# Energy-Efficient Processor for Blind Signal Classification in Cognitive Radio Networks

Eric Rebeiz, Student Member, IEEE, Fang-Li Yuan, Student Member, IEEE, Paulo Urriza, Student Member, IEEE, Dejan Markovic, Member, IEEE, Danijela Cabric, Member, IEEE

Abstract—Blind modulation classification is of vital importance in spectrum surveillance applications and future heterogeneous wireless networks. In standardized wireless systems, modulation classification can be performed through exhaustive search of known signal features. Most commonly used classifiers are based on the detection of cyclostationary features, which are secondorder moments of a signal, related to its carrier and symbol rate. However, when the signal parameters are unknown, an exhaustive search for cyclostationary features is energy inefficient due to high computational complexity. In this paper, we present a reconfigurable processor architecture that can blindly classify any linearly modulated signal (M-QAM, M-PSK, M-ASK, and GMSK) in addition to multi-carrier signals and spread spectrum signals. The contributions of this work are twofold. First, we analyze the complexity tradeoffs among different dependent signal processing kernels in order to minimize the total processing time and energy. Second, we optimize the processor architecture by the co-design methodology to enhance block reusability and reconfigurability. The proposed processor has been verified and synthesized in a 40-nm CMOS technology with core area of 0.06 mm<sup>2</sup> and power dissipation of 10 mW under 0.9 V supply voltage at 500 MHz. Under a 500 MHz wide-band signal at 10 dB SNR, a complete blind classification process consumes 10.37  $\mu$ J to meet 95% of classification accuracy.

## I. INTRODUCTION

**B** LIND modulation classifiers have numerous applications in current and future wireless networks. From an electronic surveillance point of view, military applications of blind modulation classifier include tracking the spectrum activity of specific users (often interferes or jammers) and learning their modulation classes. Blind modulation classification is therefore vital to electronic countermeasures in such hostile environments.

Additionally, with the recent deployment of heterogeneous networks (HetNet) such as Long Term Evolution (LTE), modulation classification becomes part of interference management [1], [2]. Multi-user detection is performed to support multiple overlapping transmissions in time and (or) frequency. Knowledge of the modulation type by means of modulation classification [3] is necessary to demodulate the interfering signal [4]. However, this application assumes that the transmitted signals are standard-compliant. In the future, as a result of spectrum under-utilization, Cognitive Radios (CRs) [5] will adaptively change their transmission parameters and modulation schemes in order to opportunistically access the unused spectrum holes. In such future wireless applications, demodulation of these adaptively modulated signals would require blind modulation classification. For these highly adaptive radios, information about transmit parameters cannot be assumed. As a result, blind modulation classification approaches are of significant research interest.

A survey of commonly used modulation classifiers is given in [6] and in the references within. The authors of [7] have proposed a hierarchical modulation classifier based on cumulants which are higher-order moments of the received information symbols. This algorithm requires perfect timing synchronization to extract information symbols, and is sensitive to the imperfect knowledge of the Signal to Noise Ratio (SNR). On the other hand, some modulation classification algorithms operate on over-sampled signals. Among such classifiers are cyclostationary-based modulation classifiers which classify signals based on the cyclostationary features [8]. For linearly modulated signals, these cyclostationary features are a function of the signal's symbol rate and carrier frequency. However, the main challenge of blind modulation classification is the absence of *a priori* information about the transmit parameters. As a result, the search for the features used for modulationtype classification<sup>1</sup> becomes very computationally demanding. One approach to efficiently solve blind classification is to first estimate signal parameters and then use cyclostationary based classifiers. In [9], a blind cyclostationary modulation classifier is proposed where the cyclostationary features are estimated using infinite number of samples. However, this approach is energy inefficient and cannot be used for real-time classification.

From the architectural point of view, although various *non-blind* classification algorithms have been studied and even implemented in Digital Signal Processing (DSP) [10] and Software-Defined Radio (SDR) platforms [11], an efficient silicon realization that classifies multi-carrier, spread spectrum, and linearly modulated signals was never realized before. In addition, these classifiers require prior knowledge of the targeted signals, which make them unsuitable for real-time blind classifiers. In order to achieve high energy efficiency, realization of Application-Specific Integrated Circuits (ASICs) is desirable. However, due to diversity of modulation classes and algorithms for their classification, a heuristic ASIC design equipped with multiple dedicated modules – one for each signal class – would result in large area and suboptimal energy consumption due to the difficulty of hardware sharing.

In this paper, we propose an implementation with high functional diversity and energy/area efficiency. By jointly considering the algorithm and architecture layers, we first select

<sup>&</sup>lt;sup>1</sup>We use the term *modulation type* to refer to the modulation scheme of the signal (e.g. QAM, PSK), and the term *modulation level* to refer to its modulation order (e.g. 4-QAM, 16-QAM).

 TABLE I

 Design Specifications of the Proposed Modulation Classifier.

Variables	Specifications			
Modulation Types	M-QAM, M-PSK, M-ASK, GMSK, OFDM, DSSS			
Probability of Correct Classification	$\geq 0.95$			
Energy Budget	15 μJ			
Proc. Time Budget	2 ms			
Channel Bandwidth	500 MHz			
Frequency Resolution	12.5 KHz			
Signal Bandwidth	$\leq$ 500 MHz			
Minimum SNR	10 dB			

computationally efficient parameter estimation and modulation classification algorithms. We then exploit the functional similarities between algorithms to build a processing architecture that maximizes hardware utilization. In addition, we carefully analyze the processing strategy of the processor in order to minimize the overall consumed energy.

The design specifications of the proposed classifier are summarized in Table I. We consider a minimum SNR of 10 dB, which is reasonable for classification of interferers in multiuser detection and blind signal demodulation applications. We note that we do not estimate the received SNR in the proposed processor. Instead, we guarantee that when the SNR exceeds 10 dB, the proposed processor will correctly classify the received signal with a minimum probability of 95%. The frequency resolution is set to 12.5 KHz in order to allow a fine spectral resolution to detect narrowband interferers. The classifier should identify multi-carrier, spread spectrum and linearly modulated signals correctly with a probability of 95%. Within the linearly modulated signals, the classifier should also distinguish the modulation types given in Table I. The proposed processor needs to meet an energy constraint of 15  $\mu$ J and a processing time of 2 ms.

This paper is organized as follows. Section II presents the overall receiver architecture and the design challenges in blind classification. Section III describes the low-complexity signal processing modules implemented in the processor. In Section IV, we present the proposed energy optimization methodology, in which we analyze the tradeoffs among dependent blocks of the architecture. A reconfigurable architecture for the classification processor is proposed in Section V. Section VI discusses the hardware emulation for functionality demonstration of the proposed architecture. Section VII concludes the paper.

# **II. DESIGN CONSIDERATIONS**

This section describes the overall receiver architecture and shows how the proposed modulation classification processor fits as part of a wideband receiver chain. We then describe the challenges in blind modulation classification and give an intuitive explanation behind the tradeoffs among different blocks of the proposed processor.

# A. System Model

We illustrate in Fig. 1 the top-level block diagram of the blind signal classifier. At the beginning, the RF front-end (not

shown) filters and downconverts a 500 MHz spectrum to baseband. The signal is then sampled and digitized for baseband processing. The digital baseband part starts with a sensing engine, referred to as the band segmentation, that detects the presence of one or more signals in the wideband channel in the presence of Additive White Gaussian Noise (AWGN) [12]. The detection is based on energy detection which estimates the spectrum of the received signal. The sensing time and threshold for detection are adjusted to meet the desired probability of detection and false alarm. The supported signals that can be classified could be of any modulation type given in Table I with bandwidth greater than 12.5 KHz, and could be located at any carrier frequency. Since this paper deals with the design of signal classification processor, we assume that the signal has already been detected. Identifying the presence of a signal during band segmentation inherently results in coarse estimates of the signal's carrier frequency and symbol rate. Using the coarse transmit parameters, the detected signal is down-converted and filtered using a reconfigurable Cascade-Integrator-Comb (CIC) filter [13]. The output of the CIC filter is fed to the modulation classifier to identify the modulation type of the signal. In the event of detecting multiple signals in the wideband channel, each signal is downconverted and processed by the CIC filter sequentially.

This work focuses on the design of an energy-efficient modulation classifier, which detects the types of signals using the optimized tree-based approach shown in Fig. 2. The proposed modulation type classifier is based on second-order cyclostationary properties of the received signal and therefore does not distinguish among different levels of a given modulation type. In particular, M-PSK (M>2) and M-QAM signals exhibit the same cyclostationary features [8]. Therefore, the modulation-type classifier can distinguish among three different classes of single-carrier modulation types: Class  $1 = \{M-\}$ PSK (M > 2), M-QAM}, Class 2 = {M-ASK}, and Class  $3 = \{GMSK\}$ . However, once the modulation type is found, the signal can then be fed to a modulation-level classifier. In our earlier work, we developed a low-complexity modulationlevel classifier [14] based on the distribution distance test that chooses the modulation level whose cumulative distribution function (CDF) is closest the received symbols CDFs. The performance of the proposed modulation-level classifier has been compared in hardware experiments [15] against the wellknown cumulants classifier [7]. Although the modulationlevel classifier can be implemented as part of the proposed processor, it is not considered in this work due to its very low computational complexity and consumed energy. The estimated energy consumed by the modulation-level classifier is around 15 nJ at a SNR of 10 dB, which is a negligible fraction of the 15  $\mu$ J energy budget.

#### B. Design Challenges

The objective of the proposed classifier is to minimize its consumed energy while achieving the required probability of correct classification. The energy minimization is achieved by 1) selecting and developing computationally efficient algorithms, and 2) by minimizing the total classification time



Fig. 1. Energy-time breakdown of the processing kernels of the proposed processor.



Fig. 2. Signal classification tree showing the possible modulation classes recognized by the proposed processor.

while meeting the classification accuracy of 95%. Although existing maximum-likelihood-based algorithms [16], [17] can meet the classification requirement, their computational complexity results in power and/or delay requirements that cannot be tolerated in real-time operating radios. In addition, blind modulation classifiers require the estimation of the signal's transmit parameters, adding to the overall complexity of the receiver. Therefore, our objectives in meeting the specifications are twofold: 1) developing low-complexity algorithms that meet the classification probability, 2) minimize the processing times of all the blocks in order to satisfy the energy budget.

As a result of the 12.5 KHz resolution of the band segmentation, the coarse estimates of the symbol rate and carrier frequency obtained from the band segmentation have estimation errors on the orders of thousands of parts per million. For instance, as a result of the transmit filter roll-off, the coarse symbol rate estimate of a 3 MHz signal can vary between 3 and 3.5 MHz, yielding an estimation error of  $1.6 \times 10^5$  ppms. As was shown in [18], [19], the features used for modulationtype classification degrade under large estimation errors of the cyclic frequencies. As a result, classification probability cannot be met under such large offsets. Therefore, coarse estimates cannot be used directly for detection of cyclic features, and hence fine estimates of the transmit parameters are needed. To address this issue, our architecture includes symbol rate and carrier frequency estimation blocks referred to as the *pre-processors*. We show that there exists an inherent tradeoff between estimation accuracies of the transmit parameters and the classification accuracy that can be achieved, which will be analyzed in Section IV.

From the architectural point of view, the design of an energy-efficient hardware to detect a variety of signal classes is not an easy task. Directly implementing a set of low-complexity algorithms as aformentioned is infeasible if those algorithms possess too little functional similarity. The lack of commonality forces the hardware to have many non-reusable modules, creating the so-called dark silicon with dominating leakage energy in deep-submicron CMOS technology. Consequently, the energy efficiency and flexibility cannot be achieved by merely mapping algorithms, but by algorithm-architecture *co-design*.

# III. LOW-COMPLEXITY BLIND CLASSIFICATION ALGORITHMS

In this section, we present the proposed algorithmic hierarchical classification tree. The design hierarchy is based on both the level of *a priori* information that a block requires and its computational complexity. In particular, the blocks that do not require *a priori* information about the signal being classified are processed first. For instance, the multi-carrier classifier employs a totally blind low-complexity algorithm, and therefore can be performed first. This design methodology dictates the order in which the classification algorithms are performed as shown in Fig. 1. In the remainder of this section, we describe each of the blocks of our processor, and specify what design variables need to be optimized in order to meet the given accuracy and energy requirements.

#### A. Multi-Carrier Classification

This block differentiates between Multi-Carrier (MC) OFDM (Orthogonal Frequency Division Multiplexing) and Single Carrier (SC) signals. The MC classifier is based on the fourth-order cumulant  $C_{42}$  [20] which is a form of a Gaussianity test. The property of  $C_{42}$  is that, it tends to zero if the input samples are approaching Gaussian distribution. The  $C_{42}$  statistic of an OFDM signal, as a result, is close to zero since the OFDM is a mixture of a large number of sub-carrier waveforms. For other narrowband SC signals, the test statistic converges to a non-zero value, thereby making it possible to separate MC from SC signals without any information about the signal's carrier frequency and symbol rate. The fourthorder cumulant is computed as follows:

$$C_{42} = \frac{1}{N_m} \sum_{n=1}^{N_m} |x[n]|^4 - |C_{20}|^2 - 2C_{21}^2, \qquad (1)$$

where  $N_m$  are the number of samples used for distinguishing MC and SC signals,  $x[\cdot]$  is the vector of samples obtained from the CIC filter,  $C_{20} = \frac{1}{N_m} \sum_{n=1}^{N_m} x[n]^2$  and  $C_{21} = \frac{1}{N_m} \sum_{n=1}^{N_m} |x[n]|^2$ . The MC detection is a threshold-based test derived by comparing  $C_{42}$  to a threshold  $\gamma_m$ . Both  $N_m$  and  $\gamma_m$  are set based on the minimum SNR requirement of 10 dB, resulting in  $N_m = 90$  samples and  $\gamma_m = -0.63$  which guarantee a correct classification probability of ~ 96% for MC signals and a misclassification probability of ~ 3%.

## B. Center Frequency and Symbol Rate Preprocessor

When the signal is classified as an SC signal, its transmit parameters need to be estimated first. Both the *pre-processors* and the modulation-type classifier for SC signals rely on the Cyclic Auto-Correlation (CAC) function to detect their cyclostationary features. Under a finite number of samples N, the conjugate and the non-conjugate CACs can be computed respectively as follows:

$$\tilde{R}_{x^*}^{\alpha}(\nu) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] x^*[n-\nu] e^{-j2\pi\alpha n T_s}, \qquad (2)$$

$$\tilde{R}_{x}^{\alpha}(\nu) = \frac{1}{N} \sum_{n=0}^{N-1} x[n]x[n-\nu]e^{-j2\pi\alpha nT_{s}},$$
(3)

where  $\nu$  is the lag variable,  $T_s$  is the sampling period, and  $\alpha$  is the cyclic frequency to be detected. Note that the conjugate CAC in (2) is used to detect cyclic frequencies close to baseband, whereas the non-conjugate CAC in (3) is used to detect the cyclostationary features at cyclic frequency  $\alpha$  related to the carrier frequency. Different modulation classes can be differentiated via the cyclostationarity test because their CACs possess cyclic peaks at different locations of cyclic frequencies  $\alpha$ , which is a function of the symbol rate (1/T) and the carrier frequency ( $f_c$ ). Table II summarizes the locations of spectral peaks for the three targeted modulation classes in this work.

However, in blind classification scenarios, the estimated cyclic frequencies might not be equal to true cyclic frequencies. It was shown in [18], [19] that computing the CAC at  $\hat{\alpha} = (1 + \Delta_{\alpha})\alpha$ , where  $\alpha$  is the true cyclic frequency and  $\Delta_{\alpha}$ 

TABLE II Cyclic features for some modulation classes that occur for conjugate  $(\cdot)^*$  and non-conjugate CAC.

Modulation	Peaks at $(\alpha, \nu)$
Class 1	$(\frac{1}{T}, 0)^*$
Class 2	$\left  \left(\frac{1}{T}, 0\right)^*, (2f_c, 0), (2f_c \pm \frac{1}{T}, 0) \right $
Class 3	$(\frac{1}{T}, 0)^*, (2f_c \pm \frac{1}{2T}, 0)$

is the cyclic frequency offset (CFO), results in performance degradation in terms of the classification accuracy. The relation between the CAC at  $\hat{\alpha}$  and that at  $\alpha$  is given by

$$\left|\tilde{R}_{x}^{\hat{\alpha}}(0)\right| = \left|\tilde{R}_{x}^{\alpha}(0)\right| \times \left|\frac{\sin(\pi\alpha N T_{s}\Delta_{\alpha})}{N\sin(\pi\alpha T_{s}\Delta_{\alpha})}\right|.$$
 (4)

Therefore, under a non-zero CFO  $\Delta_{\alpha}$ , increasing the number of samples (N) does not improve the detection accuracy but instead degrade the cycliclostationary feature. This in turn motivates the need for accurate estimates of the transmit parameters in order to minimize the CFO  $\Delta_{\alpha}$  and improve the performance of the modulation-type classification.

With respect to the symbol rate estimation, we note that all SC modulation classes considered in this work exhibit a cyclostationary feature at cyclic frequency  $\alpha = 1/T$ . Therefore, detecting the presence of this cyclostationary feature would inherently estimate the symbol rate of the signal. The coarse estimate of the symbol rate from the band segmentation can be used to set the search window  $W_T$ , within which the cyclic peak at the symbol rate will be located. The detection of the cyclostationary feature at 1/T is therefore obtained by solving the following optimization problem:

$$\max_{\alpha_i \in \mathcal{W}_T} \left| \sum_{n=0}^{N_T - 1} |x[n]|^2 e^{-j2\pi\alpha_i nT_s} \right|,\tag{5}$$

where  $N_T$  is the number of samples per CAC computation used to estimate the signal's symbol rate.

Given that not all classes have the cyclostationary feature related to their carrier frequency, the CACs given in (2) and (3) cannot be directly used to estimate the signal's carrier frequency. Estimation of the carrier frequency of the incoming signal can be performed by detecting the cyclic feature at  $\alpha = 4f_c$  after squaring the incoming samples [21], [22]. We denote the search window by  $W_f$  within which the cyclic peak at  $4f_c$  occurs. The estimation is therefore obtained by solving the following optimization problem:

$$\max_{\alpha_i \in \mathcal{W}_f} \left| \sum_{n=0}^{N_f - 1} x[n]^4 e^{-j2\pi\alpha_i n T_s} \right|, \tag{6}$$

where  $N_f$  is the total number of samples per CAC computation used to estimate the signal's carrier frequency. Note that with increasing number of samples over which the CAC is computed, the noise is suppressed and the features of interest become prominent. As a result, both  $N_T$  and  $N_f$  are a function of the SNR of the received signal.

Solving the optimizations given in (5) and (6) requires infinite computational complexity. As a result, the search space for the maximum cyclic feature has to be discretized. We denote by  $\Delta_{\alpha_T}$  and  $\Delta_{\alpha_f}$  the resolutions for the symbol rate and carrier frequency estimators. As a result, there are two degrees of freedom in the design of each of the algorithms: 1) the step size  $\Delta_{\alpha_T}$  and  $\Delta_{\alpha_f}$  within the window  $\mathcal{W}_T$  and  $\mathcal{W}_f$  respectively, and 2) the number of samples  $N_T$  and  $N_f$  required for the computation of every CAC at the cyclic frequency  $\alpha_i$  of interest. The symbol rate and the carrier frequency estimation algorithms cannot yield estimation accuracies smaller than their respective step size  $\Delta_{\alpha_T}$  and  $\Delta_{\alpha_f}$ .

Also, the number of CAC computations required in (5) and (6) are equal to the cardinality of the discretized search windows  $S_T = \lceil W_T / \Delta_{\alpha_T} \rceil$  and  $S_f = \lceil W_f / \Delta_{\alpha_f} \rceil$  respectively. Given that both estimators use the CAC signal processing kernel, their consumed energy per sample is therefore the same, with the exception of the energy consumed for squaring the samples which is negligible compared to the CAC energy consumption. As a result, the total consumed energy of the *preprocessors* is proportional to  $(S_T N_T + S_f N_f)T_s$ . The choice of the design parameters  $(\Delta_{\alpha_T}, \Delta_{\alpha_f}, N_T, N_f)$  and their relationship to the required classification accuracy is explained in Section IV.

## C. Modulation-Type Classifier

After estimating the signal parameters, the proposed modulation-type classifier computes the CAC at cyclic frequencies within the union of possible cyclostationary features in Table II, resulting in a six-dimensional feature vector [23] given by

$$\mathbf{F} = \left[ |\tilde{R}_{x^*}^{1/T}(0)|, |\tilde{R}_x^{2f_c - 1/T}(0)|, |\tilde{R}_x^{2f_c - 1/2T}(0)|, \\ |\tilde{R}_x^{2f_c}(0)|, |\tilde{R}_x^{2f_c + 1/2T}(0)|, |\tilde{R}_x^{2f_c + 1/T}(0)| \right].$$
(7)

Because each element in the feature vector  $\mathbf{F}$  is proportional to the received signal power, we normalize the feature vector to unit power, and compare this normalized feature vector  $\mathbf{\bar{F}}$ to asymptotic normalized feature vectors  $\mathbf{\bar{V}}_i$ ,  $i \in [1, 2, 3]$ , for each of the classes considered. For instance, the normalized asymptotic feature vector for signals belonging to Class 1 is  $\mathbf{\bar{V}}_1 = [1, 0, 0, 0, 0, 0]$  as only one cyclic feature is present at the signal's symbol rate.

The resulting normalized feature vector is compared to each feature vector  $\bar{\mathbf{V}}_i$ , and the classifier picks the modulation class  $\hat{C}$  whose feature vector is closest to one of the received signal in the least square sense [23], namely

$$\hat{C} = \arg\min_{i \in [1,2,3]} ||\bar{\mathbf{F}} - \bar{\mathbf{V}}_i||^2.$$
(8)

In contrast to the *pre-processors*, the only degree of freedom in the design of the modulation type classifier is the number of samples  $N_c$  required to compute each of the six CACs that form the feature vector. Given SNR of the received signal and the estimation accuracies of the *pre-processors*,  $N_c$  is chosen accordingly to meet the desired classification probability. As a result of the six CACs required for classification, the processing time for modulation-type estimation is equal to  $6N_cT_s$ . The six CACs are computed sequentially to enable high degree of hardware reuse without violating the processing time budget and compromising the total energy consumption. While the cyclic features that the considered modulation types exhibit are known and can be used for parameter estimation, an energy efficient method to estimate the symbol rate and carrier frequency has not been proposed before. Further, the authors are not aware of any work that ties the required symbol rate and carrier frequency accuracies to meet the modulation classification probability. As will be shown in Section VI, the *pre-processors* consume most of the processor's energy, and therefore a careful selection of the step sizes for  $W_T$  and  $W_f$  is necessary to achieve an energy efficient solution.

#### D. Spread Spectrum Classification

Within the SC class, we distinguish between BPSK and direct sequence spread spectrum (DSSS) signals based on the variance  $\rho(\tau)$  of the signal's autocorrelation at a given lag  $\tau$  [24]. The received signal is divided into non-overlapping windows of  $N_d$  samples each. For each window, we compute the estimate of the autocorrelation for the possible expected lags. The fluctuations of the autocorrelation value for each  $\tau$  of interest is then measured over  $M_d$  windows. It was shown [24] that these fluctuations have peaks at a lag equal to the code length. The algorithm has further been optimized to only search for code lengths that are a power of two as these are most commonly used. With this approach, the presence of a DSSS signal as well as its code length can be determined in a single step.

Mathematically, the autocorrelation function is approximated using  $N_d$  samples over all lags of interest  $\tau \in 2^{[1,...,6]}$  for each frame  $m \in [1,...,M_d]$  of input samples  $x_m[\cdot]$ , resulting in

$$r_x(m,\tau) = \frac{1}{N_d} \sum_{n=1}^{N_d} x_m[n] x_m[n-\tau].$$
 (9)

The variance of the autocorrelation function is computed at every lag given  $M_d$  realizations of  $r_x(m, \tau)$ 

$$\rho(\tau) = \frac{1}{M_d} \sum_{m=1}^{M_d} r_x(m,\tau)^2 - \left(\frac{1}{M_d} \sum_{m=1}^{M_d} r_x(m,\tau)\right)^2.$$
 (10)

Finally, in order to detect if the received signal is a spread spectrum signal with code length  $\tau$ , the statistic  $\rho(\tau)$  is compared to threshold  $\gamma_d$ . Using  $\gamma_d = 4.25$  at SNR of 10 dB,  $N_d = 32$  samples per frame and  $M_d = 4$  averages are required for each lag  $\tau$  of interest to meet a correct classification probability of DSSS signals of 98% and a misclassification probability of 1%.

## E. Example of Classification Flow

We consider the classification of a DSSS signal with an underlying BPSK modulation scheme that is spread with a code of length 8. The DSSS signal has a symbol rate of 5 MHz, and is centered at 125 MHz at SNR of 10 dB. After detecting the presence of the signal in the band segmentation, the CIC filter downconverts the signal to a center frequency of 16 MHz and decimates it resulting in 4 samples per symbol. Fig. 3 shows the output of each of the algorithms discussed in



Fig. 3. Classification example of a 5 MHz DSSS signal with underlying BPSK modulation scheme.

this section. In the first block of the classification tree, the  $C_{42}$ cumulant is computed and compared against a threshold. We show that setting  $N_m = 90$  samples is sufficient to separate SC and MC classes with a probability of 95%. In this case, the DSSS signal being a SC signal will be classified as SC, and its transmit parameters will be computed next using (5) and (6). Using  $N_T = N_f = 400$  samples, the *pre-processors* estimate the symbol rate and carrier frequency of the DSSS signal. Using these estimates, the modulation type classifier computes the normalized feature vector  $\mathbf{F}$  which is compared to the theoretical normalized feature vector of BPSK signals plotted in solid lines in Step 4 of Fig. 3 for different realizations of the feature vector <sup>2</sup>. Finally, after being classified as a BPSK signal, the variance of the auto-correlation function  $\rho(\tau)$  given in (10) is computed using  $N_d = 32$  samples per frame and  $M_d = 4$  averages, and is compared against the threshold  $\gamma_d$ . Given the oversampling ratio of 4, the peak of  $\rho(\tau)$  will occur at lag  $\tau = 8 \times 4 = 32$ . Therefore, detecting the presence of this peak inherently asserts the presence of the DSSS signal and estimates the code length simultaneously. Note that the design variables in this example are selected so that the estimation accuracies of the pre-processors are on the order of 100 ppm. However, such small estimation accuracies might not be required to meet the desired classification accuracy. The aim of the next section is to analyze the maximum tolerable

estimation accuracies in order to minimize the total consumed energy and processing time.

## IV. ENERGY MINIMIZATION METHODOLOGY

In this section, we proceed with the optimization of the design parameters in order to minimize the total consumed energy while meeting the desired classification probability.

In order to minimize the consumed energy, we split the signal processing blocks into *dependent* blocks, whose design variables are a function of the output of previous signal processing stages, and *independent* blocks, whose design variables can be set independently of the output of other blocks. For instance, the design variables of both the multi-carrier and DSSS classifiers do not depend on the output of any other stage in the classification, and are therefore labeled as independent blocks. On the other hand, the modulation type classifier block relies on the outputs of the pre-processors, and the choice of the number of samples spent for modulation type classification  $N_c$  is tightly related to the estimation accuracies of the transmit parameters. These blocks are therefore labeled as *dependent*. It is clear that the *independent* blocks consume a fixed amount of energy regardless of the other blocks, and therefore are not jointly optimized with the rest of the blocks. On the other hand, a joint optimization of the total consumed energy of the dependent blocks is possible. A summary of the dependent and independent blocks and their respective design variables are depicted in Fig. 4.

<sup>&</sup>lt;sup>2</sup>We only require one realization of the feature vector to perform modulation-type classification, but the average detection performance is computed using multiple realizations of the feature vector



Fig. 4. Proposed processor showing dependent blocks in gray and their design variables to be optimized.

# A. Energy Optimization of Dependent Blocks

In order to optimize the energy consumption of the proposed pre-processor and classifier, we note that all three blocks make use of the CAC statistic in (3). Thus, minimizing the total number of samples spent for classification is equivalent to minimizing the total consumed energy. Note that minimizing the total number of samples is also equivalent to minimizing the processing time given by  $(6N_c + S_T N_T + S_f N_f)T_s$ , where  $S_T = \left[ \mathcal{W}_T / \Delta_{\alpha_T} \right]$  and  $S_f = \left[ \mathcal{W}_f / \Delta_{\alpha_f} \right]$ . The search windows  $W_T$  and  $W_f$  are obtained from the band segmentation and are SNR dependent, and are therefore not optimized. Similarly, the number of samples per CAC computation  $N_T$ and  $N_f$  are also SNR dependent since they are the minimum required number of samples to push the noise level below the feature to be detected. At SNR of 10 dB,  $N_T = N_f = 320$ samples are required to correctly estimate the symbol rate and carrier frequency. Therefore, the only variables to optimize over are  $N_c, S_T$ , and  $S_f$ , which in turn is equivalent to optimizing over  $N_c, \Delta_{\alpha_T}$ , and  $\Delta_{\alpha_f}$ .

The objective function that minimizes the total consumed energy can therefore be formulated as follows

$$\min_{N_c, S_T, S_f} \quad 6N_c + S_T N_T + S_f N_f$$
  
such that  $\mathcal{P}(\hat{C} = i \mid \Delta_{\alpha_f}, \Delta_{\alpha_T}, N_c, C = i) \ge 0.95$   
 $\forall i \in [1, 2, 3].$  (11)

It is important to note that the result of the optimization problem (11) is a function of the coarse estimate windows  $W_T$  and  $W_f$ . In fact, the wider the windows are, the larger the number of CAC computations  $S_T$  and  $S_f$  are required for a given step size  $\Delta_{\alpha_T}$  and  $\Delta_{\alpha_f}$ , respectively. Therefore, the optimum choice of the design variables is inherently tied to the coarse estimation accuracy from the band segmentation. Next, we study the tradeoffs between the symbol rate and carrier frequency estimation errors under a given probability of classification constraint. We show that there exists a region of *pre-processor* ( $\Delta_{\alpha_T}, \Delta_{\alpha_f}$ ) pairs that satisfy the classification probability requirement.

#### B. Tradeoffs Between Pre-Processor Accuracies

Given that signals belonging to Class 1 only exhibit a cyclostationary feature at their symbol rates, the requirement for the maximum tolerable  $\Delta_{\alpha_T}$  is determined by signals belonging to this class. The classification accuracy for Class



Fig. 5. Probability of correct classification of M-QAM signals as a function of number of samples for different cyclic frequency offsets at SNR of 10 dB.

1 signals is shown in Fig. 5 as a function of the number of samples used for classification under different  $\Delta_{\alpha\tau}$  values. It can be seen that the classification accuracy of QAM signals is below the desired probability of 0.95 under CFO  $\Delta_{\alpha\tau}$  greater than 1000 ppm at SNR of 10 dB even when the number of samples is increased. We refer to the SNR-dependent maximum tolerable cyclic frequency offset as  $\Delta_{\max \alpha_T}$ . At SNR of 10 dB,  $\Delta_{\max \alpha_T} = 1000$  ppm. Therefore, as long as the symbol rate estimator guarantees an accuracy less than 1000 ppm, signals belonging to Class 1 can meet the required classification accuracy. Further, since the cyclostationary feature at the symbol rate is the weakest among all cyclostationary features [25], it requires the most number of samples to be detected. Therefore, the number of samples spent during classification  $N_c$  is determined by signals of Class 1 for every  $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}.$ 

The accuracy of the carrier frequency estimation error  $\Delta_{\alpha_f}$  is determined by the modulations that exhibit a cyclostationary feature at the carrier frequency, namely signals belonging to Class 2 and 3. However, unlike the accuracy requirement for the symbol-rate estimate which is governed by signals belonging to Class 1,  $\Delta_{\alpha_f}$  has to be jointly determined for every  $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}$ . As a result, for every  $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}$  that guarantees proper classification of Class 1 signals, there exists a maximum estimation error  $\Delta_{\max \alpha_f}$  that can be

tolerated by Class 2 and 3 signals. Therefore, in order to understand the tradeoffs between the accuracies of both *preprocessors*, we obtain the feasible region in the  $(\Delta_{\alpha_T}, \Delta_{\alpha_f})$  coordinate system under which the classification accuracy for all classes is met.

For every  $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}$  and  $N_c$  that meet the classification accuracy of Class 1 signals, the maximum tolerable CFO  $\Delta_{\max \alpha_f}$  is the result of the following optimization:

$$(\Delta_{\max \alpha_f} \mid N_c, \Delta_{\alpha_T}) = \max \quad \Delta_{\alpha_f}$$
such that  $\mathcal{P}(\hat{C} = i \mid \Delta_{\alpha_f}, \Delta_{\alpha_T}, N_c, C = i) \ge 0.95,$ (12)

where C is the correct class to which the received signal belongs to, and  $i \in [2, 3]$ . Therefore, for every  $\Delta_{\alpha_T} \leq \Delta_{\max \alpha_T}$ , there exists a maximum  $\Delta_{\max \alpha_f}$  under which classification requirement of 95% is met.

This tradeoff among different set of triplets is illustrated in Fig. 6 for SNR of 10 dB. We note the tradeoff between accuracies of the two pre-processors, and their respective impact on  $N_c$ . It turns out that setting a stricter requirement on the symbol-rate estimator relaxes the required accuracy of the carrier frequency estimator. As expected, changing  $\Delta_{\alpha\tau}$  results in different number of samples required for classification as discussed earlier. It is important to note that the tradeoff saturates after a certain point. In fact, spending more energy in the symbol rate estimator to push  $\Delta_{\alpha_T}$  below 700 ppm does not result in a relaxation of the carrier frequency estimator requirement. As a result, the cyclostationary features at a function of the carrier frequency cannot be detected reliably with an offset larger than 5400 ppm at SNR of 10 dB. In addition, the maximum tolerable estimation accuracy for the carrier frequency  $\Delta_{\alpha_f}$  given the accuracy of the symbol rate estimation  $\Delta_{\alpha_T}$  is denoted in Fig. 6 by markers. From an energy point of view, for a given  $\Delta_{\alpha_T}$  and the corresponding  $N_c$  samples spent in the modulation classification, setting  $\Delta_{\alpha_f} = \Delta_{\max \alpha_f}$  minimizes the total consumed energy of the pre-processor. Therefore, although there exists an infinite number of  $(\Delta_{\alpha_T}, \Delta_{\alpha_f}, N_c)$  triplets that meet the required classification probability, the most energy-efficient triplets lie on the boundary of the feasible region shown in Fig. 6.

# V. DESIGN METHODOLOGY AND HARDWARE ARCHITECTURE

This section presents a reconfigurable architecture for the blind classification flow. Unlike traditional ASICs that only focus on a single algorithm, the proposed reconfigurable hardware is co-optimized in both algorithmic and architectural design spaces, making it able to perform a variety of classification tasks yet still achieve high energy efficiency.

#### A. Algorithm-Architecture Co-design

The algorithm-architecture co-design methodology is applied to realize the proposed reconfigurable classifier, as illustrated in Fig. 7. Table III depicts the list of algorithms used by the proposed processor which were chosen for their algorithmic robustness, good classification accuracy and architectural similarity to enable high degree of hardware reuse.



Fig. 6. Tradeoff between the symbol rate estimator and carrier frequency estimator accuracies at SNR of 10 dB in order to meet a classification probability of 0.95 for all classes with the corresponding number of samples.



Fig. 7. Algorithm-architecture co-design framework delivers optimized hardware as well as processing strategies.

These algorithms, although employed to perform distinct types of tasks, are algorithmically similar. All input samples undergo the complex multiplication-and-accumulation (MAC) followed by a magnitude computation. However, the post-processing on the computed magnitude is different among algorithms. For instance, the CAC for the *pre-processors* simply performs the *argmax* function which chooses the cyclic frequency which maximizes the objective function, while the CAC for modulation type classification needs Euclidean distance calculation and *argmin* to detect the signal class whose theoretical feature vector is closest to the computed feature vector.

The selection of algorithms directly affects the implementation strategy. From functionality points of view, the implementation can be partitioned into two parts. We call the first part as the degree-of-freedom (DOF) operation, meaning that this type of computation is required by all algorithms. The second piece is the non-degree-of-freedom (NDOF) operation, whose hardware cannot be efficiently shared by different algorithms. In this sense, the MAC and the magnitude computation are categorized as DOF, while the post-processing is viewed as NDOF. Another aspect of algorithm-architecture trade-off is described by the workload requirements. Considering the processing time *along* an algorithm in Table III, we can see that the MAC is active for >95% of the time, while the magnitude computation and the post-processing only work for a few clock



Fig. 8. Proposed reconfigurable classification processor.



Fig. 9. Multi-mode multiplication-and-accumulation (MM-MAC) unit. The DFF denotes the D-flip-flop, and the two highlighted multipliers represent the complex multipliers. Detailed architecture of the coefficient generator (Coef. Gen.) is shown in Fig. 10.

cycles, having very low utilization. On the other hand, if we focus on the workload requirements across the algorithms, we see that the parameter estimations and the modulationtype classification take up a majority of the time and energy (>99%). Since all these three algorithms are realized by similar versions of CAC functions, the architecture for DOF operations has to be optimized in favor of the CACs instead of other functions (e.g.  $C_{42}$ ) to have strong connection to the energy minimization methodology proposed in Sec. IV. Distinct hardware design constraints for each of these components are therefore derived. The MAC has to support high-throughput with minimized dynamic energy which can be accomplished by applying parallelism and aggressive voltage scaling at the circuit level. In addition, the magnitude computation and the post-processing need to have minimized leakage when staying idle due to their low utilization. Combined with the algorithmlevel energy minimization strategy in Section IV, the entire co-design framework is able to deliver high energy efficiency from both the algorithm and circuit perspectives.

#### B. Proposed Reconfigurable Classification Processor

The proposed reconfigurable classification engine, as shown in Fig. 8, consists of a multi-mode MAC (MM-MAC), a magnitude computation unit (MCU), a post-processing unit (PPU), a  $64 \times 16b$  register bank, a  $128 \times 26b$  instruction and signal database memory, and a system controller that decode and deploy the control signals. The sizes of the register bank and the memory are decided to properly fulfill the classification tasks. Unlike traditional processors that unify their datapaths, the proposed classifier is a hybrid-datapath system, doing complex-valued computation in MM-MAC and MCU, but real-valued processing in PPU. Each processing block is individually optimized with particular design constraints derived from its workload requirements. The architectures of complex multipliers in MM-MAC are carefully selected based on their propagation delay and area cost. The scalingtype coordinate digital computer (CORDIC) realizes the MCU with better numerical accuracy than direct squaring followed by square-root operations. Detailed implementation issues are presented as follows.

1) MM-MAC: Figure 9 shows the architecture of the multimode MAC unit, with its internal bit-width optimized by an in-house analysis tool [26]. Consisting of a coefficient generator, several multipliers/squarers and well-designed datapath, the MM-MAC unit is particularly dedicated to the critical operations of selected classification algorithms. It catches the complex-valued data (x[n]) directly from the chip interface and pass them through a series of multiplier and/or squarer to generate their second- or fourth-order products. The products are then optionally passed through another complex multiplier (in CAC mode) before reaching the final accumulation stage. The formula  $C_{42}$  for multi-carrier classification is decomposed into three parts  $(\frac{1}{N_m}\sum |x[n]|^4$ ,  $C_{20}$  and  $C_{21}$ ), separately computed by MM-MAC and stored in the register bank, and finally combined by the post-processing unit. The two-mode squarer is flexible to perform either the square or the absolutesquare of a complex number a+jb efficiently by the following reformulation:

$$(a+jb)^{2} = (a+b) (a-b) + j (2ab), |a+jb|^{2} = (a+b) (a+b) - (2ab).$$
 (13)

Compared to the direct-mapping approach that requires three 8b multipliers and two 12b adders, the proposed method only uses two 8b multipliers, two 8b and one 12b adders, saving 28% of area.

The CAC coefficient generator, as shown in Fig. 10, generates complex exponential terms for CAC functions. It starts with a free-running angle accumulator whose step size equals the product of cyclic frequency and sampling rate  $(\alpha_i T_s)$ . Note that the accumulator doesn't need to be reset before each CAC computation because any of its initial phase offset will be eventually eliminated through MCU. Following the accumulator is the angle synthesizer. It is realized in an areaefficient way by the piecewise-linear approximation method [27], plus a re-mapping circuit to generate sine/cosine values whose angles are out of the range between 0 and  $\pi/4$ . The area efficiency from the piecewise-linear approximator comes with the loss of accuracy. The synthesizer suffers a mean-squareerror (MSE) of -40 dB when it generates certain angles, meaning that it won't perform any better even in floatingpoint systems. However, such error is below the noise floor at 10 dB SNR and therefore produces negligible effects on the classification performance.

The two complex multipliers in MM-MAC are realized using the traditional four real multiplications and two additions  $(4\times, 2+)$  rather than the method suggested in [28] that uses  $(3\times, 5+)$  due to several reasons. Conventionally, trading one multiplier for three adders in the  $(3\times, 5+)$  approach is beneficial since the complexity of multipliers is usually much



Fig. 11. Programming of MM-MAC unit via simple request-acknowledge protocol. The program counter (PC) is halted during MAC operations, and resumes to access the next instruction address after receiving the acknowledgement signal.



Fig. 10. The coefficient generator in MM-MAC unit provides the complex exponential terms for CAC by using only simple adders and shifters. The numbers inside the squares denote the amount of right or left shifting with sign extension.

higher than that of adders for general-purpose processors. However, since the wordlength of complex multiplication in MM-MAC is small, the original form is simpler. To see the tradeoff between  $(4\times, 2+)$  and  $(3\times, 5+)$  regarding their area estimates, we use the array-multiplier approximation for firstorder comparison. Without loss of generality, the normalized size of an array multiplier can be estimated by the product of wordlengths of the multiplier and the multiplicand [29]. The area estimate of a  $(3\times, 5+)$  complex multiplier is thus generalized by the following equation

$$Area_{3\times 5+} = 3L^2 + 10L, \tag{14}$$

where L denotes the wordlength. On the other hand, the area of a  $(4\times, 2+)$  multiplier can be formulated as

$$Area_{4\times 2+} = 4L^2 + 4L.$$
(15)

Solving these two equations shows that  $(3\times, 5+)$  can only be noticeably better (by 20%, for example) when the wordlengths of its operands are greater than 20 bits. In our case, these two candidates for an 8b multiplier realization only differs by 5.5%. The other concern to the argument is about the propagation delay. It is obvious that the  $(3\times, 5+)$  approach is slower than  $(4\times, 2+)$  due to the delay from an additional adder stage. As a consequence, the  $(4\times, 2+)$  complex multiplier can use smaller logic gates to achieve the same delay as  $(3\times, 5+)$ , or it can exploit the advantageous timing slack to allow more voltage scaling, further minimizing its power consumption.

2) MCU: The scaling-type CORDIC is used to compute the magnitude of a complex number due to its robust numerical stability. The core building block of a scaling CORDIC consists of adders and shifters. The output precision depends on the number of CORDIC iteration stages  $N_i$ . There are three different types of architecture to implement CORDIC, i.e. fully pipelined, fully folded, and a hybrid between these two. Pipelined CORDIC achieves the highest throughput with high area and leakage penalty. The folded architecture takes  $N_i$  cycles to calculate the magnitude with around  $N_i$ -times lower area and leakage cost. Since the magnitude computation is highly underutilized and is only required at the end of each MAC operation, the fully folded CORDIC architecture is implemented.

3) PPU: The post-processing unit is a real-valued, onecycle-latency processor with specialized arithmetic logic units (ALUs). The ALU consists of a comparator (for threshold comparison, argmin and argmax functions), an 8-bit right/left-shifter (for power-of-two normalization), a 16-bit adder/subtractor and a 16-bit multiplier. For most of the time, PPU executes the normalization and/or the threshold comparison on the MCU outputs. The real-valued adder/subtractor and multiplier are occasionally used to compute the Euclidean distance required by the modulation-type classification. Instead of using a divider to normalize the computed CAC feature vector, the multiplier is employed to de-normalize the theoretical feature vector before substracting it by the computed one. The same multiplier is then reused to perform the squaring operation to complete the Euclidean distance calculation. The ALU operations are executed sequentially, one in each clock cycle, to realize the complex operations in an area-efficient way. Although the average operational latency from this approach is much longer than the one which does all operations in parallel, the cycle-time overhead is still negligible since the PPU is only active <1% of the total processing time. The slowest yet simplest PPU architecture minimizes the area and leakage.

4) Data Transfer and Control: The proposed processor employs a  $64 \times 16b$  one-write-two-read (1W2R) register bank for

inter-block data transfer, a 128×26b memory for instruction and signal database storage, and a central controller to decode and deploy the control signals. Note that the contents in the register bank can also be monitored from the chip outputs at any time for functional verification. The instruction-set architecture (ISA) supports regular register-type instructions for the PPU and the register bank, and special instructions to control the MM-MAC and the MCU. The ISA also implements loop and jump instructions to efficiently use the memory space. The program counter continues to accumulate when it executes the regular one-cycle-latency instructions, but is halted during the long-latency operations of MAC and MCU. All of the processing blocks use the simple request-acknowledge protocol to communicate with the central controller, telling the controller when to let the register bank access their outputs. For illustration, the programming of MM-MAC is depicted in Fig. 11. The controller first sends the request signal REQ and the initialization information INIT (i.e. the mode, frame length and cyclic frequency, based on the contents of the 26b instruction) to activate the MM-MAC. The MM-MAC then starts the computation and generates an acknowledgement signal ACK along with the outputs when it finishs. Upon catching the acknowledgement signal, the halted program counter resumes to access the new instruction, allowing the controller to store the MAC outputs into the register bank for later use. The MM-MAC finally returns to the idle state, preparing to accept another request whenever needed. Since most of the processing time is spent on the internal datapath of MM-MAC, the overhead from the data transfer is negligible, having limited effects on the total processing time and energy consumption. The instruction and memory overhead, however, cannot be ignored since they are active every cycle to control the processing blocks. The energy cost from the ISA and the processing blocks, as a result, are jointly considered for more accurate energy estimates, as will be seen in the next section.

Lastly, since the classification tasks are already determined at the co-design stage, there is no need to use generic software compilers (e.g. C++) to program the processor. An assembly code, instead, is manually written and converted to machine code by an in-house assembler for design verification. Implemented using Excel, the assembler provides a series of dropdown boxes that contain the user-friendly syntaxes (e.g. ADD, SUB, etc.) to compose the instructions. The tool then parses the syntaxes into binary machine codes using the VLOOKUP and the CONCATENATE functions embedded in Excel.

## VI. DESIGN VERIFICATION

This section discusses the Simulink-based functional verification of the proposed processor. We then use the power estimates from synthesized RTL code to compute the total consumed energy of the processor at a SNR of 10 dB.

# A. Simulink Design Environment

We have developed a Simulink-level experiment to run Monte-Carlo simulations in order to verify the functionality of the proposed processor, the hierarchy of which is similar to the one shown in Fig. 1. This processor is connected to the emulated RF front-end and band segmentation, which digitizes and senses the 500 MHz wide spectrum respectively. Each of the signal processing kernels has been implemented using Simulink embedded functions which effectively proves the low algorithmic complexity of our implementation. The Simulink design environment was used to test different signal types under different scenarios. Further, the Simulink environment is used to generate test vectors for testing the RTL code as well as demonstrate the functionality of the different algorithms that compose the processor.

#### B. Chip Implementation

An integrated chip design flow is adopted to incorporate algorithm, architecture, and circuit implementation in a highly automated environment. The graphical Simulink development environment offers high-level floating-point and fixed-point modeling for simulation. Specifically, the in-house analysis tool [26] is employed to minimize the internal data bitwidth subject to the classification accuracy of 95% at 10 dB SNR. The Simulink description can also be used to automatically generate the hardware description for gate-level synthesis. Cycle-accurate power estimate is done by static-timing analysis tool using true test vectors exported from Simulink model. A script-based backend place-and-route flow is also developed to shorten the design cycle and enhance the chip reliability. Implemented in a 40-nm CMOS technology, the classifier takes 0.06 mm<sup>2</sup> (85k equivalent 2-input NAND including the memory and the register bank) and consumes 10 mW at 500 MHz from a supply voltage of 0.9 V. Detailed energy breakdown shows the MM-MAC consumes 20 pJ/cvcle, and the MCU and PPU take 10 pJ/cycle. Each of these average energy costs include 1) the active energy from arithmetic logic, and 2) the energy of the control circuits that configure the block. In other words, it represents the energy consumed whenever a block is used. Extra clock cycles due to interblock data transfer and latencies are also considered but are found to be negligible because the processor spends most of the time within the processing blocks. One thing to be noted is that, since the MM-MAC is fully pipelined, it handles one sample per clock cycle from a throughput perspective. As a result, the average energy of 20 pJ/cycle from the MM-MAC is equivalent to 20 pJ/sample. The MCU and PPU, on the other hand, don't have this property as they are either fully folded or take multiple instruction cycles to process one sample.

## C. Total Processing Time and Consumed Energy

In order to solve the optimization (11) and obtain the consumed energy of the *pre-processor* and modulation-type classifier, we use values of  $W_T$  and  $W_f$  that correspond to the band segmentation processor [12]. We implemented the band segmentation in [12] and obtained coarse estimate windows for both the symbol rate and carrier frequency at SNR of 10 dB, given by  $W_T = 150$  KHz, and  $W_f = 260$  KHz respectively. To illustrate the benefits of energy minimization across the design space composed of the triplets  $(N_c, \Delta_{\alpha_T}, \Delta_{\alpha_f})$ , we compute the total energy for some of the triplets in the feasible region. Since most of the processing time is spent on the

 TABLE III

 Summary of Classification Blocks, Task Partitions, Processing Times and Consumed Energies.

Task	Task Partition of Processing Blocks		Total Num. of Cycles		Proc. Time	Energy
	MM-MAC & MCU	PPU	MM-MAC	MCU & PPU	@500 MHz	Consump
			(20pJ/Cycle)	(10pJ/Cycle)	(ms)	(µJ)
Mult. Car.	$C_{42} = \frac{1}{N_m} \sum_{n=1}^{N_m}  x[n] ^4 -  C_{20} ^2 - 2C_{21}^2$	$C_{42} \gtrless$ Threshold	300 (97.7  0.05)	7 (2.3  0.10)	0.001	0.006
Carrier Freq. Est.	$ R_{x^2}(\alpha_i)  = \left  1/N_f \sum_{n=0}^{N_f - 1} x[n]^4 e^{-j2\pi\alpha_i nT_s} \right $	$\max_{\alpha_i}  R_{x^2}(\alpha_i) $	401k (98.8  77.0)	5024 (1.2  75.6)	0.800	8.000
Symbol Rate Est.	$ R_{x^*}(\alpha_i)  = \frac{1}{N_T} \sum_{n=0}^{N_T - 1}  x[n] ^2 e^{-j2\pi\alpha_i nT_s}$	$\max_{\alpha_i}  R_{x^*}(\alpha_i) $	116k (98.8  22.3)	1456 (1.2  21.9)	0.230	2.300
Mod. Type	$ R_x(\alpha_i)  = \frac{1}{N_c} \sum_{n=0}^{N_c-1} x[n]^2 e^{-j2\pi\alpha_i nT_s}$	Euc. Dist. + argmin	3k (95.3  0.60)	148 (4.7  2.20)	0.006	0.060
DSSS	$r_x(m,\tau) = 1/N_d  \sum_{n=1}^{N_d} x_m[n] x_m[n-\tau] $	$\max_{\tau} \rho(\tau)$	230 (94.3  0.05)	14 (5.7  0.20)	0.001	0.005
Note: Numbers inside the parenthesis represent the workload (along $\ across$ ) the classification tasks (in %)			Sum	1.037	10.37	

Note: Numbers inside the parenthesis represent the workload (along || across) the classification tasks (in %)



Fig. 12. Combined energy consumed by the *pre-processor* and modulation type classifier at different  $(N_c, \Delta_{\alpha_T}, \Delta_{\alpha_f})$  in the feasible region with a consumed energy of 20 pJ per sample.

parameter estimation and the modulation-type classification that use the MM-MAC block, the average consumed energy of the entire system is dominated by these two tasks and can be approximated as 20 pJ/sample. Substituting this number into the proposed tradeoff analysis framework, the energy consumed by the *pre-processor* and the modulation-type classifier is computed as

$$E_T(N_c, \Delta_{\alpha_T}, \Delta_{\alpha_f}) = 20(6N_c + S_T N_T + S_f N_f) \quad \text{pJ.}$$

Based on the above equation, we compute the energy spent by the *pre-processor* and the modulation-type classifier for a range of  $(N_c, \Delta_{\alpha_T}, \Delta_{\alpha_f})$  triplet. We show in Fig. 12 the combined consumed energy of the *pre-processor* and modulation type classifier at SNR of 10 dB. It can be seen that the triplets that lie on the boundary of the feasible region consume the minimum amount of energy of 10.3  $\mu$ J, as compared to 24.9  $\mu$ J and 59.5  $\mu$ J for example if more processing time is spent on either of the two *pre-processors*. It is worth noting that the reason why there doesn't exist a unique optimum point for the optimization is because the  $6N_c$  term in (11) is negligible compared to the other two terms, and therefore, there exists a range of  $N_c$  values for which the objective function is minimized.

After minimizing the energy consumed by the *pre-processor* and the modulation-type classifier, we compute a breakdown

of the total energy and processing time, as shown in Table III. As can be seen, at a clock rate of 500 MHz, the proposed processor can meet the classification requirement of 95%while consuming a total of 10.37  $\mu$ J, and a processing time of around 1 ms, meeting both energy and processing time requirements given in Table I. Further, it is worth noting that the energy consumed by the pre-processor and modulation type classifier constitute 99.33% of the total consumed energy which validates our reasoning behind optimizing the *dependent* blocks. Note that the proposed classifier can also perform nonblind signal classification, if the transmit parameters are known and stored in the signal database. In such an application, the total consumed energy and processing times drop dramatically to around 71 nJ and 8  $\mu$ s, respectively. These numbers can be obtained by adding the processing times and consumed energy in Table III with the exception of both the carrier frequency and symbol rate estimation blocks.

#### VII. CONCLUSION

A low-complexity blind modulation classification processor that operates without the knowledge of any of the parameters of the signal being processed is presented. The processor is composed of low-complexity hierarchical signal processing kernels that can classify single-carrier and multi-carrier signals. With respect to single-carrier signals, increasing the processing time during the modulation type classification does not necessarily increase the classification probability under large parameter estimation errors. As a part of the design strategies, the tradeoffs between the pre-processor and modulation type classifier are analyzed, and an optimization framework is formulated to minimize the total consumed energy and processing time. The reconfigurable hardware consists of degree-of-freedom operations and dedicated operations. Degree-of-freedom operations are optimized for highthroughput and energy efficiency, while the area of the nondegree-of-freedom operations is optimized to minimize the leakage spent during idle cycles. The proposed classifier achieves power consumption of 10 mW at 500 MHz, 0.9 volt nominal supply voltage. Its total energy consumption is further optimized using algorithm-level tradeoff analysis, yielding a total of  $10.37\mu$ J at a SNR of 10 dB to meet the classification accuracy of 95%.

## ACKNOWLEDGMENT

This work was supported by the DARPA CLASIC program under grant A002069701.

#### REFERENCES

- J. Lee, D. Toumpakaris, and W. Yu, "Interference mitigation via joint detection," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 6, pp. 1172 –1184, Jun. 2011.
- [2] D. Bai, J. Lee, S. Kim, and I. Kang, "Near ML modulation classification," in *IEEE VTC Fall*, 2012.
- [3] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *Communications, IET*, vol. 1, no. 2, pp. 137 –156, april 2007.
- [4] S. Moshavi, "Multi-user detection for ds-cdma communications," *IEEE Commun. Mag.*, vol. 34, no. 10, pp. 124–136, oct 1996.
- [5] B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection," *IEEE Circuits Syst. Mag.*, vol. 9, pp. 27 -45, 2nd Quarter 2009.
- [6] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Blind modulation classification: a concept whose time has come," in Advances in Wired and Wireless Communication, 2005 IEEE Sarnoff Symposium on, Apr. 2005.
- [7] A. Swami and B. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, Mar. 2000.
- [8] W. Gardner, W. Brown, and C.-K. Chen, "Spectral correlation of modulated signals: Digital modulation," *IEEE Trans. Commun.*, vol. 35, no. 6, pp. 595–601, Jun. 1987.
- [9] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Cyclostationarity-based blind classification of analog and digital modulations," in *Proc. IEEE MILCOM*, 2006.
- [10] N. Kim, N. Kehtarnavaz, M. Yeary, and S. Thornton, "DSP-based hierarchical neural network modulation signal classification," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1065–1071, Sep. 2003.
- [11] J. Xu, W. Su, and M. Zhou, "Software-defined radio equipped with rapid modulation recognition," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1659–1667, May 2010.
- [12] T.-H. Yu, O. Sekkat, S. Rodriguez-Parera, D. Markovic, and D. Cabric, "A wideband spectrum-sensing processor with adaptive detection threshold and sensing time," *IEEE Trans. Circuits Syst. I*, vol. 58, no. 11, pp. 2765–2775, Nov. 2011.
- [13] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 2, pp. 155–162, Apr. 1981.
- [14] P. Urriza, E. Rebeiz, P. Pawelczak, and D. Cabric, "Computationally efficient modulation level classification based on probability distribution distance functions," *IEEE Commun. Lett.*, vol. 15, no. 5, pp. 476–478, may 2011.
- [15] P. Urriza, E. Rebeiz, and D. Cabric, "Hardware implementation of kuiper-based modulation level classification," in *Proc. ACSSC*, Pacific Grove, CA, USA, Nov. 6–9, 2011.
- [16] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 189–193, Feb. 2000.
- [17] Q. Shi and Y. Karasawa, "Noncoherent maximum likelihood classification of quadrature amplitude modulation constellations: Simplification, analysis, and extension," *IEEE Trans. Wireless Commun.*, vol. 10, no. 4, pp. 1312–1322, Apr. 2011.
- [18] E. Rebeiz, P. Urriza, and D. Cabric, "Experimental analysis of cyclostationary detectors under cyclic frequency offsets," in *Proc. IEEE Asilomar Conf. on Signals and Systems*, 2012.
- [19] Y. Zeng and Y. C. Liang, "Robustness of the cyclostationary detection to cyclic frequency mismatch," in *Proc. IEEE PIMRC*, Istanbul, Turkey, Sep. 26–29, 2010.
- [20] M. Shi, A. Laufer, Y. Bar-Ness, and W. Su, "Fourth order cumulants in distinguishing single carrier from OFDM signals," in *Proc. IEEE MILCOM*, San Diego, CA, USA, Nov. 17–19, 2008.
- [21] M. Morelli and U. Mengali, "Feedforward frequency estimation for psk: A tutorial review," *European Transactions on Telecommunications*, vol. 9, no. 2, pp. 103–116, Mar. 1998.
- [22] P. Ciblat and E. Serpedin, "A fine blind frequency offset estimator for ofdm/oqam systems," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 291–296, Jan. 2004.

- [23] E. Rebeiz and D. Cabric, "Low complexity feature-based modulation classifier and its non-asymptotic analysis," in *Proc. IEEE GLOBECOM*, Houston, TX, USA, Dec. 5–9, 2011.
- [24] G. Burel, C. Bouder, and O. Berder, "Detection of direct sequence spread spectrum transmissions without prior knowledge," in *Proc. IEEE GLOBECOM*, San Antonio, TX, USA, Nov. 25–29 2001.
- [25] W. A. Gardner, *Statistical spectral analysis: a non probabilistic theory*. Prentice Hall, 1988.
- [26] C. C. Wang, C. Shi, R. W. Brodersen, and D. Markovic, "An automated fixed-point optimization tool in matlab xsg/syndsp environment," in *ISRN Signal Processing, Article ID 414293*, 2011.
- [27] K.-H. Lin, "Design of a baseband receiver for DVB-T standard," Master's thesis, National Taiwan University, 2004.
- [28] A. Wenzler and E. Luder, "New structures for complex multipliers and their noise analysis," in *IEEE International Symposium on Circuits and Systems*, 1995.
- [29] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits:* A Design Perspective. Prentice Hall, 2003.



Eric Rebeiz (S'06) received his B.S. degree (Summa Cum Laude) from the University of Massachusetts Amherst in 2008 and his M.S. degree from the University of Southern California in 2009, both in Electrical Engineering. He is currently a Ph.D. Candidate at the University of California Los Angeles advised by Prof. Danijela Cabric. His current research interests pertain to the field of cognitive radios and include low power cyclostationary-based spectrum sensing, compressive sensing, and modulation classification in wideband channels.



Fang-Li Yuan (S'10) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, in 2006 and 2008, respectively. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, University of California, Los Angeles. His research interests include the flexible DSP architectures and the VLSI circuit designs for communication signal processing, with particular focus on the blind signal classification and the multi-antenna digital baseband. Mr. Yuan was awarded the Broadcom Fellowship in 2012

for his research on the multi-core processors for software-defined radios.



**Paulo Urriza** (S'07) received the B.S. degree in Computer Engineering and the M.S. degree in Electrical Engineering from the University of the Philippines Diliman, Quezon City, Philippines in 2007 and 2009 respectively. In 2009, he joined the Electrical Engineering Department of the University of California Los Angeles with his research advisor Prof. Danijela Cabric. His research interests include modulation classification, traffic prediction, localization and other Primary User parameter estimation techniques for advanced Cognitive Radio systems.



**Dejan Markovic** (S'96–M'06) received the Dipl.Ing. degree from the University of Belgrade, Serbia, in 1998 and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 2000 and 2006, respectively, all in electrical engineering. In 2006, he joined the faculty of the Electrical Engineering Department at the University of California, Los Angeles where he is now an Associate Professor. Since 2009, he has been affiliated with the Biomedical Engineering Interdepartmental Program at UCLA as a co-chair

of the Neuroengineering field. He is also a director of the Integrated Circuits track within the UCLA Master of Science in Engineering Online Program. His current research is focused on integrated circuits for emerging radio and healthcare systems, programmable ICs, design with post-CMOS devices, optimization methods and CAD flows.

Dr. Markovic was awarded the CalVIEW Fellow Award in 2001 and 2002 for excellence in teaching and mentoring of industry engineers through the UC Berkeley distance learning program. In 2004, he was a co-recipient of the Best Paper Award at the IEEE International Symposium on Quality Electronic Design. In recognition of the impact of his Ph.D. work, he received 2007 David J. Sakrison Memorial Prize at UC Berkeley. He received an NSF CAREER Award in 2009. In 2010, he was a co-recipient of ISSCC Jack Raper Award for Outstanding Technology Directions and a winner of the DAC/ISSCC Student Design Contest.



**Danijela Cabric** (S'96–M07) received the Dipl. Ing. degree from the University of Belgrade, Serbia, in 1998, and the M.Sc. degree in electrical engineering from the University of California, Los Angeles, in 2001. She received her Ph.D. degree in electrical engineering from the University of California, Berkeley, in 2007, where she was a member of the Berkeley Wireless Research Center. In 2008, she joined the faculty of the Electrical Engineering Department at the University of California, Los Angeles as an Assistant Professor. Dr. Cabric received the Samueli

Fellowship in 2008, the Okawa Foundation Research Grant in 2009, Hellman Fellowship in 2012 and the National Science Foundation Faculty Early Career Development (CAREER) Award in 2012. She serves as an Associate Editor in IEEE Journal on Selected Areas in Communications (Cognitive Radio series) and IEEE Communications Letters, and TPC Co-Chair of 8th International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM) 2013. Her research interests include cognitive radio systems and spectrum sensing, VLSI architectures of signal processing and digital communication algorithms, and their performance analysis and experiments on embedded system platforms.