International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 3, June 2017, pp. 1316~1325
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i3.pp1316-1325

High Speed Under-Sampling Frequency Measurements on FPGA

Seyed Ehsan Yasrebi Naeini, Ali Maroosi
Departement of Electrical and Computer Engineering, University of Torbat Heydarieh, Iran

Article Info

Article history:

Received Feb 10, 2017
Revised May 6, 2017
Accepted May 21, 2017

Keyword:

Chinese reminder theorem
Field programmable gate arrays
Frequency measurement
Signal under-sampling
Sub-nyquist sampling

ABSTRACT

A Sampling rate is less than Nyquist rate in some applications because of hardware limitations. Consequently, extensive researches have been conducted on frequency detection from sub-sampled signals. Previous studies on under-sampling frequency measurements have mostly discussed under-sampling frequency detection in theory and suggested possible methods for fast under-sampling frequencies detection. This study examined few suggested methods on Field Programmable Gate Array (FPGA) for fast under-sampling frequencies measurement. Implementation of the suggested methods on FPGA has issues that make them improper for fast data processing. This study tastes and discusses different methods for frequency detection including Least Squares (LS), Direct State Space (DSS), Goertzel filter, Sliding DFT, Phase changes of Fast Furrier Transform (FFT), peak amplitude of FFT to conclude which one from these methods are suitable for fast under-sampling frequencies detection on FPGA. Moreover, our proposed approach for sub-sampling detection from real waveform has less complextity than previous approaches from complex waveform.

Corresponding Author:

Seyed Ehsan Yasrebi Naeini,
Departement of Electrical and Computer Engineering,
University of Torbat Heydarieh,
Torbat Heydarieh, Iran.
Email: e.yasrebi@torbath.ac.ir

1. INTRODUCTION

Because of hardware limits, the sampling rate should be below the Nyquist rate in some applications. In these cases, the input frequencies should be determined from their sub-sampled waveforms. The analog to digital converters (ADCs) and other processing parts can work at low sampling rates. Thus, the cost is low, and management and processing of the sampled data are easier [1], [2]. Determination of frequency from its multiple under-sampled waveforms has been considered for various applications, such as sensor networks [2], [3] and phase unwrapping [4]. In addition, it has been used in synthetic aperture radar (SAR) imaging of moving targets [5]. It has been used for wide frequency band, for which the available band frequency is greater than the sampling rate [6]. Many attempts have been made to use the Chinese reminder theorem that reconstructs a large integer from its remainder modules for under-sampling frequency detections [7-9]. The mentioned approaches discuss estimation frequencies from under-sampled complex waveforms. In an earlier work, we introduced a formula to determine frequencies from under-sampled real waveforms [1].

For high-speed processing of signals, it is needed to use embedded systems like the FPGA. This study adopted the under-sampling frequency detection on the FPGA. Some experimental notes for implementation are discussed in this paper. In this work, different methods for realization of under-sampling frequency detection were tested on the FPGA and the most appropriate one was chosen. These notes can also be useful in other similar implementations, such as filter implementation on the FPGA.

2. PROBLEM STATEMENT

Assume there is a signal with frequency F_k, complex waveforms ($S_k(t) = A_k e^{j2\pi F_k t+\varphi}$) or real waveforms ($S_k(t) = A_k \cos(2\pi F_k t + \varphi)$ [1], where A_k is the amplitude of signal with frequency F_k and phase ϕ. However, the proposed implementation of under-sampling frequencies detection on the FPGA is useful for both real waveforms [1] and complex waveforms [8]. Here, we present the formula for real waveforms. This signal is sampled with p ADCs (sensors) at rate $f_{s1}, f_{s2},...,f_{sp}$ that signal frequency may be less than all sampling frequencies, that is $f_{si} < 2F_k$, $i = 1,...,p$ (multiple sampling rates are below the Nyquist rate). If sampling frequencies are chosen properly, the unambiguous analog frequency estimation $\hat{F}_k$ is achieved.

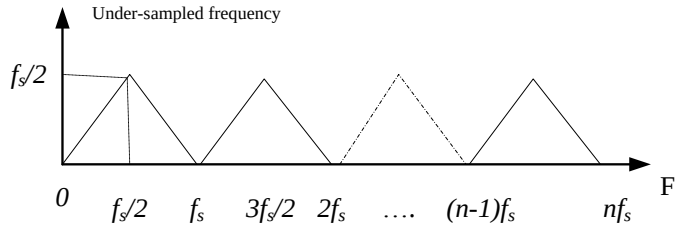

Figure 1. Aliased frequency as a function of analog frequency after sampling at f_s

According to Figire 1, the relationship between frequency of under-sampled waveform f_u ($f_{undersampling}$) and analog frequency F can be obtained as follows:

$$f_u = F - mf_s \text{ or } f_u = -F + nf_s \quad (1)$$

In (1), m and n are integer numbers and f_s is sampling frequency.

This study aimed to understand how to implement sub-Nyquist frequency detection approach on the FPGA to find a frequency more than 1 *GHz*. Thus, we needed a high-speed processing approach (around 200 *MHz*) with low use of FPGA resources. Usually, when a signal is processed, many parameters should be extracted, and frequency is among them. Hence, resources of FPGA should be reserved for other required processing (besides frequency measurement) on the received signal as much as possible.

3. TESTING AND DISCUSSION OF DIFFERENT METHODS ON FPGA

Frequency estimation plays an important role in many digital signals processing applications. There are different methods for frequency estimation, such as the Discrete Fourier Transform (DFT), the Least Squares (LS), and the Direct State Space (DSS) [10], [11] that are applied in sequential platforms like microprocessors. Different methods of under-sampling frequencies detection on the FPGA have not discussed in previous studies that we are tested and discussed below.

3.1. Limitation of DSS and LS methods on FPGA

The DSS and LS methods appear to work better than the DFT algorithm since the algorithm does not have discrete bin sizes. But the computational intensity creates a challenging practical real time; therefore, we did not use these two methods.

3.2. Limitations of Goertzel filter and Sliding DFT on FPGA based on our testing

Several properties of the DFT make it suitable for a parallel implementation. There are different kinds of DFT realization, such as the Goertzel filter, the Sliding DFT (SDFT) and the Fast Fourier Transform (FFT) and those related to the Fourier Transform. We compared several methods of Fourier Transform for extracting frequencies of signals, and chose the method that occupied small area in the FPGA (gates) and yielded an accurate frequency, while kept the processing speed high.

The Goertzel filter is typically implemented as a second-order IIR band pass filter [12]. The Goertzel algorithm can extract arbitrary frequency components from a given signal. Transformation of the output response for the Goertzel algorithm is as:

$$H_k(z)=\frac{1-e^{j2\pi k/N}\,z^{-1}}{1-2\cos(\frac{2\pi k}{N})z^{-1}+z^{-2}};k=1,\ldots,N \tag{2}$$

The following Equations can be derived from (2):

$$\begin{aligned} v[n] &= x[n]+2\cos(2\pi k/N)v[n-1]-v[n-2], \quad n=1,\ldots,N \\ y_k[n] &= v[n]-e^{j2\pi k/N}x[n-1] \end{aligned} \tag{3}$$

The Goertzel filter is impressive for fast processing implementation, because regardless of N, it requires a constant number of operations to compute a successive DFT output. The filter can be realized without input buffering, because each sample can be processed as received. We can relate the DFTs of two successive windowed sequences, each of length N as the Sliding DFT (SDFT). The SDFT is appropriate, because regardless of N, it requires a constant number of operations to compute a successive DFT output [13], two real adds and one complex multiply from the following Equation:

$$y_k(n)=[y_k(n-1)-x(n-N)+x(n)]e^{j2\pi k/N} \tag{4}$$

We tried to implement the SDFT and the Goertzel filter techniques (with similar structures) on the FPGA by the Xilinx ISE software. It seems that implementation of these two techniques requires small quantity of gates in the FPGA. However, we encountered some problems, solving of which increased usage of gates. Thus, amount of gates usage makes the SDFT or the Goertzel filter impractical. The following discusses problems in implementation of the SDFT or Goertzel filter on the FPGA.

The quantization effects in digital filters realized with fixed-point arithmetic were analyzed by Beraldin and Steenaart (1989) [14]. Due to the coefficient quantization, the poles and zeros of the system function move from their ideal positions to the quantized positions in the z-plane. Therefore, for decreasing this effect, registers bits number should be increased. In simulation, it was found that for a tolerable error we should choose registers of sin and cos ($e^{j2\pi k/N}$) more than 12 bits, and register of the previous value of SDFT ($y_k(n-1)$) should be more than 36 bits. We used the register with more bits since increased number of register bits will decrease error of quantization. Quantization effect of implementing the SDFT caused error; therefore, it was updated by non-recursive FFT output after some samples periodically. We used the FFT core with Radix-2 Lite, Burst I/O structure for updating the SDFT. The FFT with Radix-2 Lite type uses DSP48Es (gates) less than other types although the latency between input and calculation of FFT is longer than other structures. We had to rise bits of registers in the SDFT, thus we could not use (4) for implementation of the SDFT in high speed usage (more than 150 MHz). The reason was that we could not calculate the previous sample of SDFT $y_k(n-1)$ in one clock sampling. We had to use a modified Equation (5), which describes the relationship between the new sample that should be calculated and three previous samples of the SDFT. As such, we solved the problem of speed, but as a result, the number of used multipliers for the implementation raised. Therefore, we used another method to detect frequency.

$$\begin{aligned} y_k(n)=&[\overbrace{y_k(n-3)}^{42bits}+\overbrace{x(n-N-2)+x(n-2)}^{15bits}]\overbrace{e^{j2\pi 3k/N}}^{18bits}+ \\ &[x(n-N-1)+x(n-1)]e^{j2\pi 2k/N}-[x(n-N)+x(n)]e^{j2\pi k/N} \end{aligned} \tag{5}$$

3.3. Limitation of Phase Changes of FFT based on our Testing

Moreover, we used the FFT for dividing total band in to a small sub-band (channelization of band), followed by phase changes of the Fourier transform output to detect frequencies [15]. The phase change for *m* sample(s) can be obtained from the following. If $F\{x[n]\}=X[k]$ is the k^{th} component of the discrete Fourier transform of x[n] with N point(s), then the DFT after *m* sample(s) is as:

$$F\{x[n+m]\}=e^{j2\pi km/N}X[k] \tag{6}$$

This technique is similar to frequency estimation technique in time domain [16], but the difference is that it is also useful for multi sinusoidal signal. After we simulated this technique in the Xilinx ISE, the extracted frequency was not accurate, because quantization effects of the FFT output caused error in its phase output. The extracted frequency from the quantized phase was not accurate; thus, we used a different method.

Problems associated with the mentioned approaches of frequency estimation on the FPGA persuaded us to use peak of FFT spectrum amplitude. This method has not shortcoming of previous methods. Thus, we described more this method in the next section.

3.4. Peak of Power Amplitude of FFT Output for Fast Undersampling Frequency Detection

In the previous parts, we practically tested different methods that seem to be usefull approaches for fast frequencies detection on FPGA. However, these approaches including, DSS, Goertzel filter, sliding DFT and phase changes of FFT have their own limitation for implementation on FPGA. Previous studies [2], [7-9] tried to extract input frequency from compex waveform. However, converting received signal to two parts Inphase (real) and Quadratire (imaginary) parts need extra equipment (See Figure 2 (a)). Since, complex waveform has two parts, inphase and quadrature; thus, two ADCs are needed for sampling at rate f_s.The proposed approach used undersampling real waveform to extract the frequency of input signal. Thus it needs one ADC per sampling frequency and no need extra equipment to convert received signal to complex waveform (See Figure 2 (b)).

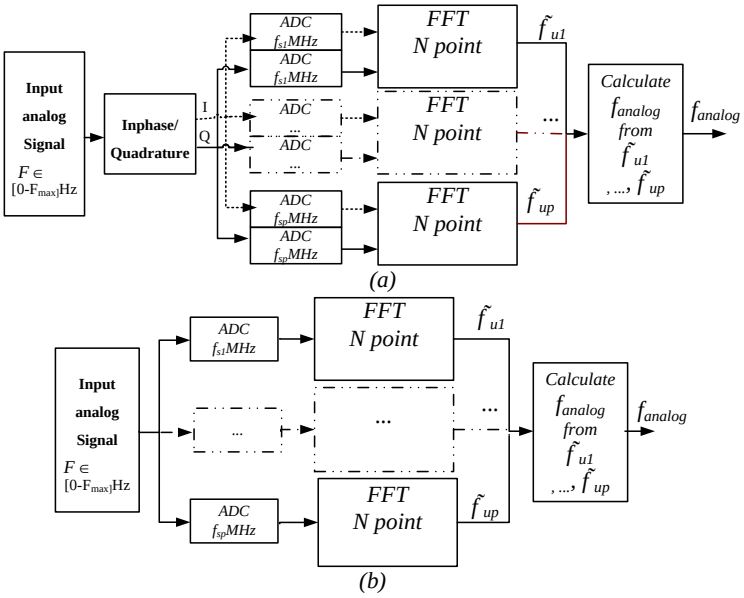

Figure 2. Block diagram of undersampling frequency detection from (a) complex waveform of previous studies [2], [7-9] and (b) proposed approach from real waveform that

Proposed frequency determination procedure for under-sampling real waveform:
By considering Figure 2, the below procedure was proposed for frequency estimation in noisy under-sampled real waveform:

Step1: Get the maximum range of input signal frequency $F_{\max}$ where $F \in [0, F_{\max}]$.

Step2: Select the desired sampling frequency band as $f_{si} \in [f_{\min}, f_{\max}]$ which the frequencies $f_{\min}$ and $f_{\max}$ are determined with respect to power limitation or processing limitation speed in FPGA, characteristic of under-sampling and other constrains that is important in practical implementation.

Step3: Determination the number of ADC and find their sampling frequencies as:

-Get the number of ADC, i.e. *p* then find the sampling frequencies that have maximum $\chi_{\min}$ [1] as follows:

$$\varepsilon_{\max} < \chi_{\min}/4 \;\; ; \;\; \chi_{\min} = \min_{k_1^{'},\ldots,k_i^{'},\ldots,k_p^{'}} [\max\{|k_1^{'} f_{s1} - k_2^{'} f_{s2}|,\ldots,|k_1^{'} f_{s1} - k_p^{'} f_{sp}|,$$
$$\ldots,|k_i^{'} f_{si} - k_{i+1}^{'} f_{s(i+1)}|,\ldots,|k_i^{'} f_{si} - k_p^{'} f_{sp}|,\ldots,|k_{p-1}^{'} f_{s(p-1)} - k_p^{'} f_{sp}|\}];$$
$$k_i^{'} = \{0,\pm 1,\ldots..,\pm 2k_i^{\max}\} \quad (7)$$

Where $\varepsilon_{\max}$ is the maximum tolerable error and $k_{\max}^{i} = \lceil F_{\max} / f_{si} \rceil$.
The frequencies that have minimum distances between their reminders [1] satisfied following relation ships:

$$F = (-\frac{\Lambda_1 f_{s1}^{G1} + \Lambda_2 f_{s2}^{G1} + \ldots + \Lambda_N f_{sN}^{G1}}{N} - \frac{\Delta_1 f_{s1}^{G2} + \Delta_2 f_{s2}^{G2} + \ldots + \Delta_M f_{sM}^{G2}}{M}) / 2,$$
$$F^{'} = (-\frac{\Lambda_1 f_{s1}^{G1} + \Lambda_2 f_{s2}^{G1} + \ldots + \Lambda_N f_{sN}^{G1}}{N} + \frac{\Delta_1 f_{s1}^{G2} + \Delta_2 f_{s2}^{G2} + \ldots + \Delta_M f_{sM}^{G2}}{M}) / 2 \quad (8)$$

Where $M + N = p$ in which $I_1 = \bigcup_{i=1}^{N} f_{si}^{G1}, I_2 = \bigcup_{i=1}^{M} f_{si}^{G2}$ then $I_1 \cup I_2 = \bigcup_{i=1}^{p} f_{si}$, $\Delta_i \in \{0,1,2,\ldots,\Delta_{\max}^{i}\}$ and $\Lambda_i \in \{0,-1,-2,\ldots,-2\Lambda_{\max}^{i}\}$ in which $\Lambda_{\max}^{i}$ and $\Delta_{\max}^{i}$ are the minimum integers that $\Lambda_{\max}^{i} f_{si}^{G1} > F_{\max}$ and $\Delta_{\max}^{i} f_{si}^{G2} > F_{\max}$.

A pseudo code for sampling frequencies selection is as follows:
Choose the f_{si} ; $i = 1,\ldots,p$ such that $f_{si} \in [f_{\min}, f_{\max}]$ were $f_{\min}$ and $f_{\max}$ are determined with respect to power limitationor processing limitation speed in FPGA, characteristic of under-sampling and other constrains that is important in practical implementation.
Begin of code:
$f_{s1} = f_{\min}$
while $f_{s1} < f_{\max}$
$f_{s1} = f_{s1} + 1 \quad f_{si} = f_{\min} \;\; ; \;\; i > 1$
while $f_{s2} < f_{\max}$
$f_{s2} = f_{s2} + 1 \quad f_{si} = f_{\min} \;\; ; i > 2$
$\vdots$
while $f_{s(p-1)} < f_{\max}$
$f_{s(p-1)} = f_{s(p-1)} + 1 \quad f_{si} = f_{\min} \;\; ; i > p-1$

while $f_{sp} < f_{\max}$
$f_{sp} = f_{sp} + 1$
Find the minimum value of (7) with respect to f_{si} ; $i = 1,\ldots,p$ and different value of k_i in (7). Save the f_{si} ; $i = 1,\ldots,p$ that maximize the minimum distance (i.e. $\chi_{\min}$) as desired sampling frequency.
End of code

Or, get the variance (σ^2) of the noise that the proposed algorithm should be extract the desired frequency uniquely, i.e. by using of (7) and $\sigma = \varepsilon_{\max}$ calculate $\chi_{\min}$ and name it as needed $\chi_{\min}$ (i.e. $\chi_{\min(needed)}$). Also, name the maximum range of input signal frequency $F_{\max}$ as $F_{\max(needed)}$ and finally find the minimum number of ADC (number of frequency sampling) by considering the maximum tolerable variance of noise as (7). To do this, we start with three ADC, i.e. p=2. Since we have the sampling frequency band as $f_{si} \in [f_{\min}, f_{\max}]$ and range of input signal as $F \in [0, F_{\max}]$. Thus, we can calculate the minimum of $\chi_{\min}$'s in (7) with respect to different values of k_i, and compare it with $\chi_{\min(needed)}$. If minimum value of $\chi_{\min}$, from (7), is smaller than

$\chi_{\min(needed)}$, in this case, the band of sampling frequencies, i.e. $f_{si} \in [f_{\min}, f_{\max}]$, should be increased. To do this, it is possible to decrease the lower band ($f_{\min}$). However, it changes the maximum range of input frequency [1] $F_{\max}$ that is obtained as follows:

$$\begin{aligned} F_{\max} &= \min_{M,N}(LCM(f_{s1}^{G1},\ldots,f_{sM}^{G1}) + LCM(f_{s1}^{G2},\ldots,f_{sN}^{G2}))/2 \\ &; \; F \in [0, F_{\max}], M+N=p, \quad M,N=1,..,p \\ &\{f_{s1}, f_{s2},\ldots,f_{sp}\} = \{\{f_{s1}^{G1}, f_{s2}^{G1},\ldots,f_{sN}^{G1}\} \cup \{f_{s1}^{G2}, f_{s2}^{G2},\ldots,f_{sM}^{G2}\}\} \end{aligned} \quad (9)$$

If $F_{\max}$ from (9) is smaller than $F_{\max(needed)}$ then increase number of ADC (number of sampling frequencies) and calculate minimum of $\chi_{\min}$ in (7) and $F_{\max}$ in (9), again. Repeat this step until minimum of $\chi_{\min}$ in (7) and $F_{\max}$ from (9) to be bigger than $\chi_{\min(needed)}$ and $F_{\max(needed)}$, respectively. Save the number of ADCs as needed number of ADCs.

Or, get the minimum SNR that you want to extract the desired frequency unambiguity and obtain minimum of $\sigma = \varepsilon_{\max}$ that is needed for this from (10) [17] and (7) and named it as $\chi_{\min(needed)}$.

$$\sigma_f = \frac{f_s\sqrt{12}}{2\pi\sqrt{N_s(N_s^2-1)SNR}} \quad (10)$$

The lower limit on frequency error for any unbiased estimation is given by the Cramer-Rao bound [17]. Here, σ_f is the standard deviation in *Hz*, f_s is the sampling frequency of signal and N_s is number of samples available for FFT estimation without compensating the frequency truncated to the nearest bins, causing the error of $\frac{f_s}{2N}$.

Then, find the minimum number of ADC that can be used to satisfy this SNR value (or $\chi_{\min(needed)}$ value). This procedure is named as determining of the minimum number of ADC from desired noise variance.

Step 4: Find the noisy under-sampled frequencies from output of p ADCs as Figure 2 (b) by N-Point FFT are $\tilde{f}_{ui}$; $i=1,\ldots,p$. and real frequency (input analog frequency) that should be extracted from these under samplings is F .

Step5: Calculate the desired frequency from its under-sampled waveforms by following pseudo code:

Extraction of the input frequency from its noisy under-sampled real waveform:

A pseudo code for extraction of a real frequency from its noisy under-sampled frequencies is obtained in this part.

Step5-1: calculate possible frequency in band, i.e. $f \in [0, F_{\max}]$, where its undersampled frequency is $\tilde{f}_{u1}$ when sampled with f_{s1} and name it $\hat{F}_1^t$ as follows:

$$\hat{F}_1^t = \{\hat{k}_1^t f_{s1} + (-1)^{\hat{v}_1^t} f_{u1};\ 0 \le \hat{k}_1^t f_{s1} + (-1)^{\hat{v}_1^t} < F_{\max}, \hat{v}_1^t \in \{1,2\}, \hat{k}_1^t = 0,1,\ldots\} \quad (11)$$

Step5-2: Find under samplings of all $\hat{F}_1^t$ when are sampled with sampling frequencies f_{si} ; $i=2,\ldots,p$ and name these undersampled walues as $\hat{f}_{ui}^t$; $i=2,\ldots,p$. Therefore, relationship between $\hat{F}_1^t$ and its corresponded undersampled $\hat{f}_{ui}^t$; $i=2,\ldots,p$ are as:

$$\hat{F}_1^t = \hat{F}_i^t = \hat{k}_i^t f_{si} + (-1)^{\hat{v}_i^t} \hat{f}_{ui}^t \quad (12)$$

Step5-3: Substitute $\hat{f}_{ui}^t$; $i=2,\ldots,p$ by their noisy undersampled $\tilde{f}_{ui}$; $i=2,\ldots,p$ in Equation (12). Then calculate :

$$\tilde{F}_i^t = \hat{k}_i^t f_{si} + (-1)^{\hat{v}_i^t} \tilde{f}_{ui}^t \tag{13}$$

Step5-4: Then find $\hat{k}_i^t$ and $\hat{v}_i^t$ that minimize the following relationship and name them as $\hat{k}_i^{t^*}$ and $\hat{v}_i^{t^*}$:

$$\begin{aligned}&\hat{k}_1^{t^*}, \hat{v}_1^{t^*}, \ldots, \hat{k}_i^{t^*}, \hat{v}_i^{t^*}, \ldots, \hat{k}_p^{t^*}, \hat{v}_p^{t^*} = \{\hat{k}_1^t, \hat{v}_1^t, \ldots, \hat{k}_i^t, \hat{v}_i^t, \ldots, \hat{k}_p^t, \hat{v}_p^t\};\\ &\min_{\hat{k}_1^t, \hat{v}_1^t, \ldots, \hat{k}_p^t, \hat{v}_p^t} \max\{\left|\tilde{f}_{u2}^t - \hat{f}_{u2}^t\right|, \ldots, \left|\tilde{f}_{ui}^t - \hat{f}_{ui}^t\right|, \ldots, \left|\tilde{f}_{up}^t - \hat{f}_{up}^t\right|,\\ &\left|\tilde{F}_2^t - \tilde{F}_3^t\right|, \ldots, \left|\tilde{F}_2^t - \tilde{F}_p^t\right|, \ldots, \left|\tilde{F}_i^t - \tilde{F}_{i+1}^t\right|, \ldots, \left|\tilde{F}_i^t - \tilde{F}_p^t\right|, \ldots, \left|\tilde{F}_{p-1}^t - \tilde{F}_p^t\right|\}\}\end{aligned} \tag{14}$$

Note that $\hat{k}_i^t$ and $\hat{v}_i^t$ are related to choose of $\hat{k}_i^1$ and $\hat{v}_i^1$.

Step5-5*:* After finding $\tilde{F}_i^t = \hat{k}_i^{t^*} f_{si} + (-1)^{\hat{v}_i^{t^*}} \tilde{f}_{ui}$ then, input analog frequency can be determined as (15) $F \approx \hat{F}$.

$$\hat{F} = (\sum_{i=1}^{p} \tilde{F}_i^{t^*}) / p \tag{15}$$

4. SIMULATIONS AND RESULTS

4.1. Performance of Peak of FFT Spectrum Amplitude based on our Testing

Different methods for sub-Nyquist frequency detection on the FPGA were tested in this research as discussed in the previous section. Finally, the direct FFT was selected for frequency estimation. The local maximum in the index bin of a n-point FFT was taken as frequency. Note that the FFT core [18], (available in the Xilinx ISE Tools) in the ISE environment. A Xilinx development board with XC5vsx95T, speed of 2 and pakage FF1136 of Virtex 5 family were used for testing. We tested sub-Nyquist frequency detection by N = *2048* FFTs points and sampling frequency of 200 *MHz* on the FPGA. With a bin size of *bs = 200 MHz*/ 2048 = 97.6 *kHz*, we hypothesized that the error would never exceed $b_s/2$*=48.8 kHz*. We chose three sampling frequencies between 180-200 MHz, because managing and processing the data with the selected technology (e.g. FPGA and ADC) was possible up to *200 MHz*. and the input signal band was between 0-2 GHz.

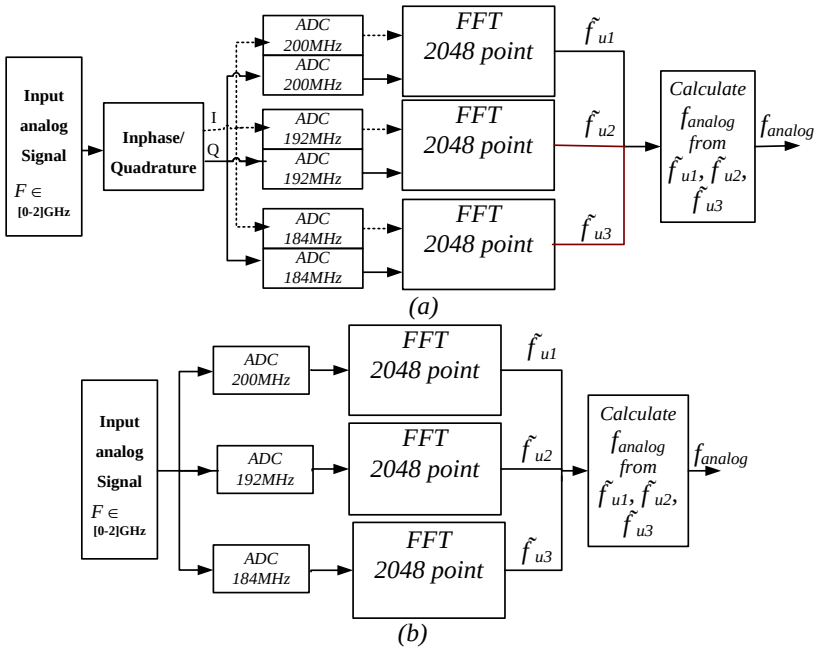

Figure 3. Under-sampling frequency detection (a) previous studies [2], [7-9] need more ADC and Inphase/Quadrature equipments (b) proposed approach.

Suitable frequencies for sampling frequency are f_{s1} = *184*, f_{s2} = *192*, f_{s3} = *200 MHz* according to Step3 of frequency estimation procedure in the previous section. We adopted sampling frequencies of *200 MHz*, *192 MHz* and *184 MHz* to maximize the minimum distance ($\chi_{\min}$) of (7). As shown in Figure 3, we used the FFT with 2048 points to extract the input frequency. Then, we used the relationship between the reminders of analog signal frequency, i.e., $\tilde{f}_{u1}$, $\tilde{f}_{u2}$, and $\tilde{f}_{u3}$ to extract the real frequency of the analog signal ($f_{analog} = F$).

The quantity of consumed gates for implementation of 2048-point FFT is presented in Table 1.

Table 1 Synthesis results of a 2048-point FFT core pipelined streaming I/O on XC5vsx95T used previous approaches [2], [7-9] and proposed approach (usages for both are same)

FPGA resource	Used	Total in FPGA	% Usage
# of slice Register	4741	58880	8
# of slice LUTs	3569	58880	6
# of DSP48Es	60	640	9
#of BRAMs	8	244	3

For extracting analog frequency *F* from the reminders $\tilde{f}_{u1}$, $\tilde{f}_{u2}$, and $\tilde{f}_{u3}$ the proposed algorithm in Step5-1 to 5-5 was used.

We tested the implementation by input signal with frequency $(-(-4f_{s1}+-4f_{s2})+2f_{s3})/2 = 2(200+192)+184 = 968MHz$ and the have minimum distance based on (8) that is equal to 8 according to Equation (7). It may be detected incorrectly as $F' = (-(-4f_{s1}+-4f_{s2})+2f_{s3})/2 = 2(200+192)-184 = 600MHz$ based on (8). It can be found from Equation (10) that good frequency estimation method (near the Cramer-Rao bound) allows correct detection of frequency by the proposed method until the point where deviation of frequency is below $\frac{8}{2\sqrt{3}}$ (65 *dB*). The minimum distance between the reminders of each frequency in the band with respect to f_{s1}=*184*, f_{s2}=*192*, f_{s3}=*200 MHz* are 8 *MHz* (7). Therefore, maximum tolerance of each frequency to detect frequency uniquely is $\frac{8}{2\sqrt{3}}$ *MHz.*

Results of testing for signals with different pulse widths and periods of repetition equal to 100 us were evaluated and the minimum power for frequency extraction uniquely (Fig. 4) was near -70 *dBm* for greater pulse width. It is shown that increased pulse width alters the minimum power of detectable pulse until the number of sample pulses is less than the FFT length. The simulation results in Fig.4 shows undersampling frequency detection for proposed approach and previous approach have same performance while proposed approach has less hardware in compared with the previous studies (see Figure 3 (a), 3(b) and Table 2).

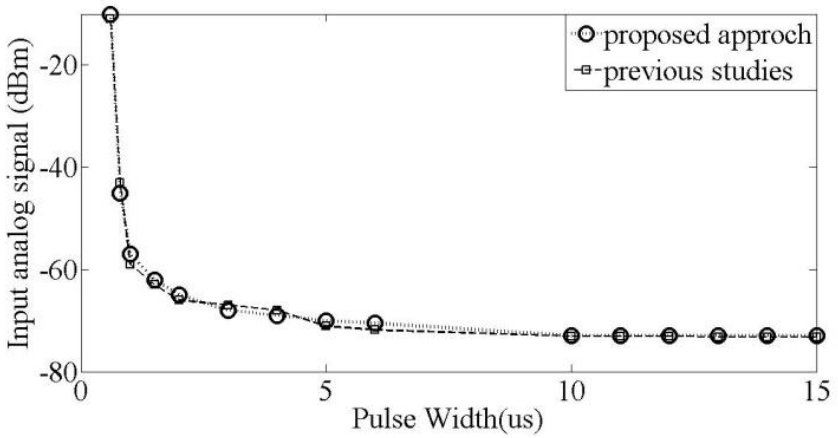

Fig 4. Minimum power detectable with new under-sampling frequency detection method on the FPGA for different pulse widths

Comparision between proposed and previous studies for undersampling frequency detection can be summarized as Table 2. As you can see number of used ADC per sampling frequencies for previous studies that used complex aveform (has Inphase and Quadrature parts) was double of proposed approach from real waveform. Furturemore, extra equipment was needed in previous studies to obtain complex waveform. Since implementation of FFT for complex and real numbers on FPGA is same and same core IP is used for them the average usage of resurces for both approaches are same.

Table 2. Comparision between proposed and previous studies approaches

Method	No. ADC per samling frequencies	Extra equipment	Average FPGA Resouce usage on on XC5vsx95T
Comlex waveform undersampling[2,7,8,9]	2	Equipment for convert received signal to complex waveform	8%
Proposed method	1	No need	8%

In sensor networks, energy consumption and cost of hardware implementation are important factores [19], [20]. Our proposed simple hardware can save cost and energy. Therefore, it is useful for sensors in sensor networks.

5. CONCLUSIONS

This paper examined and discussed different methods for estimation of sub-sampled frequency on the FPGA. We implemented different methods on the FPGA for Sub-Sampling frequencies detection, including Least Squares, Direct State Space, Goertzel filter, Sliding DFT, Phase changes of FFT, peak amplitude of FFT. Shortcoming and advantages of each method during the implementation on FPGA were discussed and tested. Note that, shortcomings of these methods are not discussed specifically in previous studies. Finally, an appropriate method for under-sampling frequency detection on the FPGA was chosen. Implementation and modification of codes on the FPGA are time-consuming procedures and require knowledge of hardware. Therefore, this work can help choose an appropriate approach for implementation of sub-sampling frequency detection on the FPGA from Least Squares, Direct State Space, Goertzel filter, Sliding DFT, Phase changes of FFT, peak amplitude of FFT spectrum. After choosing an appropriate method that is Peak of FFT spectrum amplitude, it is possible to use usual ADCs, which usually work under 1 Giga sample/sec to detect frequencies much higher than its sampling rate. Detection of higher frequency by lower sampling rate has been applied to various areas, such as sensor networks and radars signal processing. Our proposed approach, undersampling frequency detection from real waveform, needs less ADC and no need equipment for converting received signal to complex waveform in the contrast with the previous studies for complex waveform.

ACKNOWLEDGMENTS

This work was supported by the Research Grant Scheme of University of Torbat Heydarieh.

REFERENCES

[1] Maroosi A, Bizaki H.K., "Digital Frequency Determination of Real Waveforms based on Multiple Sensors with Low Sampling Rates", *IEEE Sensors Journal*, 2012; 12(5): 1483-1495
[2] Xiao L, Xia X.G, Huo H, "New Conditions on Achieving the Maximal Possible Dynamic Range for a Generalized Chinese Remainder Theorem of Multiple Integers", *IEEE Signal Processing Letters*, 2015; 22(12): 2199-2203.
[3] Li W. C, Wang X. Z, Wang X. M, Moran B, "Distance Estimation using Wrapped Phase Measurements in Noise", *IEEE Trans. Signal Process*, 2013; 61: 1676-1688.
[4] Akhlaq A, McKilliam R, Subramanian R, "Basis Construction for Range Estimation by Phase Nwrapping", *IEEE Signal Process. Letter*, 2015; 22: 2152-2156.
[5] Marques P. A. C, Dias J. E. M. B, "Velocity Estimation of Fast Moving Targets using a Single SAR Sensor", *IEEE Trans. Aerosp Electron Syst*, 2005; 41: 75–89.
[6] Zoltowski M. D, Mathews C. P., "Real-Time Frequency and 2-D Angle Estimation with Sub-Nyquist Spatio-Temporal Sampling", *IEEE Trans. Signal Process*, 1994; 42: 2781–2791.
[7] Xiao L, Xia X.-G., "A Generalized Chinese Remainder Theorem for Two Integers", *IEEE Signal Process.* Lett, 2014; 21: 55–59.

[8] Wang W, Li X. P, Xia X.-G, Wang W. J., "The Largest Dynamic Range of a Generalized Chinese Remainder Theorem for Two Integers", *IEEE Signal Process. Lett*, 2015; 22: 254–258.
[9] Xiao L, Xia X.G., "A New Robust Chinese Remainder Theorem with Improved Performance in Frequency Estimation from Undersampled Waveforms" *Signal Processing*, 2015; 117: 242-246.
[10] Barbosa D, Monaro R.M, Coury D.V, Oleskovicz M, "Modified Least Mean Square Algorithm for Adaptive Frequency Estimation in Power Systems", *IEEE Power and Energy Society General Meeting* 2008; pp.1-6.
[11] Jing D, Jun W, Han C, Dalu L, Han W, *"Estimating the Frequency in Power System based on State Space Recursive Least Squares"*, ICIEA 2008, IEEE Conf on Industrial Electronics and Applications, 2008; 1444-1447.
[12] Oppenheim A. V, Schafer R.W, Buck J. R., "Discrete-Time Signal Processing", Pearson Education, Inc., 2nd edition, 1999.
[13] Jacobsen E, Lyons R, "The Sliding DFT", *IEEE Signal Process Magazine*, 2003; 20: 74-80.
[14] Beraldin J, Steenaart W, "Overflow Analysis of a Fixed-Point Implementation of the Goertzel Algorithm, *IEEE Tran. Circuits and Systems*, 1989; 36: 322-324.
[15] Brown, J. C, Puckette M.S., "A High Resolution Fundamental Frequency Determination based on Phase Changes of the Fourier Transform", The Journal of the Acoustical Society of America, 1993; 94: 662-667.
[16] Herselman P. L, Cilliers J. E., "A Digital Instantaneous Frequency Measurement Technique using High-Speed Analogue-to-Digital Converters and Field Programmable Gate Arrays", *South African Journal of Science*, 2006; 102: 345–348.
[17] Holm S, "Optimum FFT-based Frequency Acquisition with Application to COSPAS-SARSAT", *IEEE Trans. Aerosp Electron Syst*, 1993; 29: 468-475.
[18] Emmert J, *"An FFT Approximation Technique Suitable for on-Chip Generation and Analysis of Sinusoidal Signals"*, In Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2003.
[19] Gupta S, Gupta G, "Simulation Time and Energy Test for Topology Construction Protocol in Wireless Sensor Networks", *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 2015; 3: 89-92.
[20] Manjunatha R.C, Rekha K.R, Nataraj K. R., "Implementation of Fuzzy based Simulation for Clone Detection in Wireless Sensor Networks", *International Journal of Electrical and Computer Engineering*, 2016; 6: 1570-1576.

BIOGRAPHIES OF AUTHORS

Seyed Ehsan Yasrebi Naeini is a lecturer at University of Torbat Heydarieh. His reaserch areas include data proicessing, data mining, etc.

Ali Maroosi is an assistant professor at University of Torbat Heydarieh. His raserch interests are data and signal processing, parallel processing, intelligent algorithms, etc. He has published many journal papers in refreed journals.