

Research Article

Challenges in Clock Synchronization for On-Site Coding Digital Beamformer

Satheesh Bojja Venkatakrishnan, Elias A. Alwan, and John L. Volakis

Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174, USA

Correspondence should be addressed to Satheesh Bojja Venkatakrishnan; bojjavenkatakrishnan.1@osu.edu

Received 30 May 2017; Revised 27 August 2017; Accepted 17 September 2017; Published 25 October 2017

Academic Editor: John Kalomiros

Copyright © 2017 Satheesh Bojja Venkatakrishnan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Typical radio frequency (RF) digital beamformers can be highly complex. In addition to a suitable antenna array, they require numerous receiver chains, demodulators, data converter arrays, and digital signal processors. To recover and reconstruct the received signal, synchronization is required since the analog-to-digital converters (ADCs), digital-to-analog converters (DACs), field programmable gate arrays (FPGAs), and local oscillators are all clocked at different frequencies. In this article, we present a clock synchronization topology for a multichannel on-site coding receiver (OSCR) using the FPGA as a master clock to drive all RF blocks. This approach reduces synchronization errors by a factor of 8, when compared to conventional digital beamformer.

1. Introduction

Synchronization techniques are required for reliable high data rate communications to avoid phase mismatches and jitter [1, 2]. For complex RF systems involving data converter arrays, different frequencies may be required implying that components are clocked at different frequencies as well. Synchronization then requires different clocking architectures depending on the system configuration [3, 4].

In complex systems, the number of clock signals can rapidly increase from just a few to hundreds. Quite often, in large scale systems, a single external clock circuit may not have enough outputs to drive all branches. To overcome this issue, various clock tree topologies are used to synchronize multiple parts, devices, or systems [5]. However, each level in the tree introduces a delay component that is fixed or undetermined. Although fixed delays can be compensated with additional effort, it is highly challenging to eliminate the undetermined delays. Further, these delays may be affected by external factors like voltage and temperature changes and device-specific variations. Altogether, inaccuracies result in intolerable timing variations in analog-to-digital converters

(ADCs) and digital-to-analog converters (DACs) that affect the clocking [3, 4].

For transceivers where the devices and components are located in proximity, sharing a common timing signal is generally the easiest and most accurate method for synchronization [5–7]. This is the case with on-site coding receiver (OSCR) [8, 9] that combines several of signals into a single ADC using coding technique, as shown in Figure 1. Since a single ADC is used to handle several antennas/paths, errors due to ADC-FPGA synchronization, interchannel skews, and variations within ADCs can be reduced. Specifically, in OSCR architecture [9–11], all the components are enclosed within a single block making synchronization easier and less prone to errors. Further, using a single FPGA as a master source for synchronization removes the clock skew errors [12]. However, to highlight the implementation challenges using different topologies, we have included the analysis of clock synchronization implementation using external distribution circuit. Preliminary work has already been presented in [13]. In this article, we present a clock synchronization implementation for an 8-channel OSCR digital beamformer using an FPGA,

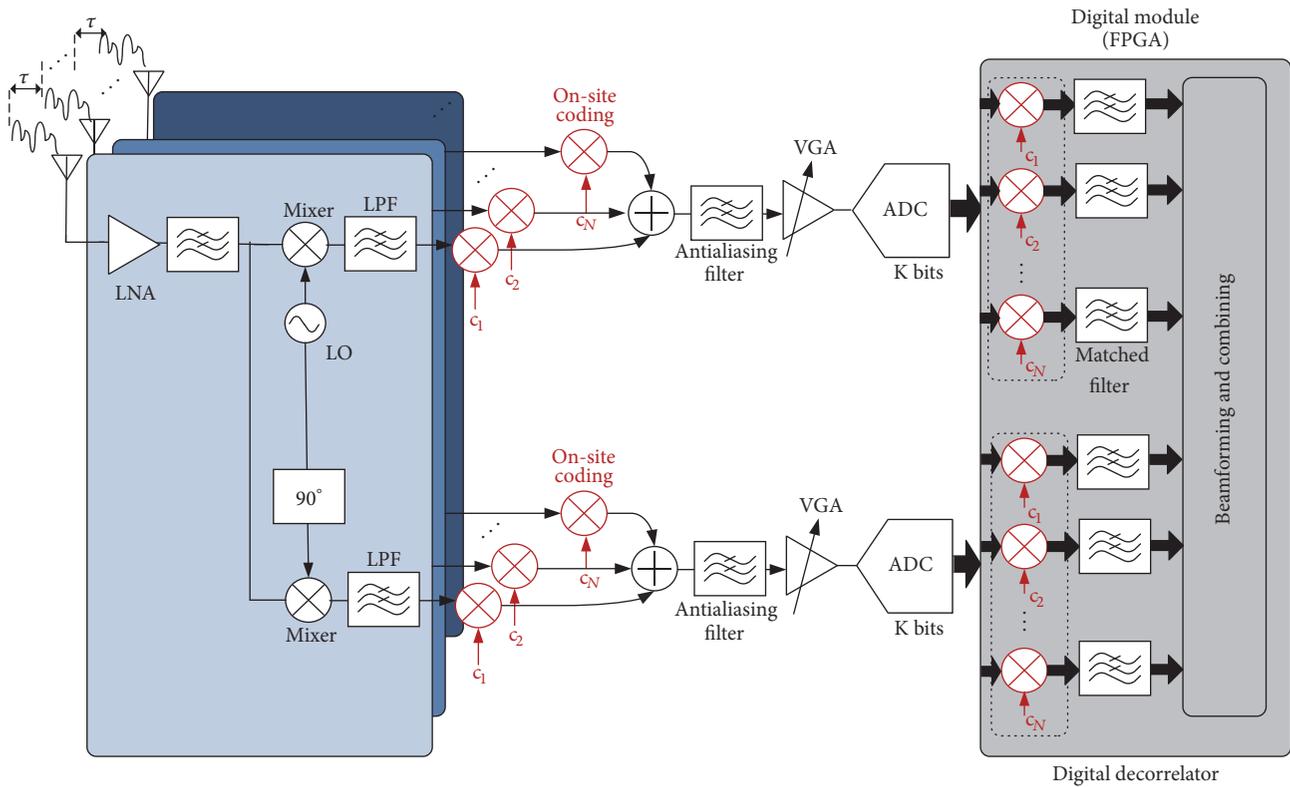


FIGURE 1: Block diagram representation of OSCR system [8, 9].

an ADC, an in-house built encoder board, and RF transceiver boards.

2. Clock Synchronization Implementation

2.1. FPGA Controlled Distribution Circuit. A general description of realized OSCR is given in [9–11]. The 8-channel OSCR employs 8 RF boards that require 8 clock signals operating at 38.4 MHz and an encoder board with 8 differential (16 signals) orthogonal codes, perfectly synchronized with each other. Also, a dual-channel ADC sampling at 256 MSPS, requiring a clock of 256 MHz is included. Thus, for an 8-channel OSCR 25 signals (8 RF clocks + 8 differential codes + 1 ADC clock) must be synchronized. To keep the jitter low and reliability high, the total number of components used for synchronization must be minimized. With this in mind, FPGA platform was chosen as a good compromise between performance and flexibility. Figure 2 shows the FPGA used to provide timing signals to all devices. Notably, using a single FPGA to generate clock signals at different rates becomes a significant design challenge. An additional complexity is that most RF front-end units rely on serial interfaces to the transmission/receive blocks, requiring that data and clock be embedded/deembedded by a digital processor or FPGA.

As can be seen from Figure 2, clock synchronization was implemented using Xilinx VC707 FPGA running ISE software. VHDL was used for programming and the “clocking wizard” IP from Xilinx was used to generate and distribute

the required clocks. To do so, a low phase noise reference signal of 12.8 MHz was fed to the FPGA. The phase noise at 10 KHz offset was found to be -80 dBc/Hz and at 1 KHz offset was -102 dBc/Hz. This clock frequency was chosen to reduce output jitter and clock skew. Using this clocking wizard, the 12.8 MHz reference clock was used to generate output clock frequencies of 38.4 MHz for the RF boards, 256 MHz for the ADC clock, and 64 MHz to be used by the FPGA for generating orthogonal codes whose maximum frequency component is 32 MHz. It should be noted that, for ADCs with high sampling rate, the sampling clock should be sharp. High phase noise will lead to degradation in signal-to-noise ratio (SNR) performance [14].

Having determined the clock frequencies, a next step is to program the clock signal assignments and code generation using VHDL. After an error free execution of the program, module level simulation was performed. Figure 3 shows the simulated output generated by the FPGA. There are 8 different orthogonal codes each synchronously triggered with reference to the rising edge of the 64 MHz internal FPGA clock. We note that each of the 8 signals represents a 16-bit orthogonal code used for encoding/decoding, as shown in Figure 1. From the simulations in Figure 3, we observe that the FPGA generates clocks and codes with perfect synchronization. This has been verified using lab measurements as presented in the later sections. From the measured lab result, it was seen that the system exhibits mesosynchronization with constant delay or phase shift between signals.

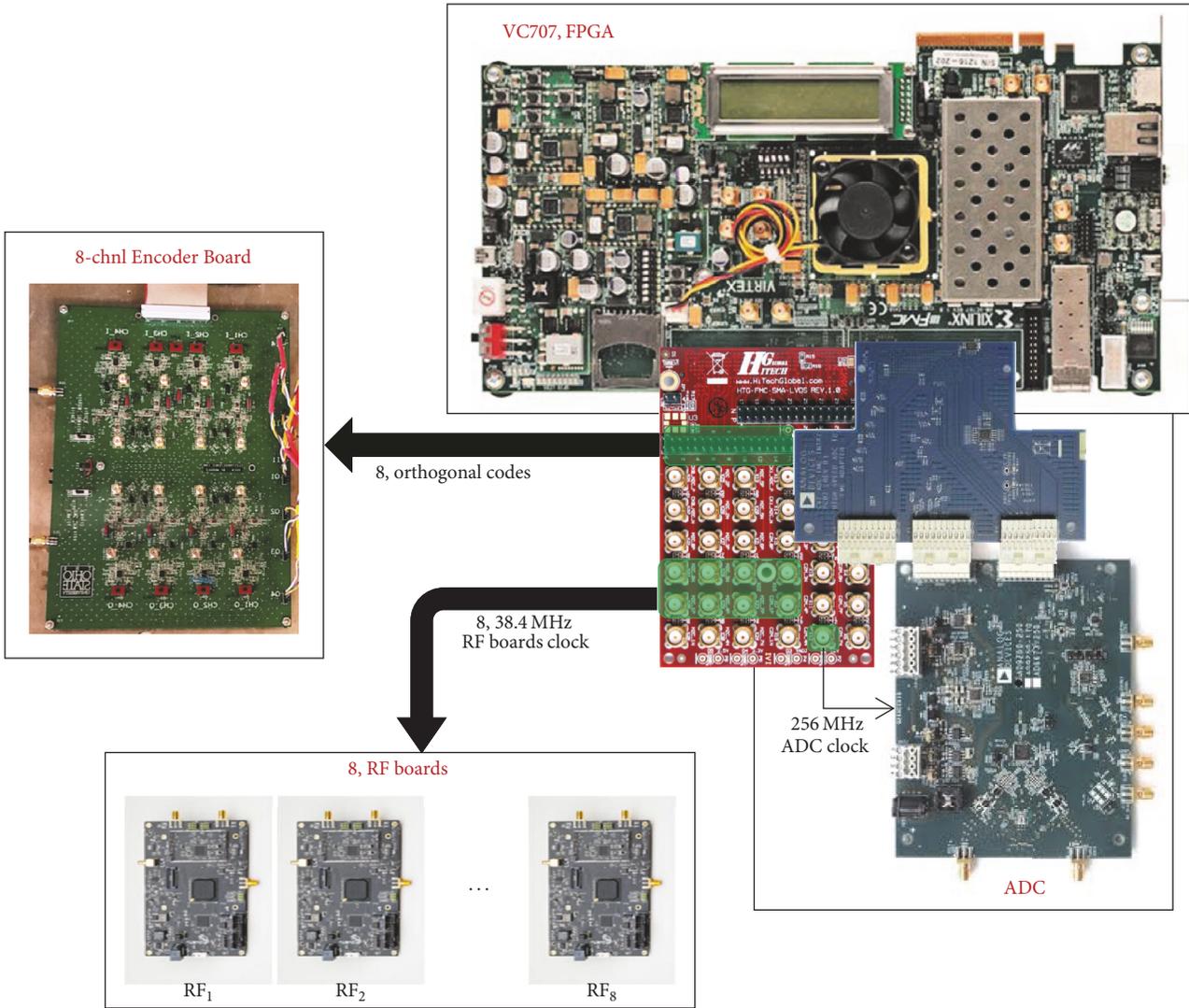


FIGURE 2: Clock generation and synchronization requirement for an 8-channel OSCR using FPGA as a master clock.

The 25 signals of the OSCR must be rerouted within the FPGA to the output pins of the VC707 board. These signals are taken from the debug card connected to the FPGA using the FPGA Mezzanine Card (FMC). FMC houses 400 pins and hence 200 signals can be generated at any time from the fully populated debug card. In the present scenario, pin placement was performed following the interface constraints. Also, like signals were placed within close proximity to reduce mismatch in the path lengths of these signal.

It is necessary to generate all 25 signals with proper logic levels. For example, to generate high-speed orthogonal code signals and still keep jitter low, a low voltage differential signaling (LVDS) output buffer was used for the FPGA outputs. Mapping is then performed after converting the register-transistor logic (RTL) description into a netlist of basic elements (BELs). This takes care of the packaging and signal placement. Subsequently, we perform routing of the signals to specified user pins. As a final step, timing analysis is

performed to ensure all signals have comparable path delays. This timing analysis considers the location of the output pins and is hence more realistic in determining the propagation delay of each signal. Timing analysis was performed only for 8 orthogonal codes and 8 RF board clocks and had to be perfectly synchronized for realizing digital beamforming. Timing analysis for ADC clock was ignored. Propagation delays for RF board clock signals and orthogonal codes are given in Figures 4(a) and 4(b), respectively. It should be noted that for OSCR, mesochronous synchronization is also considered to be perfectly synchronized, since the phase difference between signals is constant without skews and can be corrected in post processing.

From Figure 4(a), we can infer that the propagation delays of the clock sources decrease uniformly. This is due to the way the clock outputs were routed. The path taken by clock is controlled by the slices location and output pins in the floor plan of the FPGA. That is, even though all 8 clock outputs

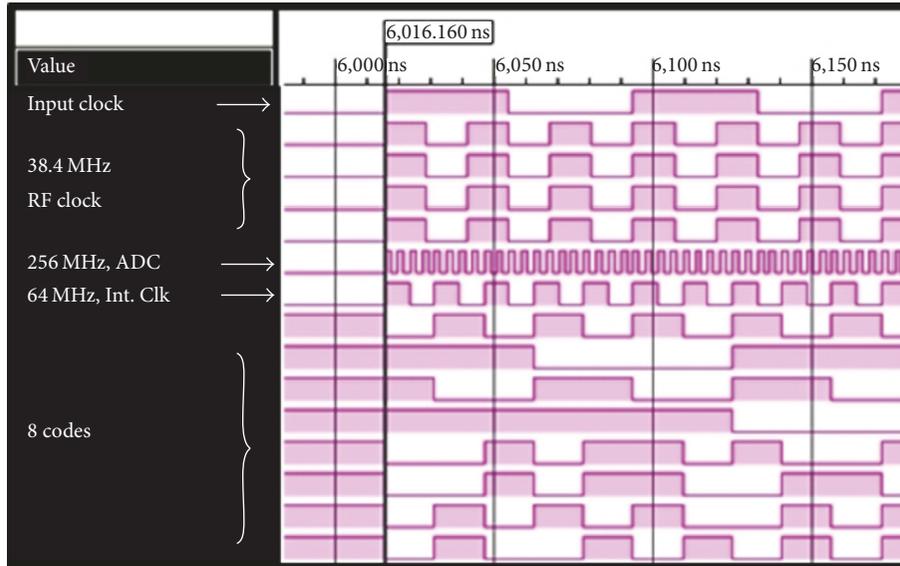


FIGURE 3: Simulated signal waveforms as required by the 8-channel OSCR system.

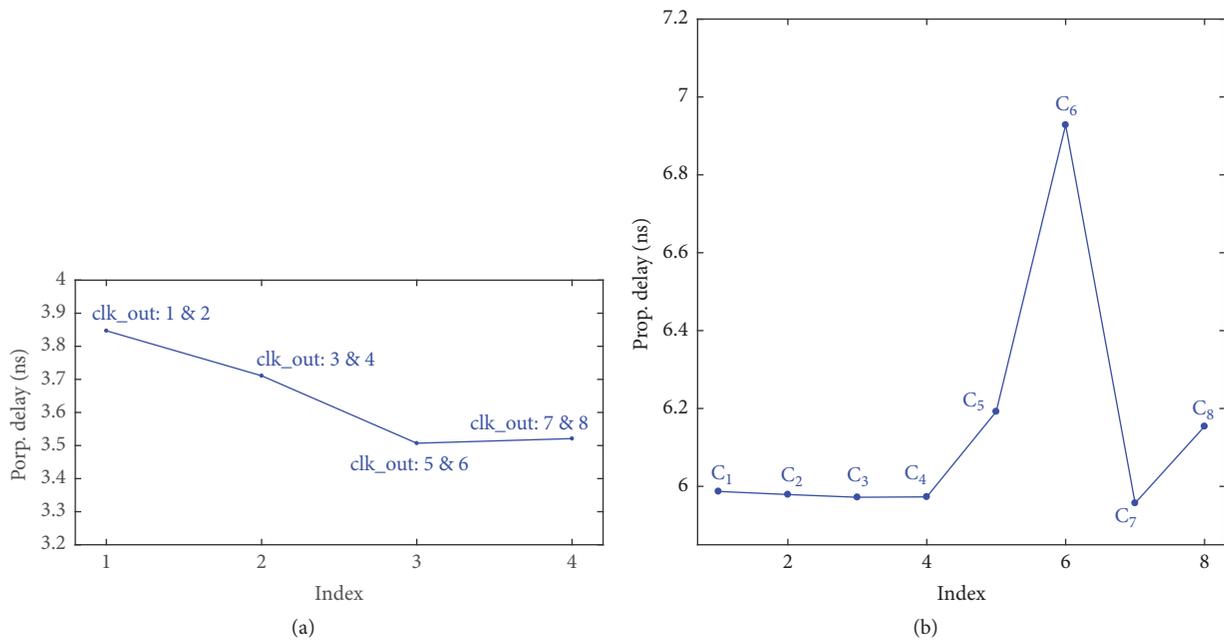


FIGURE 4: Simulated propagation delay of signals based on timing analysis. (a) Propagation delay of 8 RF board clock signals at 38.4 MHz and (b) Propagation delay of 8 orthogonal codes.

are at the same frequency of 38.4 MHz, their signal paths are different and hence they experience different propagation delays. Clock outputs 1, 2, 3, and 4 have a longer propagation path and clock outputs 5, 6, 7, and 8 have shorter paths. Therefore, longer propagation delays were associated with clock outputs 1 and 2. Similarly, for the orthogonal code signals, 7 sets of codes were placed within close proximity of each other except for the code pair 6. As a result, propagation delays of this code were higher as compared to the others (see Figure 4(b)). It should be noted that the mismatch in the paths is due to design constraint of the OSCR architecture.

Generally, the propagation delays between the board and the FPGA paths are in “ns” range. Correspondingly, the delay differences among the clock signals are in the “ps” range. Similarly, from the analysis of the orthogonal code signals, the worst-case scenario arises when we compare code 6 (C6) with code 4 (C4). The propagation delay difference between C6 and C4 is approximately 0.9 ns. These delays can be corrected within the FPGA. The reason is that VC707 can generate signals up to 933 MHz and corresponding clock of 1.07 ps. Hence, any clock signal with time period greater than 1.07 ps can be easily synthesized. Consequently, delays

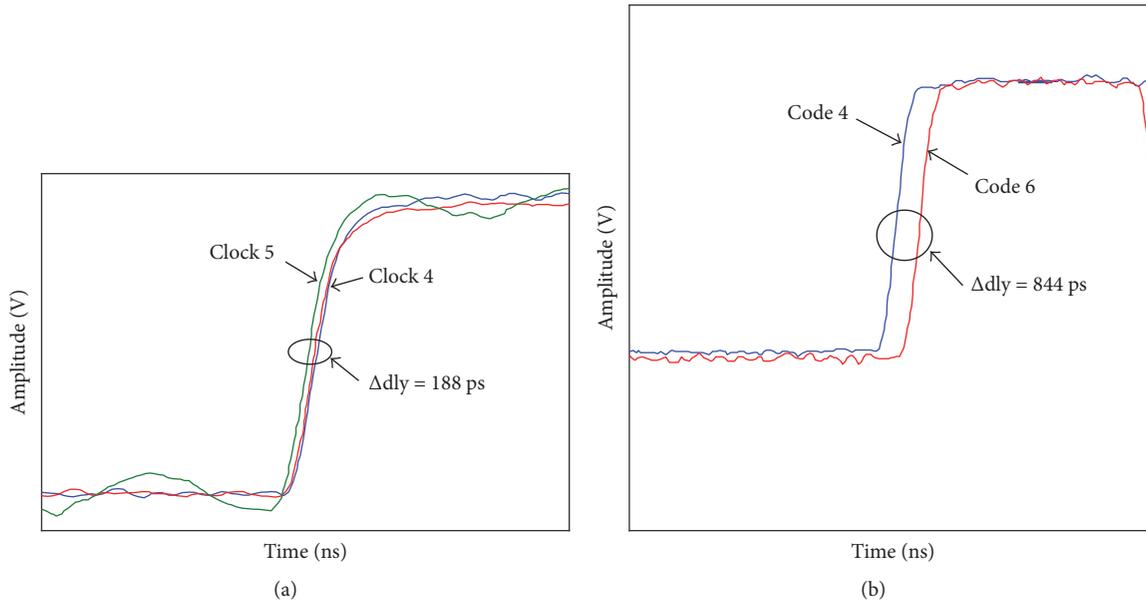


FIGURE 5: Generated signals by the FPGA measured using oscilloscope. (a) Generated clock signals (Clock 4 and Clock 5) and (b) Generated codes (C4 and C6).

also of the order “ps” can be synthesized and accounted within the FPGA using VHDL coding. Notably, differences of this magnitude have minimum effect on the on-site coding performance.

Figure 5(a) shows the captured clock signals on an oscilloscope for clock outputs 4 and 5. Similarly, Figure 5(b) shows the captured codes of C4 and C6 generated by the FPGA. We note that the reason for considering these specific signals is because they exhibited the largest delay. As can be seen from Figures 5(a) and 5(b), the maximum delay between clock signals used for triggering the RF boards is ~ 188 ps and the maximum delay between codes is ~ 844 ps.

For the direction of arrival application using current OSCR system, the RF boards are clocked at 38.4 MHz or 26.04 ns. Since the maximum delay between these clock signals is 0.188 ns, which falls within the rise time/fall time of the specification of the clock generator chip, the delay has no significant impact on the system performance. Similarly, the maximum delay for the codes is ~ 0.9 ns. It should be noted that the same codes generated by the FPGA are used for encoding and decoding. Hence, by delaying the digitized coded signals for processing within the FPGA, it is possible for us to compensate for the delay caused by the codes. Alternatively, the FPGA has the capability to generate delays as low as 1.07 ps. Hence, corresponding delays between codes can be generated and corrected within the FPGA itself before encoding.

Using the above synchronization topology, OSCR performance was measured and it was found that there is no overall degradation in OSCR performance due to synchronization issues, as detailed in [9, 10, 15]. Using the above synchronization scheme, bit error rate (BER) and SNR was simulated. Based on the results in [9, 10, 15], it can be concluded that

above synchronization scheme results in no performance degradation.

2.2. Clock Synchronization Using External Distribution Circuit. It is also imperative to analyze the clock synchronization implementation using an external distribution clock. In complex systems involving multiple components to be synchronized, a single external clock generation integrated circuits may not have enough outputs to drive all branches. This problem can be overcome using clock tree topology to synchronize multiple devices [4]. However, in such topologies each level of distribution introduces a delay component that is the result of fixed and undetermined delays that are also affected by external factors. The inaccuracies add up causing intolerable timing variations which drastically affect the high frequency components clocking. Although it is relatively easy to compensate for fixed delays, undetermined delays cannot be corrected for within the system [4]. In addition to these constraints, the tree structure should be accommodating when the number of branches is increased.

Despite these drawbacks, analysis was performed for 8-element OSCR clock synchronization using external source and distribution circuit to compare it with our proposed synchronization method using FPGA for clock distribution. In order to do so, the clock tree should have at least 10 clock outputs (8 for RF boards, 1 for ADC, and 1 for FPGA), as shown in Figure 6. Commercially off-the-shelf (COTS) components were considered for this case. AD9576 by analog devices was the right choice providing up to 11 clock outputs and also has the capability to generate clocks. It has built-in dividers that enable generation of multiple frequencies. Despite the capability of the evaluation board (AD9576) to generate low jitter clock signals over wide frequency range,

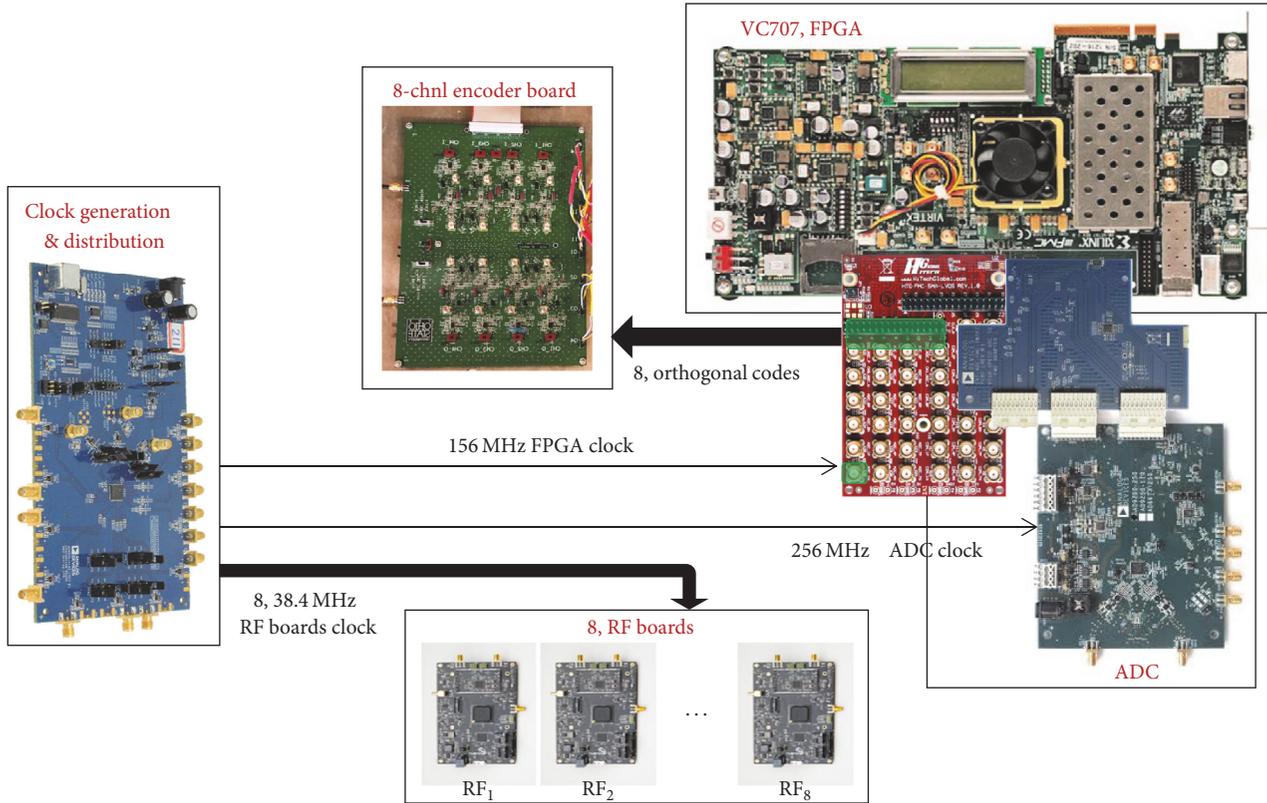


FIGURE 6: Clock generation and synchronization requirement for an 8-channel OSCR using external source.

it still is not desirable for synchronization implementation in OSCR. By employing external distribution circuit, the hardware requirement increases, thus defeating the novelty of OSCR [8–11]. Further, the existing FPGA generates signals which results in the required synchronization without any performance degradation.

Nevertheless, referring to the block diagram in Figure 6, an external source can be used to achieve clock synchronization. This evaluation board has a maximum jitter of up to 50 ps which is better than the previous architecture discussed. The output signaling modes may be changed by software control independent from each other, which gives the option of trade-off between power consumption and drive strength versus frequency. The clock signals generated by the COTS evaluation board suffered from overshoot at high frequencies. When sampled at incorrect instance, this leads to significant synchronization error.

Orthogonal codes used for encoding/decoding had to be generated using FPGA. Hence, it was imperative to employ FPGA in any clocking architecture [8–11]. Further, multiple AD9576 distribution boards are required to achieve clock synchronization for a 64-element OSCR system. Apart from increased implementation complexity, this also requires synchronization of the distribution circuit. It should be noted that OSCR would behave in a similar manner if external clock synchronization scheme is implemented. Doing so will lead to increased power and cost, thus defeating the sole purpose of OSCR. It also should be noted that the scope of this paper

was to prove that clock synchronization was in fact possible without any additional components. Hence, this architecture is not suited for OSCR, since the drawbacks outweigh its low jitter performance.

2.3. Measurements and Results. Using the setup in Figure 7, various multibeam and multifrequency measurements are performed in the anechoic chamber using the constructed 8 channels [9, 15]. For the measurement, 8 RF boards were used as receivers and an encoder board was employed for performing on-site coding. A single ADC was used for digitization and the FPGA employed at the digital backend for post processing. For the setup shown in Figure 7, all the digital components are synchronized using the initial scheme. The measurements in the anechoic chamber included five different test cases, namely, $(\Theta_{s1,in}, \Theta_{s2,in}) = (0^\circ, 50^\circ), (-10^\circ, 40^\circ), (-20^\circ, 30^\circ), (-30^\circ, 20^\circ),$ and $(-40^\circ, 10^\circ)$ at frequencies $f_1 = 1350$ MHz and $f_2 = 1800$ MHz and SNR of 22 dB. The main goal of the measurement is to verify the faithful recovery of the signal phase after on-site coding/decoding. That is, OSCR was experimentally verified by accurately estimating location of two beams simultaneously and performing digital beamforming.

Table 1 gives the measured angle of incidence $(\Theta_{s,in})$ and decoded angles $(\Theta_{s,out})$ for various cases. For evaluating the OSCR, phases were estimated for two different incoming beams at different frequencies. To evaluate the accuracy of our approach, we computed the maximum phase error

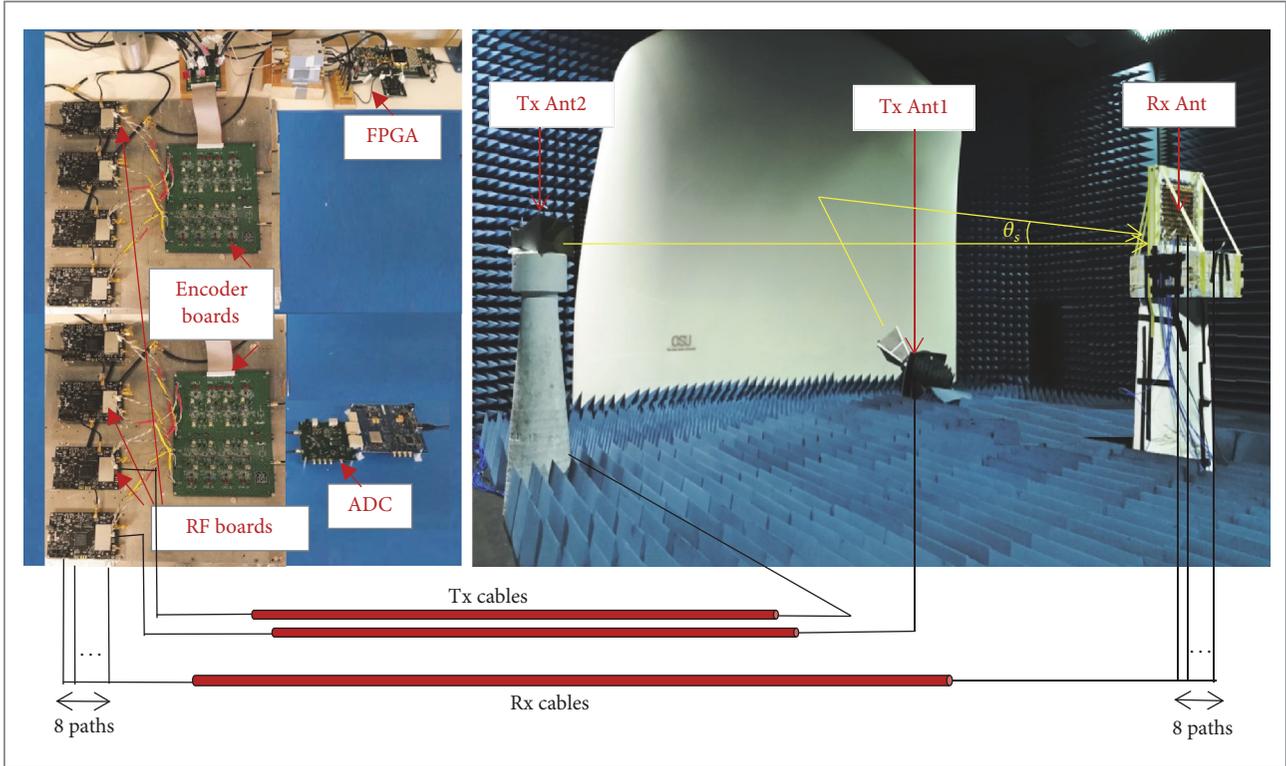


FIGURE 7: Eight-channel OSCR on the workstation outside the chamber. Also shown is the setup inside the anechoic chamber with two transmitting horn antennas at different locations and 64-element receiver array mounted on a rotating foam column.

TABLE I: Angle estimation.

Cases	Incident angle		Measured angle		Max error $\Theta_{s, \text{err}}$
	$\Theta_{s1, \text{in}}$	$\Theta_{s2, \text{in}}$	$\Theta_{s1, \text{out}}$	$\Theta_{s2, \text{out}}$	
1	0°	50°	0°	50°	0°
2	-10°	40°	-9.8°	41.8°	1.8°
3	-20°	30°	-19.6°	31.4°	1.4°
4	-30°	20°	-29.1°	20.4°	0.9°
5	-40°	10°	-38.1°	10.4°	1.9°

($\Theta_{s, \text{err}}$) defined as the maximum of $|\Theta_{s, \text{in}} - \Theta_{s, \text{out}}|$. Thus from Table 1, the maximum computed phase error using 8-channel OSCR is $\Theta_{s, \text{err}} = 1.9^\circ$. This error is mostly attributed to system hardware component nonidealities and hence can be removed via calibration. Also, it should be noted that multiple measurements were performed for repeatability and OSCR performance remained the same.

Further, from the phase distribution plot shown in Figure 8, phase variance (σ_φ) can be estimated. The latter is used to compute the SNR using [11],

$$\text{SNR} = \frac{1}{\sigma_\varphi^2}. \quad (1)$$

Based on the decoded data, the phase variance was found to be 0.083 rad, which corresponds to SNR of 21.6 dB. This

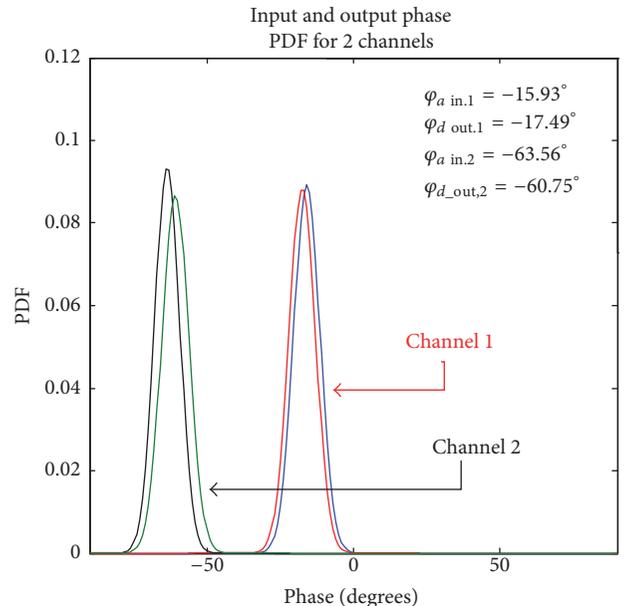


FIGURE 8: Phase distribution of decoded data of 2 channels used for quantifying the performance of OSCR.

results in 0.4 dB degradation as compared with the input SNR of 22 dB. As stated earlier, this degradation could be attributed to the hardware component nonidealities.

Due to hardware limitations of our OSCR system, BER was computed indirectly from the estimated SNR using,

$$\frac{E_b}{N_0} = \text{SNR} \times \frac{1}{\log_2 M}, \quad (2)$$

where E_b is the energy per bit, N_0 is the noise spectral density, and $M = 4$ for QPSK. Hence, BER for the corresponding E_b/N_0 can be estimated from the theoretical AWGN BER graph. From the theoretical BER plot for QPSK [9, 11], the BER is estimated to be $\sim 5e - 7$ for the computed E_b/N_0 of 10.8 dB. Thus, it can be concluded that with the above synchronization scheme, OSCR performance was not degraded.

3. Conclusion

Utilizing state-of-the-art FPGA circuits, we designed a purely digital clock synchronization system, without analog components (i.e., delay lines) that would require a time-consuming calibration and lead to increasing jitter for long delay ranges. Thus, the entire functionality, such as selecting trigger and clock source, defining trigger threshold level, setting delay, and defining the polarity and width of the output, is programmed using VHDL. Clock synchronization was implemented using the above setup for 8-channel OSCR system and various measurements and performance analysis were conducted. It was observed that system worked in a perfect synchronous manner and signals were detected and decoded with no synchronization errors. Alternatively, in this article we showed that implementing clock synchronization using COTS distribution board suffers from many drawbacks as compared to FPGA distribution. Further, when expanding OSCR to 64 channels, establishing clock synchronization using the above approach leads to more complex problems. It can be concluded that using FPGA as a master source for synchronization proves to be a better choice owing to its performance and the flexibility it offers.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Office of Naval Research, Arlington, VA, USA, under Grant no. N00014-16-1-2253.

References

- [1] A. A. Nasir, S. Durrani, H. Mehrpouyan, S. D. Blostein, and R. A. Kennedy, "Timing and carrier synchronization in wireless communication systems: a survey and classification of research in the last 5 years," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, article 180, 2016.
- [2] C. Winstead and M. E. Hamoui, "Reducing clock jitter by using Muller-C elements," *IEEE Electronics Letters*, vol. 45, no. 3, pp. 150-151, 2009.
- [3] P. Poshala and P. Shetty, "Synchronizing the giga-sample ADCs interfaced with multiple FPGAs," Application Report SLAA643, Texas Instruments, Dallas, Tex, USA, 2014.
- [4] K. Pekar and A. Oz, "Synchronizing sample clocks of a data converter array," Technical Article, Analog Devices, Norwood, Mass, USA, <http://www.analog.com/en/technical-articles/synchronizing-sample-clocks-of-a-data-converter-array.html>.
- [5] J. Dedić, A. Hasanović, D. Golob, and M. Pleško, "Extremely low-jitter FPGA based synchronization timing system," in *Proceedings of the IEEE Particle Accelerator Conference (PAC '07)*, pp. 296-298, Albuquerque, NM, USA, June 2007.
- [6] C. Y. Wu, J. Chen, K. H. Hu et al., "RF reference distribution and timing system for the taiwan photon source," in *Proceedings of the 10th European Workshop on Beam Diagnostics and Instrumentation for Particle Accelerators (DIPAC '11)*, Hamburg, Germany, May 2011.
- [7] S. Li, W. Wan, and L. Pan, "A practical method of clock synchronization in 2-out-of-3 system," in *Proceedings of the IET International Communication Conference on Wireless Mobile and Computing (CCWMC '11)*, pp. 304-306, Shanghai, China, November 2011.
- [8] E. A. Alwan, A. A. Akhiyat, W. Khalil, and J. L. Volakis, "Analytical and experimental evaluation of a novel wideband digital beamformer with on-site coding," *Journal of Electromagnetic Waves and Applications*, vol. 28, no. 12, pp. 1401-1429, 2014.
- [9] S. B. Venkatakrishnan, D. K. Papantonis, A. A. Akhiyat, E. A. Alwan, and J. L. Volakis, "Experimental validation of on-site coding digital beamformer with ultra-wideband antenna arrays," *IEEE Transactions on Microwave Theory and Techniques*, vol. 99, pp. 1-10, 2017.
- [10] S. B. Venkatakrishnan, A. A. Akhiyat, E. A. Alwan, and J. L. Volakis, "Dual-band validation of on-site coding receiver using ultra-wideband antenna array at C, X and Ku-bands," in *Proceedings of the 2016 IEEE Antennas and Propagation Society International Symposium (APSURSI '16)*, pp. 1655-1656, Fajardo, Puerto Rico, July 2016.
- [11] E. A. Alwan, S. B. Venkatakrishnan, A. A. Akhiyat, W. Khalil, and J. L. Volakis, "Phase error evaluation in a two-path receiver front-end with on-site coding," *IEEE Access*, vol. 3, pp. 55-63, 2015.
- [12] J. M. Castillo-Secilla, F. León, J. Olivares, and J. M. Palomares, "Multiplicative composition of clock-skew components for improving time synchronisation," *IEEE Electronics Letters*, vol. 51, no. 13, pp. 991-993, 2015.
- [13] S. B. Venkatakrishnan, E. A. Alwan, and J. L. Volakis, "Clock synchronization challenges for on-site coding digital beamformer," in *Proceedings of the 2017 IEEE International Symposium on Antennas and Propagation (APSURSI '17)*, San Diego, Calif, USA, July 2017.
- [14] W. Kester, "Converting oscillator phase noise to time jitter," Tutorial MT-008, Analog Devices, Norwood, Mass, USA, <http://www.analog.com/media/en/training-seminars/tutorials/MT-008.pdf>.
- [15] S. B. Venkatakrishnan, A. Akhiyat, E. A. Alwan, and J. L. Volakis, "Multiband and multibeam direction of arrival estimation using on-site coding digital beamformer," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 2332-2335, 2017.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

