# A Novel Method Improving Nonuniform Sampling Instant Placement

F. Papenfuß, D. Timmermann
Institute of Applied Microelectronics and Computer Engineering
College of Computer Science and Electrical Engineering, University of Rostock
Richard Wagner Str. 31, 18119 Rostock
GERMANY
frank.papenfuss@uni-rostock.de, http://www-md.e-technik.uni-rostock.de

*Abstract:* – In a world of fast evolving standards in wireless communication systems it is desirable to build flexible radio front–ends that have to be reconfigured rather then redesigned if a standard is changing or a new standard is introduced. This is the driving idea behind *software defined radio* (SDR). SDR systems today usually employ several parallel ADC sampling channels for direct down conversion to be able to process various radio channels at once (see e. g. [1]). Although greatly benefiting from the use of dedicated ASICs and FPGAs these systems are still very complex. An alternative approach to such architectures is offered by using a single (or at most dual) ADC direct down conversion system exploiting *nonuniform sampling* instead of uniform sampling. This enables *alias–free signal processing* in a frequency range determined by the systems time quantum. We briefly explain the motivation behind nonuniform sampling in our introduction. The challenge when sampling radio signals nonuniformly lies in the generation of precise, predictable (i. e. pseudo random) sampling instances. These instances have to be produced with picosecond precision like required in uniform undersampling applications mentioned above. Deliberate nonuniform sampling requires a *sampling driver* (SD) conveniently implemented in FPGAs. The FPGA has to be clocked by a low jitter clock generator (exposing 3 ps RMS cycle-to-cycle *jitter* or less). Such clock sources are not available for all frequency ranges and are often very expensive. Therefore it is inevitable to use a delay locked loop (DLL) or phase locked loop (PLL), respectively, to generate correct sampling pulse timings which unfortunately adds unacceptable degree of jitter to the sampling pulses.

In this paper we propose a remedy to the added DLL jitter. The proposed jitter reducing methodology is robust and will work in a variety of nonuniform sampling driver designs. The obtained results are verified by *jitter measurements* based on an algorithm introduced in [12].

*Key-Words:* – Software defined radio, Nonuniform sampling, alias-free signal processing, sampling theorem, radio signals, sampling system, sampling driver, jitter, jitter measurement

## 1 Introduction

A sampling driver is a device connected directly to the encode input of an ADC. In contrast to traditional sampling systems applying a fixed frequency clock to the ADC encode input our approach applies a randomised train of sampling pulses to the ADC. Assuming that the SD system clock period $T_{clk}$ is matched to the minimum ADC conversion time the alias-free bandwidth gain is given by

$$G_{BW} = \frac{T_{clk}}{T_q} \tag{1}$$

where $T_q$ is the time quantum realised using a DCDL. It is the least significant delay step (see [10] for details). Earlier publications [10,11] showed efficient architectures to construct a suitable sampling scheme modelled by (2) taking $N$ nonuniform samples. The statistical properties (i. e. PDF of sampling instances $\mathbf{t}_n$ and intersample intervals) of a randomised sampling scheme must be designed carefully to avoid spurious frequencies in the spectrum of randomly sampled signals (see [10] and [9]). The sampling instances $\mathbf{t}_n$ are random variables and are placed on a grid with spacing $T_q$.

$$\mathbf{s}\left(\omega^{(s)}, t\right) = \sum_{n=0}^{N-1} \delta\left(t - \mathbf{t}_n\right) \tag{2}$$

The main goal is to generate a suitable *sampling point density function* (SPDF) $D_s(t)$. $D_s(t)$ possesses unit $[1/s]$ and its magnitude is determined by the probability that time point *t* will become a sampling instance. A suitably constructed random sampling scheme will possess a constant SPDF with a value given by $T_q/T_m$ (where $T_m$ is the mean sampling period). This implies that the sampling process (2) has a constant mean sampling rate $f_m = 1/T_m$, a valid assumption for a SD system with deliberate randomised sampling. For a more

complete coverage of nonuniform sampling and the details on how to construct a suitable nonuniform sampling scheme for alias suppression see [2-4,6,7,13]. A favourable scheme producing a constant SPDF is *additive random sampling* (ARS) explained in [3,15]. It is implemented by the architecture referred to in this paper.

In an actual implementation of a nonuniform sampling system an ARS scheme will be applied to a RF signal $x_{RF}(t)$. The only condition that the signal has to fulfil is that the sum of all the bands in which it is present must not exceed $[0, f_m/2]$. These subbands may be distributed arbitrarily within the alias free range $f \in [0, f_q/2]$ as depicted in Fig. 1b. This distinguishes nonuniform from uniform sampling. When sampling uniformly the signal's allowed total bandwidth is $[0, f_{clk}/2]$ with $f_{clk}$ being the clock frequency of the uniform sampling system. If $X_{RF}(f)$ is chosen to be present above $f_{clk}/2$ then this is referred to as undersampling. Bands possessing the same alias inside the baseband have to be avoided, as they cannot be distinguished. Fig. 1a illustrates this. It gives an example where band $B_3$ marked with X cannot be chosen because it would possess the same baseband alias as $B_2$.

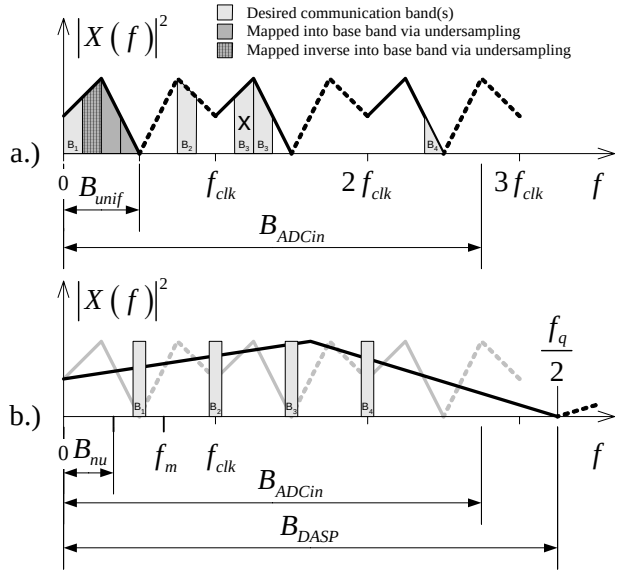


Fig. 1: Placement of subbands in a system utilising a.) uniform (under)sampling and b.) nonuniform additive random sampling ($B_{ADCin}$ – analogue bandwidth of used ADC, $B_{unif}$ – total bandwidth provided by uniform sampling, $B_{nu}$ – total bandwidth provided by nonuniform sampling, $B_{DASP}$ – digital alias–free signal processing (DASP) bandwidth usable by subbands of $B_{nu}$).

Thus, for a fixed sampling frequency $f_{clk}$ there are always a lot of invalid subband combinations for $X_{RF}(f)$ which in a world of an ever growing demand for bandwidth and demand to dynamically configure the radio front–end of a communication system represents a serious limitation in terms of configurability. The advantage of using nonuniform sampling as opposed to uniform undersampling is obvious. In a radio front–end of software defined radio systems nonuniform sampling gives more flexibility to select multiple bands in a reconfigurable RF system.

It should be noted that in the case of nonuniform sampling the subbands, as indicated in Fig. 1b, when spaced uniformly along the frequency axis, can be used to analyse a periodic signal. The harmonic components of which in the case of uniform sampling would alias into baseband $B_{unif}$. This technique is utilised in sampling scopes to analyse wideband periodic signals (described e. g. in [14]). Herley [5, p. 1562] defines a measure of *sampling efficiency* (SE) using a partitioning into $N_B$ subbands as

$$SE = \frac{2\sum_{i=0}^{N_B-1} \tilde{B}_i}{2\sum_{i=0}^{N_B-1} B_i} = \frac{B_{eff}}{B_{max}}. \qquad (3)$$

Where $B_{eff}$ is the effective bandwidth of the signal and $B_{max}$ the maximum allowable bandwidth according to the sampling theorem with $B_{max} = \frac{1}{2} f_m$ in the nonuniform and $B_{max} = \frac{1}{2} f_{clk}$ in the uniform case, respectively. The notion $B_i$ and $\tilde{B}_i$ refer to the width of the *i*th subband reserved for analysis and the portion of the *i*th subband actually occupied by the signal, respectively. Thus $B_i \geq \tilde{B}_i$ and $SE \leq 1$ always hold and, as long as $f_m$ can be tailored to the effective bandwidth of the analysed signal by a nonuniform sampling driver, achievable sampling efficiencies for nonuniform sampling prevail over the more restricted case of uniform sampling as long as placement of sampling instants is at least as accurate as in the uniform undersampling case.

## 2 Sampling Instance Creation

The sampling point creation process is implemented using a SD comprising from a control unit, a pseudo random number generator (PRNG) and a digitally controllable delay line (DCDL). The architecture we use is sketched in Fig. 2.

The DCDL is the most vital part of the SD since it realises the time quantum step $T_q$. PRNG and control unit reside in a FPGA but for the DCDL an external specialised chip is used. It would be beneficial to have the DCDL integrated onto the SD chip but such functionality is currently not available in FPGA chips. The sampling instant creation algorithm is

implemented in the control unit as a finite state machine (FSM). It can be summarised as

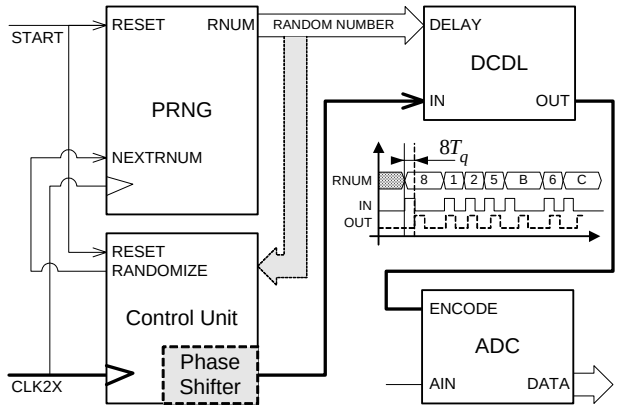


Fig. 2: Fundamental structure of an ARS sampler based on a random number generator coupled with a phase shifter.

given in Table 1. Actually, the algorithm in Table 1 is a little more complicated because in our prototype the attached ADC is a pipelined design and needs maintenance pulses

| | |
|---|---|
| 1. | If *start* signal is or becomes asserted issue a pulse to the DCDL, instruct PRNG to generate next random number (will occur on next rising clock edge). |
| 2. | If MSB of random number is asserted insert a 180° phase shift (effectively leaving out one clock cycle for the sampling pulse to follow). |
| 3. | Start over with first step. |

Table 1: Sampling instance creation algorithm.

while the SD is in idle state. However, this is left out here to focus on nonuniform sampling instance creation.

### *2.1 Sampling Pulse Jitter*

It has been pointed out that nonuniform sampling is most reasonably applied to direct digital signal processing of radio signals. Our reference design therefore uses a time quantum $T_q$ of 625 ps delivering an alias-free bandwidth of 800 MHz suitable to directly scan a larger portion of the RF spectrum by digital means. The bandwidth of 800 MHz is well tuned to the analogue bandwidth of the used ADC (AD9433) of 750 MHz. The time quantum of 625 ps represents a challenge in terms of chip and PCB design. Hence, the sampling system must be carefully constructed with regard to signal integrity issues. The jitter sensitive path connecting the components involved (i. e. FPGA, DCDL and ADC) is drawn bold in Fig. 2. Actually, the path from the clock generator to the FPGA is involved too but not covered by Fig. 2. We use a low jitter (≤ 1 ps RMS), 100 MHz oscillator to clock the FPGA. But as indicated in Fig. 2 (CLK2X) this clock has to be doubled inside the FPGA to drive the FSM of the sampling driver. Two standard approaches exist to achieve such clock doubling; either through use of a DLL or PLL, respectively. We use a digital clock management (DCM) unit featuring a DLL.

The random numbers along with the sampling pulses coming from the control unit are passed to the DCDL. The DCDL delays incoming pulses according to the applied random number and eventually drives the ADC encode input. The ideal delay step width of 625 ps and integer multiples of it can practically not be realised due to fluctuations in the DCDL delay steps caused by the chip production process. Therefore, a calibration procedure is vital to determine the true delay step values of a particular DCDL. Unfortunately delay step values are also temperature dependent. The errors of the sampling instants produced by the SD can thus be partitioned into two categories: First, systematic errors, delay fluctuations due to manufacturing deviations on a per DCDL chip basis. Errors due to temperature drift also fall into this category. Second, random errors, these comprise from random shifts on a per cycle basis stemming from different error sources. This error is usually referred to as cycle-to-cycle jitter. Jitter coming from the clock source and jitter added by the FPGA fall into this category.

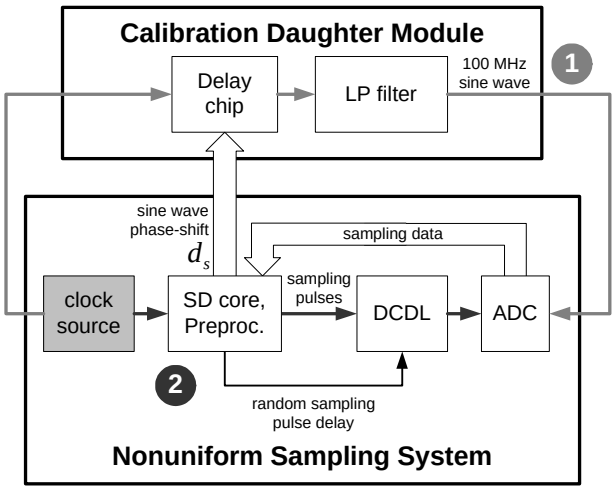


Fig. 3: Sampling driver with calibration module.

In order to quantify both systematic and random errors we developed a calibration algorithm presented in [12]. This algorithm uses the setup shown in Fig. 3 requiring an additional calibration daughter module attached to our digitiser. The advantage of such a structure is that the clock which the ADC uses for encoding as well as the test signal

are derived from the same clock source via the two paths labelled ❶ and ❷ in Fig. 3, respectively. Therefore, both signals possess the same medium and long-term phase drift (often referred to as wander) seen over several hundreds or thousands of clock cycles. This is important because it eliminates errors due to asynchronous phase drift when measuring in picosecond range.

The sampled calibration sine wave is depicted in Fig. 4. One would expect a flat line with some added noise since the SD is working in uniform mode when calibration data is gathered. As is clearly observed, besides the expected noise, there are odd transitions. It looks as if the signal makes sudden jumps. More precisely, it is like it would not be exactly the sampled sine wave but the sine wave plus some added signal. As it turns out the deviations are produced due to the clock phase corrections carried out by the digital clock management unit (DCM) inside the FPGA.

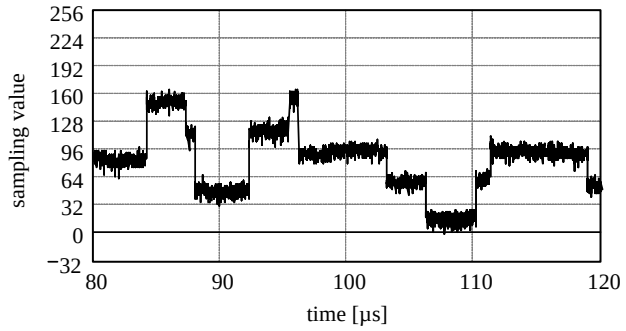


Fig. 4: Excerpt from DCM "corrupted" data, 100 MHz sine wave test signal sampled uniformly at 100 MSPS.

This unit is responsible for creating the FPGA internal phase aligned 200 MHz and 100 MHz clock signals. We use a model of the calibration signal $x_c$

$$x_c(t) = \hat{a}_c \cos\left(2\pi f_c t + \varphi_c(d_s)\right) + \hat{o}_c \qquad (4)$$

were $\hat{a}_c$ represents the estimated amplitude and $\hat{o}_c$ being the offset estimate, respectively. The actual delay steps of the DCDL being part of the nonuniform sampling system can thus be determined by statistical means. The phase of the calibration signal is a function of the delay value $d_c$ passed to the delay chip on the calibration module. A sampling pulse phase shift of the SD between two consecutive sampling pulses is, assuming that $d_s$ remains unchanged, given by

$$\Delta\hat{t} = \frac{1}{2\pi f_c}\left(\arccos\left(\frac{x_2 - \hat{o}_c}{\hat{a}_c}\right) - \arccos\left(\frac{x_1 - \hat{o}_c}{\hat{a}_c}\right)\right). \qquad (5)$$

With estimates $\hat{a}_c = 1807$ and $\hat{o}_c = -5.2$ obtained from delay determination as well as $\max\{x_2\} = 174$ and $\min\{x_1\} = -4$ obtained from a representative data set we obtain a maximum phase shift estimate of $\Delta\hat{t} = 157$ ps for our prototype. This is well within the specified value of ±150 ps provided in [16, M3, p.32], yet, represents a serious source of sampling instance placement error (about 25% of the time quantum we use). Though most of the time the shift will be less than this maximum the whole measurement is contaminated by this random error source. In fact in a lot of data sets gathered with our prototype board it is evident that the DCM unit produces a phase modulation frequency of around 125 kHz. Phase jumps with same polarity are approximately 8 µs apart. This is reflected in the spectrum by the presents of two sidebands ±125 kHz accompanying the fundamental of our calibration signal. The resulting spectrum is depicted in Fig. 5.

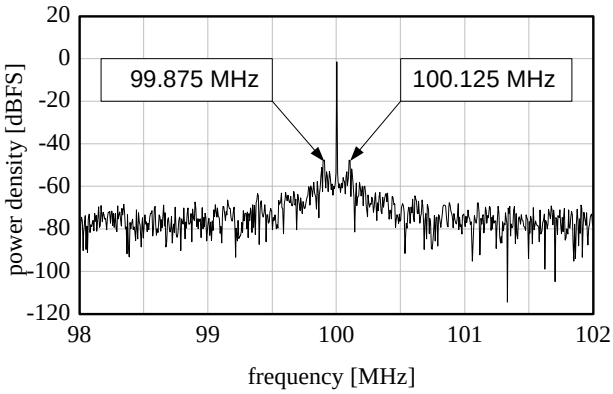


Fig. 5: PSD of 100 MHz calibration signal sampled nonuniformly while DCM jitter is present. The 125 kHz modulation side bands are easily spotted. Magnitude of the side lobes varies with degree of DCM phase correction present during sample acquisition.

Note that the spectrum of the calibration signal is only revealed using advanced signal processing

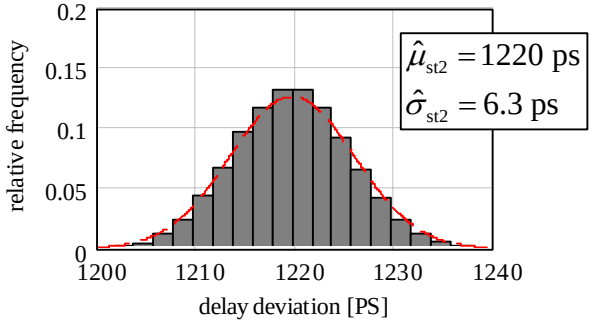


Fig. 6: Statistics of DCDL step delay for tap delay two whith DCM jitter present at the ADC *ENCODE* input. Statistic estimates were obtained from 65536 samples.

tailored to nonuniform sampled data such as the *sequential component extraction* (SeCoEx) algorithm

described in [8] and also [12]. Only a selected portion of interest is show in Fig. 5. In Fig. 6 we present representative statistics of DCDL delay when digital code two is applied to the *DELAY* tap. Standard deviations of other delay steps are around 6 ps too.

## 3 Sampling Instance Jitter Reduction

Clearly, for any serious measurement the error source presented in the previous section has to be suppressed as much as possible. We therefore propose a simple yet effective circuitry as a remedy to the jitter problem depicted in Fig. 7.

After the SD core has generated the sampling pulses they should be passed through a "clock gate". The idea is that the high precision clock that entered the FPGA will also be used to clock this „clock gate" when pulses are leaving the chip.

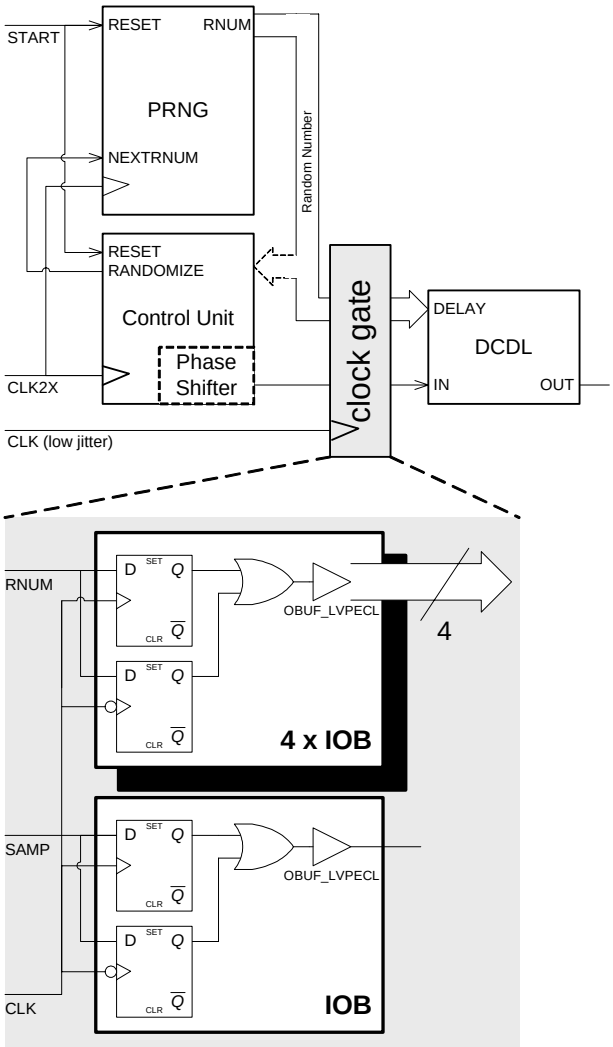


Fig. 7: DDR solution to DCM sampling jitter corruption.

This will only work if the clock signals CLK and CLK2X (see Fig. 7) are aligned. This is true for the used DCM. But there is another problem which waiting to be solved. The original system clock is 100 MHz yet the SD clock is 200 MHz and edges appear on a 5 ns grid. Thus, when clocking the "clock gate" with only 100 MHz, sampling pulses will eventually get lost invalidating the whole sampling scheme!

The perturbation presented in Fig. 4 can be overcome using *double data rate* (DDR) flip flops (see e. g. [16, M2, p.3]). DDR flip flops are realised using two interleaved signal paths for one output. The delays for both paths are matched with high precision in the *input output blocks* (IOBs) of the FPGA. DDR flip flops can be instantiated as black box components and serve as clock gate virtually operating at the required rate of 200 MHz. The double data rate components have to be inferred at the IOBs of the FPGA to ensure optimal alignment of edges between signals leaving the FPGA chip (single bits of the random number as well as the sampling pulses). Fig. 8 shows the 100 MSPS

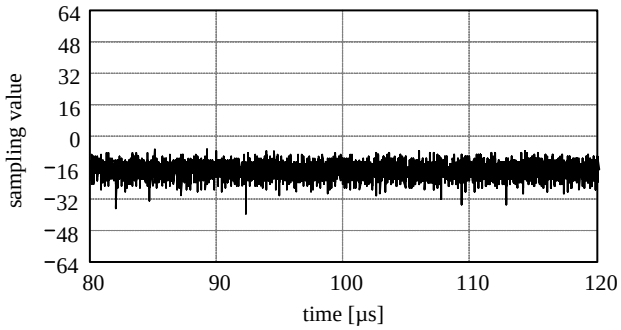


Fig. 8: Excerpt from sampling data, 100 MHz sine wave test signal sampled uniformly at 100 MSPS. The DCM jitter evident in Fig. 4 has completely been canceled out.

uniform sampled test signal after the described modification was made to the SD VHDL design. As is easily observed the corruption by the added DCM jitter has been removed completely. Fig. 9

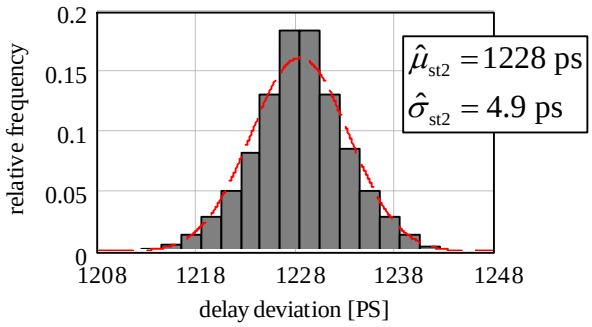


Fig. 9: Statistics of DCDL step delay for tap code two when DCM jitter is eliminated. Statistic estimates were obtained using 65536 samples.

shows the impact on the statistical estimation. Given the fact that the precision of delay estimation is already high with respect to the used time quantum the relative reduction of the delivered standard deviation by 22% is impressive. The improvement is also evident in the spectrum of the nonuniform sampled calibration signal as Fig. 10 reveals. Modulation side lobes present in Fig. 5 have disappeared all together.

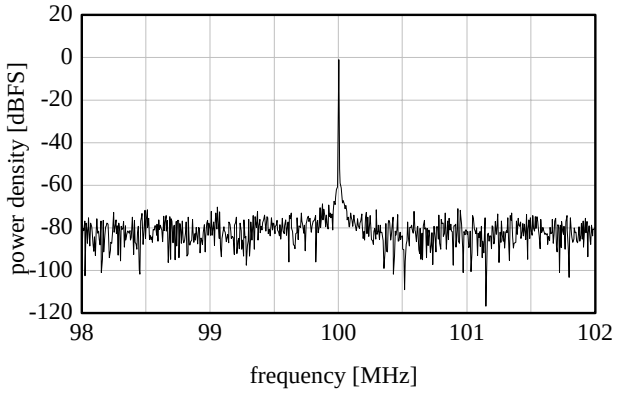


Fig. 10: PSD of the 100 MHz calibration signal (only selected section around fundamental displayed) sampled nonuniformly with $f_m = 66MHz$. Data gathered after DCM jitter had been eliminated.

## 4 Conclusions

If nonuniform sampling techniques are to make their way into mainstream designs then one utterly important component is the calibration module (see Fig. 3). Such a module was introduced in [12]. It enables estimation of the actual delay steps of the sampling driver. For this algorithm to work most effectively it is vital that the SD design presents a pseudo-randomised sampling to the ADC with the least jitter possible. It has been found by implementation of a prototype board featuring a Xilinx Virtex-II FPGA that a considerable source of cycle-to-cycle jitter is the internally provided digital clock manager (which is using a DLL to control the frequency of the internal clock/clocks). The DCM is provided to perform certain tasks with regard to the system clock like clock doubling, phase shifting etc. In our sampling driver design we require a doubled system clock in order to produce the nonuniform sampling scheme that will remove alias or spurious spectral content up to the equivalent frequency $f_q$. The benefit of sampling nonuniformly is a considerably increased freedom to choose direct digital processing bands for direct processing of radio signals e. g. inside SDR applications.

What we have shown in this paper is that a “clock gate” can be used efficiently to remove jitter added by an internal DLL. The proposed methodology is applicable also when using a PLL for clock doubling. The proposed method enabled us to reduce the standard deviation of our delay step determination algorithm by as much as 22%.

This leads to the conclusion that a “clock gate” should always be used for SD chip generated sampling pulses when clock doubling techniques are used involving clock edge controlling elements such as a DLL or PLL.

## 5 References


[1] A. Avanzado, *Model 6230 - 32-Ch. Narrowband Receiver with A/D and FPGA - VIM-4.* Pentek. Jun. 15, 2006. http://www.pentek.com/products/Products.cfm

[2] I. Bilinskis and I. Mednieks, *Introduction to digital alias-free signal processing*, Riga : Institute of Electronics and Computer Science, 2001.

[3] I. Bilinskis and A. Mikelsons, *Randomized Signal Processing*, Prentice Hall International, 1992. ISBN: 0137510748.

[4] D. M. Bland and A. Tarczynski, “Optimum nonuniform sampling sequence for alias frequency suppression,” in *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on*, Hong Kong, vol. 4, pp. 2693-2696, Jun, 1997.

[5] C. Herley and P. W. Wong, “Minimum rate sampling and reconstruction of signals with arbitrary frequency support,” *Information Theory, IEEE Transactions on*, vol. 45, no. 5, pp. 1555-1564, 1999. ISSN: 0018-9448.

[6] F. A. Marvasti, *Nonuniform sampling, theory and practice*, New York : Kluwer Academic, 2001. ISBN: 0306464454.

[7] I. Mednieks, “Methods for spectral analysis of nonuniformly sampled signals,” in *Sampling Theory and Applications, 1999. SAMPTA '99., Proceedings of the International Workshop on*, Loen, Norway, pp. 190-193, Aug, 1999.

[8] A. Mikelsons, “Alias-free spectral estimation of signals with components of arbitrary frequencies,” in *COST 229 Workshop*, Ljubljana, Slovenia, pp. 105-108, Apr, 1994.

[9] F. Papenfuss , Y. Artyukh, E. Boole, and D. Timmermann, “Optimal sampling functions in nonuniform sampling driver designs to overcome the Nyquist limit,” in *Acoustics, Speech, and Signal Processing (ICASSP), Proceedings of the IEEE*

*International Conference on*, Hong Kong, China, vol. 6, pp. 257-260, Apr, 2003.

[10] F. Papenfuss, Y. Artyukh, E. Boole, and D. Timmermann, "Nonuniform sampling driver design for optimal ADC utilization," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, Bangkok, Thailand, vol. 4, pp. 516-519, May, 2003.

[11] F. Papenfuss, R. Hecht, and D. Timmermann, "Scalable VHDL architectures for non-uniform sampling driver design," in *Proceedings of the 8th Biennial Baltic Electronic Conference, BEC2002*, Tallinn, Estonia, pp. 249-252, 2002.

[12] F. Papenfuss, F. Olbricht, Y. Artyukh, and D. Timmermann, "A Calibration Scheme for a Nonuniform Sampling Driver Architecture," in *Global Embedded Signal Processing & Expo Conference (GSPX), Proceedings of the* , Santa Clara, CA, 2004.

[13] A. Tarczynski, D. M. Bland, and T. I. Laakso, "Spectrum estimation of non-uniformly sampled signals," in *Industrial Electronics, 1996. ISIE '96., Proceedings of the IEEE International Symposium on*, Warsaw, Poland, vol. 1, pp. 196-200, Jun, 1996.

[14] Tektronix, Inc., "Sampling oscilloscope techniques," *Technique Primer TEK47W-7209*, 1989. Internet.

[15] J. J. Wojtiuk, *Randomised sampling for radio design.* Mawson Lakes, Univ. of South Australia. [online], Mar, 2000. http://www.itr.unisa.edu.au/~steven/thesis/jjw2.ps.gz.

[16] Xilinx, Inc., *DS031: Virtex-II Platform FPGAs Advance Product Specification (Complete).* rev. 1.9, Xilinx, Inc. Apr. 15, 2003. last update: Dec. 28, 2002. www.xilinx.com