TriCore™ TC1791/93/98

AP 32185

Frequency Modulated PLL

Application Note

V1.0 2011-09

Microcontrollers

Edition 2011-09

Published by
Infineon Technologies AG
81726 Munich, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

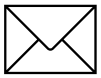
Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision History:	
Previous Version:	
Page	**Subjects (major changes since last revision)**

We Listen to Your Comments

Is there any information in this document that you feel is wrong, unclear or missing?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:



mcdocu.comments@infineon.com

Table of Contents

1 Preface

1.1 How to use this application note

Frequency-modulated clocks are an efficient way to reduce electromagnetic emission significantly. Unfortunately, clock modulation implies clock edge offset (accumulated jitter, time interval error) which might inhibit correct operation of asynchronous data interfaces. Infineon's "clipped-FM" technology solves this problem by limiting the accumulated jitter while providing the full emission reduction as it is known from existing spread-spectrum systems.

This application note should provide the reader with:

Chapter 2:

- Basic knowledge on clock frequency modulation
- Infineon microcontrollers' frequency-modulated PLL (FMPLL) implementation and its benefits:
  - Limitation of the accumulated jitter / time interval error for proper asynchronous data transfers
  - Calculation of clock frequency offsets
  - Limit curves for the accumulated jitter

Chapter 3:

- Programming the Infineon microcontroller FMPLL

Chapter 4:

- Emission reduction from FMPLL, validated by measurements

Chapter 5:

- FMPLL parameter values and trends, validated by measurements:
  - Accuracy of the mean system frequency (target clock reference)
  - Physical clock frequency offset for proper FM operation; limitations for TC1798
  - Accuracy of modulation frequency and modulation amplitude
  - Accumulated jitter in clipped-FM mode
  - Introduction of fitting curve for easy J_{ACC} limit calculation
  - Mean system frequency deviation in clipped-FM mode
  - Maximum time interval error

Chapter 6:

- PLL control register settings and maximum accumulated jitter values for recommended FMPLL configurations

1.2 Abbreviations

BISS	IC EMC Test Specification, download from http://www.zvei.org/IC_EMC_Test_Specification
Center-spread	Symmetric frequency modulation around a center frequency.
EMC	Electromagnetic compatibility, the ability of a system to not disturb any other systems and being not disturbed by other systems.
EME	Electromagnetic emission, RF noise generated by (synchronous) switching activity.
FM	Frequency modulation, a periodic change of a clock rate.
FMPLL	Frequency-modulated phase-locked loop, an emission-reducing clock generator for ICs.
f_{CPU}	Operating clock for the microcontroller's central processing unit. f_{CPU} is derived from f_{PLL}.
f_{MOD}	Modulation frequency, determines the duration of one full modulation period.
f_{OSC}	Oscillator frequency, determined by the crystal connected to the microcontroller's oscillator.
f_{PLL}	PLL output frequency, used as input clock for the respective clock domain dividers.
J_{ACC}	Accumulated jitter, the maximum expected offset of the real clock edge over an infinite time towards the nominal (unmodulated) clock edge without noise. Note that in the microcontrollers' data sheets the accumulated jitter is abbreviated as J_{TOT}.
LF	Low frequency, audio-range frequency.
MA	Modulation amplitude, half frequency shift between minimum and maximum frequency; for a symmetrical center-spread modulation this is the frequency shift between the center frequency and the maximum/minimum frequency, respectively.
MTIE	Maximum time interval error, the maximum expected offset of the real clock edge towards the nominal (unmodulated) clock edge without noise after a defined time interval.
PLL	Phase-locked loop, a VCO which generates a low-jitter high-frequency clock by synchronizing to a low-frequency reference clock.
RF	Radio frequency, high frequency used as radio carrier.
Upspread	Frequency modulation above a nominal frequency.
VCO	Voltage-controlled oscillator, used in a PLL to compensate frequency drifts of the high-frequency clock.
VDD	Core supply voltage, powering the digital logic of the microcontroller.
VDDP	Pad supply voltage, powering the I/O stages of the microcontroller.

2 Introduction to frequency-modulated clocks

Frequency-modulated clocks are also known as spread spectrum clocks. They are well established in communication techniques. FM radio uses an RF carrier which is modulated by the LF audio signal. As a result, the carrier frequency moves around its nominal frequency. The modulation amplitude is defined by the LF signal amplitude, the modulation frequency is identical to the LF frequency. Both parameters are varying over time.

Frequency modulation can be used to intentionally spread the carrier energy around the nominal carrier frequency. Since the overall energy stays constant, the carrier energy is distributed over a frequency band instead of staying in one discrete frequency. As a result, the peak emission caused by the carrier is reduced. Typical applications of this spread-spectrum technique are EMC-critical applications such as automotive electronic control units.

Although 20 dB emission reduction can be reached using spread-spectrum clocks, this technique is not yet very common for automotive microcontroller systems. The main reason is the danger of inhibiting real-time functions like asynchronous data communication or timer capture sequences. To understand this danger, let us have a closer look at the determining parameters for frequency-modulated clocks.

Similar to radio FM, a frequency-modulated microcontroller clock is controlled by its modulation frequency (f_{MOD}) and its modulation amplitude (MA).

Typically, the modulation frequency should be selected to be approximately factor 1000 below the clock frequency. This distance is required by the PLL loop filter which must prevent the modulation clock to be coupled to the VCO. Typical values for f_{MOD} range between 50 kHz and 200 kHz.

The modulation amplitude defines the amount of frequency shift in one direction from the mean clock frequency (“carrier”). An MA value of 1% on a 100 MHz clock means a frequency variation between 99 MHz and 101 MHz.

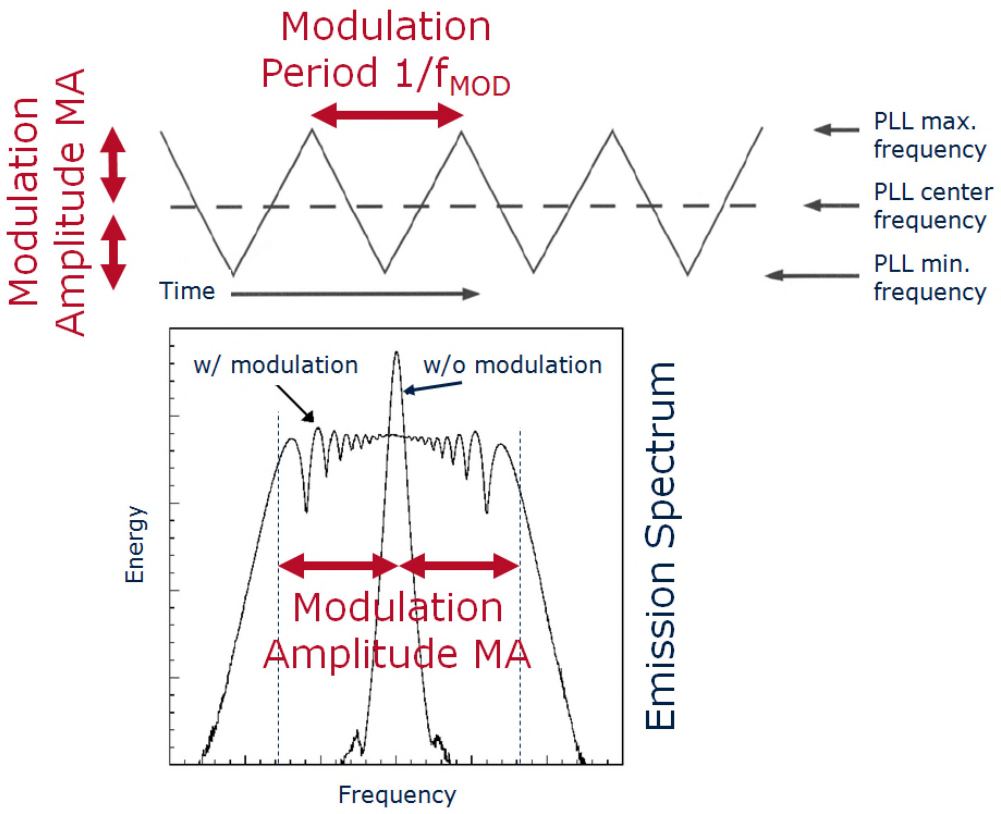


Figure 1: *FMPLL function*

Higher values of MA and lower values of f_{MOD} lead to less electromagnetic emission (EME). Bigger MA values lead to wider sidebands where the carrier energy is distributed. Slower f_{MOD} values lead to a smoother distribution of sideband energy, i.e. any discrete sideband frequency is activated less often. Chapter 4 provides EME measurement results without and with activated frequency modulation of the system clock.

Unfortunately, the trend of less emission together with higher MA and lower f_{MOD} is accompanied by an increasing accumulated jitter. This parameter is also known as maximal time interval error (MTIE). It means the the amount of time shift between the unmodulated clock edges and the modulated clock edges over a certain time interval.

In order to calculate the accumulated jitter over n clock periods, the duration of n mean clock periods (i.e. n clock periods of the mean frequency) is subtracted from the duration of those n real clock periods under observation, see Equation 1.

$$J_{ACC}(n) = \sum_{i=1}^{n} P_i - n \cdot P_{mean} \quad \text{(Equation 1)}$$

Since this value for $J_{ACC}(n)$ will vary for different selections of consecutive clock periods, the calculation of $J_{ACC}(n)$ must be refined. Therefore a data base of much more than n clock periods should be sampled. $J_{ACC}(n)$ is then calculated for every n-cycle cluster within this data base. The longest ($J_{ACC\text{-}max}$ according Equation 2a) and shortest ($J_{ACC\text{-}min}$ according Equation 2b) found duration of n clock cycles is calculated, and the higher absolute value of those is defined as $J_{ACC}(n)$.

s is the starting period from which the next n periods are concatenated.

$$J_{ACC\_max} = \max\left(\sum_{i=s+1}^{s+n} P_i\right) - n \cdot P_{mean} \quad with\ s = 0,1,2,\ldots\ and\ n = 1,2,3,\ldots \quad \text{(Equation 2a)}$$

$$J_{ACC\_min} = \min\left(\sum_{i=s+1}^{s+n} P_i\right) - n \cdot P_{mean} \quad with\ s = 0,1,2,\ldots\ and\ n = 1,2,3,\ldots$$

(Equation 2b)

For a center-spread triangular modulation, $J_{ACC\text{-}max}$ is always posive and J_{macc_min} is always negative, but the absolute values of both are (ideally) equal. However, they may differ by a small amount due to noise influence.

Finally, the value of J_{ACC} should be independent of n. Therefore, from all $J_{ACC\text{-}max}$ values found for all tested values for n, the total maximum is selected. Also, from all $J_{ACC\text{-}min}$ values found for all tested values for n, the total minimum is selected. From these to values, the larger absolute value is defined as the general accumulated jitter J_{ACC}, see Equation 3.

$$J_{ACC} = \max(|J_{ACC\_max}|, |J_{ACC\_min}|) \quad \text{(Equation 3)}$$

For a real FMPLL, J_{ACC} is composed of a random noise part and a deterministic FM part $J_{ACC\text{-}FM}$. The noise part is resulting from electrical disturbance.

Now let us select m in a way that m clock cycles match the duration of exactly one or multiple modulation periods. Then the resulting $J_{ACC}(m)$ is ideally zero. This is because shorter and longer periods than the mean period cancel each other. However, in a real noisy system, even $J_{ACC}(m)$ is non-zero due to the overlaid intrinsic jitter, but anyway it shows local minima for multiples of one modulation period.

Generally, the variation of J_{ACC} over n shows a periodical sinusoidal envelope curve with the period length of the modulation frequency, see middle diagram in Figure 2.

The $J_{ACC\text{-}FM}$ value which is determined by the frequency modulation ranges from ca. 20 ns up to well above 100 ns for typical FM settings. For a center-spread triangular modulation, $J_{ACC\text{-}FM}$ is a function of the modulation frequency f_{MOD} and the modulation amplitude MA according to Equation 4:

$$J_{ACC\_FM}[ns] = \frac{2500 \cdot MA[\%]}{f_{MOD}[kHz]}$$

(Equation 4)

Due to the periodic modulation of clock periods, the maximum value of $J_{ACC\text{-}FM}$ occurs for $J_{ACC\text{-}FM}(2*k-1)*m/2)$, assuming that k=1,2,3,... and m clock cycles cover exactly one full modulation period.

Figure 2 shows some important frequency modulation parameters. As an example, a 100 MHz clock was taken. This clock is modulated with f_{MOD} = 100 kHz and MA = 1.0%. Additionally, some random noise was added which leads to additional noise jitter.

The “Period Sequence” diagram shows the periods of 10000 clock cycles, covering 10 modulation periods. The triangular modulation is seen clearly. The modulation-related and the noise-related period duration swings are marked in the diagram.

The “Accumulated Jitter” diagram shows the clock edge offset for period groups of 1 to 10000. For every multiple of 1000 clock periods (n=1000, 2000, 3000, ...), J_{ACC} shows a local minimum. However, J_{ACC} does not

return to zero due to the random noise which was added to the clock signal. The theoretical J_{ACC} value of 25 ns according to Equation 4 is nearly met, the offset is again due to the random noise.

The “MTIE” diagram shows the maximum time interval error for data rates between 30 kHz and 1 MHz. Maxima occur for all time intervals covering (n-1/2) modulation periods; minima occur for all time intervals covering n modulation periods, with n=1,2,3,...

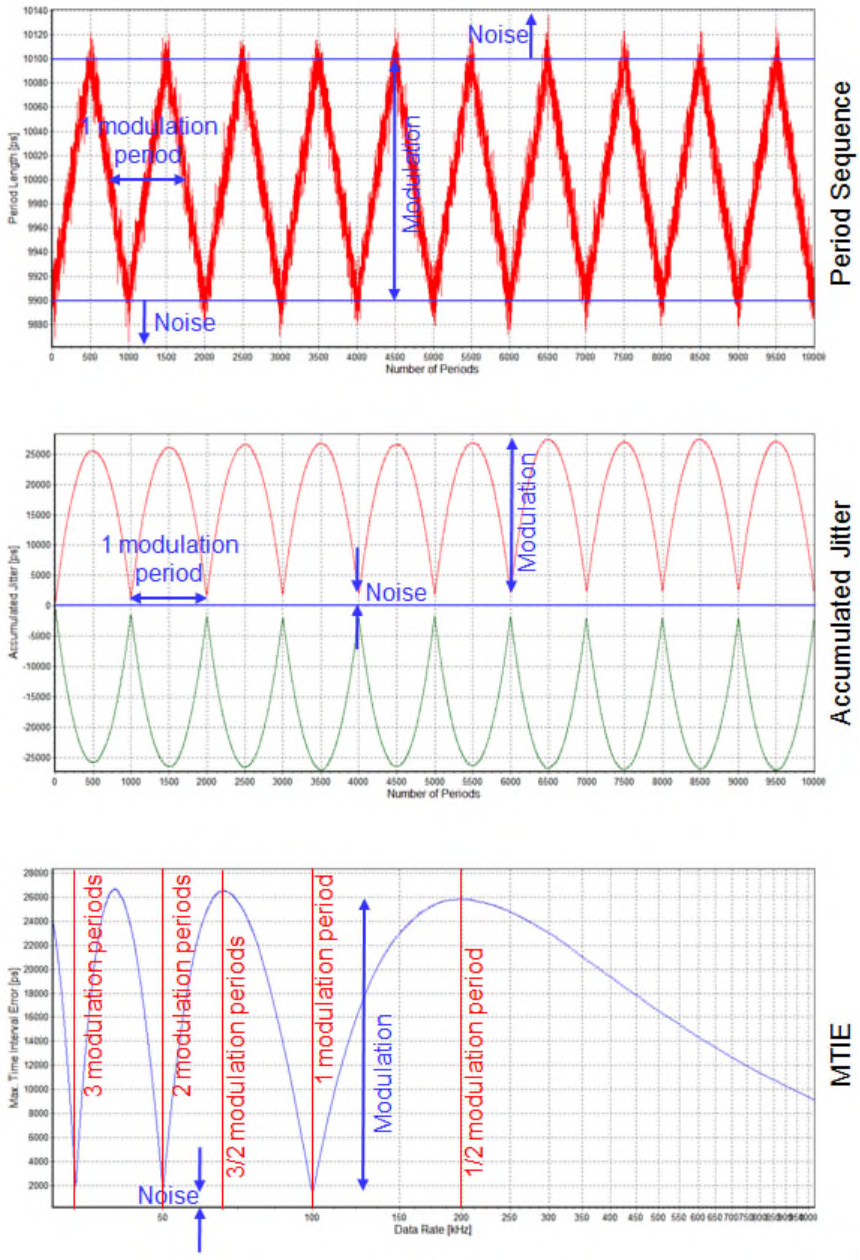


Figure 2: *Important FM parameters*

In order to guarantee reliable function of asynchronous interfaces and real-time data capture functions, the MTIE along an asynchronous frame has to stay below a certain limit defined for these interfaces and timer modules operated with this FM clock. This requirement can be achieved by using a fast f_{MOD} and a small MA, however these settings result in higher electromagnetic emission than a slow f_{MOD} and a large MA.

Infineon microcontrollers solve this MTIE problem by introducing a so-called “clipped-FM” PLL. It operates an upspread FM, i.e. the real FM clock is always higher than the desired mean system clock. A dedicated circuitry determines whether the accumulated jitter is exceeding one VCO clock period. An appropriate number of VCO clocks is suppressed over time in a way that the resulting number of clocks equals the number of unmodulated clocks. By this clock clipping, the accumulated jitter is limited to small values which should meet the jitter/MTIE specifications of commonly used data interfaces, regardless of the selected f_{MOD} and MA values.

Figure 3 compares the modulation technique and resulting accumulated jitter of “classical” and “clipped” FM.

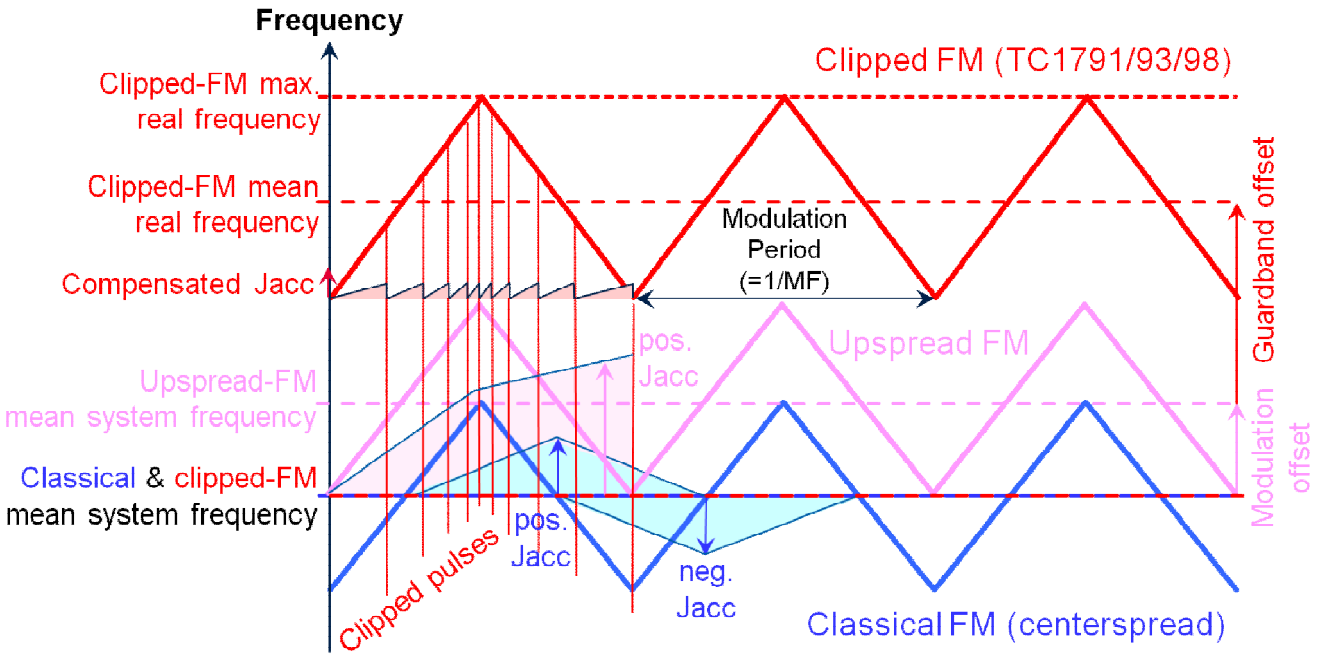


Figure 3: *Classical and clipped FM*

The clipped-FM mode uses an upspread modulation. The real FMPLL frequency stays above the target system frequency at any time. This is necessary to meet the same target (average) system frequency as in case of an unmodulated PLL, see Equation 5a. Over time, the system sees an average system frequency $f_{PLL\text{-}nom}$ according to Equation 5a for clipped-FMPLL mode, named “Clipped-FM mean system frequency” in Figure 3. f_{OSC} is the crystal frequency, P is the oscillator divider, N is the PLL feedback divider and K2 is the system clock divider according the clock system specification of the microcontrollers.

$$f_{PLL\_nom} = \frac{f_{OSC} \cdot N}{P \cdot K2} \qquad \text{(Equation 5a)}$$

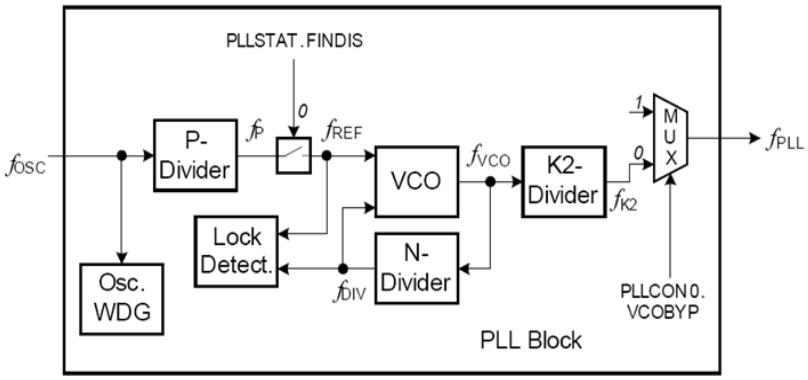
To ensure reliable operation under all conditions, an additional guardband offset for the real system frequency was introduced, see the red curve in Figure 3. The absolute maximum frequency which occurs during clipped-FM operation (named “Clipped-FM max. real frequency” in Figure 3) is calculated according Equation 5b. It shows additional dependencies from the N-divider, an additional N_{offset} value, and the modulation amplitude MA. An additional small frequency offset caused by noise is not considered in Equation 5b.

$$f_{PLL\_max} = \frac{f_{osc} \cdot N}{P \cdot K2} \cdot \left[1 + \frac{N_{offset} + \mathrm{int}(MA \cdot N/51) + \mathrm{int}(N/64)}{2 \cdot N} \right] \cdot (1 + MA/100) \qquad \text{(Equation 5b)}$$

3 FMPLL programming

Prior to FM configuration, the system PLL must be set to the desired mean system frequency f_{PLL}. Four parameters determine the system frequency, see Figure 4:

- Crystal frequency f_{OSC}
- P-divider
- N-divider
- K2-divider



$$8MHz \le \frac{f_{OSC}}{P} \le 16MHz$$

$$f_{PLL} = \frac{N}{P \cdot K2} \cdot f_{OSC}$$

Figure 4: *System PLL block diagram*

The corresponding bit fields are located in registers PLLCON0 and PLLCON1, see Figures 5 and 6.

PLLCON0
PLL Configuration 0 Register **(018$_H$)** **Reset Value: 0001 C600$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				PDIV				0					RES LD	0	1
rw				rw				r					w	r	rw

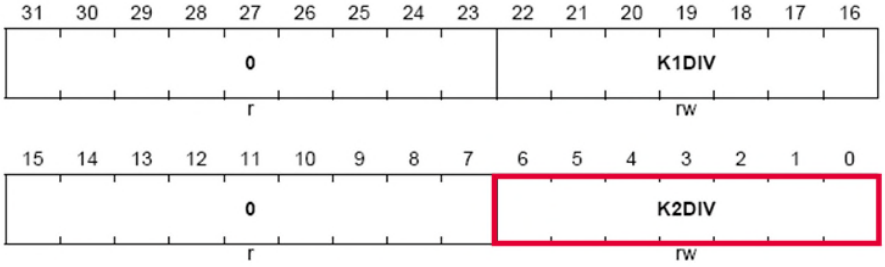
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDIV							0	0	OSC DISC DIS	CLR FIN DIS	SET FIN DIS	0	MOD EN	VCO PWD	VCO BYP
rw							rw	r	rw	w	w	rw	rw	rw	rw

Field	Bits	Type	Description
VCOBYP	0	rw	**VCO Bypass** 0_B Normal operation, VCO is not bypassed 1_B Prescaler Mode; VCO is bypassed
VCOPWD	1	rw	**VCO Power Saving Mode** 0_B Normal behavior 1_B The VCO is put into a Power Saving Mode and can no longer be used. Only Prescaler Mode are active if previously selected.
MODEN	2	rw	**Modulation Enable** This bit controls the activation of the frequency modulation of the PLL. 0_B Frequency modulation is not activated 1_B Frequency modulation is activated
SETFINDIS	4	w	**Set Status Bit PLLSTAT.FINDIS** 0_B Bit PLLSTAT.FINDIS is left unchanged 1_B Bit PLLSTAT.FINDIS is set. The input clock from the oscillator is disconnected from the VCO part.
CLRFINDIS	5	w	**Clear Status Bit PLLSTAT.FINDIS** 0_B Bit PLLSTAT.FINDIS is left unchanged 1_B Bit PLLSTAT.FINDIS is cleared. The input clock from the oscillator is connected to the VCO part.

Field	Bits	Type	Description
OSCDISCDIS	6	rw	**Oscillator Disconnect Disable** This bit is used to disable the control PLLSTAT.FINDIS in a PLL loss-of-lock case. 0_B In case of a PLL loss-of-lock bit PLLSTAT.FINDIS is set 1_B In case of a PLL loss-of-lock bit PLLSTAT.FINDIS is cleared
NDIV	[15:9]	rw	**N-Divider Value** The value the N-Divider operates is NDIV+1.
RESLD	18	w	**Restart VCO Lock Detection** Setting this bit will clear bit PLLSTAT.VCOLOCK and restart the VCO lock detection. Reading this bit returns always a zero.
PDIV	[27:24]	rw	**P-Divider Value** The value the P-Divider operates is PDIV+1.
0	3, 8, [31:28]	rw	**Reserved** Have to be written with 0.
1	16	rw	**Reserved** Should be written with 1.
0	7, 17, [23:19]	r	**Reserved** Read as 0; should be written with 0.

Figure 5: *PLL/FMPLL-relevant register PLLCON0*

PLLCON1
PLL Configuration 1 Register (01C$_H$) Reset Value: 0002 000F$_H$

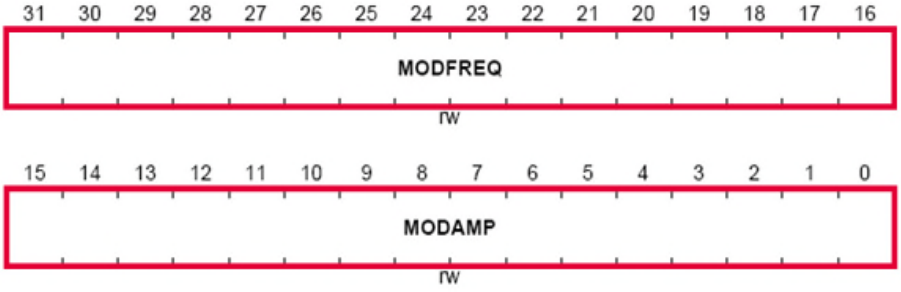


Field	Bits	Type	Description
K2DIV	[6:0]	rw	**K2-Divider Value** The value the K2-Divider operates is K2DIV+1.
K1DIV	[22:16]	rw	**K1-Divider Value** The value the K1-Divider operates is K1DIV+1.
0	[15:7], [31:23]	r	**Reserved** Read as 0; should be written with 0.

Figure 6: *PLL-relevant register PLLCON1*

In a second step, the FM parameters need to be programmed in registers PLLCON2, see Figure 7. PLLCON2 contains the bit patterns for the modulation frequency f_{MOD} and the modulation amplitude MA in two 16-bit fields. Finally, the FMPLL must be activated by setting the MODEN bit (= bit 2) in register PLLCON0.

PLLCON2
PLL Configuration 2 Register (020$_H$) Reset Value: 0000 0000$_H$



Field	Bits	Type	Description
MODAMP	[15:0]	rw	**Modulation Amplitude** This bit field defines the amplitude of the modulation.
MODFREQ	[31:16]	rw	**Modulation Frequency** This bit field defines the modulation frequency.

Figure 7: *FMPLL-relevant register PLLCON2*

For any desired f_{MOD} and MA values, the corresponding MODFREQ and MODAMP bitfields can be calculated according Equations 6a and 6b:

$$MODFREQ = \text{int}\left(\frac{5.14 \cdot MA \cdot N \cdot f_{MOD} \cdot P}{f_{OSC}}\right) \quad \text{(Equation 6a)}$$

$$MODAMP = \text{int}(161 \cdot MA \cdot N) \quad \text{(Equation 6b)}$$

Vice versa, for any given desired MODFREQ and MODAMP bitfields, the corresponding f_{MOD} and MA values can be calculated according Equations 6c and 6d:

$$f_{MOD} = \frac{MODFREQ \cdot f_{OSC} \cdot 31.32}{MODAMP \cdot P} \quad \text{(Equation 6c)}$$

$$MA = \frac{MODAMP}{161 \cdot N} \quad \text{(Equation 6d)}$$

In an optional step, the FM parameter Noffset may be programmed in the memory-mapped address [0xF000 0650H], see Figure 8.

For proper clipped-FM operation, the bits in this memory location have to be set to the values shown in Table 1. All bits not listed in Table 1 should not be changed according to their reset values.

Bit location	Name	Recommended value (binary)
16	-	1B
14:12	-	010B
11:9	NOFFSET[1]	100B for TC1791 and TC1793 100B or 011B for TC1798
8	-	1B

Table 1: *Recommended bit settings in memory-mapped address [0xF000 0650H]*

Note:

[1] The 3-bit field NOFFSET is located in bits [11:9]. For TC1791 and TC1793, Noffset should be programmed to Noffset=4. For the TC1798, Noffset may be programmed to Noffs=3, depending on the desired modulation amplitude. Please refer to Table 3 in chapter 5.4 for details.

4 FMPLL settings and resulting electromagnetic emission

Both modulation frequency f_{MOD} and modulation amplitude MA influence the reduction of electromagnetic emission (EME). Figure 8 shows the EME trend for selected f_{MOD} and MA settings (gray lines). The resulting accumulated jitter is also shown (black lines). Note that these J_{ACC} values are according Equation 4, and are valid for the classical FM, but not for the clipped FM. Pulse clipping reduces the accumulated jitter significantly for all these f_{MOD} and MA values. Therefore, the maximal EME reduction (up to ca. 20 dB) can be achieved only with clipped-FM, without limiting the proper function of critical asynchronous serial interfaces or real-time capture modes.

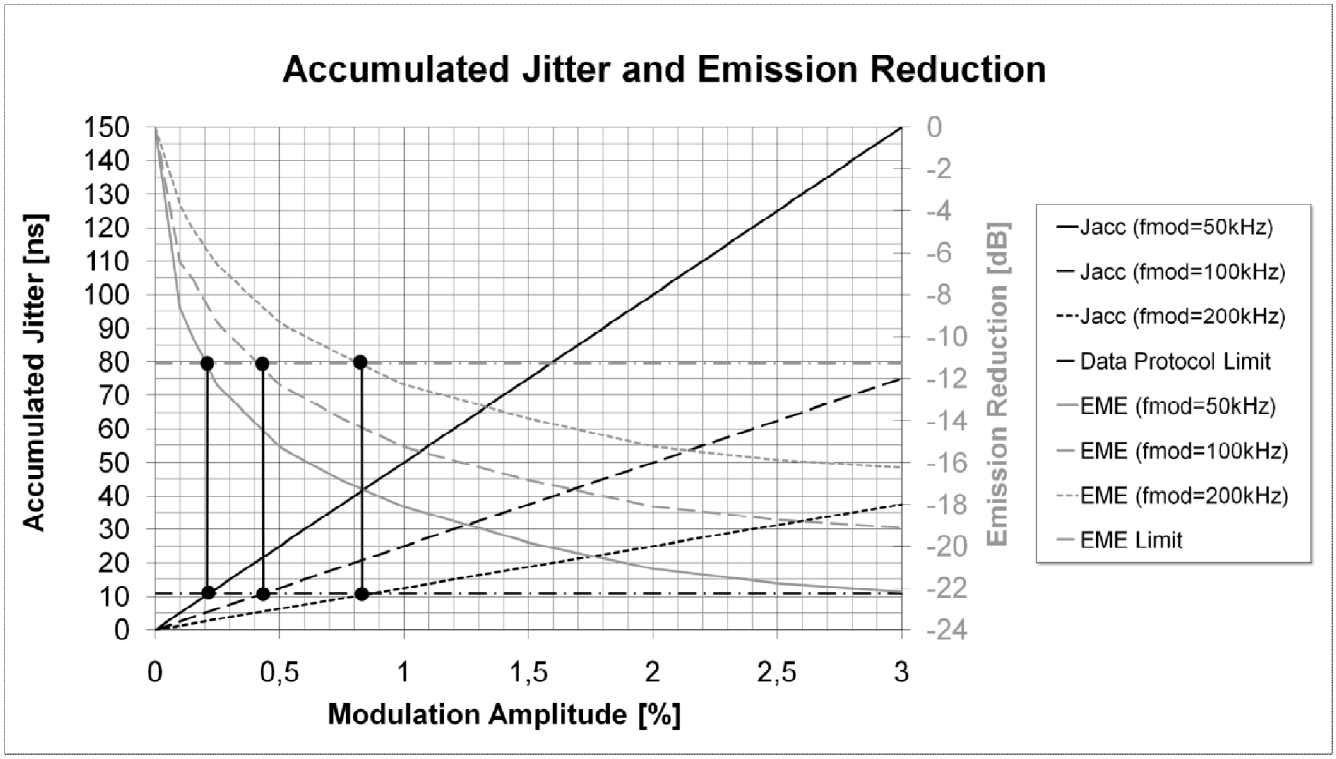


Figure 8: *EME reduction trend as function of f_{MOD}, MA and MTIE*

In a noisy microcontroller environment, the typical intrinsic PLL accumulated jitter may rise to 4 ns under worst-case conditions. To stay below an overall accumulated jitter of e.g. 15 ns, the J_{ACC} contribution caused by clock modulation must stay below 11 ns. The corresponding reachable emission reduction is ca. 11 dB (follow the black lines with a ball on each end from the 11 ns “Data Protocol Limit” to the “EME Limit”). This means that ca. 11 dB emission reduction can be reached by using a conventional FMPLL without clock clipping to maintain proper asynchronous data transfer. When using Infineon’s clipped-FM PLL, the accumulated jitter stays sufficiently low even for higher modulations, providing an emission reduction up to ca. 20 dB without violation the accumulated jitter requirements of common data interfaces.

4.1 Electromagnetic emission measurement

Figures 9-11 show the measured emission spectra of the microcontroller TC1793 running at a system clock f_{PLL} = 270 MHz for disabled FM and for f_{MOD} = 100 kHz and MA = 1 and 3%, respectively. The red curve indicates the BISS[1] emission limit.

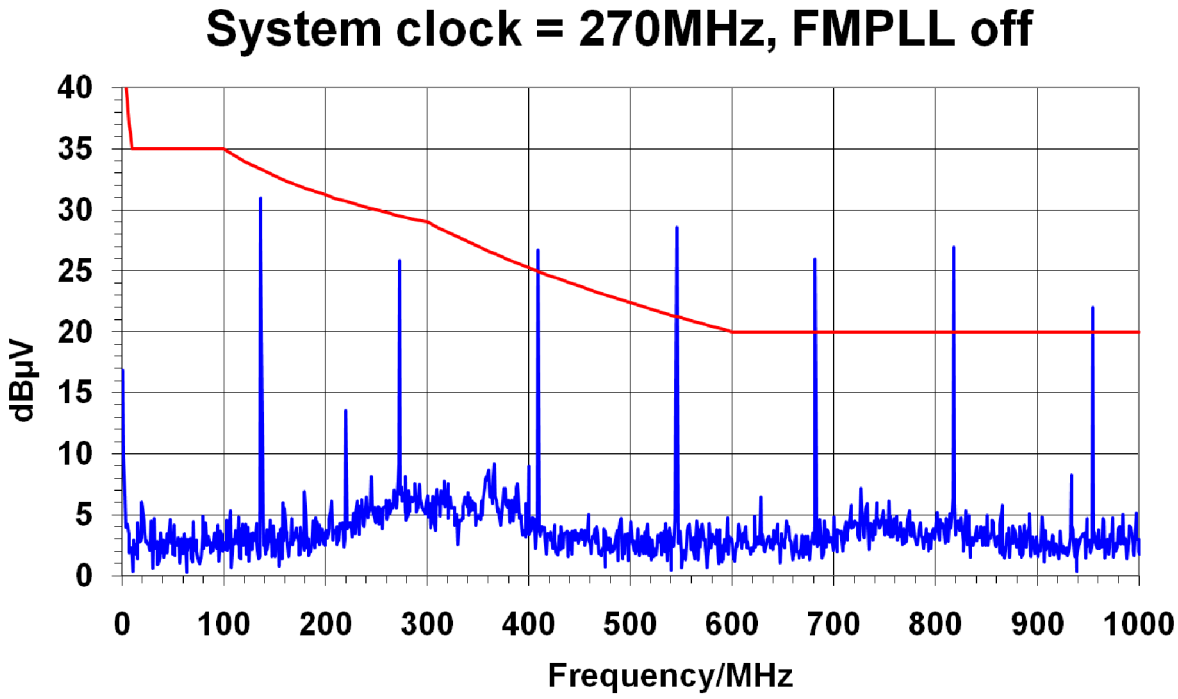


Figure 9: *Reference emission without FM*

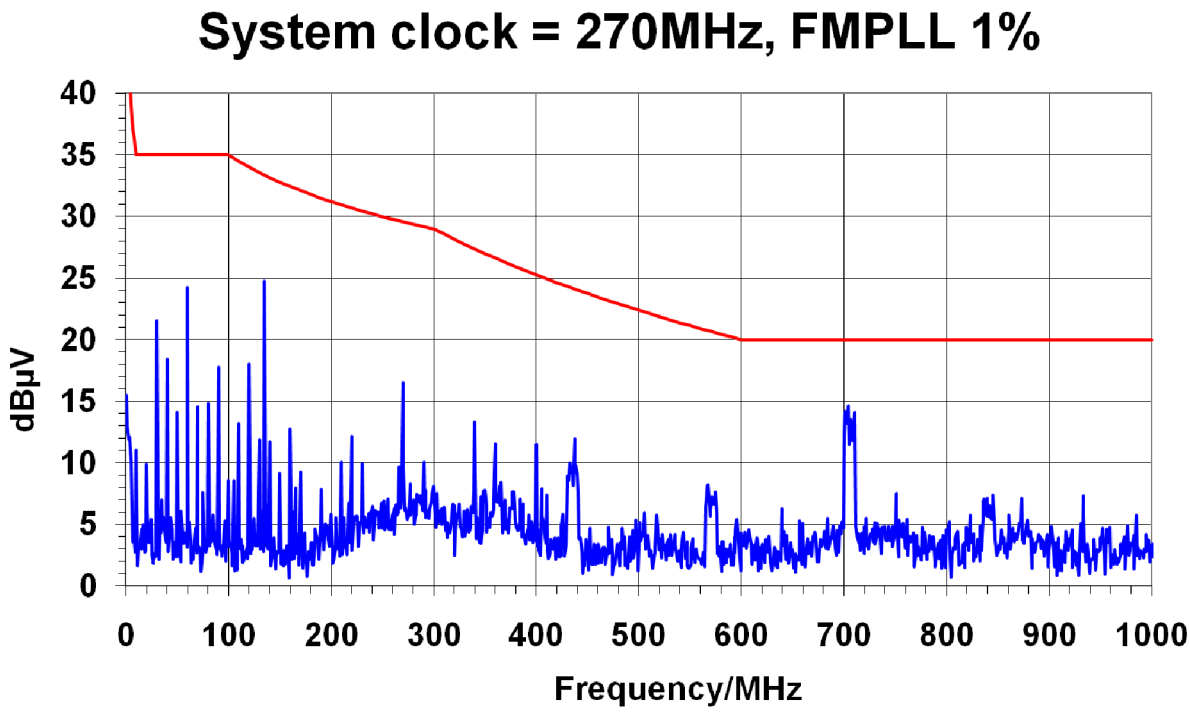


Figure 10: *Emission with f_{MOD}=100kHz and MA=1%*

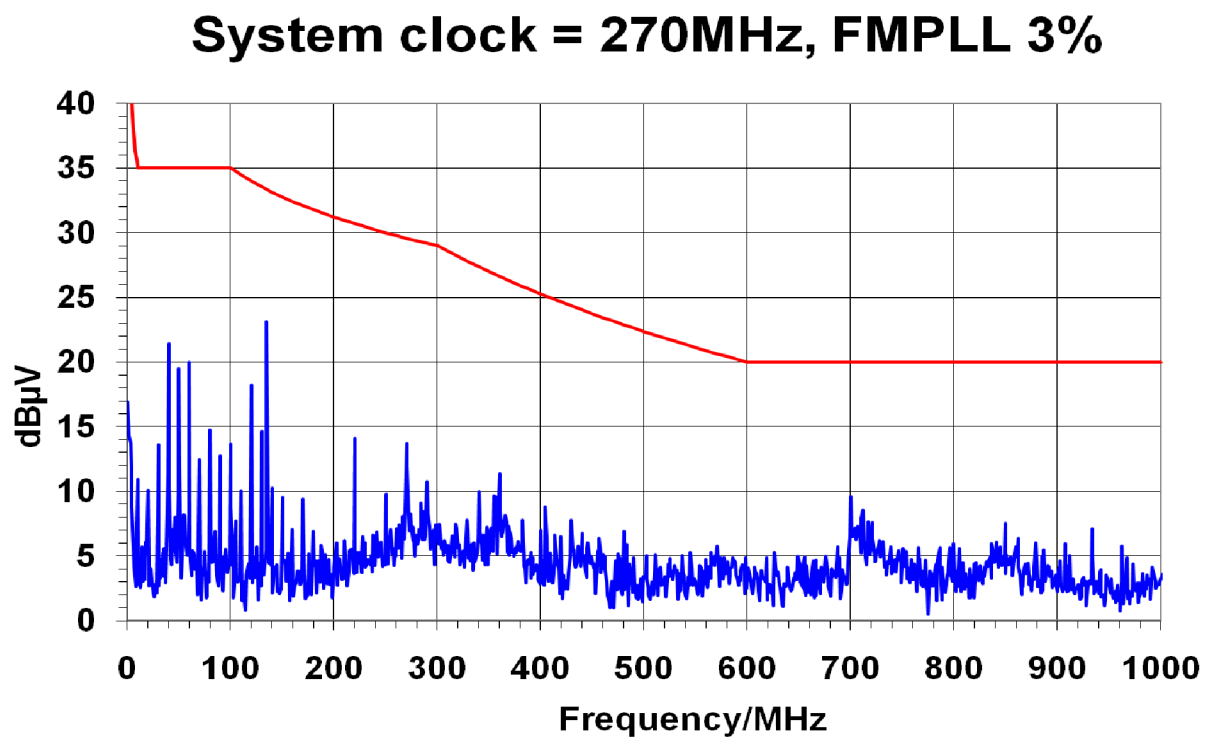


Figure 11: *Emission with f_{MOD}=100kHz and MA=3%*

Note:

[1] BISS = IC EMC test specification; download link: http://www.zvei.org/IC_EMC_Test_Specification

Even when considering different f_{MOD} settings (not shown in the figures), there is no significant difference caused by the modulation frequency. Only the modulation amplitude determines the emission reduction. Therefore, it is recommended to use f_{MOD} = 100 kHz. Already 1% modulation amplitude reduces the clock harmonics significantly. MA = 3% reduces the clock harmonics to or below the measurement noise floor.

Table 2 shows the absolute peak emission values for 0.5xf_{PLL}, f_{PLL}, 1.5xf_{PLL}, 2xf_{PLL}, 2.5xf_{PLL}, 3xf_{PLL} and 3.5xf_{PLL} as a function of MA:

Harmonic	Frequency [MHz]	Peak emission [dB]		
		FM off (MA=0%)	Clipped-FM MA=1%	Clipped-FM MA=3%
0.5 x f_{PLL}	135	31	25	23
f_{PLL}	270	26	17	14
1.5 x f_{PLL}	405	27	12	9
2 x f_{PLL}	540	28	8	6
2.5 x f_{PLL}	675	26	15	9
3 x f_{PLL}	810	27	7	7
3.5 x f_{PLL}	945	22	7	7

Table 2: *Emission peak values for the clock harmonics*

5 FMPLL parameter evaluation

5.1 Characterization scope

The modulation mode is clipped-FM.

All recommended system clocks for the AudoMax derivatives have been evaluated:

- TC1798: f_{PLL} = 300MHz nominal
- TC1793: f_{PLL} = 270 MHz nominal
- TC1791: f_{PLL} = 240 MHz nominal
- TC1791: f_{PLL} = 200 MHz nominal

The full specified process/voltage/temperature variations have been evaluated.

FMPLL settings:

- Unmodulated and clipped-FM
- f_{OSC} = 20 MHz, P = 2, N = 60/54/48, K2 = 2
- Modulation frequency f_{MOD}: 50 kHz, 100 kHz, 200 kHz
- Modulation amplitude MA: 0.5%, 1.0%, 1.5%, 2.0%, 2.5%, 3.0%[1]

The following parameters have been characterized:

- Mean system frequency accuracy
- Real mean system frequency offset
- Maximum modulation amplitude determined by maximum system frequency
- Real system frequency offset determined by Noffset
- Maximum system frequency determined by modulation amplitude
- Modulation frequency accuracy
- Modulation amplitude accuracy
- Accumulated jitter reduction by clipped-FM mode
- Absolute accumulated jitter

Finally, the parameter results are compared between nominal and worst-case activity at 300 MHz.

Note:

[1] The modulation amplitude MA = 3.0% is out of specification, but was considered in the validation to ensure reliable FMPLL operation over the full specified range up to MA = 2.5 %.

5.2 Real system frequency offset

The maximum offset for the real system frequency stays below 7%.

Conditions: clipped-FM, f_{MOD} = 50 kHz and MA ≤ 2.5 %.

The mean system frequency is the effective system frequency determined by the values of the P, N and K2 dividers in the PLL. The clock clipping circuit requires a higher “real” VCO frequency to be able to suppress clock pulses in a way that the resulting clock frequency matches the target frequency which is determined by the PLL dividers P, N and K2. This increased “real” system frequency leads to a limitation of the modulation amplitude for the TC1798 when it is operated faster than 285 or 290 MHz. For details see chapter 5.4.

5.3 Maximum modulation amplitude determined by system frequency

For system performances in the upper specification range of the microcontroller, Infineon specifies a maximum mean system frequency, e.g. 300 MHz for the TC1798. This frequency must not be exceeded in any application.

When the TC1798 is operated at f_{CPU} ≥ 290 MHz, the modulation amplitude is limited. Table 3 lists the maximum possible modulation amplitudes as a function of microcontroller products with their respective system frequency f_{CPU}. Note that the N divider value is related to a crystal f_{OSC} = 20 MHz, a P-divider of 2, and a CPU clock divider of 2. The setting for Noffset is described in Table 1 in chapter 3.

Product	System frequency f_{CPU}	N divider	Noffset	Maximum modulation amplitude
TC1791	240 MHz	48	4	2.5%
TC1793	270 MHz	54	4	2.5%
TC1798	300 MHz	60	3	0.8%
TC1798	300 MHz	60	4	No modulation allowed
TC1798	295 MHz	59	3	1.6%
TC1798	295 MHz	59	4	0,8%
TC1798	290 MHz	58	3	2.5%
TC1798	290 MHz	58	4	1,7%
TC1798	285 MHz	57	4	2,5%

Table 3: *Product- and clock-related maximum specified modulation amplitude*

Note: The limitation of the modulation amplitude for high system frequencies is caused by the fact that the maximum clock frequency which appears in reality is higher than the mean system frequency. The system frequency f_{CPU} listed in Table 3 is the mean system frequency provided by the clipped-FM operating mode. The maximum real system frequency is calculated by Equation 5b in chapter 2, considering an additional clock divider. Since clock clipping occurs in irregular intervals, several successive short clock pulses may occur. This fact may influence any system timing constraints.

5.4 Modulation frequency accuracy

Modulation frequencies 50 kHz and 100 kHz are reached with an accuracy better than 1%, whereas 200 kHz are reached with an accuracy better than 5%.

Conditions: clipped-FM, f_{MOD} = 50 kHz and MA ≤ 2.5 %.

The PLLCON2 register bit field for the modulation frequency f_{MOD} is calculated using the linear Equation 6a:

$$MODFREQ = \text{int}\left(\frac{5.14 \cdot MA \cdot N \cdot f_{MOD} \cdot P}{f_{OSC}}\right)$$

In reality, the behavior of the FMPLL is not linear over the full f_{MOD} range. This is indicated by an f_{MOD} shift for f_{MOD} values >100 kHz.

However, this effect is not rated critical because of the negligible EMI impact of small modulation frequency deviations. The clipped-FM specific accumulated jitter limitation is working independent from the modulation frequency, therefore no impact on the accumulated jitter exists.

5.5 Modulation amplitude accuracy

The real modulation amplitude is met with an accuracy better than 10%.

Conditions: clipped-FM, f_{MOD} = 50 kHz and MA ≤ 2.5 %.

The PLLCON2 register bit field for the modulation amplitude MA is calculated using the linear equation 6b:

$$MODAMP = \text{int}(161 \cdot MA \cdot N)$$

This amount of deviation is however not rated critical because of the negligible EMI impact of small modulation amplitude deviations.

The clipped-FM specific accumulated jitter limitation is working independent from the modulation amplitude, therefore no impact on the accumulated jitter exists.

5.6 Accumulated jitter in clipped-FM mode

Under all specified process/voltage/temperature conditions the accumulated jitter stays below 12.7 ns.

Conditions: clipped-FM, f_{MOD} = 50 kHz and MA ≤ 2.5 %.

The accumulated jitter of the conventional unclipped triangular modulation is calculated according equation 4:

$$J_{acc\_FM}\ [ns] = \frac{2500 \cdot MA\ [\%]}{f_{MOD}\ [kHz]}$$

This behavior was verified by measurement. J_{ACC} follows this equation with good accuracy in case of unclipped-FM mode – results are not shown in this application note. However, the additional noise-related jitter increases this value slightly. Using the clipped-FM mode significantly reduces the accumulated jitter significantly. J_{ACC} increases with higher modulation amplitude MA. The clipped-FM mode limits the accumulated jitter independent of the modulation frequency and the PLL output frequency f_{PLL}. Since the accumulated jitte increases with f_{MOD} for unclipped modulation, the relative J_{ACC} reduction in clipped-FM mode is higher for lower f_{MOD} values. This explains why for f_{MOD} = 50 kHz the J_{ACC} is reduced by ca. 95%, compared to ca. 90% for f_{MOD} = 100 kHz and ca. 80% for f_{MOD} = 200 kHz.

A quadratic fitting curve is introduced to the accumulated jitter measurement results. This upper limit curve for J_{ACC} is valid for all specified fab process variations, supply voltages and ambient temperatures. The fitting quadratic polynom calculates the maximum expected accumulated jitter as a function of the modulation amplitude MA.

The clipped-FM implementation causes J_{ACC} to increase with rising modulation amplitude. This effect is due to distributed clock clipping to avoid spurious emission peaks. The maximum expected J_{ACC} value as a function of the modulation amplitude MA can be calculated using a fitting quadratic polynom according to Equations 7a-c. Note that Equations 7a-c are valid for modulation amplitudes up to 3.0%; they are based on validation results.

Equations 7a-c correspond to different minimum microcontroller supply voltages:
VDD = 1.3 V - 10 | 5 | 2 % and VDDP = 3.3 V – 10 | 5 | 2 %.
Every 5% higher supply voltage leads to a decreased accumulated jitter of 200ps.
The value for MA must be given in [%]; the result for $J_{ACC\text{-}max}$ is in [ns].

$$J_{ACC\_max} < 3.9 + 4.5 \cdot MA - 0.4 \cdot MA^2; MA \le 3\%; VDDx = VDDx\_nom - 10\% \quad \text{(Equation 7a)}$$

$$J_{ACC\_max} < 3.7 + 4.5 \cdot MA - 0.4 \cdot MA^2; MA \le 3\%; VDDx = VDDx\_nom - 5\% \quad \text{(Equation 7b)}$$

$$J_{ACC\_max} < 3.6 + 4.5 \cdot MA - 0.4 \cdot MA^2; MA \le 3\%; VDDx = VDDx\_nom - 2\% \quad \text{(Equation 7c)}$$

The maximum accumulated jitter values for these supply voltage limitations are given in Figure 12.

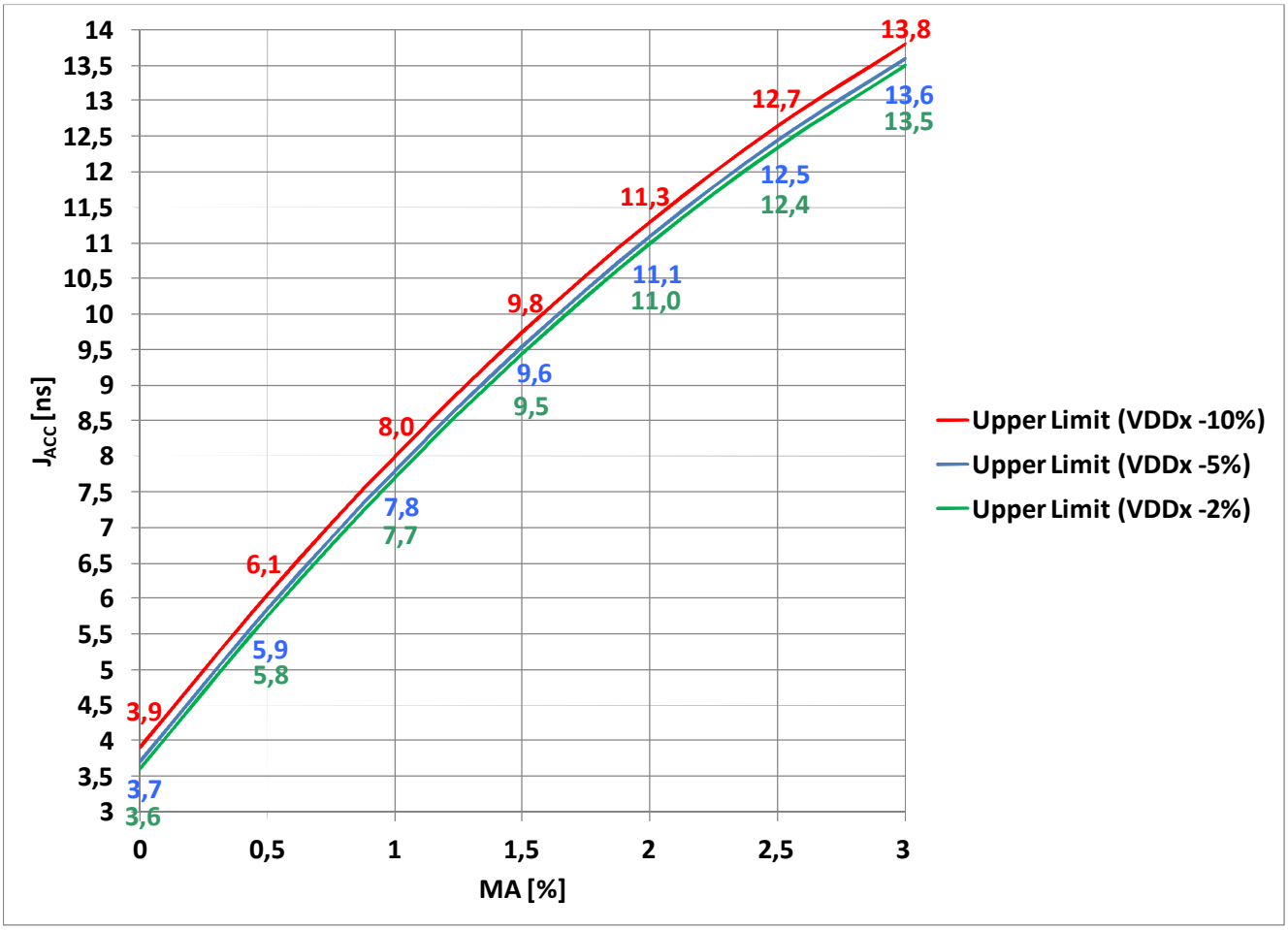


Figure 12: *2nd order fitting polynom for accumulated jitter in clipped-FM mode over the full spec range*

5.7 System frequency deviation for clipped-FM mode

Under all specified process/voltage/temperature conditions the mean system frequency deviation stays below 0.01%.

Conditions: clipped-FM, f_{MOD} = 50 kHz and MA ≤ 2.5 %.

The largest mean frequency deviation occurs for hot ambient temperature. Note that this value includes already the inaccuracy of the crystal time base.

The effective system frequency is determined by the values of the P, N and K2 dividers in the PLL. First of all, the accuracy of the mean system frequency is determined by the crystal accuracy. Second, the clock clipping circuit must suppress the exact number of clock pulses such that the real clock frequency is reduced exactly to the target clock frequency.

Example: For a 200 MHz unmodulated system clock, exactly 200 million clock pulses are expected to happen per second. In clipped-FM mode, the clock frequency is increased to 215 MHz, i.e. 215 million clock pulses are generated per second. From these 215 million pulses, exactly 15 million pulses must be suppressed by the clock clipping circuit. This measurement evaluates the deviation of the number of clock pulses per second in clipped-FM mode from the expected number of clock pulses per second.

The FMPLL's clock clipping circuit keeps track of the amount of pulses to clip and time slots for clipping. Thus it is interesting to validate the accuracy of the resulting mean system frequency. Since the clipping logic cuts the exact number of required clock pulses, the accuracy of the mean frequency is similar to the accuracy of an unclipped FMPLL.

5.8 Maximum Time Interval Error (MTIE)

Under all specified process/voltage/temperature conditions the worst-case MTIE for TC1793 stays below 12.7 ns. The contribution of a constant deviation between the real mean frequency and the target frequency to the MTIE stays below 1 ns.

Conditions: clipped-FM, f_{MOD} = 50 kHz and MA ≤ 2.5 %.

The worst-case MTIE is the MTIE for the data rate which is double the modulation frequency. Consequently, MTIE was calculated for 100 kHz data rate.

The clock frequency modulation results in an accumulated jitter. This accumulated jitter is a maximum limit for the MTIE value for any data rate. However, the MTIE for an actual data rate depends strongly on the modulation frequency. A data rate whose period covers 0.5, 1.5, 2.5 etc. of the modulation period will have the highest MTIE value. A data rate whose period covers one or multiple full modulation periods will have the lowest MTIE value (zero in ideal case without noise). For a 50 kHz modulation period, a 100 kHz data rate will have the highest MTIE. This effect is for example interesting for CAN communication – a 1 MBd CAN transfer rate together with clock synchronization after 10 bits results in a 10 µs synchronization interval, similar to a 100 kHz data rate.

Any mean frequency deviation less than 0.01% will result in an additional accumulated jitter of less than 1 ns. Validation measurements proved that the system frequency deviation stays below 900 ps.

6 Sample FMPLL register settings

To simplify FMPLL programming, a lookup table is provided for the following PLL settings:

- Target system frequency f_{CPU} = 200 MHz ; Crystal frequency f_{OSC} = 20 MHz
  Divider P = 2 ; Multiplier N = 40 ; CPU-Divider[1] = 2
- Target system frequency f_{CPU} = 240 MHz ; Crystal frequency f_{OSC} = 20 MHz
  Divider P = 2 ; Multiplier N = 48 ; CPU-Divider[1] = 2
- Target system frequency f_{CPU} = 270 MHz ; Crystal frequency f_{OSC} = 20 MHz
  Divider P = 2 ; Multiplier N = 54 ; CPU-Divider[1] = 2
- Target system frequency f_{CPU} = 300 MHz ; Crystal frequency f_{OSC} = 20 MHz
  Divider P = 2 ; Multiplier N = 60 ; CPU-Divider[1] = 2

Additional settings for TC1798:

- Target system frequency f_{CPU} = 295 MHz ; Crystal frequency f_{OSC} = 20 MHz
  Divider P = 2 ; Multiplier N = 59 ; CPU-Divider[1] = 2
- Target system frequency f_{CPU} = 290 MHz ; Crystal frequency f_{OSC} = 20 MHz
  Divider P = 2 ; Multiplier N = 58 ; CPU-Divider[1] = 2
- Target system frequency f_{CPU} = 285 MHz ; Crystal frequency f_{OSC} = 20 MHz
  Divider P = 2 ; Multiplier N = 57 ; CPU-Divider[1] = 2

Note:

[1] The CPU-Divider can either be realized by K2=2 (in register PLLCON1) and SRIDIV=1 (in register CCUCON0) or by K2=1 (in register PLLCON1) and SRIDIV=2 (in register CCUCON0), which is the normal application case. In Tables 4-18, K2=2 is selected to achieve a PLL output clock f_{PLL} which is similar to the desired CPU clock f_{CPU}. Therefore, SRIDIV is supposed to be 1.

Tables 4-18 list accordingly the recommended register contents for:

- PLLCON0
- PLLCON1
- PLLCON2
- Memory location [0xF000 0650$_H$][2]

for various combinations of:

- f_{MOD} = 50 / 100 / 200 kHz
- MA = 0.0 / 0.5 / 1.0 / 1.5 / 2.0 / 2.5 %

The register contents calculation is based on equations 6a and 6b. In addition, the expected maximum accumulated jitter values according to Equation 7a are listed.

Note:

[2] Referring to Table 1 in chapter 3, only the bits [16, 14:8] in memory location [0xF000 0650$_H$] should be re-programmed for proper FMPLL setting. All other bits should not be touched. The memory location [0xF000 0650$_H$] is ENDINIT protected.
In the Tables 4-18, the values given for the memory location [0xF000 0650$_H$] should be treated as follows:
There are only two cases for the memory content: 1) 0x???129??; 2) 0x???127??.
For case 1) this means: bits [31:17] untouched; [16]=1$_B$, [15] untouched, [14:8]=0101001$_B$, [7:0] untouched.
For case 2) this means: bits [31:17] untouched; [16]=1$_B$, [15] untouched, [14:8]=0100111$_B$, [7:0] untouched.
A proper programming of the memory location [0xF000 0650$_H$] can be done by ANDing all bit locations [16, 14:8] which should get a ‘zero’ value with 0$_B$. All bit locations [16, 14:8] which should get a ‘one’ value can be ORed with 1$_B$.
For memory content 0x???129?? the AND and OR sequence should be:
AND [0xF0000650$_H$],#0xFFFFA9FF;
OR [0xF0000650$_H$],#0x00012900;
For memory content 0x???127?? the AND and OR sequence should be:
AND [0xF0000650$_H$],#0xFFFFA7FF;
OR [0xF0000650$_H$],#0x00012700;

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	50	50	50	50	50
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	40	40	40	40	40	40
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**200**	**200**	**200**	**200**	**200**	**200**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?4E?0**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x02020C94**	**0x04041928**	**0x060525BC**	**0x08083250**	**0x0A0A3EE4**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 4: *FMPLL register settings for f_{CPU}=200 MHz, f_{MOD}=50 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	100	100	100	100	100
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	40	40	40	40	40	40
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**200**	**200**	**200**	**200**	**200**	**200**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?4E?0**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x04040C94**	**0x08081928**	**0x0C0B25BC**	**0x10103250**	**0x14143EE4**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 5: *FMPLL register settings for f_{CPU}=200 MHz, f_{MOD}=100 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	200	200	200	200	200
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	40	40	40	40	40	40
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**200**	**200**	**200**	**200**	**200**	**200**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?4E?0**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**	**0x?10?4E?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x08080C94**	**0x10101928**	**0x181725BC**	**0x20203250**	**0x28283EE4**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 6: *FMPLL register settings for f_{CPU}=200 MHz, f_{MOD}=200 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	50	50	50	50	50
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	48	48	48	48	48	48
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**240**	**240**	**240**	**240**	**240**	**240**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?5E?0**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x02680F18**	**0x04D11E30**	**0x073A2D48**	**0x09A33C60**	**0x0C0B4B78**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 7: *FMPLL register settings for f_{CPU}=240 MHz, f_{MOD}=50 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	100	100	100	100	100
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	48	48	48	48	48	48
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**240**	**240**	**240**	**240**	**240**	**240**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?5E?0**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x04D10F18**	**0x09A31E30**	**0x0E742D48**	**0x13463C60**	**0x18174B78**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 8: *FMPLL register settings for f_{CPU}=240 MHz, f_{MOD}=100 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	200	200	200	200	200
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	48	48	48	48	48	48
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**240**	**240**	**240**	**240**	**240**	**240**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?5E?0**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**	**0x?10?5E?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x09A30F18**	**0x13461E30**	**0x1CE92D48**	**0x268C3C60**	**0x302F4B78**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 9: *FMPLL register settings for f_{CPU}=240 MHz, f_{MOD}=200 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	50	50	50	50	50
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	54	54	54	54	54	54
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**270**	**270**	**270**	**270**	**270**	**270**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?6A?0**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x02B510FB**	**0x056B21F6**	**0x082132F1**	**0x0AD743EC**	**0x0D8D54E7**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 10: *FMPLL register settings for f_{CPU}=270 MHz, f_{MOD}=50 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	100	100	100	100	100
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	54	54	54	54	54	54
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**270**	**270**	**270**	**270**	**270**	**270**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?6A?0**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x056B10FB**	**0x0AD721F6**	**0x104332F1**	**0x15AF43EC**	**0x1B1B54E7**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 11: *FMPLL register settings for f_{CPU}=270 MHz, f_{MOD}=100 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1
f_{MOD}	kHz	0	200	200	200	200	200
MA	% +/-	0	0,5	1	1,5	2	2,5
Noffset	dec	0	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20
P	dec	2	2	2	2	2	2
N	dec	54	54	54	54	54	54
K2	dec	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**270**	**270**	**270**	**270**	**270**	**270**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**8**	**9,75**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?6A?0**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**	**0x?10?6A?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x0AD710FB**	**0x15AF21F6**	**0x208632F1**	**0x2B5E43EC**	**0x363654E7**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 12: *FMPLL register settings for f_{CPU}=270 MHz, f_{MOD}=200 kHz, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1	1
f_{MOD}	kHz	0	50	50	100	100	200	200
MA	% +/-	0	0,5	0,8	0,5	0,8	0,5	0,8
Noffset	dec	0	3	3	3	3	3	3
f_{OSC}	MHz	20	20	20	20	20	20	20
P	dec	2	2	2	2	2	2	2
N	dec	60	60	60	60	60	60	60
K2	dec	2	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**300**	**300**	**300**	**300**	**300**	**300**	**300**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**7,244**	**6,05**	**7,244**	**6,05**	**7,244**
PLLCON0	**hex**	**0x?10?76?0**	**0x?10?76?4**	**0x?10?76?4**	**0x?10?76?4**	**0x?10?76?4**	**0x?10?76?4**	**0x?10?76?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x030212DE**	**0x04D11E30**	**0x060512DE**	**0x09A31E30**	**0x0C0B12DE**	**0x13461E30**
0xF0000650H	**hex**	**n/a**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**

Table 13: *FMPLL register settings for f_{CPU}=300 MHz, Noffset=3, MA=0.0%..0.8%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1	1
f_{MOD}	kHz	0	50	50	100	100	200	200
MA	% +/-	0	1	1,6	1	1,6	1	1,6
Noffset	dec	0	3	3	3	3	3	3
f_{OSC}	MHz	20	20	20	20	20	20	20
P	dec	2	2	2	2	2	2	2
N	dec	59	59	59	59	59	59	59
K2	dec	2	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**295**	**295**	**295**	**295**	**295**	**295**	**295**
J_{ACC} Limit	**ns**	**3,9**	**8**	**10,076**	**8**	**10,076**	**8**	**10,076**
PLLCON0	**hex**	**0x?10?74?0**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x05EC251B**	**0x097A3B5E**	**0x0BD8251B**	**0x12F43B5E**	**0x17B1251B**	**0x25E83B5E**
0xF0000650H	**hex**	**n/a**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**

Table 14: *FMPLL register settings for f_{CPU}=295 MHz, Noffset=3, MA=0.0%..1.6%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1	1
f_{MOD}	kHz	0	50	50	100	100	200	200
MA	% +/-	0	2	2,5	2	2,5	2	2,5
Noffset	dec	0	3	3	3	3	3	3
f_{OSC}	MHz	20	20	20	20	20	20	20
P	dec	2	2	2	2	2	2	2
N	dec	58	58	58	58	58	58	58
K2	dec	2	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**290**	**290**	**290**	**290**	**290**	**290**	**290**
J_{ACC} Limit	**ns**	**3,9**	**11,3**	**12,65**	**11,3**	**12,65**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?72?0**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x0BA548F4**	**0x0E8E5B31**	**0x174A48F4**	**0x1D1D5B31**	**0x2E9448F4**	**0x3A3A5B31**
0xF0000650H	**hex**	**n/a**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**	**0x???327??**

Table 15: *FMPLL register settings for f_{CPU}=290 MHz, Noffset=3, MA=0.0%..2.5%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1	1
f_{MOD}	kHz	0	50	50	100	100	200	200
MA	% +/-	0	0,5	0,8	0,5	0,8	0,5	0,8
Noffset	dec	0	4	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20	20
P	dec	2	2	2	2	2	2	2
N	dec	59	59	59	59	59	59	59
K2	dec	2	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**295**	**295**	**295**	**295**	**295**	**295**	**295**
J_{ACC} Limit	**ns**	**3,9**	**6,05**	**7,244**	**6,05**	**7,244**	**6,05**	**7,244**
PLLCON0	**hex**	**0x?10?74?0**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**	**0x?10?74?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x02F6128D**	**0x04BD1DAF**	**0x05EC128D**	**0x097A1DAF**	**0x0BD8128D**	**0x12F41DAF**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 16: *FMPLL register settings for f_{CPU}=295 MHz, Noffset=4, MA=0.0%..0.8%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1	1
f_{MOD}	kHz	0	50	50	100	100	200	200
MA	% +/-	0	1	1,7	1	1,7	1	1,7
Noffset	dec	0	4	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20	20
P	dec	2	2	2	2	2	2	2
N	dec	58	58	58	58	58	58	58
K2	dec	2	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**290**	**290**	**290**	**290**	**290**	**290**	**290**
J_{ACC} Limit	**ns**	**3,9**	**8**	**10,394**	**8**	**10,394**	**8**	**10,394**
PLLCON0	**hex**	**0x?10?72?0**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**	**0x?10?72?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x05D2247A**	**0x09E63E02**	**0x0BA5247A**	**0x13CC3E02**	**0x174A247A**	**0x27983E02**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 17: *FMPLL register settings for f_{CPU}=290 MHz, Noffset=4, MA=0.0%..1.7%*

Phys. Param.	Unit	Value	Value	Value	Value	Value	Value	Value
Clipped-FM active	none	0	1	1	1	1	1	1
f_{MOD}	kHz	0	50	50	100	100	200	200
MA	% +/-	0	2	2,5	2	2,5	2	2,5
Noffset	dec	0	4	4	4	4	4	4
f_{OSC}	MHz	20	20	20	20	20	20	20
P	dec	2	2	2	2	2	2	2
N	dec	57	57	57	57	57	57	57
K2	dec	2	2	2	2	2	2	2
Mean f_{CPU}	**MHz**	**285**	**285**	**285**	**285**	**285**	**285**	**285**
J_{ACC} Limit	**ns**	**3,9**	**11,3**	**12,65**	**11,3**	**12,65**	**11,3**	**12,65**
PLLCON0	**hex**	**0x?10?70?0**	**0x?10?70?4**	**0x?10?70?4**	**0x?10?70?4**	**0x?10?70?4**	**0x?10?70?4**	**0x?10?70?4**
PLLCON1	**hex**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**	**0x??????01**
PLLCON2	**hex**	**n/a**	**0x0B7147B2**	**0x0E4E599E**	**0x16E347B2**	**0x1C9C599E**	**0x2DC747B2**	**0x3939599E**
0xF0000650H	**hex**	**n/a**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**	**0x???329??**

Table 18: *FMPLL register settings for f_{CPU}=285 MHz, Noffset=4, MA=0.0%..2.5%*

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG