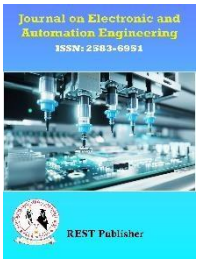
Journal on Electronic and Automation Engineering

Vol: 2(2), June 2023

REST Publisher; ISSN: 2583-6951 (Online)

Website: https://restpublisher.com/journals/jeae/

DOI: https://doi.org/10.46632/jeae/2/2/20



Enhanced OMP for Compressible Sensing Synthesis Using FPGA Implementation

V. Senthil Kumaran, K. Giri, S. Saraswathi, R. Priyanka

Mahendra Engineering College, Namakkal, Tamil Nadu, India.

*Corresponding author: senthilkumaranv@mahendra.info

Abstract: *The research work offers a unique -created building for reconstructing compressively sensed signals using the Orthogonal Matching Pursuit (OMP) method. This research work investigated the difficulty of computation and data dependencies among the various stages of the OMP algorithm in order to build a structure that gives greater speed while consuming less space. The present paper presents an enhanced OMP technique and its employment using Xilinx Vivado High-Level Fabrication (VHLF). This study employs Gram-Schmidt orthogonalization to enhance signalling residue updating so that sign restoration individual requirements to run the lowest-squares explanation once, subsequent in a substantial decrease in the overall sum of vector processes in an electronic design. Results from simulations demonstrate that our OMP method is similarly precise in signal restoration as the initial OMP technique. Our strategy offers a quick and adaptable deployment for a variety of signal sizes, measuring grid sizes, and sparseness levels. The suggested approach may recuperate a 128-length indicate with evaluation the amount M = 32 and sparseness K = 8 and K = 5 in 21 and 13.2 µs, indicating at least a 20.9% and 21.3% enhancement contrasted against the present HLS-based works. The findings are comparable to those of typical hardware description language (HDL) employments. The findings demonstrate, enhanced OMP method not merely has a faster rebuilding time compared to comparable current HLS-based works, but it can additionally contend with previous attempts developed via the classic FPGA construction path.*

Keywords*: Compressed sensing (CS), VHLF, FPGA, OMP, sparse signal recovery*

1. INTRODUCTION

Conventional signal processing (SP) relies around the Nyquist-Shannon sampling theorem [1]. To properly maintain all of the data contained in the initial analog signal, the digital frequency of sampling needs to be no less than twice that of the highest possible frequency. Tao et al. [2] showed that in the event the signal is insufficient, it can be rebuilt with considerably less examples than needed by the Nyquist theorem. The notion of CS has been officially introduced in 2006. It is a technique with possible uses in a variety of domains, including image processing [3], [4], in which CS is rummage-sale to rebuild images from a limited amount of data. It container also be used for mapping, spectral imagination [5], and virtual opening radar scanning. In all real-world scenarios, one essential challenge that C must address is the ability to reclaim the underlying signals being detected in an effective manner. Numerous techniques have been developed for reconstructing signals from CS samples. There are other techniques, including MP [7] and Simultaneous OMP (SOMP) [8] [9]. The MP algorithm continuously searches for the dimension medium row that appears greatest important for the present signal estimating before performing a simple computation to continually modify the predicted signal. The OMP technique uses a least-squares fitting phase for pattern estimate, making it more precise than MP. The lowest-mean phase in OMP lowers the sum of repetitions while increasing the level of difficulty of each cycle. The SOMP uses several observation matrices based on OMP, however it considerably raises complication as well as costs. Because of its balance among cost and precision, the OMP process is an excellent candidate for hardware-based application to achieve actual time signal rebuilding. The difficulty of each successive repetition of the OMP method stems primarily from the huge amount of inner product and transposal matrix processes. The scope of the matrix that measures and the level of sparseness also have a significant impact on the OMP's general complexity. This algorithm's bottleneck is it's the lowest-squares result. The classic normalization matrices solution approach requires inverting the matrix immediately, which is exceedingly difficult and unsuitable for the use of hardware. The most common lowest-

squares solutions rely on matrices breakdown, involving Cholesky factorization, Singular Value Decomposition (SVD) and QR decay. The SVD is an extremely precise distillation method, but it is complex and difficult to implement in hardware. The majority of them Least Square (LS) method is founded on QR breakdown and Cholesky factorization, which optimizes the effectiveness of calculation of matrices reversal. yet remain significant obstacles regarding the way the computer hardware is applied. This paper offerings an innovative method to calculating the OMP, carried out with HLF, that needs fewer mathematical results in smaller rebuilding times than preceding HLF applications, and produces outcomes similar to conventional FPGA deployments using HDLs.

2. LITERATURE SURVEY

The research has recently suggested numerous hardware design options for OMP implementations. There are two main categories: classic FPGA development using HDLs like Verilog and VHDL, and HLS development. Instead of implementing a design architectural with HDL, HLS uses a detailed functional C/C++ overview of a scheme to build an HDL layout explanation instantly. HLS represents a new style paradigm. HLS provides a benefit of working at a greater level of conceptualization, allowing design investigation and modeling for multiple systems easier. Then initially offer HDL versions with an emphasis on FPGAs. Septimus et al. [10] demonstrated single of the primary computer hardware renditions of the OMP method. They, like us, use Gram-Schmidt orthogonalization to improve hardware installation efficiency. They showed that this technique outperforms CPUs and GPUs in a particular arrangement, but not much data is provided regarding the architecture, such as how it represents numbers and its impact on accuracy. We conducted a more thorough examination of a bigger setup and area for design offered by an HLS methodology. Bai et al. [11] created a fixed-point version of OMP on an FPGA using QR decay and a specific spectrum of acceptable waveform sparseness. Blache et al. [12] proposed a system in which the matrix invert is calculated using the Coordinate Rotation Digital Computation (CORDIC) technique. Many operations, including matrix-vector addition, can be repeated to improve together area and implementation time. Ren et al. [13] presented a FPGA system that uses adjustable computation components to distribute computational resources between various OMP workloads. Rabah et al. [14] improved OMP implementation time by optimizing matrix inversion utilizing the Newton-Raphson loop and leveraging computer hardware for matrix vector additions. The work in [15] is an FPGA application that uses a method known as Matrix Inversion Bypass (MIB) to separate the calculations of interim indication estimations and diagonal transposals, reducing computing efficiency. Over the past few years, there have emerged several HLS-based CS implementations. A Xilinx Vivado HLS application for OMP by means of QR deconstruction is provided in [16], which resolves the mean squares issue while avoiding squared-root computations. Kerdjidj et al. [17] devised a computer hardware design that compresses and recovers electrocardiogram (ECG) signals. Korrai et al. [18] used the QR decomposed to create a least-squares fitting technique for estimating mmWave indoor scattered channels. Knoop et al. [19] developed a quick the fields of digital design process for OMP using HLS. Kim et al. [20] presented an HLS-based FPGA architecture for the OMP method that is enhanced by a unique separated inverting method. The linked research stated previously all aim to simplify the difficult the least-squares issue related to OMP. However, since the LSs problem has to be performed in each repetition, the amount of computing effort is substantial. In this study, we present a successful OMP version on FPGA for CS recovery. The main point to make advance is that, unlike previous studies, we incorporate Gram-Schmidt perpendicularity [21] into the indication remnant inform, allowing us to remove the mean squares resolution in each recurrence. The mean squares solution only requires to be run once during the last iteration. In the final phase, this paper create the mean squares procedure by means of the Cholesky decomposition.

3. DESCRIPTION OF THE OMP THE TECHNIQUE AND ITS DIFFICULTY ANALYSIS

Assuming a signal is limited on a certain the foundation, CS assumes that only a limited number of nonadaptive linear measurement techniques may record its information. An n-sparse signal vector contains no less than m non-zero scalar elements. A sign vector $\mathbf{x} \in R^N$ developed via rectilinear capacities is specified by equation (1).

$$y = \phi x + n \quad (1)$$

Where $\phi \in R^{KN}$ is a rectangular sample background demonstrating the sample scheme, $y \in R^K$ is the dimension trajectory, and n represents *K*-point path that signifies the dimension sound. The supports of medium ϕ signified $\phi 1, \phi 2, \ldots \phi n$ specifies *K*-point trajectories *(K < N),* also named atoms. The measurement of sizes trajectory y is in over-all expected to be significantly lesser than the distance of gesture trajectory x.

4. DESCRIPTION OF OMP ALGORITHM

The first method represents the OMP method proposed in [22]. The function receipts the dimension medium ϕ and the acquired trajectory y as inputs and returns a rough approximation $\tilde{x}$ of the actual signal x. This is a continuous algorithm. In each iteration, it selects one of the rows of ϕ that has the greatest correlation with the remaining of observations y. It then eliminates the impact of that row in order to calculate a novel enduring. It also determines an additional approximation of the initial indication; after m rounds, the method returns the last estimation of the unique signal.

The task of optimization in step 2 is handled by computing the Pearson relationship matrix w as trails in equation (2).

$$w = \phi^T r_i - 1 \quad (2)$$

where $r_i - 1$ remains the outstanding trajectory of $(i - 1)^{th}$ repetition. The directory λ_i of the constituent of **w** consuming thoroughgoing complete rate is recognized, and the consistent support is removed from ϕ to establish average $\tilde{\phi}$ of such removed columns. Rendering to step 4, an approximation of the rebuilt signal $\tilde{x_i}$ is gotten by resolution the subsequent equation (3).

$$y = \tilde{\phi}\, \tilde{x} \quad (3)$$

Where $\tilde{\phi}$ represent *(K × m)* quadrangular matrix with *K > m*. A mutual method to upset such quadrilateral matrix is to usage the Moore–Penrose pseudo-inverse, stated by equation (4).

$$\tilde{\phi} \dagger = (\tilde{\phi}^T \tilde{\phi})^{-1} \tilde{\phi}^T \quad (4)$$

Where $\tilde{\phi}$† denotes inverse of Moore–Penrose. Consequently, the explanation of (3) is gotten by resolution the subsequent equation (5).

$$w = C\tilde{x} \quad (5)$$

Where **C** = $\tilde{\phi}^T \tilde{\phi}$ is a symmetrical matrix ∈ R $^{m \times m.}$ Equation (5) canister be resolved by medium overturn or by advancing/retrograde replacement [12].

A matrix's inversion can be found using a variety of approaches, including Cholesky factorization, LU, and QR decomposition. For matrix tilting, we employed the modified Cholesky factorization approach described in [18], which requires no square-root processes. The extended Cholesky factorization enables the C matrix to be represented as a combination of three matrix structures, as shown in equation (6).

$$C = LDL^T \quad (6)$$

The lowermost trapezoidal medium L and diagonally medium D are constructed by means of the relationships shown in equations (7a) and (7b).

$$L_{i.j} = \frac{1}{D_{j,j}} \left\{ C_{i,j} - \sum_{k=1}^{j-1} (L_{i,k} L_{j,k} D_{k,k}) \right\}, i > j \quad (7a)$$

$$D_{i,i} = C_{i,i} - \sum_{k=1}^{i-1} (L_{i,k}^2\, D_{k,k}) \quad (7b)$$

The reverse of medium C is calculated as trails in equation (8).

$$C^{-1} = (L^{-1})^T D^{-1} L^{-1} \quad (8)$$

The inverse transformation of matrix D is achieved by inverting each of its slanting mechanisms, though the reverse transformation of medium L is carried out continuously using the relative in equation (9).

$$L_{i,j}^{-1} = -\sum_{k=i}^{i-1} L_{i,k} L_{k,j}^{-1} \quad (9)$$

For *i > j*. In step 6, the remainder of the vector r is modified for the following cycle by means of the relationship in equation (10)

$$r_i = y - \tilde{\phi}\, \tilde{x} \quad (10)$$

To make use of the backward/forward replacement strategy, matrix C must be decomposed into three matrix structures, L, D, and LT, as illustrated in (6). Resolution (5) is equivalent to completing 3 equations: Lz = w using advancing replacement, Dr = z, and L^Tx = r using retrograde conversation, wherever z and r represents transitional trajectories. Matrix C is symmetrical, and increasing in size from 1 × 1 to m × m with each repetition. A modest examination demonstration that the scheming of L and D receipts m(m+1)/2 sequences form repetitions of L^{-1} and D^{-1} can be achieved in equivalent and takings the similar quantity of series. As determination be confirmed later in this research work, the scheming of C^{-1} will yield (m + 1) m/2 and that of C^{-1}w can yield 1, m, or m^2, dependent on the gradation of correspondence of the accepted building. In this case, through m series, this paper finds a total amount of series equivalent to (m + 2) m for environment downturn method. In the second approach, the scheming of L and D, z, and x determination yield (m + 1) m/2 series respectively, which parallels to an entire of 3(m + 1) m/2. As a result, we elected to design the primary technique, which has additional opportunities for concurrency. Furthermore, we used this method to compare to comparable proposed designs [10], [11].

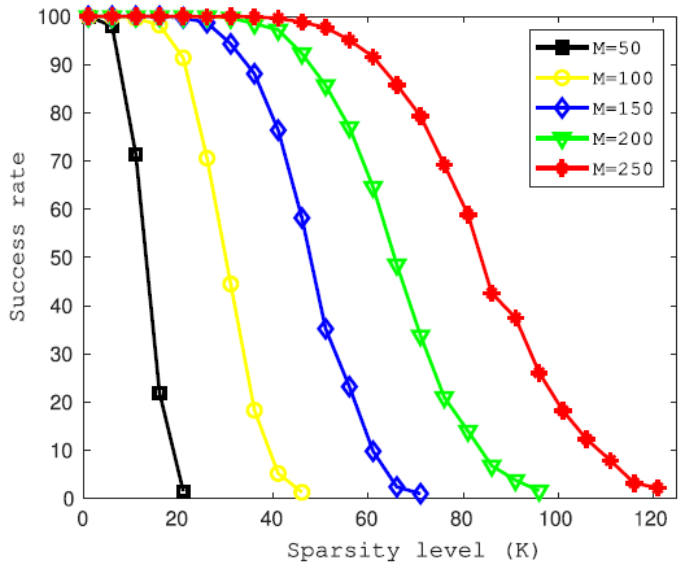
5. DIFFICULTY EXAMINATION OF OMP PROCESS

The OMP technique is reiterative and has six sorted purposes. The six variables are computed consecutively and performed with each iteration. Functions 1 and 2 compute the coefficient vector (2), compare it to determine the row of ϕ with the greatest correlation, then modify the matrix $\tilde{\phi}$ by reinforcing the obtained supports to produce matrix. In functions 3 and 4, medium C can be constructed from matrix $\tilde{\phi}$, and its converse is determined using the Cholesky extraction approach described in (7) - (9). The approximation $\tilde{x}$ of the rebuilt signal x is obtained using (5), and the remainder vector has been modified for the following repetition.
We assessed the level of complexity of the purposes used in the OMP process. The total amount of multipliers, improvements, contrasts, negations, and divides, respectively, $7m^3/6 + (K + 1)m^2 + (NK + K - 2)m$, $4m^3/6 + (K - 1)m^2 + (NK - N - 2/3)m$, $(N - 1)m$, $m(m - 1)/2$, and m. As $N > K \gg m$, Functions 1, 3, and 6 contribute significantly to the general complexity of the OMP algorithm.

The fundamental issue with expediting the OMP method is that because of its repetitive its very nature, it does not support simultaneous execution. Furthermore, the mathematical stream of the six purposes in every repetition of the method depends on data. As a result, calculating six functions at the same time is impossible.
Yet certain tasks can be deliberately parallelized to accelerate processing. We can conduct N IPC in parallel and quickly compare IP standards to discover the most associated column of ϕ. This will necessitate enormous possessions. As a result, we created an automated IPC unit that executes one IPC and then compares it in individually sequence. The purposes 1, 3, and 6 share the similar IP unit, resulting in substantial efficiency of resources. According to present concepts, Cholesky decomposition needs an extremely large amount of clock cycles while doing less work compared to other OMP procedure functions. This is mostly due to the splitting process and the complicated data interdependence of the mediums L and D. In this paper, we provide an improved equivalent framework for inverted matrices based on Cholesky reduction. The section that follows describes these structures in depth, as well as the proposed design for the OMP method.

6. SIMULATION RESULTS

The following section demonstrates the effectiveness our OMP technique is for signal restoration. While it reduces computing difficulty, it maintains the algorithm's correctness. Simulation studies are carried out using MATLAB. Our OMP algorithm relies heavily on the values of K and M. In Figure 1, we plot the achievement degree of the enhanced OMP as an indicator of sparseness, with N set to 256 and M varying. It demonstrates that for a specific amount M, the success rate declines as sparseness grows. For an established sparseness, further measurements yield a better result. Figure 2 depicts the fraction of signals accurately rebuilt as a function of observation number under different sparsity's. It demonstrates that when sparsity rises, additional measurements are required to keep pattern reconstruction efficient.



176-191.

FIGURE 1. The rate of success as a purpose of sparseness, for N = 256 and variable M, the amount of observations.

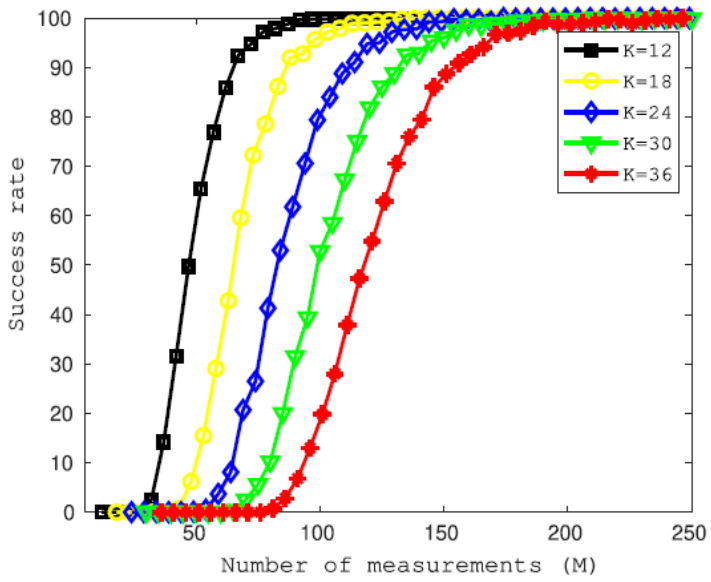
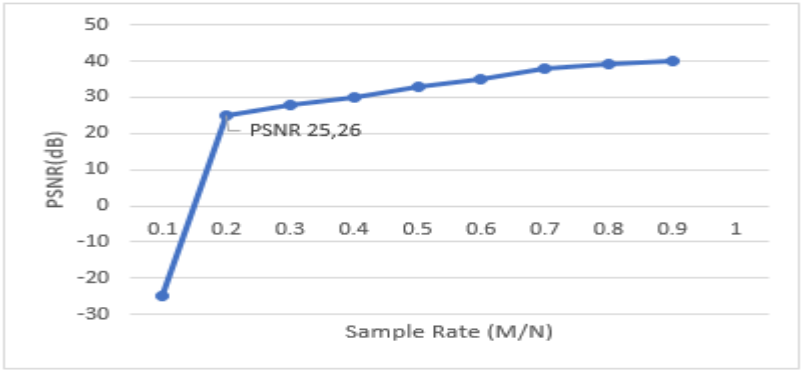


FIGURE 2. Rate of success as a purpose of observation amount, for N=256 and variable K, the sparseness level

OMP improves practical signal processing in applications other than signal rebuilding, such as image, acoustic and ECG. Figure 3 depicts the restoration of a 256 × 256 picture "Lena" and its PSNR efficiency at various sample rates. It can be expressed as the number of measurements (M) divided by the signal length (N). Image reconstruction performance improves as the sampling rate increases. As can be seen, reconstruction of images occurs when M is too tiny. When the collection rate is 0.2, an inflection points forms, indicating that the quantity of observations meets the Restricted Isometry Property (RIP). When the amount of capacities$M > K * \log(\frac{N}{K})/\delta^2$, OMP can accurately recover unlimited signals, where is an isometry variable

(a) (b)

FIGURE 3.



(c)

FIGURE 4. Image rebuilding findings (a) Original image. (b) Reconstructed image. (c) PSNR at different sample rates

7. RESULT AND ANALYSIS

We conducted our trials using the Xilinx Vivado HLS tool version 2018.3. All of the analysis findings presented in this paper are for a Xilinx Zynq UltraScale+ MPSoC working at 113 MHz. For maximum accuracy, all calculations use single-precision floating point numbers procedures. To describe our OMP algorithm, we typically refer to as: Table 1 shows three categories: small, medium, and big. The parameters relate to small- to massive amounts signal restoration for a variety of uses, including medical signal analysis, mmWave channels estimates and infrared signal processing. As a result, this architecture is suitable for a wide range of CS restoration applications.

TABLE 1. Parameters Settings

Sets	M x N	K	Application example
Small	32 x 128	5/8	e-health monitoring
Medium	256 x 256	100	ECG signal Analysis
Large	256 x 1024	12/36	AIC

The efficiency of our approach is assessed in relation to known versions of the OMP algorithm (Table 2). Vivado HLS is used by all of the tiny implementations in Table 2, with the exception of Septimus's [10] and Polat. Rabah [14] employed MATLAB-Simulink[25]. Except for our OMP in the small and huge groups, every other of our implementations use HDL.

TABLE 2: Comparison of proposed OMP results with existing works

Implementation	**M x N**	**K**	**FPGA Type**	**Frequency**	**Computation Precision**	**Block RAM's**	**DSP**	**FF**	**LUT**	**Reconstruction time**
Septimus [10]	32 x 128 (small)	5	Virtex-5	39MHz	Fixed point	-	-	-	-	$24\mu s$
Knoop[19]			Virtex-7	100MHz	Fixed point Q4.14	4	130	13.7 k	22.3 k	$16.9\mu s$
Polat [28]			Virtex-5	115MHz	Fixed point 18-bit	42	43	8.2k	14.9 k	$6.26\ \mu s$
Proposed OMP		5	Zynq Ultrascale	113 MHz	Floating point 32- bit	68	254	55.7 k	61.1 k	$13.2\ \mu s$
Proposed OMP		8	Zynq Ultrascale	113 MHz	Floating point 32-bit	70	254	55.1 k	61.3 k	$21\ \mu s$
Stanislaus [27]	64 x 256 (Medium)	8	Virtex-5	85MHz	-	-	-	-	-	$27.1\ \mu s$
Polat [28]			Virtex-5	110MHz	Fixed point 18-bit	80	81	13.5 k	22.6 k	$23.2\ \mu s$
Polat [28]		12	Virtex-5	101MHz	Fixed point 18-bit	88	89	24.7 k	40.1 k	$39.56\ \mu s$
Proposed OMP		8	Zynq Ultrascale	113MHz	Floating point 32-bit	132	446	150k	114k	$20.6\ \mu s$
Proposed OMP		12	Zynq Ultrascale	113MHz	Floating point 32-bit	134	446	151k	116k	$33.54\ \mu s$
Bai[11]	256 x 1024 (Large)	12	Virtex-6	100MHz	Fixed point 18-and 42- bit	-	-	-	-	$158.7\ \mu s$
Bai [11]		36	Virtex-6	100MHz	Fixed point 18-and 42- bit	258	261	-	96k	$630\ \mu s$
Rabah[14]			Virtex-6	120MHz	Fixed point 18-bit	576	589	-	18k	$340\ \mu s$
Proposed OMP		12	Zynq Ultrascale	113MHz	Floating point 32-bit	518	1.7k	635k	453k	$150.3\ \mu s$
Proposed OMP		36	Zynq Ultrascale	113MHz	Floating point 32-bit	521	1.7k	596k	440k	$423\ \mu s$

This may explanation for a considerable portion of the outcome alteration. In Row 4, we compared with Polat [28] and find that the process of rebuilding time is shorter than ours, probably since Polat uses fixed point instead of variable point, which has substantially greater potential processes. The findings also reveal that proposed OMP has the shortest rebuilding period among all of the compact strategies [24]. This applies level when some other designs utilize fixed-point integers while the proposed approach uses values that are floating point, as well as after adjusting for clock rate changes. Nevertheless, compared Rows 2 and 6 reveals a large energy expense when matching propsed OMP to Knoop [19]. Several of this can be attributable to our usage of decimal point, while the remainder is owing to the method, requiring meaningfully more block RAM. Block RAMs are mostly employed for storing the arrays of vectors and indices utilized in the method, such as y, r, and A. To facilitate cooperation, vectors and multipliers become available in block RAMs using cyclical matrix splitting. The single-block RAMs are also set up for simultaneously availability, which allows for speedier computations by reading many subsequent scalar and matrix entries in concurrently. For intermediate assessments, there is Stanislaus [27] and Polat [26] in A series 8-10 with respect to proposed OMP. Our reconstructive time is 24% faster than Stanislaus', however both times are equal when the clock frequency are adjusted for normality. Stanislaus fails to offer anything about reserve utilization. For Polat, we can see that proposed OMP is approximately 11.2% and 15.2% improved than Polat with K = 8 and K = 12, although it takes much more hardware. In the subsequent section, this paper talk about hardware size while considering huge layouts.

8. CONCLUSION

This work introduces an enhanced OMP technique for CS reconstruction. In proposed OMP approach, Gram-Schmidt symmetry is used to modify the remaining rather than running the mean squares solution at each repetition. Due to the enormous amount of inner products, the original OMP technique has a higher computing efficiency for the mean squares issue. As a result, the improvements presented in this paper can significantly lower the computationally challenging nature of matrix computations. We tested proposed OMP algorithm with MATLAB. The simulation consequences demonstration that proposed OMP technique minimizes computational burden while maintaining responsiveness. To address the requirements for applications that operate in real time, proposed OMP's hardware architecture has been applied on an FPGA with the Xilinx Vivado HLS design package [23]. The suggested architecture utilizes HLS and careful parallelism design to enable various measuring matrix capacities and sparsity. The execution consequences show that proposed OMP algorithm has a faster rebuilding time associated to other current HLS-based works and compares in instantaneous performance compared to conventional HDL-based methods for huge signal input sizes, but at a significant rate in terms of hardware expenses due to the manner in which proposed HLS tool presently operates. Unlike most prior research, our design works for signals of arbitrary size and sparsity, rather than predefined characteristics. However, proposed HLS technique causes substantial equipment expansion as M and K rise, therefore it is best suited for modest devices from an economic standpoint. We suspect that increased equipment sharing might significantly decrease hardware utilization in large as well as medium designs, however the existing HLS implementation has no capacity to do so.

REFERENCES

[1]. S. H. W. Oppenheim, A. S. Alan, and A. V. Nawab, Signals and Systems. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[2]. D. L. Donoho, “Compressed sensing,” IEEE Trans. Inf. Theory, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[3]. R. G. Baraniuk, T. Goldstein, A. C. Sankaranarayanan, C. Studer, A. Veeraraghavan, and M. B. Wakin, “Compressive video sensing: Algorithms, architectures, and applications,” IEEE Signal Process. Mag., vol. 34, no. 1, pp. 52–66, Jan. 2017.

[4]. J. Li, M. Song, and Y. Peng, “Infrared and visible image fusion based on robust principal component analysis and compressed sensing,” Infr. Phys. Technol., vol. 89, pp. 129–139, Mar. 2018.

[5]. K. Gunasheela and H. Prasantha, “Compressive sensing approach to satellite hyperspectral image compression,” in Information and Communication Technology for Intelligent Systems. New York, NY, USA: Springer, 2019, pp. 495–503.

[6]. J. Yang, T. Jin, C. Xiao, and X. Huang, “Compressed sensing radar imaging: Fundamentals, challenges, and advances,” Sensors, vol. 19, no. 14, p. 3100, Jul. 2019.

[7]. S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” IEEE Trans. Signal Process., vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[8]. J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” IEEE Trans. Inf. Theory, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[9]. J. A. Tropp, A. C. Gilbert, and M. J. Strauss, “Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit,” Signal Process., vol. 86, no. 3, pp. 572–588, Mar. 2006.

[10]. A. Septimus and R. Steinberg, “Compressive sampling hardware reconstruction,” in Proc. IEEE Int. Symp. Circuits

Syst., May 2010, pp. 3316–3319.
[11]. L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, "High-speed compressed sensing reconstruction on FPGA using OMP and AMP," in Proc. 19th IEEE Int. Conf. Electron., Circuits, Syst. (ICECS), Dec. 2012, pp. 53–56.
[12]. P. Blache, H. Rabah, and A. Amira, "High level prototyping and FPGA implementation of the orthogonal matching pursuit algorithm," in Proc. 11th Int. Conf. Inf. Sci., Signal Process. Their Appl. (ISSPA), Jul. 2012, pp. 1336–1340.
[13]. F. Ren, R. Dorrace, W. Xu, and D. Markovic, "A single-precision compressive sensing signal reconstruction engine on FPGAs," in Proc. 23rd Int. Conf. Field Program. Log. Appl., Sep. 2013, pp. 1–4.
[14]. H. Rabah, A. Amira, B. K. Mohanty, S. Almaadeed, and P. K. Meher, "FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 10, pp. 2209–2220, Oct. 2015.
[15]. G. Huang and L. Wang, "An FPGA-based architecture for high-speed compressed signal reconstruction," ACM Trans. Embed Comput. Syst., vol. 16, no. 3, p. 89, May 2017.
[16]. [16] Z. Yu et al., "Fast compressive sensing reconstruction algorithm on FPGA using orthogonal matching pursuit," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2016, pp. 249–252.
[17]. [17] O. Kerdjidj, N. Ramzan, A. Amira, K. Ghanem, and F. Chouireb, "Design and evaluation of vivado HLS-based compressive sensing for ECG signal analysis," in Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput., 16th Int. Conf. Pervas. Intell. Comput., 4th Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech), Aug. 2018, pp. 457–461.
[18]. P. K. Korrai, K. Deergha Rao, and C. Gangadhar, "FPGA implementation of OFDM-based mmWave indoor sparse channel estimation using OMP," Circuits, Syst., Signal Process., vol. 37, no. 5, pp. 2194–2205, May 2018.
[19]. B. Knoop, J. Rust, S. Schmale, D. Peters-Drolshagen, and S. Paul, "Rapid digital architecture design of orthogonal matching pursuit," in Proc. 24th Eur. Signal Process. Conf. (EUSIPCO), Aug. 2016, pp. 1857–1861.
[20]. S. Kim et al., "Reduced computational complexity orthogonal matching pursuit using a novel partitioned inversion technique for compressive sensing," Electronics, vol. 7, no. 9, p. 206, Sep. 2018.
[21]. D. Clayton, "Algorithm AS 46: Gram-schmidt orthogonalization," J. Roy. Stat. Soc. C (Appl. Statist.), vol. 20, no. 3, pp. 335–338, 1971.
[22]. J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Inf. Theory, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
[23]. D. O'Loughlin, A. Coffey, F. Callaly, D. Lyons, and F. Morgan, "Xilinx vivado high level synthesis: Case studies," in Proc. 25th IET Irish Signals Syst. Conf. China-Ireland Int. Conf. Inf. Commun. Technol. (ISSC/CIICT), Limerick, Republic of Ireland, 2014, pp. 352–356, doi: 10.1049/cp.2014.0713.
[24]. S. Liu, N. Lyu, and H. Wang, "The implementation of the improved OMP for AIC reconstruction based on parallel index selection," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 2, pp. 319–328, Feb. 2018.
[25]. J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, "High-level synthesis for FPGAs: From prototyping to deployment," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 4, pp. 473–491, Apr. 2011.
[26]. Ö. Polat and S. K. Kayhan, "High-speed FPGA implementation of orthogonal matching pursuit for compressive sensing signal reconstruction," Comput. Electr. Eng., vol. 71, pp. 173–190, Oct. 2018.
[27]. J. L. V. M. Stanislaus and T. Mohsenin, "High performance compressive sensing reconstruction hardware with QRD process," in Proc. IEEE Int. Symp. Circuits Syst., May 2012, pp. 29–32.
[28]. Ö. Polat and S. K. Kayhan, "High-speed FPGA implementation of orthogonal matching pursuit for compressive sensing signal reconstruc tion," Comput. Electr. Eng., vol. 71, pp. 173–190, Oct. 2018.

