

Research Article

FPGA Implementation of Real-Time Compressive Sensing with Partial Fourier Dictionary

Yinghui Quan, Yachao Li, Xiaoxiao Gao, and Mengdao Xing

National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Yinghui Quan; quanyinghui@126.com

Received 18 July 2015; Accepted 13 December 2015

Academic Editor: Atsushi Mase

Copyright © 2016 Yinghui Quan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a novel real-time compressive sensing (CS) reconstruction which employs high density field-programmable gate array (FPGA) for hardware acceleration. Traditionally, CS can be implemented using a high-level computer language in a personal computer (PC) or multicore platforms, such as graphics processing units (GPUs) and Digital Signal Processors (DSPs). However, reconstruction algorithms are computing demanding and software implementation of these algorithms is extremely slow and power consuming. In this paper, the orthogonal matching pursuit (OMP) algorithm is refined to solve the sparse decomposition optimization for partial Fourier dictionary, which is always adopted in radar imaging and detection application. OMP reconstruction can be divided into two main stages: optimization which finds the closely correlated vectors and least square problem. For large scale dictionary, the implementation of correlation is time consuming since it often requires a large number of matrix multiplications. Also solving the least square problem always needs a scalable matrix decomposition operation. To solve these problems efficiently, the correlation optimization is implemented by fast Fourier transform (FFT) and the large scale least square problem is implemented by Conjugate Gradient (CG) technique, respectively. The proposed method is verified by FPGA (Xilinx Virtex-7 XC7VX690T) realization, revealing its effectiveness in real-time applications.

1. Introduction

Compressive sensing (CS) is a novel technology which allows sampling of sparse signals under sub-Nyquist rate and reconstructing the signal using computational intensive algorithms. It has received considerable attention and has been successfully applied in many fields, such as signal/image processing, radar imaging, communication, geophysics, and remote sensing [1–6]. In CS, it has been shown that a signal which is sparse or has a sparse representation in some bases can be recovered from a small number of random nonadaptive linear measurements. Unfortunately, high-performance sparse signal recovery algorithms typically require a significant computational effort [7]. While the computational complexity is not a major issue for applications where offline processing on central processing units (CPUs) or graphics processing units (GPUs) can be afforded (e.g., in MRI) [8–10], it becomes extremely challenging for applications requiring real-time processing at high throughput (e.g., in radar detection and imaging). Hence, to meet

the stringent throughput, latency, and power-consumption constraints of real-time CS-based radar systems, developing dedicated hardware implementations, such as application specific integrated circuits (ASICs) or field-programmable gate arrays (FPGAs), is of paramount importance [11].

A common approach to sparse reconstruction is known as Basis Pursuit (BP) [12]. This method uses convex optimization to find a signal representation in an overcomplete dictionary that minimizes l_1 norm of the coefficients in the representation. While being known to achieve accurate signal reconstruction, BP is more computationally intensive and has been shown to be significantly slower than other methods [13]. Orthogonal matching pursuit (OMP) which is proposed by Tropp and Gilbert [14] is an efficient and reliable reconstruction algorithm. OMP is a greedy method which identifies the location of one nonzero component of dictionary at a time. In order to converge, it requires a minimum number of samples in the order of $O(k \log N)$, where k is the signal's sparsity and N is the original dimension of the problem. Tropp and Gilbert show that, by performing

enough nonadaptive measurement, signal recovery is possible with high probability. OMP is widely used in CS signal reconstruction due to its computing efficiency and relative simplicity. Even so, the computation load is very large for real-time applications, in which the signal reconstruction should be done within specified time constraints.

Software implementation of these algorithms is time consuming since they often require massive matrix multiplications. These signal processing applications which require intense computation and simultaneous processing of large amount of data in real-time can make use of large scale field-programmable gate array (FPGA) platform for hardware acceleration. Modern high-capacity FPGA is an attractive alternative to accelerate scientific and engineering applications [15] due to the possible utilization of massive parallelism.

In order to speed up the complex reconstruction algorithms, a number of implementations on GPU, [8] ASICs, or reconfigurable FPGAs have been reported so far. The first ASIC implementation of CS reconstruction algorithms including matching pursuit (MP), gradient pursuit (GP), and OMP is presented in [16, 17] for channel estimation in wireless communication systems. FPGA implementation of OMP for generic CS problems of dimension 32×128 is developed in [18], processing signals with the sparsity of $k = 5$. In [19], a reconstruction algorithm similar to OMP is implemented on an FPGA to reconstruct band-sparse signals acquired by the modulated wideband converter. OMP-like implementation for problems of size 64×256 is proposed in [20], which, however, does not orthogonalize the estimation in every iteration. The first approximate message passing (AMP) designs [21] perform audio restoration and solve CS problems of size 512×1024 . Highly parallel FPGA implementation of OMP and AMP reconstruction algorithms are presented in [22], which run on a Xilinx Virtex-6 FPGA. The high speed architecture optimized based on [20] is discussed in [23], in which an architectural design and FPGA implementation of low-complexity compressive sensing reconstruction hardware are proposed. The proposed architecture supports vectors of length 256. And a thresholding method is applied for reducing the computation latency of dot product. In [24], a single-precision floating-point CS reconstruction engine implemented on a Kintex-7 FPGA is presented. In order to achieve high performance with maximum hardware utilization, a highly parallel architecture that shares computing resources among different tasks of OMP by using configurable processing elements (PEs) is presented.

OMP reconstruction can be divided into two main stages: optimization which finds the closely correlated atoms and least square problem. For large scale dictionary, the implementation of correlation is time consuming since it often requires a large number of matrix multiplications. The architectures listed above will take a lot of time for computing due to the path delay in processing the dot product and performing the matrix inverse. This paper aims to optimize the computational complexity according to the characteristics of the dictionary of CS-based radar applications. Some orthonormal transformations are utilized for CS-based radar signal reconstruction and image processing, such as Fourier basis,

Discrete Cosine Transform (DCT) basis, and wavelet basis. In radar applications, the partial Fourier dictionary is widely applied for spectrum reconstruction and radar imaging. In [25–27], a framework of high-resolution inversed synthetic aperture radar (ISAR) imaging with limited measured data is presented. During CS framework, the ISAR imaging is converted into a problem of signal reconstruction with orthogonal Fourier basis. Novel step-frequency radar (SFR) systems are proposed in [28, 29], which achieve the same resolution as conventional SFRs, while using significantly reduced bandwidth. This bandwidth reduction is accomplished by employing compressive sampling ideas and exploiting the sparseness of targets in the range-velocity space with redundancy Fourier basis. Gurbuz et al. proposed a compressive sensing data acquisition and imaging method for step frequency continuous wave (SFCW) ground penetrating radar (GPR) [30, 31], where the sparsity property and limited number of buried objects are successfully utilized for improving the performance of target detection. In [32–34], similar data acquisition and target reconstruction strategies are applied for SFCW through-the-wall radar imaging. The above-mentioned compressive GPR algorithms are discussed in the framework of SFCW radar. In these systems, the dense partial Fourier dictionary or modulated partial Fourier dictionary is adopted for target reconstruction. In [35], a novel velocity ambiguity resolving method is proposed for moving target indication (MTI), in which the compressive sensing (CS) is applied to recover the unambiguous Doppler spectrum of targets from the random pulse repetition frequency- (PRF-) jittering pulses. In [36], high-frequency (HF) over-the-horizon radar (OTHR) spectrum reconstruction of maneuvering target is proposed. The spectrum is reconstructed from incomplete measurements via CS by using a redundant Fourier-chirp dictionary. Therefore, the real-time hardware implementation of OMP for radar applications based on Fourier or modulated Fourier dictionary is very meaningful.

Based on the procedure of standard orthogonal matching pursuit (OMP) algorithm and the characteristics of Fourier-class dictionary, an improved solver named as IOMP is developed to optimize the computation complexity of optimization, which implements the correlation by fast Fourier transform (FFT) and the least squares by Conjugate Gradient (CG), respectively.

This paper is organized as follows. Section 2 reviews the CS theory, the traditional OMP algorithm, and the improved OMP algorithm for partial Fourier dictionary. Section 3 describes the circuit optimization and FPGA architecture for IOMP. In Section 4, several experiments using hardware acceleration are conducted. Then, the calculation efficiency, accuracy, and resources utilization are reported. Finally, Section 5 concludes the paper.

2. CS Theory and IOMP Algorithm

2.1. CS Theory. The developing theory of compressive sensing indicates that an unknown sparse signal can be exactly recovered from a very limited number of measurements with high probability by solving an optimization problem when some special conditions are met [1–6]. Consider

$\mathbf{x} \in C^N$, and suppose there exists a basis $\Phi = \{\phi_1, \phi_2, \dots, \phi_N\}$ satisfying $\mathbf{x} = \Phi\theta$, where θ is a sparse vector. Making a linear measurement process to \mathbf{x} , we have $\mathbf{y} = \mathbf{F}\mathbf{x} = \mathbf{F}\Phi\theta + \mathbf{e}$, where \mathbf{F} is a $M \times N$ ($M < N$) random measurement matrix and $\Psi = \mathbf{F}\Phi$ is defined as dictionary and \mathbf{e} is additive noise. The CS theory indicates that if the matrix Ψ has optimal restricted isometry property (RIP), accurate recovery of θ with high probability can be achieved by solving a convex problem as

$$\begin{aligned} \min \quad & (\|\hat{\theta}\|_1), \\ \text{subject to} \quad & \|\mathbf{y} - \Psi\hat{\theta}\|_2 \leq \varepsilon, \end{aligned} \quad (1)$$

where $\|\cdot\|_p$ denotes l_p norm and $\min(\cdot)$ denotes minimization. In the case of low signal to noise ratio (SNR), we should set a high noise level ε for good estimation. The noise level ε estimation of radar applications can be found in [25, 26].

2.2. OMP Algorithm. Consider a k -sparse signal \mathbf{x} sampled using a random matrix \mathbf{F} and \mathbf{y} is the sampled data. Our ultimate goal is to find k columns of dictionary Ψ which mostly contributed to \mathbf{y} . To begin with, residual \mathbf{R} is initialized to \mathbf{y} . For each iteration, a column of Ψ is chosen which has the best correlation with \mathbf{R} . The residual \mathbf{R} is then updated by subtracting the correlation from \mathbf{R} for next iteration. This is repeated for k times to find k columns of Ψ and the estimated signal $\hat{\mathbf{x}}$ is obtained by solving an overdetermined least square problem. The procedure of original OMP algorithm [14] is given below:

- (a) Initialize the residual $\mathbf{R}_0 = \mathbf{y}$, the index set $\Lambda = \emptyset$, the signal set $\hat{\Psi}_0 = \emptyset$, and the iteration counter $t = 1$.
- (b) Find the index λ_t which is most correlated to Ψ by solving the optimization problem:

$$\lambda_t = \arg \max_{j=1 \dots N} |\langle \mathbf{R}_{t-1}, \psi_j \rangle|, \quad (2)$$

where ψ_j is the j th column of Ψ .

- (c) Update the index set Λ_t and signal set $\hat{\Psi}_t$:

$$\begin{aligned} \Lambda_t &= \Lambda_{t-1} \cup \{\lambda_t\}, \\ \hat{\Psi}_t &= [\hat{\Psi}_{t-1} \quad \Psi_{\lambda_t}]. \end{aligned} \quad (3)$$

- (d) Solve a least square problem to obtain a new signal estimate:

$$\hat{\mathbf{x}}_t = \arg \min_{\mathbf{x}} \|\mathbf{y} - \hat{\Psi}_t \mathbf{x}\|_2. \quad (4)$$

- (e) Calculate the new residual according to

$$\mathbf{R}_t = \mathbf{y} - \hat{\Psi}_t \hat{\mathbf{x}}_t. \quad (5)$$

- (f) Increment t and return to step (b) if t is less than k or the residual error \mathbf{R}_t is larger than a preset noise level ξ .

There are three main tasks performed in each iteration: atom searching (AS) for identifying the active set (step (b)),

least square (LS) solving for computing the new signal estimation (step (d)), and residual update (step (e)). At iteration t , the number of complex floating point operations (FLOPs) required by each task is $O(MN)$, $O(t^3)$ (realize LS by Gram-Schmidt based QR decomposition), and $O(Mt)$, respectively [37]. The AS task is found to be the performance bottleneck as it contributes the most computation load in original OMP. Simultaneously, the LS task executed in each loop plays a significant role in the hardware utilization efficiency. This is because the matrix factorization in the LS task involves a variety of operations with complex data flows, which will introduce extra hardware complexity in terms of control and scheduling [38].

2.3. AS Improvement for Partial Fourier Dictionary. The OMP algorithm mentioned above can be implemented efficiently. The LS performed in each iteration in original OMP algorithm (step (d)) can be substituted by Gram-Schmidt orthogonalization [18]. Then, step (d) to step (f) of original OMP algorithm can be realized by the following steps:

- (d) Perform modified Gram-Schmidt orthogonalization by using the λ_t column of Ψ and $\mathbf{q}_{1 \dots t-1}$ in order to determine \mathbf{q}_t .
- (e) Calculate the new residual according to

$$\mathbf{R}_t = \mathbf{R}_{t-1} - \mathbf{q}_t \cdot \mathbf{q}_t' \cdot \mathbf{R}_{t-1}. \quad (6)$$

- (f) Increment t and return to step (b) if t is less than k or the residual error \mathbf{R}_t is larger than a preset value ξ .
- (g) Solve the least square problem to find $\hat{\mathbf{x}}$ for the indices in Λ_t :

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \hat{\Psi}_t \mathbf{x}\|_2, \quad (7)$$

where $\hat{\Psi} \in C^{M \times k}$ consists of the k relevant columns of Ψ .

By doing so, the architecture explained above is more efficient because the LS should only be executed once, rather than the original OMP algorithm, in which the LS should be taken in each iteration. Then, the most time consuming steps are AS for selecting the atoms (step (b)) and the LS (step (g)) for estimating the final $\hat{\mathbf{x}}$ in large sparsity situation. As we know, most of the reconstruction time is consumed in the optimization problem for finding k columns of Ψ . When the dimension of dictionary Ψ is large, matrix-vector multiplications are time and resource demanding to implement in hardware in real time. The algorithm proposed here is based on partial Fourier basis or modulated Fourier basis which is less complex for implementing in hardware. For radar detection and imaging, it is preferable to employ a structured basis, such as partial Fourier basis, Discrete Cosine Transform (DCT) basis, and wavelet basis. In this case, the OMP implementation can be processed efficiently. Most significantly, it is possible to compute the maximum correlation between a signal and the columns of the matrix in real time by using fast transforms. Second, the matrix can

be preconstructed and it is only necessary to store only one vector from dictionary matrix.

First, we discuss the AS optimization of partial Fourier dictionary. Define the redundant time frequency dictionary as

$$\Psi = \mathbf{F}\Phi, \quad (8)$$

where Φ and \mathbf{F} are defined as the Fourier basis of size $N \times N$ and random measurement matrix of size $M \times N$ ($M < N$), respectively. \mathbf{F} is constructed by randomly selecting M rows from an $N \times N$ identity matrix. The schematic of partial Fourier dictionary construction is expressed as

$$\Psi = \left[e^{j2\pi f_1 t} \mid e^{j2\pi f_2 t} \mid \dots \mid e^{j2\pi f_N t} \right] \quad (9)$$

$$= \begin{bmatrix} e^{j2\pi f_1 t_0} & e^{j2\pi f_2 t_0} & \dots & e^{j2\pi f_N t_0} \\ \vdots & \vdots & \dots & \vdots \\ e^{j2\pi f_1 t_i} & e^{j2\pi f_2 t_i} & \dots & e^{j2\pi f_N t_i} \\ & & \dots & \\ \vdots & \vdots & \dots & \vdots \\ & & \dots & \\ e^{j2\pi f_1 t_j} & e^{j2\pi f_2 t_j} & \dots & e^{j2\pi f_N t_j} \\ \vdots & \vdots & \dots & \vdots \\ e^{j2\pi f_1 t_M} & e^{j2\pi f_2 t_M} & \dots & e^{j2\pi f_N t_M} \end{bmatrix}_{M \times N}.$$

$$\Psi = \left\{ \underbrace{\varphi(1, 1) \dots \varphi(1, L)}_L \dots \dots \underbrace{\varphi(N, 1) \dots \varphi(N, L)}_L \right\}_{M \times (N \cdot L)}. \quad (12)$$

The Fourier-chirp basis can be formulated as

$$\varphi(u, l) = \alpha_u \odot \beta_l,$$

$$\alpha_u = \left[p^u \dots p^{um} \dots p^{uM} \right]_{M \times 1}^T, \quad m \in \mathbf{I}', \quad (13)$$

$$\beta_l = \left[q^l \dots q^{lm^2} \dots q^{lM^2} \right]_{M \times 1}^T, \quad m \in \mathbf{I}',$$

where “ \odot ” denotes Hadamard product, $p = \exp(-j(2\pi/N))$, $q = \exp(-j(\pi/N^2)\Delta l)$, and Δl is the grid step of chirp rate. $\mathbf{I} = [1 \ 2 \ \dots \ N]_{N \times 1}^T$ and \mathbf{I}' is a subset of \mathbf{I} . L is the dimension of chirp rate.

Suppose we need to estimate the t th signal components of \mathbf{y} . Its Doppler frequency u_t and chirp rate l_t are achieved when the inner product of \mathbf{y} and the basis in Ψ reaches its maximum:

$$\langle u_t, l_t \rangle = \arg \max_{u, l} |\langle \mathbf{R}_{t-1}, \varphi(u, l) \rangle| \quad (14)$$

Then, we apply IFFT instead of matrix-vector multiplications directly to enhance efficiency in inner product computation

Rewrite the optimization formula of finding the index λ_t which is most correlated to Ψ by solving the following problem:

$$\lambda_t = \arg \max_{j=1 \dots N} \left| \langle \mathbf{R}_{t-1}, \Psi_j \rangle \right|. \quad (10)$$

Instead of calculating it one by one by vector dot product, we may obtain a column through the following fast Fourier transform computation:

$$\lambda_t = \arg \max_{j=1 \dots N} \text{IFFT}(\widetilde{\mathbf{R}}_{t-1}), \quad (11)$$

where $\widetilde{\mathbf{R}}_{t-1}$ is a $N \times 1$ vector based on \mathbf{R}_{t-1} , with the missing element zero padded according to measurement matrix \mathbf{F} .

Next, we discuss the AS optimization of Fourier-chirp dictionary which is adopted in [36]. By this way, the spectrum is reconstructed from incomplete measurements via CS by using a redundant Fourier-chirp dictionary for maneuvering targets. For our convenient derivation, let u and l stand for the Doppler frequency and chirp rate, respectively. And then the partial Fourier-chirp dictionary is able to be constructed:

in (14). Denote the inner product matrix corresponding to the dictionary and the measurement \mathbf{y} as

$$\mathbf{D}(u, l) = \varphi(u, l)^H \cdot \mathbf{y}. \quad (15)$$

Apparently, \mathbf{D} is an $N \times L$ matrix. Instead of calculating its element one by one by vector dot product, we may obtain a column through the following fast Fourier transform computation:

$$\mathbf{D}(:, l) = \text{IFFT} \left\{ \widetilde{\mathbf{R}}_{t-1} \odot \widetilde{\beta}_l \right\}. \quad (16)$$

$\widetilde{\mathbf{R}}_{t-1}$ and $\widetilde{\beta}_l$ are $N \times 1$ vectors based on \mathbf{R}_{t-1} and β_l , with the missing data zero padded. By seeking the maximum element in \mathbf{D} , we can get the estimation of $\langle u_t, l_t \rangle$. Then, we may achieve the signal estimation.

2.4. LS Improvement. There are two categories of methods for solving linear systems. The first is direct method, where the solution is computed through lower/upper triangular decomposition and solving triangular system. The second

```

Input:
A (priori known matrix)
b (priori known vector)
x (initial value)
e (normalized residual, iterative condition)
Output:
 $\hat{\mathbf{x}}$  (sparse solution)
function  $\hat{\mathbf{x}} = \text{cg}(\mathbf{A}, \mathbf{b}, \mathbf{x}, e)$ 
     $\mathbf{r} = \mathbf{b}$ ;
     $\mathbf{d} = \mathbf{r}$ ;
     $\delta_{\text{new}} = \mathbf{r}^H \mathbf{r}$ ;
     $\delta_{\text{old}} = \mathbf{b}^H \mathbf{b}$ ;
    while  $\delta_{\text{new}} > e^2 \cdot \delta_{\text{old}}$ 
         $\mathbf{q} = \mathbf{A}\mathbf{d}$ ;
         $\alpha = \delta_{\text{old}} / (\mathbf{d}^H \mathbf{q})$ ;
         $\mathbf{x} = \mathbf{x} + \alpha \mathbf{d}$ ;
         $\mathbf{r} = \mathbf{r} - \alpha \mathbf{q}$ ;
         $\delta_{\text{new}} = \mathbf{r}^H \mathbf{r}$ ;
         $\beta = \delta_{\text{new}} / \delta_{\text{old}}$ ;
         $\mathbf{d} = \mathbf{r} + \beta \mathbf{d}$ ;
    end

```

ALGORITHM 1: Pseudocode of CG algorithm.

is iterative method, where the solution is approximated by performing iterations from an initial vector. Direct method is only feasibly applied for small systems. In contrast, the iterative methods are suited for solving larger scale problem. One well-studied method that has been proven to be very efficient in software and robust at solving large sparse linear systems is the Conjugate Gradient (CG) algorithm. In this paper we present a hardware architecture of CG method which takes advantage of wide parallelization and deep-pipelining of FPGAs.

The pseudocode of the corresponding CG algorithm is listed in Algorithm 1.

In summary, the flowchart of OMP and IOMP algorithm is shown in Figure 1. The IOMP is optimized for partial Fourier dictionary or modulated partial Fourier dictionary which is always adopted in radar detection and imaging.

3. Proposed FPGA Architecture

In this section, FPGA-based hardware architecture is proposed to take a compromise on speed, resource utilization, and accuracy in the eventual circuit implementation. The newly available Xilinx FPGA (XC7VX690T) is utilized to validate the implementation of proposed approach. The block diagram of top-level design is illustrated in Figure 2.

As shown in Figure 2, the hardware design principally comprises two components: atom searching (AS) and least square solving (LS). AS is used to find out the atoms which are most correlated to residual \mathbf{R} in dictionary Ψ . LS is adopted to figure out sparse solutions based on CG iteration method. Firstly, zero padding is conducted to residual \mathbf{R} in corresponding positions so as to allow for carrying out IFFT operations which are capable of calculating the correlation

between residual \mathbf{R} and dictionary Ψ . Followed by the most correlated atoms which are found out, the Gram-Schmidt orthogonalization is in succession executed for updating the residual \mathbf{R} . Repeat the aforementioned steps until all atoms are found out. Finally, the CG iteration method is used for quickly figuring out the sparse solution $\hat{\mathbf{x}}$.

Figure 3(a) illustrates the hardware architecture of AS, whose input parameters are sampled data \mathbf{y} and dictionary Ψ which are stored in external double-data-rate (DDR3) synchronous dynamic random access memory (SDRAM). As depicted in Figure 3(a), processing element (PE) is a modularized circuit which is shared and available for calculating the product of two complex vectors. Its inner structure is depicted in Figure 3(b). Multiply accumulator (MAC) is a Xilinx LogiCORE IP core which provides multiply-accumulate implementations of two fixed-point vectors. FP1 and FP2 are floating-point IP cores serving as floating-to-fixed point conversion and fixed-to-floating point conversion, respectively. The output data of FP1 hold a 32-bit width with 16 bits reserved for the integer part and 16 bits for the fractional part. To offer sufficient bit width for multiply accumulator operations, the accumulation width of MAC is configured to the default value of 64 bits.

Based on comprehensive consideration of computation time and precision, data to be processed by FFT IP core are converted to 32-bit fixed point data with 16 bits for the integer part and 16 bits for the fraction part. FFT IP core utilizes XtremeDSP slices to calculate and block RAMs to store intermediate data. Full-precision unscaled arithmetic is chosen for the accurate computation. To perform 2048-point transform, the output data of FFT IP core are 44-bit width with 28 integer bits and 16 fractional bits. Consider the real-data test and redundant sign bits; integer bits are intercepted to 20 bits. As shown in Figure 3(a), the output data of FFT are converted to floating-point format by using FP2, whose input integer width and fraction width are configured to 20 bits and 16 bits, respectively. Error caused by the conversion between fixed-point and floating point is about $1.05e - 3$, which is negligible.

Ultimately, AS outputs the most correlated atoms which are subsequently utilized for LS calculations.

The hardware structure of LS is depicted in Figure 4(a). Data to be processed are received from sampled data \mathbf{y} and the output of AS, that is, matrix $\hat{\Psi}_t$. As illustrated in Figure 4(a), matrix-matrix and matrix-vector multiplications are implemented by using paralleled PEs which have the same structure as that illustrated in Figure 3(b). The inner structure of basic IP which is utilized for updating sparse solution \mathbf{x} and residual \mathbf{r} is depicted in Figure 4(b). Finally, LS outputs a sparse vector $\hat{\mathbf{x}}$ which is the optimal solution to the problem with partial Fourier dictionary.

Floating-Point Operator v5.0 is utilized here to perform floating-point arithmetic on selected FPGA device. For carrying out addition/subtraction and multiply operations, the core is configured to usage of $2 \times \text{DSP48E}$ and $3 \times \text{DSP48E}$, respectively. Both fix-point division and MAC operations require no DSP48Es, and so does the conversion between floating point and fixed point. PE module which consists of

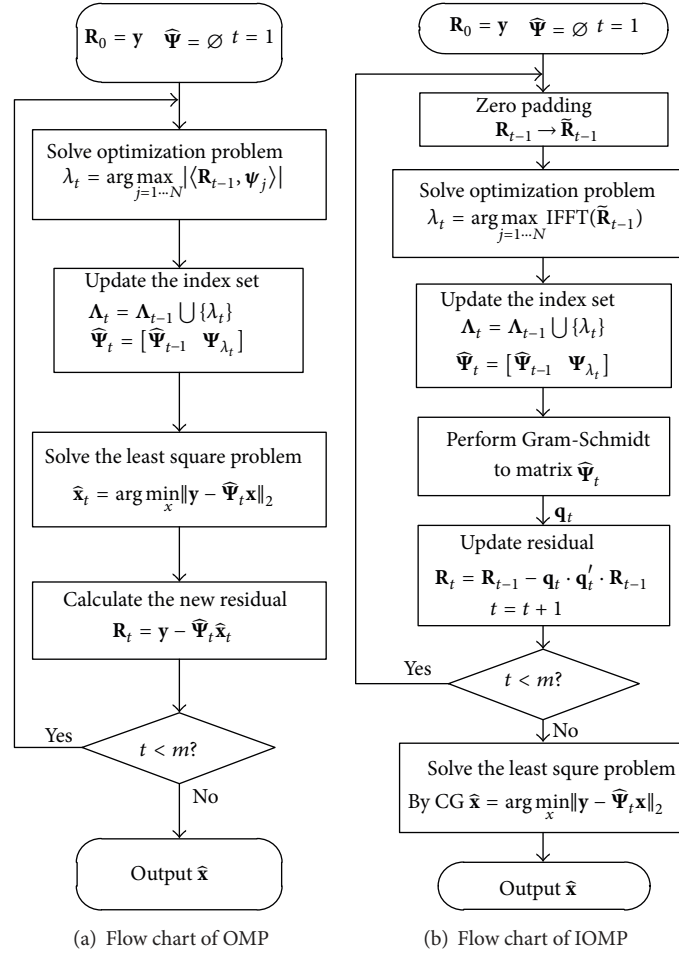


FIGURE 1: Flow chart of OMP and IOMP algorithm.

10 floating-point cores and 4 MACs utilizes only 4 DSP48Es despite the increase of input vector length.

As analyzed above, we optimized the circuit in resource utilization and computation speed, respectively. FPGA's embedded dual-port block RAMs are utilized for expediently reading/writing data, reducing data transfer delay, and simplifying sequential control.

4. Experimental Results

In this section, we will provide the comparison between other CS reconstruction algorithms and proposed IOMP algorithm in basic aspects such as processing speed, resource utilization, and computation precision.

4.1. Comparison between Traditional OMP and Proposed IOMP. The Verilog HDL program developed for IOMP algorithm is simulated and implemented based on XC7VX690T FPGA and runs at 165 MHz. In proposed technique, calculations of the correlations between dictionary Ψ and residual \mathbf{R} are conducted by IFFT operations instead of inner product. Assume the number of columns of matrix Ψ is four times more than that of rows. Based on the utilization

of DSP48Es of IFFT IP core provided by ISE14.3, vector-vector multiplications are calculated in parallel to ensure the same utilization of DSP48Es by inner product and IFFT. Under the aforementioned conditions, experimental results concerning clock cycles consumed by IFFT and inner product are described in Figure 5.

Figure 5(a) illustrates the comparative results. Red curve which rises rapidly with the increase of column vector length N illustrates the computation time of inner product, while blue curve which depicts the computation time of IFFT operations increases lentamente. Figure 5(b) illustrates the result of enlarged drawing of Figure 5(a) when the vector length is less than 256. Obviously, inner product will take far more time than IFFT does when the vector length is larger than 128. IFFT operations will achieve the speedup of 12 \times over dot product calculations under the condition of same resource utilization (DSP48E) when the vector length is 2048.

CG iteration method which has excellent properties of small memory space requirement and high iteration speed is utilized for solving the least square problem. As an iteration method, the precision of CG is determined by bit width and iteration times. Normalized mean square error (MSE) of reconstruction result is utilized as the evaluation criterion of

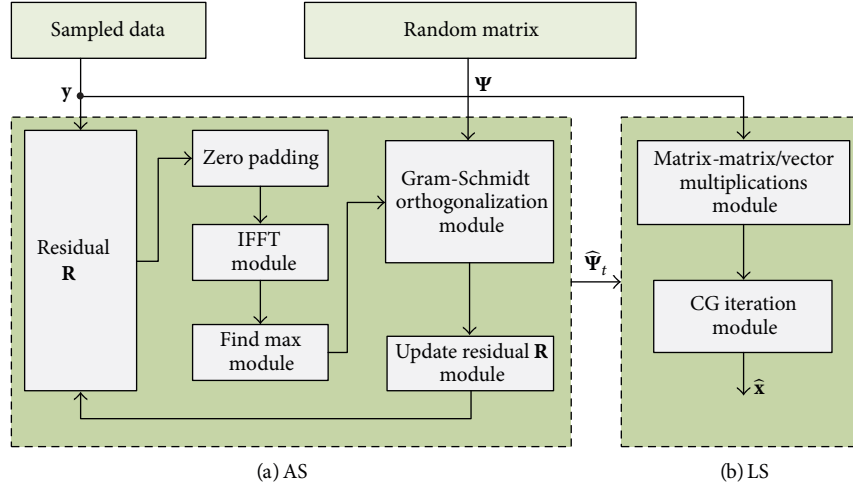


FIGURE 2: Detailed architecture diagram of top-level design.

computation precision, and $1.0e^{-4}$ is required in this design. Experiment about the relationship between precision and iteration times is carried out. The experimental result is depicted in Figure 6, where k is the sparsity. Generally six-time iterations can exactly achieve the required precision under the condition of $k \leq 15$.

Mathematical calculations, for example, floating-point multiply and add/subtract operations, are realized jointly by DSP48Es and LUTs, and the latency T_1 is one clock cycle based on the maximum usage of DSP48Es. Floating-point division is realized by LUTs and its computing latency T_2 is 18 clock cycles. As components of PE module, FP1/FP2 and MAC have a latency of one clock cycle and $(m + 1)$ clock cycles, respectively, where m is the input vector length. Thus, PE module totally requires $T'_m = m + 4$ clock cycles to accomplish the complex vector multiplications. To accomplish one iteration of CG, totally T^1 clock cycles described as in (17) are required:

$$T^1 = 3T'_m + 10T_1 + 2T_2 = (3k + 58) T_{\text{clk}}, \quad m = k, \quad (17)$$

where $k \geq 1$ is the sparsity and T_{clk} represents one clock cycle.

As discussed above, to reconstruct the signal of sparsity 12, six-time iterations can exactly satisfy the demand for precision. Totally T clock cycles are required to accomplish the whole iterations:

$$T = 6 \cdot T^1 = (18k + 348) T_{\text{clk}}. \quad (18)$$

QR decomposition is another widely used and effective least square method. Unfortunately, because of the considerable computation complexity, its computation load and resource requirement will have a sharp growth with the increase of matrix size. Even though parallel architecture is considered in the hardware implementation, with respect to computation time and resource utilization, QR decomposition can hardly obtain the excellent performance that CG iteration creates. Figure 7 gives the clock cycles that CG iteration took with the increase of sparsity.

TABLE 1: Resources utilization.

Logic utilization	Used	Available	Utilization
Registers	193,053	866,400	22%
LUTs	282,332	433,200	65%
Block RAM/FIFO	573	1,470	39%
DSP48Es	1,745	3,600	48%

According to above-mentioned experimental results, the proposed technique is more feasible for processing large-scale sparse dictionary and reconstructing signals with large sparsity.

4.2. Calculation Latency and Resource Utilization of Proposed IOMP Algorithm. In this paper, IOMP algorithm is utilized for reconstructing sparse signals. To process a 512-length measured vector of sparsity 12, assume the size of Fourier basis matrix Φ and dictionary Ψ to be 2048×2048 and 512×2048 , respectively. It requires 4245 clock cycles to find out one column of dictionary, 1053 clock cycles for Schmidt orthogonalization and updating residual \mathbf{R} , and 1080 clock cycles for LS computation, respectively. Thus, totally 64656 clock cycles, that is, $391.8 \mu\text{s}$, are required to accomplish the whole reconstruction work.

Table 1 provides the synthesis and implemented result reported by the Xilinx ISE14.3. Notice that FPGA resources are sufficient for processing reconstruction algorithm in this example.

Radar echoes are complex data containing phase and amplitude information. Theoretically, to accomplish the multiply operations of two complex numbers, totally four multiplications, one addition, and one subtraction are conducted. Relatively, only one multiplication is needed for the multiplication of two real numbers. Thus, processing complex data has considerable computation load over processing real data.

Many works about OMP accelerated reconstruction, including both software implementation and hardware

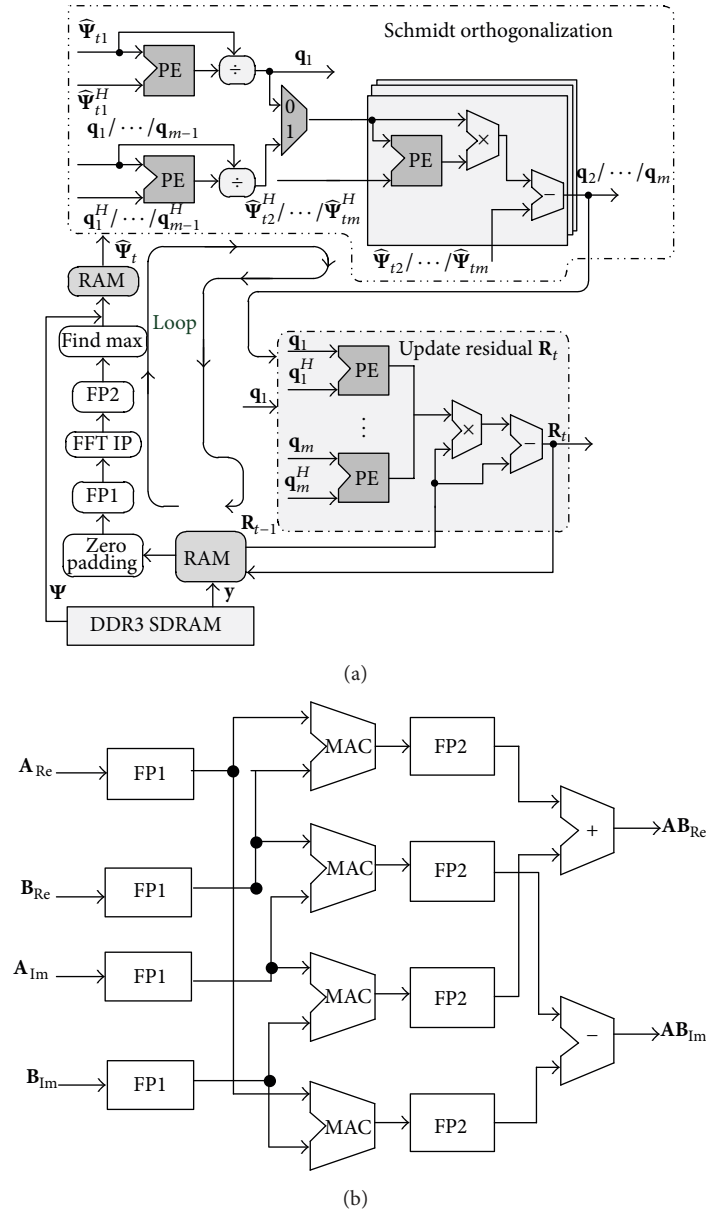


FIGURE 3: (a) Hardware structure of AS. (b) Inner structure of PE.

implementation, have been published [9, 18, 22, 24, 39, 40]. Comparisons between them and this work are carried out with respect to computing speed, maximum working frequency, computation accuracy, and so forth. Table 2 expatiates on the comparative results. Compared to other works based on FPGA, a higher working frequency of 165 MHz is achieved in this work. Authors in [18] present VLSI implementation of an optimized OMP algorithm to process the dictionary of size 32×128 with sparsity of $k = 5$, which totally takes $24 \mu s$. In [22], highly parallel FPGA implementations of two CS reconstruction algorithms OMP and AMP are proposed, which run on a Xilinx Virtex-6 FPGA. The hardware realization discussed in [24] totally takes $39.9 \cdot k \mu s$, where k is the sparsity. It totally takes $478.8 \mu s$ to accomplish CS reconstruction when $k = 12$.

Besides, the reconstruction work is also executed on CPU and GPU. A parallel architecture of implementing OMP algorithm based on GTX480 GPU is proposed in [39]. To reconstruct a signal of 8192 points with sparsity 32, totally 15 ms is required. Work [9] proposes a CS method for many-core architectures, for example, the cell processor, GPUs, and CPUs. Compared to the implementation by Intel Core i7 whose computing speed is limited by available dominant frequency, apparently a higher reconstruction speed can be achieved by this work. Meanwhile, CPU and GPU exactly consume much more power. Work [40] executed a MATLAB code of OMP algorithm on Intel Core Duo CPU at 2.8 GHz. To reconstruct a 128-length signal of sparsity 5, totally $606 \mu s$ is required. In contrast, our work obtains a speedup of 33 times.

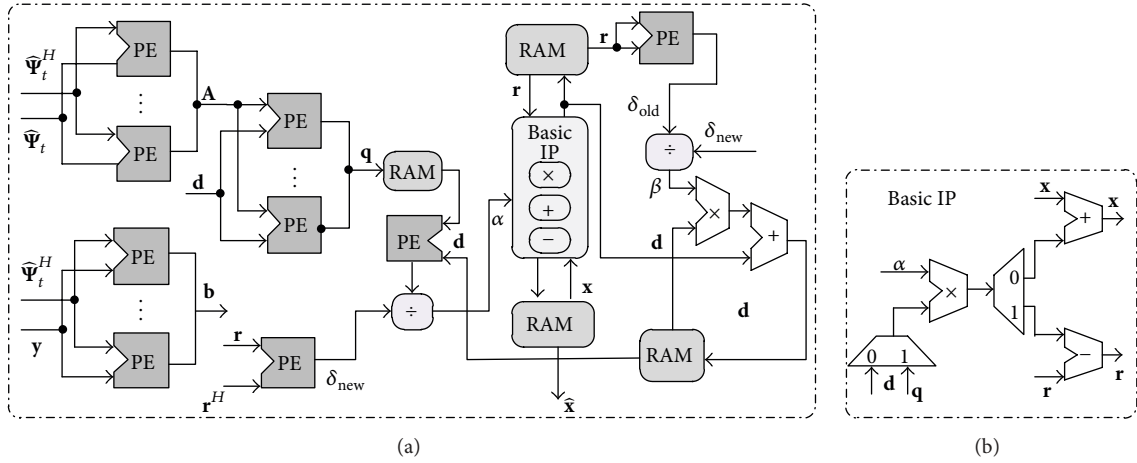


FIGURE 4: (a) Hardware structure of LS. (b) Inner structure of basic IP.

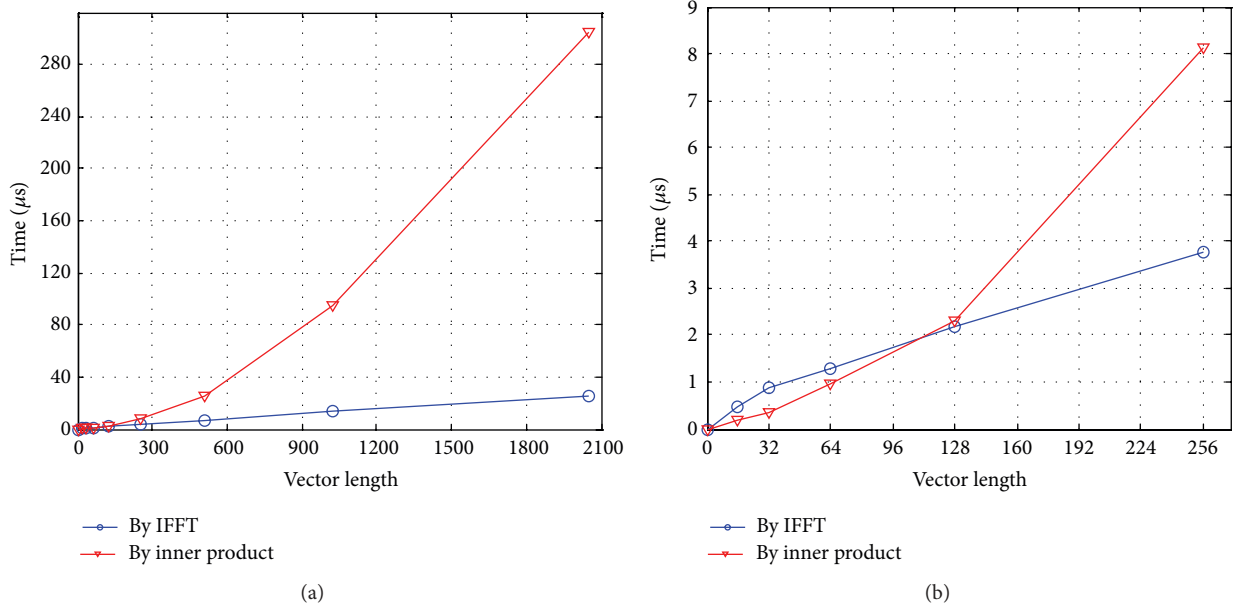


FIGURE 5: (a) Computing time of IFFT and inner product. (b) Partial enlarged drawing of (a).

TABLE 2: Comparison between proposed approach and some other works.

	Size of dictionary	Sparsity	Frequency (MHz)	Time	Accuracy	Data format
FPGA Virtex-5 [18]	32×128	5	39	$24 \mu\text{s}$	—	32-bit fixed point real data
FPGA Virtex-6 [22]	256×1024	12	100	$158.7 \mu\text{s}$	—	18-bit fixed point real data
Kintex-7 [24]	640×1470	≤ 320	53.7	$39.9 \cdot k \mu\text{s}$	—	Single-precision real data
NVIDIA GTX480 [39]	512×8192	64	—	15 ms	—	—
Intel Core i7 [9]	64×512	12	3000	25 ms	$1.24e-03$	Single-precision real data
Intel Core DUO [40]	32×128	5	2800	$606 \mu\text{s}$	—	32-bit fixed point real data
FPGA Virtex-7	32×128	5	165	$18.3 \mu\text{s}$	$1.2e-03$	Single-precision and fixed-point hybrid complex data
This work	512×2048	12	—	$391.8 \mu\text{s}$	—	—

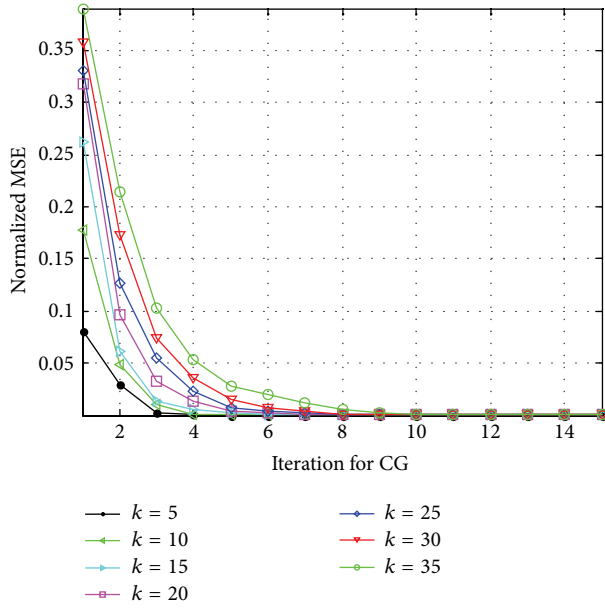


FIGURE 6: Relationship between precision and iteration times.

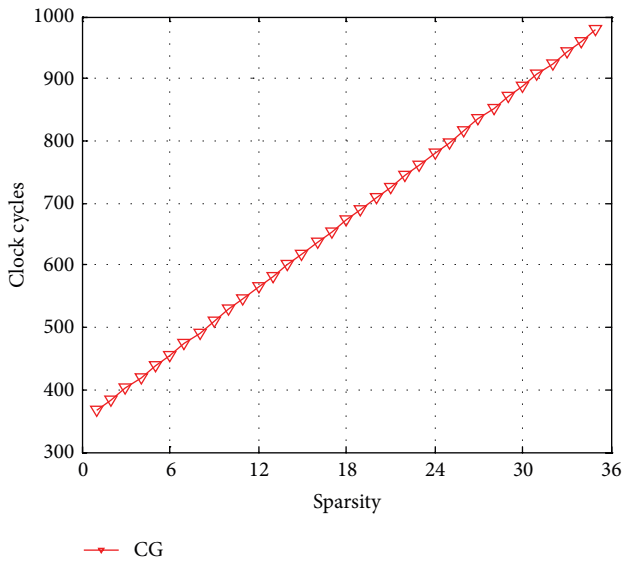


FIGURE 7: Clock cycles of CG iteration method.

It is worth mentioning that the data processed in our architecture is complex data format, while the architectures proposed in references are processing the real data. In summary, compared to other works, the proposed CS reconstruction technique in this paper has an excellent performance with respect to computing speed, accuracy, and flexible applicability in case that partial Fourier dictionary or modulated partial Fourier dictionary is adopted.

5. Conclusion

In this paper, we focus on the real-time implementation of compressive sensing with partial Fourier dictionary which is

always adopted for radar applications. And the high density FPGA is used for hardware acceleration. According to the characteristics of the dictionary, an improved orthogonal matching pursuit algorithm is proposed to solve the sparse optimization efficiently. In this scheme, the correlation is implemented by FFT and the least square is realized by CG, respectively. Fast and area-efficient FPGA realization is provided to meet the real-time requirement of CS-based radar. The hardware architecture, the resource utilization, the computation latency, and the computation precision are analyzed in detail. Finally, a comparison with other works is made to evaluate the effectiveness of proposed approach.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

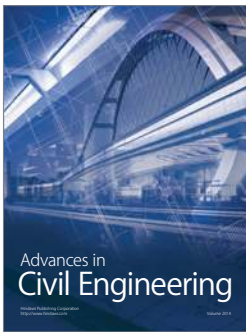
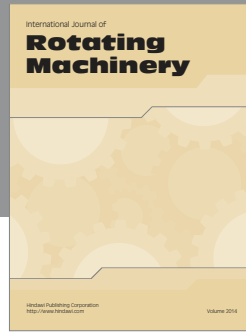
This work was supported by National Natural Science Foundations of China (Grants nos. 61303035 and 61471283).

References

- [1] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [2] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [3] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 5406–5425, 2006.
- [4] L. Zhang, Z.-J. Qiao, M.-D. Xing, J.-L. Sheng, R. Guo, and Z. Bao, "High-resolution ISAR imaging by exploiting sparse apertures," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 2, pp. 997–1008, 2012.
- [5] W. Zhang, M. G. Amin, F. Ahmad, A. Hoorfar, and G. E. Smith, "Ultrawideband impulse radar through-the-wall imaging with compressive sensing," *International Journal of Antennas and Propagation*, vol. 2012, Article ID 251497, 11 pages, 2012.
- [6] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling: a sensing/sampling paradigm that goes against the common knowledge in data acquisition," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [7] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.
- [8] M. Andrecut, "Fast GPU implementation of sparse signal recovery from random projections," <http://arxiv.org/abs/0809.1833v2>.
- [9] A. Borghi, J. Darbon, S. Peyronnet, T. Chan, and S. Osher, "A simple compressive sensing algorithm for parallel many-core architectures," CAM Report 08-64, UCLA, 2008.
- [10] D. Yang, G. D. Peterson, and H. Li, "Compressed sensing and cholesky decomposition on FPGAs and GPUs," *Parallel Computing*, vol. 38, no. 8, pp. 421–437, 2012.
- [11] M. Karkooti, J. R. Cavallaro, and C. Dick, "FPGA implementation of matrix inversion using QRD-RLS algorithm," in

- Proceedings of the 39th Asilomar Conference on Signals, Systems and Computers*, pp. 1625–1629, November 2005.
- [12] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
 - [13] S. Kunis and H. Rauhut, “Random sampling of sparse trigonometric polynomials, II. Orthogonal matching pursuit versus basis pursuit,” *Foundations of Computational Mathematics*, vol. 8, no. 6, pp. 737–763, 2008.
 - [14] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
 - [15] O. Storaasli and D. Strenski, “Exploring accelerating science applications with FPGAs,” in *Proceedings of the 3rd Annual Reconfigurable Systems Summer Institute (RSSI '07)*, pp. 1–30, Urbana, Ill, USA, July 2007.
 - [16] P. Maechler, P. Greisen, N. Felber, and A. Burg, “Matching pursuit: evaluation and implementation for LTE channel estimation,” in *Proceedings of the IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems (ISCAS '10)*, pp. 589–592, Paris, France, May–June 2010.
 - [17] P. Maechler, P. Greisen, B. Sporrer, S. Steiner, N. Felber, and A. Burg, “Implementation of greedy algorithms for LTE sparse channel estimation,” in *Proceedings of the 44th Asilomar Conference on Signals, Systems and Computers*, pp. 400–405, IEEE, Pacific Grove, Calif, USA, November 2010.
 - [18] A. Septimus and R. Steinberg, “Compressive sampling hardware reconstruction,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '10)*, pp. 3316–3319, Paris, France, May–June 2010.
 - [19] M. Mishali, R. Hilgendorf, E. Shoshan, I. Rivkin, and Y. C. Eldar, “Generic sensing hardware and real-time reconstruction for structured analog signals,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '11)*, pp. 1748–1751, IEEE, Rio de Janeiro, Brazil, May 2011.
 - [20] J. L. V. M. Stanislaus and T. Mohsenin, “High performance compressive sensing reconstruction hardware with QRD process,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '12)*, pp. 29–32, Seoul, Republic of Korea, May 2012.
 - [21] P. Maechler, C. Studer, D. E. Bellasi et al., “VLSI design of approximate message passing for signal restoration and compressive sensing,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 579–590, 2012.
 - [22] L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, “High-speed compressed sensing reconstruction on FPGA using OMP and AMP,” in *Proceedings of the 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS '12)*, pp. 53–56, IEEE, Seville, Spain, December 2012.
 - [23] J. L. V. M. Stanislaus and T. Mohsenin, “Low-complexity FPGA implementation of compressive sensing reconstruction,” in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC '13)*, pp. 671–675, San Diego, Calif, USA, January 2013.
 - [24] F. B. Ren, R. Dorrace, W. Y. Xu, and D. Marković, “A single-precision compressive sensing signal reconstruction engine on FPGAs,” in *Proceedings of the 23rd International Conference on Field Programmable Logic and Applications (FPL '13)*, pp. 1–4, IEEE, Porto, Portugal, September 2013.
 - [25] L. Zhang, M. D. Xing, C.-W. Qiu, J. Li, and Z. Bao, “Achieving higher resolution ISAR imaging with limited pulses via compressed sampling,” *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 3, pp. 567–571, 2009.
 - [26] L. Zhang, M. D. Xing, C. W. Qiu, J. L. Sheng, Y. C. Li, and Z. Bao, “Resolution enhancement for inverse synthetic aperture radar imaging under low SNR via improved compressive sensing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 10, pp. 3824–3838, 2010.
 - [27] H. X. Wang, Y. H. Quan, M. D. Xing, and S. H. Zhang, “ISAR imaging via sparse probing frequencies,” *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 451–455, 2011.
 - [28] S. Shah, Y. Yu, and A. Petropulu, “Step-frequency radar with compressive sampling (SFR-CS),” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 1686–1689, IEEE, Dallas, Tex, USA, March 2010.
 - [29] L. Zhang, Z.-J. Qiao, M. D. Xing, Y. C. Li, and Z. Bao, “High-resolution ISAR imaging with sparse stepped-frequency waveforms,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4630–4651, 2011.
 - [30] A. C. Gurbuz, J. H. McClellan, and W. R. Scott Jr., “Compressive sensing for subsurface imaging using ground penetrating radar,” *Signal Processing*, vol. 89, no. 10, pp. 1959–1972, 2009.
 - [31] A. C. Gurbuz, J. H. McClellan, and W. R. Scott, “A compressive sensing data acquisition and imaging method for stepped frequency GPRs,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2640–2650, 2009.
 - [32] Y. S. Yoon and M. G. Amin, “Compressed sensing technique for high-resolution radar imaging,” in *Signal Processing, Sensor Fusion, and Target Recognition XVII*, vol. 6968 of *Proceedings of SPIE*, Orlando, Fla, USA, March 2008.
 - [33] L. C. Potter, E. Ertin, J. T. Parker, and M. Çetin, “Sparsity and compressed sensing in radar imaging,” *Proceedings of the IEEE*, vol. 98, no. 6, Article ID 5420035, pp. 1006–1020, 2010.
 - [34] Q. Huang, L. Qu, B. Wu, and G. Fang, “UWB through-wall imaging based on compressive sensing,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 3, pp. 1408–1415, 2010.
 - [35] Y. H. Quan, L. Zhang, M. D. Xing, and Z. Bao, “Velocity ambiguity resolving for moving target indication by compressed sensing,” *Electronics Letters*, vol. 47, no. 22, pp. 1249–1251, 2011.
 - [36] Y. Quan, L. Zhang, Y. Li, H. Wang, and M. Xing, “OTHR spectrum reconstruction of maneuvering target with compressive sensing,” *International Journal of Antennas and Propagation*, vol. 2014, Article ID 870352, 10 pages, 2014.
 - [37] F. B. Ren, R. Dorrace, W. Y. Xu, and D. Marković, “A single-precision compressive sensing signal reconstruction engine on FPGAs,” in *Proceedings of the 23rd International Conference on Field Programmable Logic and Applications (FPL '13)*, pp. 1–4, IEEE, Porto, Portugal, September 2013.
 - [38] C. K. Singh, S. H. Prasad, and P. T. Balsara, “VLSI architecture for matrix inversion using modified Gram-Schmidt based QR decomposition,” in *Proceedings of the 20th International Conference on VLSI Design. Held Jointly with 6th International Conference on Embedded Systems*, pp. 836–841, IEEE, Bangalore, India, January 2007.
 - [39] Y. Fang, L. Chen, J. Wu, and B. Huang, “GPU implementation of orthogonal matching pursuit for compressive sensing,” in *Proceedings of the 17th IEEE International Conference on Parallel and Distributed Systems (ICPADS '11)*, pp. 1044–1047, IEEE, Tainan, Taiwan, December 2011.

- [40] P. Blache, H. Rabah, and A. Amira, "High level prototyping and FPGA implementation of the orthogonal matching pursuit algorithm," in *Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA '12)*, pp. 1336–1340, IEEE, Montreal, Canada, July 2012.



Hindawi

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

