizing the tracking skew of an external PLL chip on the board. Tracking skew is the amount of skew that appears in the output of a PLL due to a change in input frequency. As shown in Figure 6, the FleXO frequency margining feature can be used to change the input frequency, while measuring the variation of the output clock skew.

Figure 6. Frequency Margining for Tracking Skew Measurement



3.4 Example 4: Production Testing

Preventing the production of a faulty device due to a manufacturing defect or handling issues is not 100 percent achievable. A faulty device, in turn, can violate system timing and possibly cause a complete system malfunction. Performing an operating frequency margin test during production can easily identify the faulty unit in production. The I^2C accessibility of FleXO offers a greater flexibility for testing.

4 Summary

Armed with a unique frequency margining feature, FleXO devices provide significant flexibility in system design, testing, and functionality. This flexibility translates directly into improved product quality and reduced design effort.

About the Author

Name: Amitava Banerjee

Title: Applications Engineer Staff

A Appendix A: Frequency Margining Using I²C (CY2XF23/4)

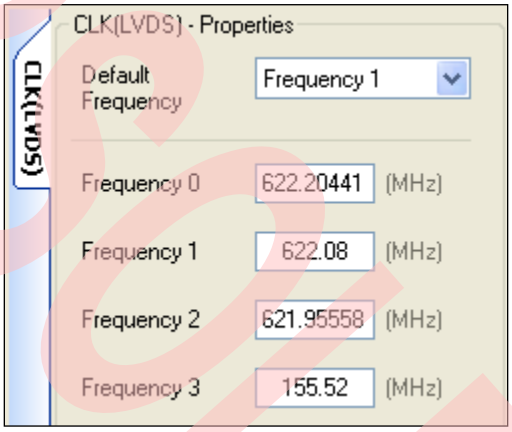
1. Generate the configuration JEDEC file with the required frequencies for the applicable device using CyClockWizard 1.0 (see the AN62914 video). The data corresponding to the four entered frequencies is written into the JEDEC file that is generated.

   Example: The application requires changing the FleXO output frequency to one of the following frequencies on the fly:

   - 622.08 MHz
   - 622.20441 MHz (+200 ppm of 622.08 MHz)
   - 621.95558 MHz (–200 ppm of 622.08 MHz)
   - 155.52 MHz

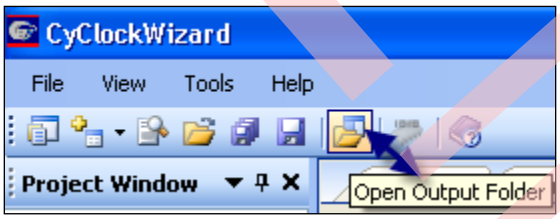
   Generate the JEDEC file using CyClockWizard by entering the frequencies, as shown in Figure 7.

   Figure 7. Frequency Options for I²C Device

2. Open the JEDEC file generated in step1 using the “Open Output Folder” button on the CyClockWizard menu bar, as shown in Figure 8.

   Figure 8. Output Folder in CyClockWizard Menu Bar

   Open the JEDEC file and locate the following data:

   L00128

   11010100101010000011010010111100000010101010011 1

   11010100101010000011010010111111111111010101011 00

   11010100011010000011001110111111111111110011010110

   11010100101010000011010010111100111111010101111 1*

3. Each row after “L00128” in the JEDEC file corresponds to one frequency word array. The data for Frequency-0 (622.204416 MHz) is written in the first row. Convert the binary data of the first row into hexadecimal format. The first row binary data and its hexadecimal form follow:

   11010100101010000011010010111100000010101010011 1 → D4 A8 34 BC 0A A7

4. To set the output frequency to 622.204416 MHz, write these six bytes (D4 A8 34 BC 0A A7) at the address of the active frequency word. The active frequency word is selected by the last two bits of the “Select Byte” register with the address as 40 h as shown in Figure 3.

   If the last two bits of the “Select Byte” register are 01b, the selected active frequency word is “Frequency Word 1,” and its array address is 16h–1Bh (Figure 3). Write 6-byte data (D4 A8 34 BC 0A A7) at address 16h–1Bh. The output frequency will be set to 622.204416 MHz.

   Similarly, write the relevant row data at the active frequency word to set a particular output frequency.

   Following are the binary data and the corresponding hexadecimal format for all four frequencies:

   L00128

   110101001010100000110100101111000000101010100111 → D4 A8 34 BC 0A A7 (622.204416M)

   110101001010100000110100101111111111101010101100 → D4 A8 34 BF FA AC (622.08M)

   110101000110100000110011101111111111110011010110 → D4 68 33 BF FC D6 (621.955584M)

   110101001010100000110100101111001111110101011111*→ D4 A8 34 BC FD 5F (155.52M)

Document History

Document Title: AN52133 - Frequency Margining Using FleXO™ and Its Applications

Document Number: 001-52133

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2677384	RICF	03/24/09	New application note.
*A	2734088	CXQ	07/09/09	Replaced all references to CY2XF44 with correct device name CY2XF34.
*B	3538384	BASH	03/01/2012	Updated title to read “Frequency Margining using FleXO™ and Its Applications”. Updated Abstract. Added FleXO – A Flexible Crystal Oscillator: Architecture and Programming. Updated Application Example. Updated in new template.
*C	4790578	XHT	06/08/2015	Added new part numbers in Table 1. Updated in new template.
*D	5848732	GNKK	08/09/2017	Updated the Cypress logo and copyright information.
*E	6372929	XHT	07/02/2018	Obsolete. Product obsolete

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709