COMPRESSIVE SENSING SPECTRUM RECOVERY FROM QUANTIZED MEASUREMENTS IN 28 NM SOI CMOS

David Bellasi[1], *Luca Bettini*[1], *Thomas Burger*[1], *Christian Benkeser*[2], *Qiuting Huang*[1], *Christoph Studer*[3]

[1]ETH Zürich, Zürich, Switzerland; {bellasid, bettini, burger, huang}@iis.ee.ethz.ch
[2]RUAG Space, Switzerland; christian.benkeser@ruag.com
[3]Cornell University, NY, USA; studer@cornell.edu

ABSTRACT

Spectral activity detection of wideband radio-frequency (RF) signals for cognitive radios typically requires expensive and energy-inefficient analog-to-digital converters (ADCs). Fortunately, the RF spectrum is—in many practical situations—sparsely populated, which enables the design of so called *analog-to-information* (A2I) converters. A2I converters are capable of acquiring and extracting the spectral activity information at low cost and low power by means of compressive sensing (CS). In this paper, we present a high-throughput spectrum recovery stage for CS-based wideband A2I converters. The recovery stage is designed for a CS-based signal acquisition front-end that performs pseudo-random subsampling in combination with coarse quantization. High-throughput spectrum activity detection from such coarsely quantized and compressive measurements is achieved by means of a massively-parallel VLSI design of a novel accelerated sparse signal dequantization (ASSD) algorithm. The resulting design is implemented in 28 nm SOI CMOS and able to reconstruct 2^{15}-point frequency-sparse RF spectra at a rate of more than 7.6 k reconstructions/second.

1. INTRODUCTION

1.1. Wideband Spectrum Sensing

Spectrum sensing aims at identifying unused frequency bands with the goal of reusing them to improve the spectral utilization [2]. Since bandwidth is a scarce and, hence, expensive resource, spectrum sensing is believed to play a major role in meeting the ever-growing demand for higher data rates in next-generation wireless systems. Conventional high-precision, high-rate analog-to-digital converters (ADCs) offer a straightforward solution for acquiring wideband signals in the GS/s regime, but they are typically energy-inefficient and expensive [3], and can result in excessive data rates (on the order of tens of Gb/s). These drawbacks prohibit their deployment in low-cost, battery-powered devices. Hence, to enable spectrum sensing at low power and low cost, novel wideband sensing techniques and corresponding VLSI circuits that are able to efficiently extract information about the spectral occupancy are necessary.

1.2. Analog-to-Information Conversion

In recent years, a number of spectrum occupancy surveys observed that the radio-frequency (RF) spectrum is sparsely populated in many practical situations [4]. Compressive sensing (CS) is a popular sampling paradigm that enables one to acquire such frequency-sparse signals at sub-Nyquist rates, while enabling their reconstruction using sophisticated sparse signal recovery algorithms [5]. Hence, CS allows the design of so-called *analog-to-information (A2I) converters*, which compressively sample sparse signals using inexpensive, energy-efficient analog circuits, while sophisticated sparse signal recovery algorithms extract the information contained in the acquired signals, such as the spectral occupancy [1, 6, 7].

Due to the high computational complexity associated with sparse signal recovery, virtually all existing CS-based A2I designs perform signal recovery off-line [6, 7]. Off-line processing, however, results in excessive I/O data-rates and prohibits the use of adaptive sensing strategies. In contrast, on-chip sparse signal recovery has the potential to avoid these drawbacks at the cost of requiring complex VLSI circuits [8].

1.3. Contributions

This paper describes a high-throughput, sparse signal recovery stage for wideband spectrum sensing in 28 nm SOI CMOS. The proposed recovery stage is part of the CS-based wideband A2I converter reported in [1], which leverages CS via randomized sub-Nyquist sampling and coarsely quantized measurements, inspired by recent results in 1-bit CS [9, 10]. For this A2I converter, we develop an efficient algorithm that is able to recover the sparse spectral information from coarsely quantized and compressive measurements. We then propose approximations on the algorithm level to enable its efficient implementation in VLSI. To achieve high recovery

Parts of this paper have been published in IEEE JETCAS [1]. The present paper describes a substantially improved version of the recovery stage and a corresponding reference VLSI design in 28 nm SOI CMOS.

throughput, we deploy a massively-parallel 2^{15}-point radix-32 fast Fourier transform (FFT) unit. We finally provide post-synthesis results in 28 nm SOI CMOS that demonstrate the efficacy of the proposed spectrum recovery unit.

2. QUANTIZED COMPRESSIVE SENSING

2.1. Compressive Sensing in a Nutshell

CS enables sub-Nyquist sampling and reconstruction of signal vectors $\mathbf{y} \in \mathbb{R}^N$ having a sparse representation $\mathbf{x}$ with only $K \ll N$ non-zero entries in an orthonormal basis $\mathbf{\Psi}$, i.e., $\mathbf{y} = \mathbf{\Psi}\mathbf{x}$. In particular, CS acquires M non-adaptive, linear measurements of the signal vector $\mathbf{y}$ as follows [5]:

$$\mathbf{z} = \mathbf{\Phi}\mathbf{y} + \mathbf{n}, \tag{1}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is a sensing matrix with fewer rows than columns ($M < N$) and $\mathbf{n} \in \mathbb{R}^M$ models noise. Given that the effective matrix $\mathbf{D} = \mathbf{\Phi\Psi}$ satisfies certain conditions [5], CS enables one to accurately recover $\mathbf{y}$ from the compressive measurements in $\mathbf{z}$. For spectrum sensing, the sensing matrix $\mathbf{\Phi}$ and the sparsifying basis $\mathbf{\Psi}$ correspond to a pseudo-random subsampling operator and to the discrete Fourier transform (DFT) matrix, respectively [1]. This combination enables the acquisition of sparse RF signals at rates well-below Nyquist.

2.2. Quantized Compressive Sensing

In practical systems, the compressive measurements are acquired by ADCs and, hence, instead of real-valued measurements as in (1), quantized measurements are acquired [9–11]

$$\mathbf{q} = \mathcal{Q}(\mathbf{z}) = \mathcal{Q}(\mathbf{D}\mathbf{x} + \mathbf{n}). \tag{2}$$

Here, $\mathcal{Q}(\cdot)\colon \mathbb{R} \to \mathcal{O}$ is a scalar quantizer (applied element-wise), which maps a real number x into $Q = |\mathcal{O}|$ ordered labels according to $\mathcal{Q}(x) = q$ if $b_{q-1} < x \leq b_q$, $q \in \mathcal{O}$, with the bin boundaries $-\infty = b_0 < \cdots < b_Q = +\infty$. In short, quantized CS recovers the sparse vector $\mathbf{x}$ from the quantized measurements in $\mathbf{q}$. The advantage of quantized CS is that it allows a further reduction of the measurement dimensionality, which enables the use of low-precision ADCs with low area and power requirements. As an example, the A2I converter in [1] takes particular advantage of quantized CS and deploys a low-cost and low-complexity, wideband 4-bit flash ADC.

2.3. Basis Pursuit De-Quantization

To recover the sparse vector $\mathbf{x}$ from the measurements $\mathbf{q}$, the method in [11] assumes that the noise vector $\mathbf{n}$ in (2) is i.i.d. zero-mean Gaussian with variance σ^2, which enables one to compute the likelihood of each measurement q_i as

$$p(q_i \mid \mathbf{d}_i^H\mathbf{x}) = \int_{\ell_i}^{u_i} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\!\left(-\frac{|\nu - \mathbf{d}_i^H\mathbf{x}|^2}{2\sigma^2}\right) d\nu, \tag{3}$$

where $u_i = b_{q_i}$ and $\ell_i = b_{q_i - 1}$ are, respectively, the upper and lower bin boundary positions associated with q_i, and $\mathbf{d}_i^H$ corresponds to the i^{th} row of $\mathbf{D} = \mathbf{\Phi\Psi}$. The idea behind the method in [11] is to find the most likely sparse vector $\mathbf{x}$ that is consistent with the quantized measurements $\mathbf{q}$, considering the system model (2). Thus, instead of using the standard, least-squares objective function that would result from the unquantized system (1), we minimize the negative log-likelihood of (3) together with an ℓ_1-norm penalty to promote sparse solutions. The resulting convex optimization problem, referred to as basis pursuit de-quantization, corresponds to

$$\text{(BPDQ)} \qquad \underset{\tilde{\mathbf{x}}}{\text{minimize}}\ \lambda\|\tilde{\mathbf{x}}\|_1 - \textstyle\sum_{i=1}^{M} \log p(q_i \mid \mathbf{d}_i^H\tilde{\mathbf{x}}),$$

where the parameter $\lambda > 0$ trades sparsity of the solution $\hat{\mathbf{x}}$ for consistency to the quantized measurements in $\mathbf{q}$.

```
$\mathbf{x}_1 = \mathbf{y}_0 = \mathbf{0}_{N\times 1}$ and $t_1 = 1$
while $k = 1, \ldots, K_{\text{max}}$ do
    $\mathbf{y}_k \leftarrow \text{shrink}\big(\mathbf{x}_k + \frac{1}{L}\mathbf{D}^H\nabla f(\mathbf{D}\mathbf{x}_k)\big)$
    $t_{k+1} \leftarrow \frac{1}{2}\big(1 + \sqrt{1 + 4t_k^2}\big)$
    $\mathbf{x}_{k+1} \leftarrow \mathbf{y}_k + \big(\frac{t_k - 1}{t_{k+1}}\big)(\mathbf{y}_k - \mathbf{y}_{k-1})$
end while
```

Algorithm 1. Accelerated sparse signal dequantization.

2.4. Accelerated Sparse Signal Dequantization

To arrive at a recovery method that enables an efficient integration in VLSI, we propose an alternative to the method in [11], referred to as *accelerated sparse signal dequantization* (ASSD). The ASSD algorithm (summarized in Alg. 1) builds on FISTA [12] and performs the following three steps until a maximum number of iterations K_{max} has been reached.

1) The gradient step enforces consistency to the quantized measurements $\mathbf{q}$. We set $w_i = \mathbf{d}_i^H\mathbf{x}$ and rewrite (3) as

$$p(q_i \mid w_i) = \Phi\big(\sigma^{-1}(u_i - w_i)\big) - \Phi\big(\sigma^{-1}(\ell_i - w_i)\big)$$

with $\Phi(a) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{a} \exp\big(-\frac{1}{2}\nu^2\big)\,d\nu$. With the definition $f(\mathbf{w}) = -\sum_{i=1}^{M} \log p(q_i \mid w_i)$, the i^{th} entry of the gradient $\nabla f(\mathbf{w})$ is given by [11]

$$[\nabla f(\mathbf{w})]_i = \frac{\exp\Big(-\frac{|u_i - w_i|^2}{2\sigma^2}\Big) - \exp\Big(-\frac{|\ell_i - w_i|^2}{2\sigma^2}\Big)}{\sqrt{2\pi\sigma^2}\,\Big(\Phi\big(\frac{u_i - w_i}{\sigma}\big) - \Phi\big(\frac{\ell_i - w_i}{\sigma}\big)\Big)}. \tag{4}$$

To ensure convergence, we use a constant step size determined by the Lipschitz constant $L = \lambda_{\text{max}}^2(\mathbf{D})/\sigma^2$, where $\lambda_{\text{max}}(\mathbf{D})$ is the largest singular value of $\mathbf{D}$. For spectrum recovery, $\mathbf{D}$ is a randomly-subsampled DFT matrix. Hence, we have $L = 1/\sigma^2$, which can be precomputed and stored in a configuration register.

2) The shrinkage step takes into account the ℓ_1-norm in (BPDQ) and enforces sparsity on the vector $\mathbf{x}$ performing element-wise *complex-valued* shrinkage as follows [12]:

$$\text{shrink}(x) = \begin{cases} \frac{x}{|x|} \max\{|x| - \lambda/L, 0\} & \text{if } x \neq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

3) The prediction step computes a new estimate of the sparse vector $\mathbf{x}_{k+1}$. The update on lines 4 and 5 of Alg. 1 yields accelerated convergence rates [12], which is key for achieving low computational complexity. To avoid costly square root and division operations, we precompute $\tau_k = (t_k - 1)/t_{k+1}$ and store them in a 128-entry look-up table (LUT).

2.5. Algorithm Approximations

To arrive at a high-throughput ASSD design, we deploy the following algorithm-level approximations.

1) The gradient step (4) requires transcendental functions. To avoid this, we use the following approximation:

$$[\nabla f(\mathbf{w})]_i \approx \begin{cases} \sigma^{-2}(u_i - w_i) & w_i > u_i \\ 0 & \ell_i \leq w_i \leq u_i \\ \sigma^{-2}(\ell_i - w_i) & w_i < \ell_i. \end{cases} \quad (6)$$

We note that the accuracy of this approximation depends on σ and improves for decreasing values of σ.

2) Complex-valued shrinkage (5) requires a division operation, which may cause issues with finite-precision (e.g., fixed-point) arithmetics. We therefore perform approximate shrinkage of $x \in \mathbb{C}$ using $\text{shrink}(x) \approx \eta(\Re\{x\}) + i\,\eta(\Im\{x\})$, where $\eta(v) = \text{sign}(v)\max\{|v| - \lambda/L, 0\}$. We note that the accuracy of this approximation depends on the ratio between $|x|$ and the threshold λ/L.

3. HIGH-THROUGHPUT ASSD ARCHITECTURE

3.1. High-Level VLSI Architecture

The proposed VLSI architecture is shown in Fig. 1(a) and comprises three main units: An *approximate gradient* unit, an *approximate shrinkage* unit, and a 2^{15}*-point radix-*32 *I/FFT* unit. The time-domain samples and the information about the (non-uniform) sample instants are stored in on-chip SRAMs ω and s_q, respectively. The q LUT holds digital representations of the upper and lower quantization bin boundaries u_i and ℓ_i, respectively. The remaining memories store intermediate results of the ASSD algorithm in Alg. 1.

The proposed architecture alternately performs forward and inverse FFTs. To ensure a low latency, the output of the inverse FFT is directly fed through the approximate gradient calculation and straight back to the FFT memory. Shrinkage and prediction process the data from the forward FFT in a similar way. In the final iteration of the ASSD algorithm, the result of the shrinkage unit corresponds to the sparse RF spectrum estimate.

3.2. High-Throughput Parallel Radix-32 I/FFT Unit

Since the number of clock cycles for one ASSD iteration is determined by the number of clock cycles for one forward and one inverse FFT, a high-throughput FFT unit is required. While aiming at a recovery bandwidth of over 3 GHz, sensing the spectral activity within the narrow bands of today's communication standards needs a resolution of at least 2^{15} points. To maximize the throughput of the ASSD unit at low silicon area, while providing a sufficient spectral resolution, we decided to implement a 2^{15}-point inverse/forward FFT unit (shown in Fig. 1(b)) containing a single radix-32 processing element (PE).

In each clock cycle, the radix-32 PE reads and writes 32 data items from the FFT's main data memory. In order to achieve such a massive parallelism without causing memory access contentions, we partition this memory into 64 independent banks (see Fig. 1(b)). To minimize silicon area, we use single-port memories in combination with a sophisticated contention-free read and write access scheme. In each clock cycle, the input data for the radix-32 PE is read from a specific set of 32 memories, whereas its output is stored in the remaining set of 32 memories.

The radix-32 PE (shown in Fig. 1(c)) is built from 31 complex-valued multipliers and a combination of a radix-16 stage and a radix-2 stage in a split-radix fashion. The radix-16 stage consists of two identical radix-16 PEs, each built from 8 multiplier-less radix-4 PEs. The inverse FFT is calculated by reversing the data path of the forward FFT with the aid of multiplexers. To maximize the throughput, the radix-32 PE features 11 pipelining stages. Post-synthesis results in 28 nm SOI CMOS show that the I/FFT unit can achieve a maximum clock frequency of more than 1 GHz, which leads to a throughput of over 309 k 2^{15}-point FFTs per second. As a comparison, the 2^{15}-point FFT in [13] uses 4 parallel radix-2 PEs and computes only 274 FFTs/second in 90 nm CMOS.

4. IMPLEMENTATION RESULTS

4.1. Sparse Spectrum Recovery Performance

To evaluate the spectral activity detection performance, we use the following metrics: (i) *True positive detection rate:* The number of correctly detected active frequency bins divided by the total number of effectively active bins. (ii) *False positive detection rate:* The number of frequency bins falsely detected divided by the total number of inactive bins. We set the spectral activity threshold to 20 dB above the quantization noise floor. The regularization parameter λ is chosen such that an optimal detection rate is obtained. Assuming a quantization-noise-limited receiver (such as the one in [1]) the noise variance σ^2 can be set to $V_{\text{LSB}}^2/12$, where V_{LSB} is the voltage difference of the least-significant-bit of the ADC.

To characterize the impact of the signal sparsity on the detection rate and the algorithm's noise sensitivity, simulations with synthetic test data were performed. The percentage of active frequency bins was set according to the desired sparsity level, while the location and spectral magnitude were

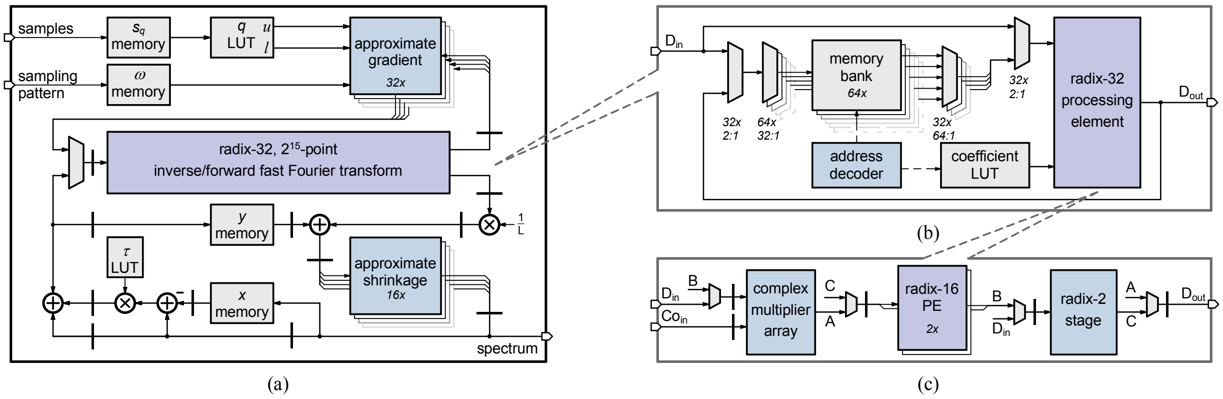

Fig. 1. Architecture of the ASSD recovery unit: (a) overview; (b) radix-32 I/FFT unit; (c) radix-32 processing element (PE).

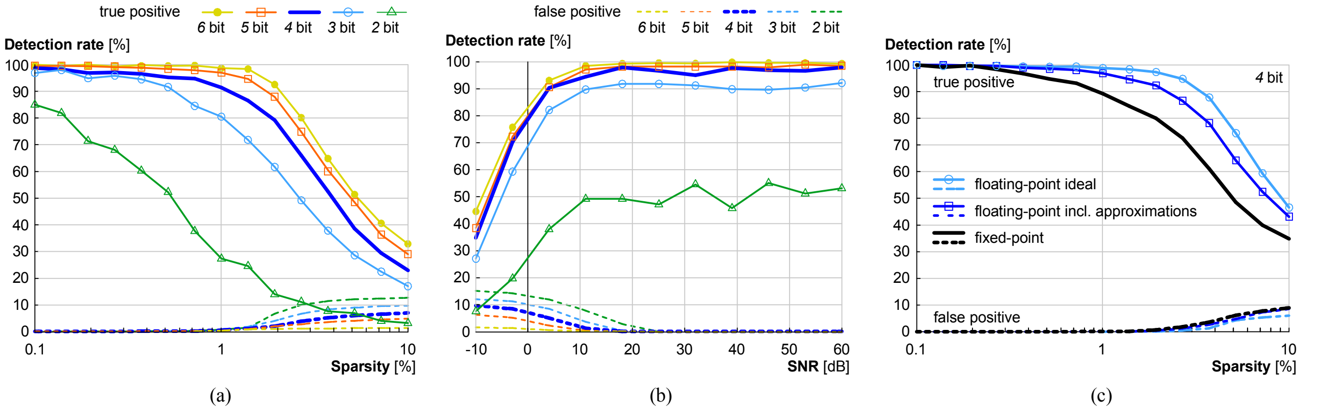

Fig. 2. Detection performance for synthetic test data: (a) varying signal sparsity levels; (b) varying input SNR; (c) comparison of an ideal (floating-point) model, a model including the approximations detailed in Sec. 2.5, and the fixed-point golden model.

both chosen at random.[1] To obtain the desired input SNR, i.i.d. zero-mean Gaussian noise with appropriate variance was added. All results were averaged over 10 Monte–Carlo trials, each running for $K_{\max} = 100$ ASSD iterations.

Fig. 2(a) characterizes the impact of the signal sparsity level on the detection rate for a different number of quantization bits B. The SNR of the input signal was set to exceed the corresponding signal-to-quantization-noise ratio by 3 dB. Reducing the signal sparsity from 10 % to 0.1 % improves the detection performance. The performance drop near 1 % active bins is due to the choice of the undersampling factor and the regularization parameter λ. Both parameters can be set at run-time to adapt the ASSD algorithm to the current sparsity level of the input signal. For the presented results, the undersampling factor was set to 11.5, while the appropriate λ values were found to be 2.3 for 2 bit, 3.7 for 3 bit, 8.8 for 4 bit, 55 for 5 bit, and 165 for 6 bit resolution.

Figure 2(b) shows the effect of the noise level on the detection rate for an input SNR from -10 dB to 60 dB; the signal sparsity level was set to 0.5 % for all trials. The true positive detection rate quickly drops for input SNRs below 0 dB, whereas larger SNRs result in good detection performance. In summary, for sufficiently high input SNRs, the ASSD algorithm achieves true and false positive detection rates close to 100 % and 0 %, respectively.

Maximum clock frequency	[MHz]	952
Maximum throughput[a]	[MS/s]	252
Memory power consumption[b]	[W]	1.53
Logic power consumption[b]	[W]	1.56

[a] At $K_{\max} = 20$ ASSD algorithm iterations.

[b] Estimated power consumption at 952 MHz, $V_{dd} = 0.92$ V, and 300 K.

Table 1. Post-synthesis results in 28 nm SOI CMOS

4.2. Fixed-Point Parameters

To minimize area and power consumption, and to maximize throughput, the proposed VLSI design uses fixed-point arithmetic. In the final design, the acquired time-domain signal is quantized to 4 bit. The real and imaginary part of the data in the radix-32 PE are represented with 24 bit each, which

[1] The locations and non-zero entries were generated using an i.i.d. uniform and i.i.d. zero-mean Gaussian distribution with unit variance, respectively.

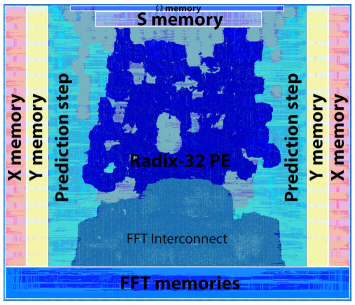

Fig. 3. ASSD chip layout.

	mm^2	%	MGE^c		mm^2	%	kBit
Gradient unit (32x)	0.05	2	0.10	S memory	0.11	5	131
Shrinkage unit (32x)	0.02	1	0.04	Ω memory	0.05	2	33
I/FFT unit	0.60	29	1.23	Y memory	0.22	11	786
– Radix-32 PE	0.38		0.78	X memory	0.22	11	786
Prediction step unit	0.34	17	0.70	FFT memories	0.45	22	1574
Total logic cells	1.01	49	2.07	Total memories	1.05	51	3310

^{c}1 GE equals 0.4896 µm^2 in the used 28 nm SOI CMOS technology.

Table 2. Area Breakdown of the ASSD Unit

determines the word-width of the memories, as well as the precision of the gradient and thresholding units. Thus, the time- and frequency-domain signals are represented with 24 bit and 48 bit, respectively. The FFT twiddle-factors use 18 bit, while the τ LUT has 8 bit entries. Figure 2(c) compares the detection rates using an ideal (i.e., floating-point) model, a model including the approximations, and the fixed-point golden model. The SNR is 3 dB below the quantization noise level, $\lambda = 2.0$, and the undersampling factor is 7.5. As shown in Fig. 2(c), the algorithm-level approximations, as well as the use of fixed-point arithmetics only entails a small loss in detection performance.

4.3. Implementation Results

The post-synthesis results for the proposed ASSD unit in 28 nm SOI CMOS are summarized in Tbl. 1. Our design achieves a maximum frequency of 952 MHz. The area breakdown in Tbl. 2 and the silicon chip layout in Fig. 3 show that the FFT unit consumes almost 30% of the total area, of which 2/3 are occupied by the radix-32 PE and 1/3 by the network connecting the PE and the FFT memories. The SRAM consumes half of the chip area, 43% of which is used for the FFT. The achievable spectral bandwidth is only limited by the employed signal acquisition front-end. Our spectrum recovery unit achieves over 7.6 k spectrum reconstructions per second, which is—to the best of our knowledge—the highest recovery throughput of a VLSI design for CS-based spectrum sensing reported in the open literature.

REFERENCES

[1] D. Bellasi, L. Bettini, C. Benkeser, T. Burger, Q. Huang, and C. Studer, "VLSI design of a monolithic compressive-sensing wideband analog-to-information converter," *IEEE JETCAS*, vol. 3, no. 4, pp. 552–565, Dec. 2013.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE JSAC*, vol. 23, no. 2, pp. 201–220, Feb. 2004.

[3] E. Janssen, K. Doris, A. Zanikopoulos, A. Murroni, G. van der Weide, Y. Lin, L. Alvado, F. Darthenay, and Y. Fregeais, "An 11b 3.6GS/s Time-Interleaved SAR ADC in 65nm CMOS," in *IEEE ISSCC*, Feb. 2013, pp. 464–465.

[4] K. Patil, K. Skouby, A. Chandra, and R. Prasad, "Spectrum occupancy statistics in the context of cognitive radio," in *IEEE PIMRC*, Oct. 2011, pp. 1–5.

[5] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE SPM*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[6] M. Wakin, S. Becker, E. Nakamura, M. Grant, E. Sovero, D. Ching, J. Yoo, J. Romberg, A. Emami-Neyestanak, and E. Candès, "A nonuniform sampler for wideband spectrally-sparse environments," *IEEE JETCAS*, vol. 2, no. 3, pp. 516–529, Sep. 2012.

[7] J. Yoo, S. Becker, M. Monge, M. Loh, E. Candès, and A. Emami-Neyestanak, "Design and implementation of a fully integrated compressed-sensing signal acquisition system," in *IEEE ICASSP*, Mar. 2012, pp. 5325–5328.

[8] P. Maechler, "VLSI architectures for compressive sensing and sparse signal recovery," Ph.D. dissertation, ETH Zürich, Switzerland, 2013.

[9] J. N. Laska and R. G. Baraniuk, "Regime change: Bit-depth versus measurement-rate in compressive sensing," *arXiv:1110.3450v1*, Oct. 2011.

[10] Rice University. 1-bit Compressive sensing. [Online]. Available: http://dsp.rice.edu/1bitCS/

[11] A. Zymnis, S. Boyd, and E. Candés, "Compressed sensing with quantized measurements," *IEEE SPL*, vol. 17, no. 2, pp. 149–152, Feb. 2010.

[12] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[13] S.-Y. Lin, C.-L. Wey, and M.-D. Shieh, "Low-cost FFT processor for DVB-T2 applications," *IEEE TCE*, vol. 56, no. 4, pp. 2072–2079, Nov. 2010.