This document is created to provide the required register information and calculations for setting PLL parameters of CY29411, CY29412, CY29421, CY29422 devices.

1 CY294xx Device Register Information

Table 1. Internal Memory Map (for Frequency Configuration) of CY294xx Device Family

Address	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x10	DIVO_FS0	DIVO < 7 : 0 >							
0x11	DIVN_INT_FS0	DIVO<8>	DIVN_INT < 6 : 0 >						
0x12	DIVO_MISC_FS0	*Reserved*	ICP_OFFSET	DIVN_INT < 8 : 7 >		ICP_BIN < 2 : 0 >			PLL_MODE
0x13	DIVN_FRAC0_FS0	DIVN_FRAC < 7 : 0 >							
0x14	DIVN_FRAC1_FS0	DIVN_FRAC < 15 : 8 >							
0x15	DIVN_FRAC2_FS0	DIVN_FRAC < 23 : 16 >							

Table 2. Internal Memory Map (for Common-mode Configuration) of CY294xx Device Family

Address	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x50	NCAL_CTRL	Reserved	I2C Bot Disable	ADC_CAPREMOVE1	HS_EN	CMOS_EN	*Reserved for Cypress factory*		
0x51	MISC_CTRL	XO_DOUBLER	VDD_DEF < 1 : 0 >		LOCK < 1:0 >		VCXO_enB	VCXO_BW <1:0 >	
0x52	VCXO_CTRL	VCXO_GINV	VCXO_GEXP < 2 : 0 >			VCXO_GSCALE < 3 : 0 >			
0x53	DRV_CTRL	OE_POL	DRV_TYPE < 2:0>			*Programmed bits by Cypress factory*			
0x54	I2C_CTRL	I2C_ADDR_EN	I2C_ADDR < 6 : 0 >						
0x55	XO_CTRL1	OT_GM <1:0>		XO_CAP < 5 : 0 >					
0x56	XO_CTRL2	XO_AMPSLICE <3:0>				XO_BYPASS	XO_CAP_X2 < 2 : 0>		
0x57	XO_CTRL3	OT_enB	OT_RES < 2 : 0>			*Programmed bits by Cypress factory*			

Note: The parameters of Table 2 are required to be configured through the ClockWizard 2.1 software.

2 Large change and small change update instructions

The change in the fractional part of the PLL division factor is referred to as a small change trigger. In the large change trigger, the PLL completely shuts down and comes up with the new frequency. When the device is in the Active state, user can change the output frequency by applying the Small Change or Large Change commands.

- Small change refers to the case in which the frequency changes within ±500 ppm. The frequency information will be loaded through I^{2}C, and the output frequency will change without any glitch from its original frequency to the new frequency. Note that the small change trigger functionality is not supported if the PLL is configured in integer mode.

  After the new configuration is written into the volatile memory, the following I^{2}C write operation needs to be executed to make the changes effective.

Write data 0x00 into the memory location 0x62

Write data 0x02 into the memory location 0x62

Write data 0x00 into the memory location 0x62

A sample i2c script to execute the command has been shown below (assuming I2C address = 55).

```
w 55 62 00
w 55 62 02
[delay=1]
w 55 62 00
```

- Large change refers to the case in which the frequency changes more than ±500 ppm, and the change is done through an I^{2}C.

Write data 0x00 into the memory location 0x62

Write data 0x01 into the memory location 0x62

Write data 0x00 into the memory location 0x62

A sample i2c script to execute the command has been shown below.

```
w 55 62 00
w 55 62 01
[delay=1]
w 55 62 00
```

Note: The parameters of Table 2 shouldn't be updated during large change and small change operation. A detail example script is shown in the section Example i2c script for programming followed by Large and Small change commands.

3 PLL Parameter Basic Calculation

DIVO, DIVN_FRAC, DIVN_INT, PLL_MODE, ICP_OFFSET calculation:

```
fOUT (in MHz), fIN (in MHz)

        VCO_min=6860 MHz

        VCO_max=8650 MHz
```

For a given `fOUT`, optimal `fVCO` needs to be chosen based on frequency planning to provide best jitter performance.

The parameters are related to the following formulae

```
        fVCO = DIVO * fOUT

        divN_cal = fVCO / fIN

        DIVN_INT = round (divN_cal)

        frac = divN_cal - DIVN_INT

        DIVN_FRAC = Round down (frac * 2^24)
```

`frac` must be within `[-0.5,0.5]`

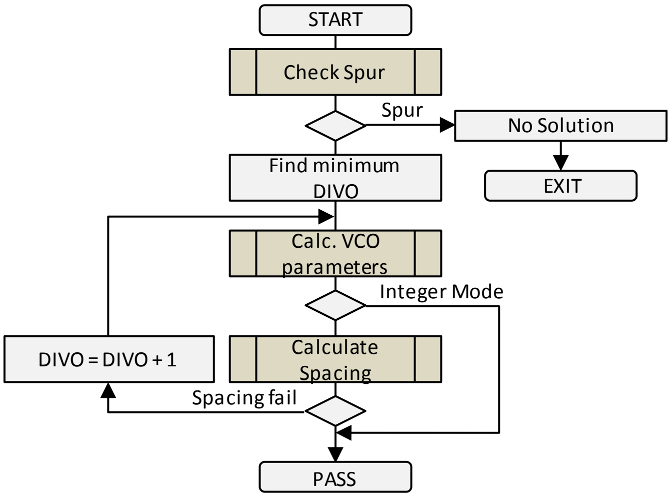
PLL_MODE:

```
If |frac / DIVN_INT| * 10^6 < 30(ppm) then set
PLL_MODE =1 (Integer-Mode)
```

4 Frequency planning algorithm (detail calculation)

This section has detail instruction of the parameters calculation for frequency registers in FS and Common registers as shown in Table 1 and 2. The algorithm is described below with parameters for Input frequency (fIN) and desired output frequency (fOUT). The best performance will be shown in Integer mode PLL and lower fVCO (lower DIVO).

Figure 1: Parameter calculation flow



4.1 Check Spurs

This routine detects the spurious on output, that is caused by fIN : fOUT coupling, before configure the device. User should choose different fIN if spur is detected.

```
        Spur = |fREF * M - fOUT * N|: for M [1:50], N[1:8]
```

if 0.01-MHz < Spur < 1.5-MHz, the combination of fIN (=fREF) and fOUT may have spur on output. There are no solution for this combination so user should change the reference frequency for the required fOUT.

Note. CY29411, CY29412, CY29421, CY29422 are Cypress packaged parts having 122.88 MHz crystal inside. Hence there is no provision to change the fIN parameter.

Example spur:

```
        fIN = 114.285, fOUT = 662.9697, Spur at M=29, N=5
        fIN = 122.88, fOUT = 1024.2396, Spur at M=25, N=3 and M=50, N=6
```

4.2 FS Parameter calculation

The following calculation will provide the parameters for:

```
        DIVO, DIVN_INT, frac, PLL_MODE, ICP_OFFSET, ICP_BIN
```

Parameter definitions:

```
        fOUT: Output frequency
        fIN: Input frequency
        fVCO: VCO frequency
```

Parameter Constant for the current revision:

```
        fVCO_min: minimum VCO frequency = 6860 (MHz)
        fVCO_max: maximum VCO frequency = 8650 (MHz)
```

fIN and fOUT constraints are described in the datasheet

Frequency parameter calculation

(DIVO, DIVN_INT, frac, DIVN_FRAC)

General Equations:

- fVCO = DIVO * fOUT or DIVO = fVCO / fOUT
- divN_cal = fVCO / fIN
- DIVN_INT = Round (divN_cal)
- frac = divN_cal - DIVN_INT
- DIVN_FRAC = Round down (frac * 2^24)

DIVN_FRAC is represented in 24-bit 2's complement for the frac

Note: frac must be within [-0.5,0.5]
Note: fVCO must be within [fVCO_min, fVCO_max]

Output frequency from the calculated parameters:

- calc_fVCO = fIN * (DIVN_INT + (DIVN_FRAC / 2^24))
- calc_fOUT = fVCO / DIVO

Find minimum DIVO

- DIVO_min = Round down (fVCO_min / fOUT) + 1

Calculation Example-1:

- fIN = 114.285-MHz
- fOUT = 622.08-MHz
- DIVO = Round down (6860 / 622.08) + 1 = Round down (11.02) +1 = 12

Frequency parameters

Calculation Example-1:

```
fIN = 114.285-MHz
fOUT = 622.08-MHz
DIVO = 12 (see previous section)
fVCO = DIVO * fOUT = 12 * 622.08 = 7464.96
divN_cal = fVCO / fIN = 7464.96 / 114.285 = 65.31880824
DIVN_INT = round (65.31880824) = 65
frac = divN_cal - DIVN_INT = 0.31880824
DIVN_FRAC = Round down (frac * 2^24) = 5348714
frac_HEX (5348714) = 519D6Ah.
```

Register setting: (X=1 for FS0, =2 for FS1 etc.,)

```
X0 = 0Ch (DIVO)
X1 = 41h (DIVN_INT)
X2 = See Misc registers
X3 = 6Ah (frac_HEX [7:0])
X4 = 9Dh (frac_HEX [15:8])
X5 = 51h (frac_HEX [23:16])
```

Note: DIVO and DIVN_INT are 9-bit expression, see register detail

Output frequency from calculated values:

```
calc_fOUT = 114.285 * (0.318808198 + 65) / 12 = 622.0799996-MHz
```

Calculation Example-2:

```
fOUT = 150.55-MHz
```

```
DIVO = (6860 / 150.55) + 1 = 46
fVCO = 6925.3
divN_cal = 60.59675373
DIVN_INT = 61
frac = -0.40324627
frac_HEX = 98C4DA (MSB = Sign bit. 2's complement expression)
```

Register setting:

```
X0 = 2E (DIVO)
X1 = 3D (DIVN_INT)
X2 = See Misc registers
X3 = DA (frac_HEX [7:0])
X4 = C4 (frac_HEX [15:8])
X5 = 98 (frac_HEX [23:16])
```

Note: DIVO and DIVN_INT are 9-bit expression, see register detail

Output frequency from calculated values:

```
98C4DAh (= 10011866d) is negative value
FRAC = 10011866 - 2^24 ==> -6765350
calc_fOUT = 114.285 * (-6765350 / 2^24 + 61) / 46 = 150.55-MHz
```

Calculate Spacing

- Calculate coff1 = | frac * fIN |
- Calculate coff2 = | (divN_cal * 2 - Round down (divN_cal) * 2 +1) * fIN |

```
If coff1 < 2 then Spacing fail
If coff2 < 1 then Spacing fail
```

if both - coff1 and coff2 - pass you can use the configuration.

```
Example for fail: fIN = 122.88 fOUT = 805.664, DIVO=9 failed
coff1 = 1.056 --Spacing fail
coff2 = 120.768 --Spacing pass
```

Solution: use DIVO=10 for clearing the spacing constraint.

4.3 Misc FS registers

(PLL_MODE, ICP_OFFSET, ICP_BIN)

PLL_MODE setting

Recommends to set the Integer mode PLL when PLL frequency differs less than 30-ppm between fractional and integer Mode, the formula is:

```
If |frac / DIVN_INT| * 10^6 < 30(ppm) then set
PLL_MODE =1 (Integer-Mode)
Set all-0 (zero) for frac_HEX if PLL_MODE = 1
```

Note. VCXO function and Small Change function is not supported for the Integer mode PLL

ICP_OFFSET

```
if PLL_MODE is Integer (PLL_MODE = 1) then
ICP_OFFSET = 0
Else
if (mod(fVCO , fIN)< 1.5 or fIN - mod(fVCO , fIN) < 1.5)
ICP_OFFSET=1
else
ICP_OFFSET=0
end if
```

ICP_BIN

Choose and set the ICP_BIN parameter according to the following Table.

Table 3: ICP_BIN chart

fVCO	ICP_BIN
fVCO < 7000	101
7000 =< fVCO < 7400	100
7400 =< fVCO < 7800	011
7800 =< fVCO	010

5 Example i2c script for programming followed by Large and Small change commands

5.1 Write a configuration into eFuse: User may choose to write a configuration to the internal eFuse (one time programmable memory) through i2c commands. The corresponding i2c script is here below. The configuration is shown in yellow.

```
; create_170621_201939
; BCP for eFuse Programming :Normal device
; device= CY5107
; ref=122.88
; fs0=100
; std=PECL
; vdd=3.3V
; vcxo=Disable

;Clear Notify command
w 55 03 ff
w 55 04 ff [Delay=1]
w 55 03 r 55 x x
; Soft reset
w 55 cf 30
w 55 cf 3f
w 55 cf 00 [Delay=1]
;
; Exit command command
w 55 63 00
w 55 63 01
[delay=1]
w 55 D7 C0
;
; Configuration details
;FS 0
w 55 10 45
w 55 11 38
w 55 12 0A
w 55 13 00
w 55 14 00
w 55 15 27
; read FS
w 55 10 r 55 x
w 55 11 r 55 x
w 55 12 r 55 x
w 55 13 r 55 x
w 55 14 r 55 x
w 55 15 r 55 x
```

```
;
; com
w 55 50 04
w 55 51 CC
w 55 53 10
w 55 54 D5
w 55 55 95
w 55 56 C5
w 55 57 50
[Delay=1]
; read COM
w 55 50 r 55 x
w 55 51 r 55 x
w 55 52 r 55 x
w 55 53 r 55 x
w 55 54 r 55 x
w 55 55 r 55 x
w 55 56 r 55 x
w 55 57 r 55 x
;
;
w 55 60 00
w 55 61 00
w 55 62 00
w 55 63 00
; Program eFuse command
w 55 61 00
w 55 61 03 [Delay=5]
w 55 61 00 [Delay=1]
;
```

5.2 **Small change update:** i2c commands to write the updated register values to change the output frequency by 100ppm. This is followed by small change update command.

```
; create_170621_201253
; BCP for small change
; fref=122.88
; fs0_org=100
; offset_ppm=100
; calc'd freq=100.01

w 69 10 45
w 69 11 38
w 69 12 0A
w 69 13 00
w 69 14 70
w 69 15 28
;
; Small change update command
w 69 62 00
w 69 62 02
[delay=1]
w 69 62 00
```

5.3 **Large change update:** i2c commands to write the updated register values to change the output frequency to 150 MHz. This is followed by large change update command.

```
; create_170622_112501
; BCP for Large change
; device= CY5107
; ref=122.88
; fs0=150
; std=PECL
; vdd=3.3V

;FS 0
w 69 10 45
w 69 11 38
w 69 12 0A
w 69 13 00
w 69 14 00
w 69 15 27
;
[delay=1]
;
; large change command
w 69 62 00
w 69 62 01
[delay=1]
w 69 62 00
```

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

Cypress Developer Community

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

Technical Support

cypress.com/support

PSoC is a registered trademark of Cypress Semiconductor Corporation.

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

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.