are required, and the use of well-formed RNS bases results in efficient realization of modular addition and multiplication required in the RNS Montgomery multiplication. Another advantage of the proposed architecture is the simple adder-based implementation of the residue- to-binary converter, compared to that in [4] which requires multiplications by large operands. The proposed ECPM architecture is implemented on Xilinx VirtexE, Virtex 2 Pro, and Altera Startix II to achieve a fair one-to-one comparison with the state-of-the-art implementations in [4], [6], [7], [8], and [5]–[15] presented in Table II. In [8], although moduli of smaller word-length are employed, a large number of moduli are used and thus the RNS base extensions are complex resulting in worse performance, as shown in Table II.

VII. Conclusion

In this brief, a new architecture for ECPM was presented, based on efficient RNS bases. Arithmetic-friendly RNS bases in conjunction with a pipelined design for RNS Montgomery multiplication resulted in noticeable improvements in terms of speed and area. Two versions of fast and area-efficient designs for RNS Montgomery multiplication in six- and four-stage pipelined architectures were presented. Comparisons with state-of-the-art implementations proved the efficiency of the proposed architecture.

References

[1] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comp.*, vol. 48, no. 177, pp. 203–209, 1987.

[2] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Adv. Cryptology LNCS*, 1986, pp. 47–426.

[3] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[4] D. M. Schinianakis, A. P. Fournaris, H. E. Michail, A. P. Kakarountas, and T. Stouraitis, "An RNS implementation of an F_p elliptic curve point multiplier," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 6, pp. 1202–1213, Jun. 2009.

[5] C. J. McIvor, M. McLoone, and J. V. McCanny, "Hardware elliptic curve cryptographic processor over $GF_{(P)}$," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 9, pp. 1946–1957, Sep. 2006.

[6] G. Orlando and C. Paar, "A scalable $GF_{(P)}$ elliptic curve processor architecture for programmable hardware," in *Proc. Workshop Cryptograph. Hardware Embed. Syst. LNCS*, 2001, pp. 348–363.

[7] S. B. Ors, L. Batina, B. Preneel, and J. Vandewalle, "Hardware implementation of an elliptic curve processor over $GF_{(P)}$," in *Proc. IEEE Appl.-Specific Syst. Arch. Process.*, Jun. 2003, pp. 433–443.

[8] N. Guillermin, "A high speed coprocessor for elliptic curve scalar multiplications over F_p," in *Proc. CHES 12th Int. Conf. Cryptograph. Hardware Embed. Syst.*, 2010, pp. 48–64.

[9] S. Kawamura, M. Koike, F. Sano, and A. Shimbo, *Cox Rower Architecture for Fast Parallel Montgomery Multiplication*. New York: Springer-Verlag, 2000, pp. 523–538.

[10] J. C. Bajard, M. Kaihara, and T. Plantard, "Selected RNS bases for modular multiplication," in *Proc. IEEE 19th Int. Symp. Comput. Arith.*, Jun. 2009, pp. 25–32.

[11] K. Navi, A. S. Molahosseini, and M. Esmaeildoust, "How to teach residue number system to computer scientists and engineers?" *IEEE Trans. Edu.*, vol. 54, no. 1, pp. 156–163, Feb. 2011.

[12] J. Bajard, L. Didier, and P. Kornerup, "An RNS Montgomery's modular multiplication algorithm," *IEEE Trans. Comput.*, vol. 47, no. 2, pp. 167–178, Feb. 1998.

[13] P. V. A. Mohan, "RNS-to-binary converter for a new three-moduli set $\{2^{n+1}-1, 2^n, 2^n-1\}$," *IEEE Trans. Circuits Syst., II*, vol. 54, no. 9, pp. 775–779, Sep. 2007.

[14] J. Yuan and C.-T. Huang, "Elixir: High-throughput cost-effective dual-field processors and the design framework for elliptic curve cryptography," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 6, no. 11, pp. 1567–80, Nov. 2008.

[15] D. M. Schinianakis, A. P. Fournaris, A. P. Kakarountas, and T. Stouraitis, "An RNS architecture of an *F(p)* elliptic curve point multiplier," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 3369–3373.

Low-Power Correlation for IEEE 802.16 OFDM Synchronization on FPGA

Thinh H. Pham, Suhaib A. Fahmy, and Ian Vince McLoughlin

***Abstract*— This brief compares the use of multiplierless and DSP slice-based cross-correlation for IEEE 802.16d orthogonal frequency division multiplexing (OFDM) timing synchronization on Xilinx Virtex-6 and Spartan-6 field programmable gate arrays (FPGAs). The natural approach, given the availability of embedded DSP blocks on these FPGAs, would be to implement standard multiplier-based cross-correlation. However, this can consume a significant number of DSP blocks, which may not fit on low-power devices. Hence, we compare a DSP48E1 slice-based design to four different quantizations of multiplierless correlation in terms of resource utilization and power consumption. OFDM timing synchronization accuracy is evaluated for each system at different signal-to-noise ratios. Results show that even relatively coarse multiplierless coefficient quantization can yield accurate timing synchronization, and does so at high clock speeds. Multiplierless designs enjoy reduced power consumption over the DSP48E1 Slice-based design, and can be used where DSP Slice resources are insufficient, such as on low-power FPGA devices.**

***Index Terms*— Correlation, cognitive radio, field-programmable gate arrays (FPGA), IEEE 802.16 standards, orthogonal frequency division multiplexing (OFDM).**

I. Introduction

Orthogonal frequency division multiplexing (OFDM) is an effective modulation technique used in both wired and wireless communication systems. Particularly, thanks to the advantages of spectral efficiency and robustness to multipath fading, OFDM was specified for multiple applications in high bit-rate wireless transmission systems such as wireless local area networks adopted by IEEE 802.11 and metropolitan area networks in IEEE 802.16d. However, OFDM performance is sensitive to receiver synchronization. Frequency offset causes inter-subcarrier interference, and errors in timing synchronization can lead to inter-symbol interference [1]. Therefore, synchronization is critical for good performance in OFDM systems.

Much research has focused on improving OFDM synchronization performance and accuracy. Cyclic prefix (CP)-based methods were introduced [2]–[4] to determine frequency offset and symbol timing, but do not themselves find the start of a frame. To assist this, all OFDM frames begin with preamble symbols which can also be used to estimate the frequency offset [5]. This relies on the characteristic of a preamble symbol with two identical halves, using autocorrelation of the received signal, which can be computed iteratively at low cost and is robust to frequency offset. However, the metric used results in a plateau which leads to some uncertainty in determining the start of a frame. Work in [6]–[9] introduced modified timing metrics based on autocorrelation and the characteristic of specific preamble symbols to reduce the ambiguity of the plateau in finding the start of frame. However, the resulting autocorrelation operation is sensitive to additive white Gaussian noise (AWGN) and frequency selectivity.

Kishore and Reddy [10] presented an algorithm that requires knowledge of the time domain preamble in the receiver to compute a cross-correlation metric between the known and received preamble

Manuscript received February 2, 2012; revised June 6, 2012; accepted July 21, 2012. Date of publication September 7, 2012; date of current version July 22, 2013.

T. H. Pham is with Nanyang Technological University, 639798 Singapore, and also with the TUM-CREATE Centre for Electromobility, 138649 Singapore (e-mail: hung3@e.ntu.edu.sg).

S. A. Fahmy and I. V. McLoughlin are with Nanyang Technological University, 639798 Singapore (e-mail: sfahmy@ntu.edu.sg; mcloughlin@ntu.edu.sg).

Digital Object Identifier 10.1109/TVLSI.2012.2210917

symbols. This can accurately determine the start of frame even at a low signal-to-noise ratio (SNR). However, the cross-correlation operation requires complex computation. Kim and Park [11] proposed an accurate synchronization method based upon the preamble symbol specified in IEEE 802.16d using two separate computation processes: first, autocorrelation is computed for coarse symbol time offset (STO) and fractional carrier frequency offset (CFO) estimation to obtain more reliable frequency synchronization and to reduce hardware cost; second, the fine STO and the integer CFO are estimated by performing cross-correlation between the received samples and known preamble.

Autocorrelation-based techniques are preferred for implementation on FPGA because of their lower hardware costs. Dick and Harris [12] reported on the FPGA implementation of an OFDM transceiver. They showed that FPGAs, with their highly parallel architecture, are suitable for the implementation of OFDM transceivers. Wang *et al*. [13] also presented an FPGA implementation of an OFDM-WLAN synchronizer. In this brief, the timing synchronization is obtained by double autocorrelation based on short training symbols that allows a reduction in the hardware cost on FPGA. Fort *et al*. [14] compared the performance and complexity of FPGA implementation of autocorrelation and cross-correlation algorithms. Their results show that the accuracy of cross-correlation algorithms is better than that of autocorrelation algorithms. However, the accuracy of cross-correlation comes at significant hardware cost. Despite proposing a new cross-correlator implementation presented in [14] to reduce hardware cost compared to a classic cross-correlation approach, it is still at least five times more complex to implement than autocorrelation, because of the fact that several multipliers are required.

Cross-correlation between received samples and a known preamble can achieve highly accurate timing synchronization; however, this requires significant resources. Multiplierless correlators for timing synchronization were introduced in [15], designed for IEEE 802.11a OFDM frames, based on expressing the correlator coefficients as sums of powers of 2 that only require shift and add operations. The authors identified a correlator that eliminates the need for multiplication, requiring only 26 additions/subtractions per output while maintaining similar synchronization accuracy as a multiplier-based implementation.

OFDM is one of the main candidate modulation schemes for cognitive radios, and we believe FPGAs are an ideal platform owing to their flexibility [16]; hence optimizing this functionality is the key. Modern FPGAs contain various resources that can be used to implement cross-correlation. This brief presents the design of several correlators for timing synchronization with preamble symbols based upon IEEE 802.16d. We compare designs using specialized digital signal processing (DSP) Slices to a multiplierless approach on Xilinx Virtex-6 and Spartan-6 FPGA devices. Attempting to implement correlation on FPGAs without considering and designing the underlying architecture results in a highly inefficient implementation. In this brief, we show optimized FPGA designs, built to fit the FPGA architecture, and evaluate performance, timing synchronization accuracy, resource utilization, and power consumption, to understand whether a multiplier-based mapping is beneficial when using modern devices.

II. Implementation of Correlators

The downlink preamble in IEEE 802.16d [17] contains two consecutive OFDM symbols, as shown in Fig. 1. The short symbol consists of four identical 64-sample fragments in time, preceded by a CP. This is followed by the long symbol which contains two repetitions of a 128-sample fragment and a CP [17].

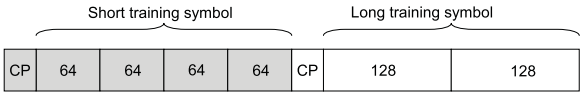

Fig. 1. Downlink preamble symbols for IEEE 802.16.

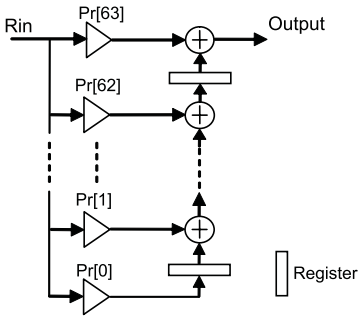

Fig. 2. Transpose direct form of the correlator.

The 64 samples in the short symbol are used to perform cross-correlation with the received samples for timing synchronization. Therefore, the correlators are designed to compute cross-correlation with 64 constant coefficients. In this brief, we explore two approaches to implement such correlators. The first is based on Xilinx Virtex-6 FPGA DSP48E1 Slices, which is the standard approach to such an implementation. The second uses multiplierless correlation implemented on both a Xilinx Virtex-6 and a low-power Xilinx Spartan-6 device. Both designs are designed to receive real and imaginary 16-b samples in Q1.15 fixed-point format. The output is the sum of 64 coefficient products, with each smaller than unity. So, the complex output words are in 21-b fixed-point Q6.15 format.

If such a design were implemented blindly, with no consideration for the FPGA architecture, the synthesis tools would infer the use of embedded DSP blocks for multiplication, but would likely achieve poor timing because of the inability to optimize the use of the DSP block and external logic elements. The DSP48E1 primitives on Xilinx Virtex-6 and later FPGAs are highly flexible and have additional circuitry within them that enables the design of optimized datapaths [18]. However, this must be done manually through writing the code in a particular style. Otherwise, the synthesis tools cannot always infer the most efficient structure [19]. In our design, we have taken into account the internal structure of the DSP block, and made the design as lean as possible. The multiplierless design is specified entirely manually as a low-level structural description.

A. Design of DSP48E1-Based Correlator

The DSP48E1 Slice inside the Virtex-6 contains a multiplier followed by a configurable arithmetic unit to provide many independent functions, e.g., multiply, multiply-accumulate, multiply-add, three-input add, and more [20]. It also allows the datapath to be configured for various input combinations and register stages; a three-stage pipeline offers maximum performance. Since the DSP Slice is designed to mirror the structure of an FIR filter tap, it is ideally suited to implement correlation, and would hence be the method of choice for this application. Our first design uses non-pipelined DSP48E1 Slices in transpose direct form, as shown in Fig. 2, with 64 coefficients, Pr corresponding to the 64 complex conjugated values of samples in the preamble. The output of the FIR filter in transpose direct form can be expressed as

$$\text{Output} = Pr[63]Ri + z^{-1}(Pr[62]Ri + z^{-1}(Pr[61]Ri + \cdots + z^{-1}(Pr[0]Ri)\cdots)). \quad (1)$$

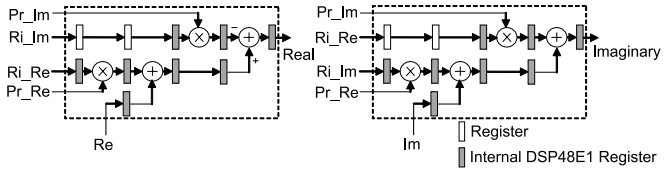

Fig. 3. Pipeline structure of the complex number multiply-add.

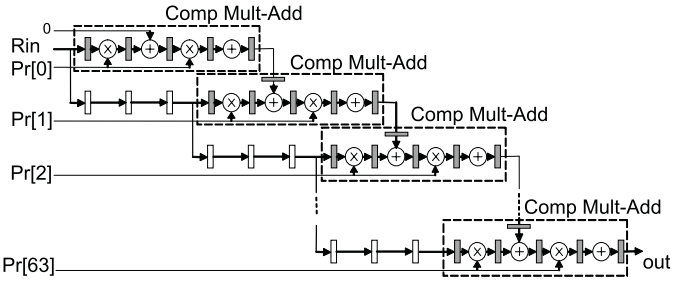

Fig. 4. Pipeline structure of correlator using DSP48E1 Slices.

The coefficients are precomputed according to the IEEE 802.16d standard. The second design spreads the complex multiply-adds in a five-stage pipeline, shown in Fig. 3, consisting of DSP48E1 Slices configured for three-stage internal pipelining. Ri_Re and Ri_Im are the real and imaginary parts of received sample, respectively. Pr_Re and Pr_Im similarly represent the complex conjugation of known preamble. The pipeline registers for the Pr_Re, Pr_Im are eliminated because they are considered to be of constant value. Re and Im are the real and imaginary parts of the previous multiply-add, MA_{n-1}. The output of these complex multiply-adds can be expressed as

$$\text{Output} = Ri\ \ Pr\ z^{-5} + z^{-4}\ MA_{n-1}. \tag{2}$$

Fig. 4 presents the pipeline structure of the correlator. The additional pipeline registers are required for handling the received sample. Adding pipeline registers should improve the performance significantly. The output of the pipelined correlator is

$$\begin{aligned}\text{Output} = {} & Pr[63]Ri(z^{-3})^{63}z^{-5} + z^{-4} \\ & \times (Pr[62]Ri(z^{-3})^{62}z^{-5} \\ & \quad + z^{-4}(Pr[61]Ri(z^{-3})^{61}z^{-5} + \cdots \\ & \quad + z^{-4}(Pr[0]Ri\ z^{-5})\cdots)) \\ = {} & (z^{-3})^{63}z^{-5}\ (Pr[63]Ri + z^{-1}(Pr[62]Ri + \cdots \\ & \quad + z^{-1}(Pr[0]Ri)\cdots)).\end{aligned} \tag{3}$$

B. Design of Multiplierless Correlator

The principle of multiplierless correlators is to represent coefficients and round them in the form of summed powers of 2. Hence, a shift-and-add is performed instead of multiplying by coefficients. It is expected that multiplierless correlation is more efficient, but with embedded hard multipliers in modern FPGAs, it is unclear whether they should still be considered favorable. Furthermore, synchronization accuracy must be considered. To explore this, four alternative multiplierless correlators are implemented using four coefficient sets with increasing degrees of rounding, to compare the cost and performance and evaluate against multiplier-based correlators. The coefficient sets are found by quantizing the 64 normalized preamble samples with quantizations of 1, 0.5, 0.25, and 0.125.

The proposed structure for multiplierless correlators is shown in Fig. 5. This structure is based on the transpose-direct form

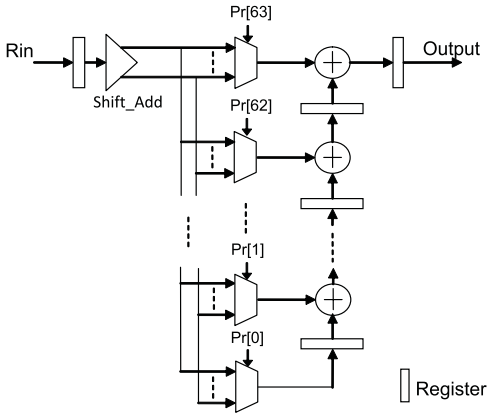

Fig. 5. Structure of multiplierless correlators.

TABLE I
IMPLEMENTATION REPORT ON VIRTEX6 AND SPARTAN6 DEVICES

Design	Occupied slices		DSP48E1s	Freqency (MHz)	
	V6	S6	V6	V6	S6
DSPc	742 (6%)	-	256 (88%)	119	-
DSPp	1110 (9%)	-	256 (88%)	398	-
ML1	661 (5%)	762 (6%)	0 (0%)	309	174
ML2	983 (8%)	1071 (9%)	0 (0%)	268	158
ML3	1191 (10%)	1257 (10%)	0 (0%)	234	136
ML4	1496 (12%)	1517 (13%)	0 (0%)	208	124

in Fig. 2. Instead of using multipliers to multiply input samples by coefficients, the Shift_Add block and multiplexers are used to perform the equivalent operation without an actual multiplication. But the Shift_Add block, multiplexers, and value of $Pr[n]$ are different depending upon the quantized coefficient set being used. The Shift_Add block performs shift and add on the received samples according to the degree of quantization that is applied. To optimize resources in the case of small numbers of bit quantization, one common Shift_Add block is used for all 64 coefficients instead of 64 separate Shift_Add blocks. This common Shift_Add block calculates all possible values for 64 coefficients. The multiplexers are used to select the corresponding values from Shift_Add to accumulate in order to generate the correlator output. These are based on the expressed coefficients $Pr[n]$ that are precomputed on the basis of quantizing the 64 preamble samples. Since the $Pr[n]$ values are constants, after synthesizing the design, the multiplexer is optimized as hard-wired logic, and the preamble cannot be changed. To support different OFDM preambles, the $Pr[n]$ could be stored in a register, and a real multiplexer used instead of hard-wired logic. This results in increased resource utilization but provides a more flexible solution.

C. Implementation Results

The designs presented were synthesized and fully implemented using Xilinx ISE 13.2, targeting Xilinx Virtex-6 (V6) and Spartan-6 (S6) devices. The results of implementation are reported in terms of the number of occupied slices, DSP48E1 Slices, and the maximum frequency, and are summarized in Table I.

DSPc and DSPp are correlator designs using DSP Slices in non-pipelined and pipelined structures, respectively. $ML1$, $ML2$, $ML3$, and $ML4$ are multiplierless correlators with coefficient quantizations of 1, 0.5, 0.25, and 0.125, respectively.

Table I reveals that the DSPp uses more logic slices because of its pipeline structure. The slices in DSP48E1-based designs are used

TABLE II
POWER CONSUMPTION AT 50 MHZ

Correlators	Quiescent (mW)		Dynamic (mW)		Total (mW)	
	V6	S6	V6	S6	V6	S6
DSPc	1312	-	846	-	2158	-
DSPp	1300	-	328	-	1628	-
ML1	1296	67	133	149	1429	216
ML2	1296	68	160	197	1456	265
ML3	1297	70	182	239	1479	309
ML4	1297	71	203	294	1500	365

for registers and route-thrus, while the slices in the multiplierless designs are mostly used as logic. The number of slices used in the multiplierless designs increases as the coefficient quantization becomes finer. The DSP48E1-based designs use 256 DSP Slices, 4 for each complex multiply plus 6%–9% of logic resources. The multiplierless designs use only logic to compute the cross-correlation with 64 complex coefficients. The total logic area is a small fraction of the whole device: around 5%–12% of total resources in the Virtex-6, and around 6%–13% of total resources in the equivalent Spartan-6. While Spartan-6 devices do include DSP Slices, their number is insufficient to implement the full 64-sample complex cross-correlation. This shows an ideal scenario where multiplierless correlation makes sense, and hence the motivation for this brief.

The maximum frequencies, reported after place and route, decrease for multiplierless designs according to the degree of coefficient quantization. Meanwhile the nonpipelined DSP48E1 design is slower than the multiplierless designs. However, the pipelined DSP48E1 design can achieve higher-frequency.

A post-place-and-route simulation in ModelSim was used to estimate the power consumption of the system using the Xilinx XPower tool. Table II shows the power dissipation of the designs running at 50 MHz. The DSP48E1-based correlators consume more power than the multiplierless correlators, but this is due primarily to increased dynamic power when using the DSP48E1s on the Virtex-6. The dynamic power of the non-pipelined DSP48E1-based correlator DSPc is greatest at 846 mW, but pipelining reduces this by a factor of more than 2.5 times, because of reduced switching activity between the multiplier and adder. The dynamic power of the multiplierless designs increases from 133 to 203 mW on Virtex-6 and from 149 to 294 mW on Spartan-6 as finer coefficient quantization is used. It is important to note that the quiescent power of the Spartan-6 is much lower by design. Hence, we can see that using this multiplierless technique allows us to implement synchronization on a Spartan-6 device, where a multiplier-based design is not possible, saving significant power and sacrificing little in terms of accuracy.

We also investigated how the total power consumption varies with the frequency, as shown in Fig. 6. As the frequency increases, the finer quantizations and DSP48E1-based designs begin to consume proportionally more power. Overall, multiplierless designs on the Spartan-6 consume 75%–85% less power than the same designs on the Virtex-6, and a 0.25 quantization design on the Spartan-6 consumes 81%–85% less power than the DSP48E1-based design on a Virtex-6.

The DPSc implementation represents how a "blind" design would be mapped. Our architecture-aware designs show significantly better performance, reduced area, and reduced power consumption.

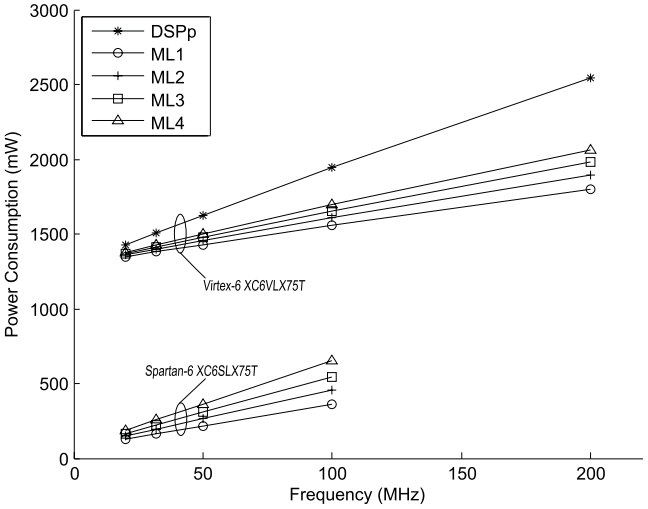

Fig. 6. Power consumption of the designs.

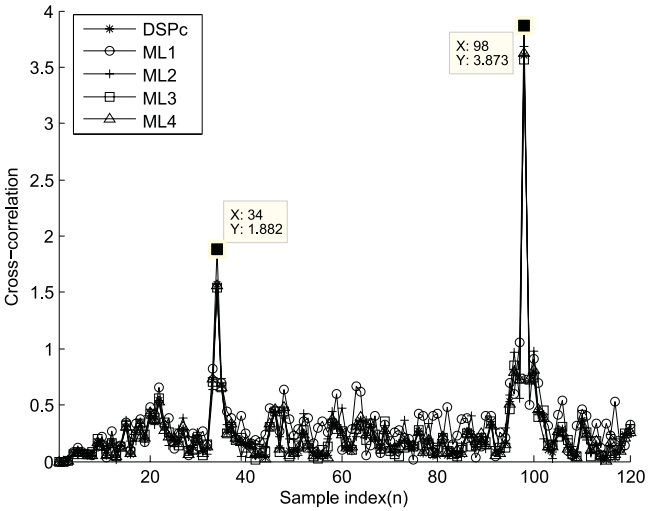

Fig. 7. Correlator output with SNR = 10 dB.

III. SIMULATIONS AND DISCUSSION

In order to validate our designs at the application level, we simulate them using ModelSim with an IEEE 802.16 OFDM frame created using MATLAB, including the preamble symbols, data symbols, and effects of an AWGN channel. Cross-correlation results using the correlator designs are compared to the corresponding results in MATLAB to verify the correctness of implementation. To evaluate the accuracy of timing synchronzsation achievable by these designs, the correlation outputs are plotted in Fig. 7 for random data frames at 10-dB SNR. The output of each correlator is slightly different because of rounding, but the timing synchronization depends upon the location of the peaks being at the position of the preamble. All the correlator designs achieve this most of the time, as shown at indices 34 and 98 for the CP samples and the first preamble samples, respectively, for a single frame.

In order to evaluate the synchronization accuracy of these approaches, we simulate 10 000 correlation operations in an AWGN channel with a detection strategy as follows. First, find the first peak $P1$ over 64 samples. Next, find the second peak $P2$ in the next 64 samples from the first peak and compute average value avg of the samples between two peaks. If $(P1 - \text{avg}) \leq 0.75 \times (P2 - \text{avg})$, the start of frame is detected and the cor-

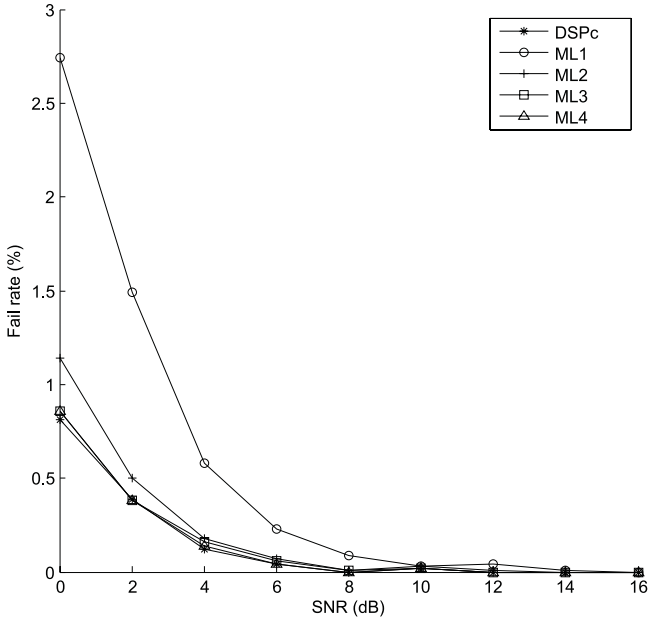

Fig. 8. Detection failure rate.

rectness of the position can be checked. It should be noted that, in all cases, the peaks were known to be located within the two search regions and that the detection strategies described above were compatible with those in [10] and [15]. Fig. 8 plots the failure rate for varying AWGN SNRs and show that the designs are able to accurately detect the start of frame even under low SNR conditions. The failure rate of *ML*1 is the highest, as expected, due to the coarse quantization. For SNRs above 4 dB, the failure rates of *ML*2, *ML*3, *ML*4, and $DSPc$ differ by less than 0.05% from each other. This suggests that sacrificing accuracy by using multiplierless cross-correlation is feasible and has only negligible impact on synchronization accuracy. Combined with the results in the previous section, we can be confident that low-power FPGAs, such as the Spartan-6, with insufficient resources for multiplier-based synchronization correlation, are still feasible for implementing robust OFDM receivers.

IV. Conclusion

The DSP48E1 Slices on modern Virtex-6 FPGA devices seem to offer the ideal resource for implementing correlation-based frame synchronizers. However, as we have discovered, in the context of synchronization for IEEE 802.16 OFDM systems, simplified multiplierless designs offer comparable synchornization performance. While the DSP48E1-based correlators can obtain higher clock speeds, this is possible only through a detailed pipelined design. Furthermore, their power consumption and resource usage are considerably greater. Since low-power low-cost devices such as the Xilinx Spartan-6 do not include sufficient DSP Slices, this suggests the adoption of multiplierless designs for low-power implementations in such devices, or whenever only insufficient DSP Slices are available. An additional benefit of multiplierless correlation is that it can be used on any FPGA architecture. We have shown that, while very low quantization resolution does impact synchronization performance, with a quantization step size of just 0.5, the synchronization accuracy is on par with multiplier-based correlation. Multiplierless correlation on a Spartan-6 can save over 85% power compared to a DSP Slice design on a Virtex-6 FPGA. Both the multiplierless and the DSP Slice-based correlators significantly exceed the timing requirements for IEEE 802.16d synchronization.

References

[1] L. Hanzo and T. Keller, *OFDM and MC-CDMA: A Primer*. New York: Wiley, 2006.

[2] J.-J. Beek, M. Sandell, M. Isaksson, and P. O. Borjesson, "Low-complex frame synchronization in OFDM systems," in *Proc. IEEE Int. Conf. Univ. Personal Commun.*, Nov. 1995, pp. 982–986.

[3] N. Lashkarian and S. Kiaei, "Class of cyclic-based estimators for frequency-offset estimation of OFDM systems," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2139–2149, Dec. 2000.

[4] T. Fusco and M. Tanda, "ML-based symbol timing and frequency offset estimation for OFDM systems with noncircular transmissions," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3527–3541, Sep. 2006.

[5] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997.

[6] H. Minn, M. Zeng, and V. Bhargava, "On timing offset estimation for OFDM systems," *IEEE Commun. Lett.*, vol. 4, no. 7, pp. 242–244, Jul. 2000.

[7] L. Schwoerer, "VLSI suitable synchronization algorithms and architecture for IEEE 802.11a physical layer," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 5. 2002, pp. 721–724.

[8] A. Langowski, "Fast and accurate OFDM time and frequency synchronisation," in *Proc. Int. Symp. Wirel. Commun. Syst.*, Oct. 2007, pp. 86–90.

[9] L. Nasraoui, L. Atallah, and M. Siala, "A very efficient time and frequency synchronization method for OFDM systems operating in AWGN channels," in *Proc. Int. Conf. Commun. Netw.*, Nov. 2010, pp. 1–5.

[10] C. N. Kishore and V. U. Reddy, "A frame synchronization and frequency offset estimation algorithm for OFDM system and its analysis," *EURASIP J. Wirel. Commun. Netw.*, vol. 2006, pp. 1–16, Mar. 2006.

[11] T.-H. Kim and I.-C. Park, "Low-power and high-accurate synchronization for IEEE 802.16d systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 12, pp. 1620–1630, Dec. 2008.

[12] C. Dick and F. Harris, "FPGA implementation of an OFDM PHY," in *Proc. Conf. Rec. Asilomar Signals Syst. Comput.*, vol. 1. Nov. 2003, pp. 905–909.

[13] K. Wang, J. Singh, and M. Faulkner, "FPGA implementation of an OFDM-WLAN synchronizer," in *Proc. IEEE Int. Workshop Electron. Des. Test Appl.*, Jan. 2004, pp. 89–94.

[14] A. Fort, J.-W. Weijers, V. Derudder, W. Eberle, and A. Bourdoux, "A performance and complexity comparison of auto-correlation and cross-correlation for OFDM burst synchronization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2. 2003, pp. 341–344.

[15] K.-W. Yip, Y.-C. Wu, and T.-S. Ng, "Design of multiplierless correlators for timing synchronization in IEEE 802.11a wireless LANs," *IEEE Trans. Consumer Electron.*, vol. 49, no. 1, pp. 107–114, Feb. 2003.

[16] S. A. Fahmy, J. Lotze, J. Noguera, L. Doyle, and R. Esser, "Generic software framework for adaptive applications on FPGAs," in *Proc. IEEE Symp. Field Programm. Custom Comput. Mach.*, Apr. 2009, pp. 55–62.

[17] *IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed Broadband Wirel. Access Systems*, IEEE Standard 802.16, 2004.

[18] H. Y. Cheah, S. A. Fahmy, D. L. Maskell, and C. Kulkarni, "A lean FPGA soft processor built using a DSP block," in *Proc. ACM/SIGDA Int. Symp. Field Programm. Gate Arrays*, 2012, pp. 237–240.

[19] B. Ronak and S. A. Fahmy, "Evaluating the efficiency of DSP block synthesis inference from flow graphs," in *Proc. Int. Conf. Field Programm. Logic Appl.*, 2012.

[20] *Virtex-6 FPGA DSP48E1 Slice: User Guide*, Xilinx Inc., San Jose, CA, 2011.