Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 1887 – 1907
© TÜBİTAK
doi:10.3906/elk-2010-108

FPGA implementation of LSD-OMP for real-time ECG signal reconstruction

Önder POLAT*, **Sema Koç KAYHAN**
Electrical and Electronics Engineering Department, Faculty of Engineering, Gaziantep University, Gaziantep, Turkey

Received: 26.10.2020 • **Accepted/Published Online:** 24.11.2020 • **Final Version:** 26.07.2021

Abstract: Compressed sensing is widely used to compress electrocardiogram (ECG) signals, but the major challenges of the compressed sensing algorithms are their highly complex signal reconstruction processes. In this paper, a reconfigurable high-speed and low-power field-programmable gate array (FPGA) implementation of the least support denoising-orthogonal matching pursuit (LSD-OMP) algorithm for the real-time reconstruction of the ECG signals is presented. The contribution of this study is two-fold: Firstly, LSD-OMP can pick more than one element at each iteration and reconstruct the sparse signal using less number of iterations as compared to the standard OMP algorithms. Latency of the proposed design is therefore reduced by exploiting the multiple index selection feature of LSD-OMP. Secondly, the proposed architecture is the first reconfigurable LSD-OMP reconstruction architecture which can take different signal sizes and different sparsity levels. The proposed design also includes an efficient inverse wavelet transform (IWT) module to convert the reconstructed signal back into the time-domain. Together with the overhead of the IWT module, the proposed design demonstrates faster execution times while consuming lower power than the existing FPGA implementations; therefore, it can be utilized in wireless body area networks as a back-end unit to reconstruct the compressed ECG signals.

Key words: Field-programmable gate array, electrocardiogram, real-time, compressed sensing

1. Introduction

Cardiovascular diseases (CVD) are the leading cause of death globally according to the World Health Organization. Patients with CVDs require extensive healthcare resources both at home and ambulant conditions. Therefore, CVDs increase the medical costs associated with monitoring, diagnosis, and treatment. Electrocardiogram (ECG) recordings are used to monitor and assess the electrical activity generated by the heart and to diagnose CVDs both manually and automatically. Since ECG monitoring is important to provide medical assistance for patients with CVDs, developing low cost, low-power, and real-time systems is important.

Wireless body area networks (WBAN) make it possible to record the ECG signal of a patient by using the biomedical sensors attached to their body. Recent advances in wireless communication and mobile devices provide an opportunity for ambulatory monitoring using WBANs. Ambulatory ECG monitoring enables the medical professionals to inspect ECG recordings of patients during their routine activity. Compared to the standard ECG, the ambulatory ECG offers precise detection of cardiac anomalies as a result of long recording periods of 24–48 h [1]. In such networks, recorded signals can be transferred to a host system in a real-time fashion to provide ambulatory monitoring for the patient. The compression of the ECG signals is therefore required for efficient storage of the large amount of ECG data and for the effective transmission over a

*Correspondence: onderp@gantep.edu.tr

communication channel. One such channel, the wireless transmission channel, is quite important for WBANs and demands efficient signal compression. Recently, compressed sensing (CS) [2] is applied to compression of the ECG signals [3, 4] by using smaller number of measurements for sampling. Using CS, it is possible to reconstruct a signal by using fewer samples than needed by the Nyquist–Shannon sampling theorem. The only condition is that the signal must show sparsity in some domain. Since it is possible to sparsely represent the ECG signals in the wavelet domain, compression utilizing the CS theory is a suitable method for ECG monitoring. In this way, compressed ECG data can be transferred over the wireless channel and reconstructed at the receiver side. In addition, clinical relevance of the ECG signals that are reconstructed using CS is investigated for detection and monitoring purposes and promising results have been reported [5, 6].

Reconstruction of a compressively sampled signal is a complex process. Algorithms are iterative and involve a lot of matrix operations (multiplication, inversion etc.). To build an effective CS reconstruction system, hardware implementations of such algorithms have been proposed in the literature.

The orthogonal matching pursuit (OMP) algorithm was implemented on a field-programmable gate array (FPGA) device in [7–11]. These studies basically differ in terms of the technique used to design the architecture (manual hardware description language (HDL) or high-level synthesis (HLS)) and the algorithm (QR decomposition (QRD) or Cholesky decomposition (CD)) used to obtain the least square (LS) solution. Design with HLS tools is performed with a programming language where the behavioral description of the architecture is given and the HLS tools automatically generate an HDL description. Compared to the manual HDL technique, it is easier and less time-consuming but the performance and the efficiency of the resultant hardware is inferior to the manual HDL design depending on the expertise of the designer. The first hardware implementation of the OMP algorithm was presented in [7]. The authors in [7] reported that their architechture performed better than the CPU counterpart. However, the design was not reconfigurable and the results were given for just one fixed configuration where the signal length was 128 for a sparsity of 5. The designs presented in [8, 9] are two similar architectures where the work in [9] sacrifices the signal quality to lower the complexity of the hardware. Both designs support the sparsity level of up to 8 and the signal length of 256 and utilize QRD for the LS solution. Another hardware architecture that implements the OMP and the approximate message passing algorithms was proposed in [10]. Their architecture can reconstruct signal with sparsity of 36 with a signal length of 1024 using QRD for the LS solution. Different from the other works, the authors in [11] proposed implementation of the OMP algorithm using the HLS technique and utilized CD for the LS solution. Compared to the other studies discussed here, the design in [11] achieves better timing and latency results. It is a reconfigurable architechture and supports sparsity levels of up to 36 and the signal length of up to 1024. The architectures presented in [7–9] are implemented on Xilinx Virtex-5 FPGA and those in [10, 11] are implemented on Xilinx Virtex-6 FPGA. Application-specific integrated circuit (ASIC) implementation of a modified algorithm, projection-based atom selection (POMP) was presented in [12]. It was implemented on 90 nm CMOS and uses QR decomposition for the LS solution.

Apart from the work presented in [12], existing implementations use the base OMP algorithm where only one element of the estimated signal is determined and none of the implementations includes an inverse wavelet transform (IWT) module. Different from the existing studies, the proposed work is tailored for the real-time ECG signal reconstruction application targeting WBAN. It contains an IWT module for fast and efficient domain transform for the ECG signals and it is the first reconfigurable FPGA implementation of the least support denoising-orthogonal matching pursuit (LSD-OMP) algorithm where multiple index selection is utilized with the novel sorting module for low power and fast signal reconstruction. The synthesis results in Section 4.3 demonstrate that the proposed work achieves better synthesis results than the discussed existing work with good reconstruction quality that is analyzed in Section 4.2.

The OMP algorithm needs at most K iterations to reconstruct a sparse signal, with K being the sparsity of the signal. The LSD-OMP algorithm [13, 14], on the other hand, can select more than one atom at each iteration without knowledge of the sparsity value and it can reconstruct the sparse signal in less than K iterations. This significantly reduces the number of iterations and the latency of the reconstruction and provides faster execution time for real-time applications. In this paper, a reconfigurable hardware implementation of the LSD-OMP algorithm, with an integrated IWT unit, has been presented for the first time. The proposed architecture is reconfigurable, that is, it can support different sparsity values (K), input signal sizes and sampling matrix sizes. The size of the input signal and the sampling matrix determine the number of the multipliers, RAM blocks, adders and layers in the adder tree and the number of the many more structural parameters. The proposed work is designed as scalable for all these parameters by using manual HDL coding. Thus, it does not require any HLS tool for reconfiguration. In addition to that, the proposed system contains an integrated IWT unit where the reconstructed signal is converted into time-domain without any additional software involved outside the hardware device.

The rest of the paper is organized as follows. Section 2 describes the LSD-OMP algorithm. The proposed hardware architecture for the LSD-OMP algorithm is presented in Section 3. Performance evaluation and implementation results are described in Section 4. The paper is concluded in Section 5.

2. LSD-OMP

A sparse signal $\mathbf{x}$ is sampled using a random matrix $\mathbf{\Phi}$ to obtain the measurement vector $\mathbf{y}$:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} \tag{1}$$

If the dimension of $\mathbf{x}$ is N and the dimension of $\mathbf{y}$ is M, $\mathbf{x}$ can be reconstructed using a $M \times N$ random matrix $\mathbf{\Phi}$ provided that $M \ll N$ and $\mathbf{x}$ is sparse in the domain indicated by the random sampling matrix $\mathbf{\Phi}$. Various algorithms are proposed in the literature to solve the system given in Equation 1. One of the widely used such algorithms is the greedy OMP [15] algorithm. The basic procedure in OMP is to find the best correlated column of the sampling matrix with the measurement vector and remove its contribution from the residual at each iteration. OMP ensures that the residual is orthogonal to the estimated signal $\widetilde{\mathbf{x}}$ to prevent selecting the same column for the following iterations.

Since the speed of the base OMP algorithm is directly dependent on the sparsity level of the signal; a lot of extensions to OMP are proposed to improve the reconstruction speed while maintaining complexity and the signal quality. LSD-OMP is one of such extensions. It reconstructs a sparse signal in L iterations by finding L indices which are maximally correlated with the residual, in each iteration. Since L is smaller than the sparsity value, LSD-OMP finishes its operation faster than the base OMP algorithm. The value L is chosen as the minimum of $K/2$ and $M/16$. Pseudocode of the LSD-OMP algorithm is given in Algorithm 1.

3. Architecture

Compared to the base OMP implementations, the main challenge for the LSD-OMP implementation is to support multiple index selection at each stage of the algorithm. These stages are interdependent on each other and involve a lot of matrix-vector multiplications and additions. The dot product between the transpose of

Algorithm 1 Least support denoising-OMP

```
Input:   $\mathbf{y}$: $M$ dimensional measurement vector.
         $\mathbf{\Phi}$: $M \times N$ dimensional sampling matrix.
         $K$: Sparsity of the signal to be reconstructed.
         $L$: Least support parameter $L$
Output:  $\widetilde{\mathbf{x}}$: $N$ dimensional estimated signal.
1: $\mathbf{r}_0 \leftarrow \mathbf{y}$,  $\mathbf{I_0} \leftarrow \emptyset$  $\mathbf{J_0} \leftarrow \emptyset$
2: for $t = 1$ to $L$ or stop condition do
3:     $J_t \leftarrow \arg \max_{t=1 \dots L} |\phi_t^T, \mathbf{r}_{t-1}|$,  $I_t \leftarrow [I_{t-1} \cup J_t]$
4:     $\widetilde{\mathbf{x}} = \arg\min \|\mathbf{y} - \widetilde{\mathbf{\Phi}}_{\mathbf{I_t}} \mathbf{x}\|$
5:     $\mathbf{r}_t \leftarrow \mathbf{y} - \widetilde{\mathbf{\Phi}}_{\mathbf{I_l}} \widetilde{\mathbf{x}}$
6: end for
7: return $\widetilde{\mathbf{x}}$.
```

the sampling matrix and the residual vector accounts for the most of the clock cycles whereas the least squares (LS) unit is responsible for the increased hardware complexity due to irregular memory access required for the matrix inversion. In addition to that, the LSD-OMP demands a sorting circuit to find the L most correlated columns at each iteration.

In this section, implementation details of the proposed design are presented. The top level diagram for the proposed design is shown in Figure 1. Both the dot product stage and the LS unit were designed to support variable number of columns for selection per iteration. Control was accomplished by finite state machines (FSM) working in cooperation with each other. Fixed point notation was used throughout the architecture to reduce the hardware complexity.

3.1. Sorting circuit and index selection

A vector multiplier is needed for every stage of the algorithm to compute multiplications between various vectors and matrices. The dot product between Φ^T and r in step 3 of Algorithm 1 requires M parallel multipliers to perform a vector-vector multiplication per clock cycle. For this task, parallel DSP slices with reconfigurable number were utilized as the vector multiplier unit and they were controlled by a main FSM to perform desired operations at each step of the algorithm. DSP slices are sophisticated logic circuits found in FPGAs that provide efficient implementation of signal processing functions such as multiplication and addition.

Multiplication results from each of these multipliers were forwarded to an adder tree implemented by instantiating 4-input individual adders. The size of the adder tree depends on the parameter M which is also the number of multipliers utilized. Each output from M multipliers must be summed up to obtain the result of dot product. Adder tree is implemented by connecting small individual 4-input adders in a tree-like structure where the outputs of one layer are the inputs for the next layer. For $M = 32$, there are 8 individual adders in the first layer. The outputs from these 8 adders are connected to 2 individual adders in the second layer and these 2 adders are connected to a single individual adder in the last layer. Thus, for $M = 32$, there are 3 layers for a total of 11 individual adders. Since each individual adder completes its operation in 2 clock cycles, addition for a single column completed in p clock cycles where p is calculated by multiplying the number of layers with 2. Since the dot product between the residual and every column of the matrix Φ^T has to be calculated, the adder tree completes its operation in $N + p + 2$ clock cycles at each iteration of the LSD-OMP algorithm for N columns. Note that the proposed design is pipelined thus at each clock cycle new data (multiplication results)

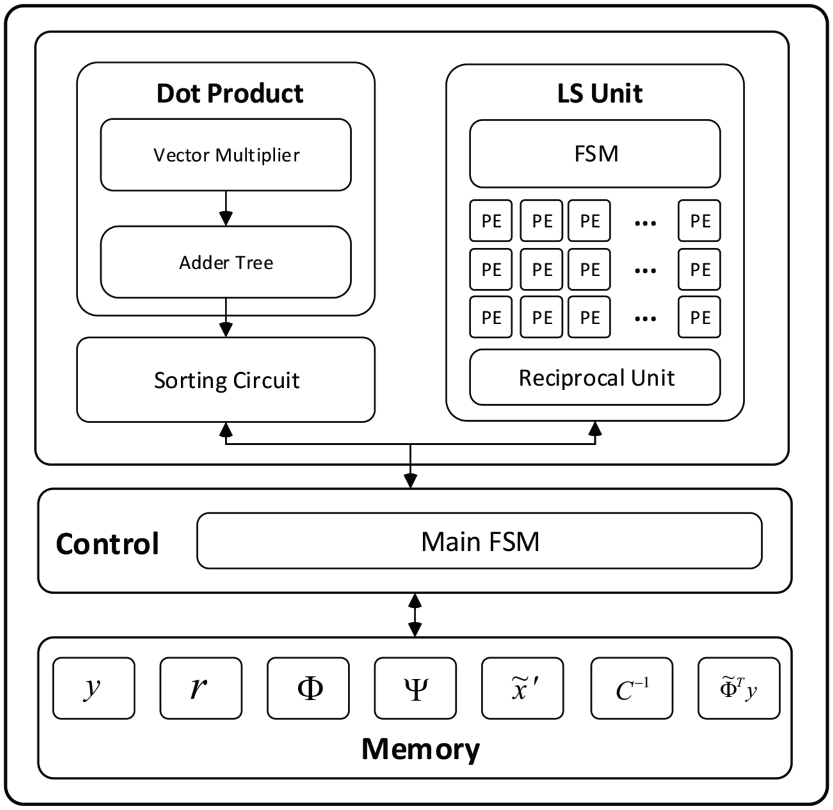


Figure 1. Top level diagram of the proposed design.

are forwarded to the adder tree without waiting for the previously inserted data to reach the final register. This greatly decreases the latency of the dot product from $N * p$ clock cycles to $N + p + 2$ clock cycles as described above.

The dot product results are forwarded to the sorter circuit in a pipelined fashion. L maximum correlated columns must be identified and sorted before sending them to the LS unit. The sorting circuit was derived from the work presented in [16]. It consists of L individual cells each holding a dot product value and the corresponding index value. Basic operation of the sorter is to perform a comparison between the input data from the adder tree with the data stored in the sorting cell. The sum value obtained from the adder tree for each column is presented to all of the sorting cells.

If the forwarded data is greater than (or equal to) the current data but smaller than the data stored in the previous cell, this new data is claimed by the corresponding cell and the current data is shifted to the next cell. In the case that the new data is greater than (or equal to) both the current cell data and the data stored in the previous cell, then the corresponding cell automatically shifts its data to next cell and claims the data shifted from the previous cell (since the previous cell will also shift its data). If the new data is smaller, the cell performs no operation. Initially all cells are cleared to zero so they will accept any data that is forwarded. Since the algorithm involves the absolute value of the dot product, there is no possibility for a negative value. First cell is configured to accept new data only if it is greater than its current data (no other check is performed). An example is shown in Figure 2 for a six-cell sorting circuit.

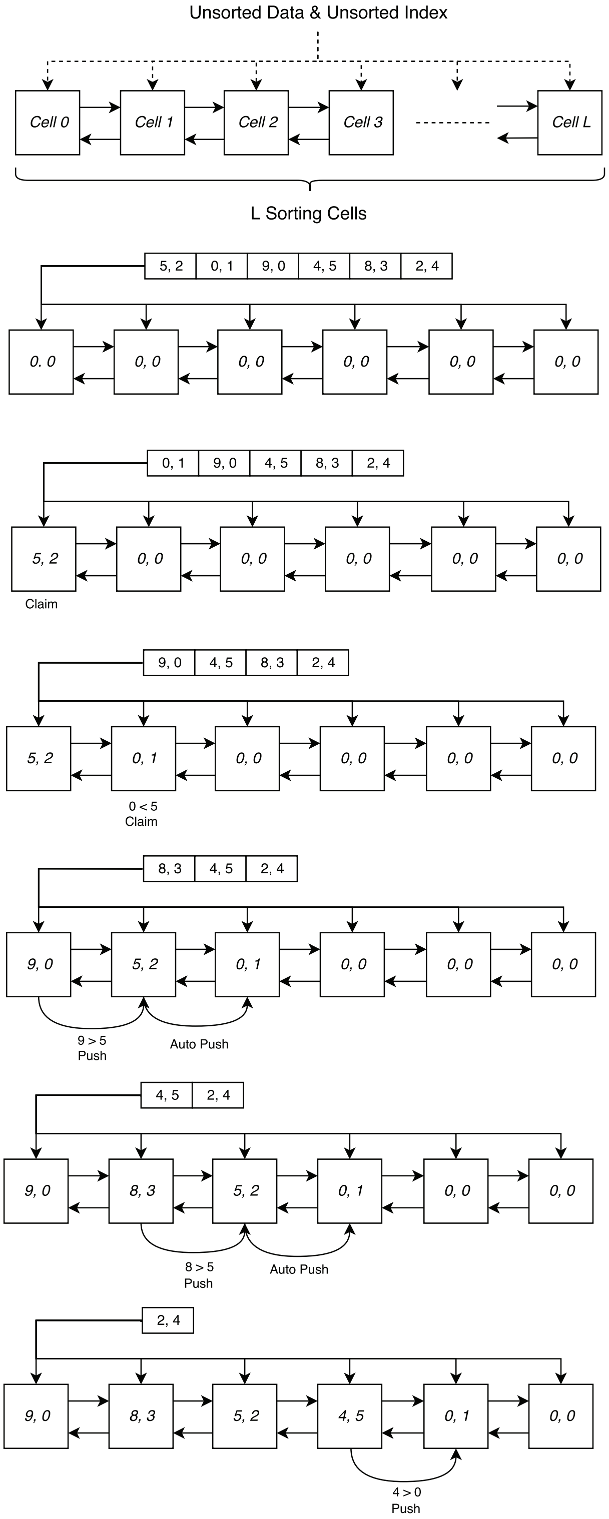


Figure 2. Sorting circuit example.

After the dot product phase is completed, index value of L most correlated columns that are identified in the sorting circuit is shifted out one by one starting from the left most cell (the largest dot product value). These index values are used to construct the augmented matrix $\widetilde{\mathbf{\Phi}}$ which is then used in the LS unit.

3.2. Least squares solution

LS estimation is used to compute the estimated signal $\widetilde{\mathbf{x}}$ from the augmented matrix $\widetilde{\mathbf{\Phi}}$ such that:

$$\mathbf{y} = \widetilde{\mathbf{\Phi}}\widetilde{\mathbf{x}} \tag{2}$$

The system in Equation 2 can be solved by computing the pseudo inverse of $\widetilde{\mathbf{\Phi}}$.

$$\widetilde{\mathbf{\Phi}}^{\dagger} = (\widetilde{\mathbf{\Phi}}^T\widetilde{\mathbf{\Phi}})^{-1}\widetilde{\mathbf{\Phi}}^T \tag{3}$$

Multiplying both sides with the pseudo inverse, the following equation is obtained:

$$(\widetilde{\mathbf{\Phi}}^T\widetilde{\mathbf{\Phi}})^{-1}\widetilde{\mathbf{\Phi}}^T\mathbf{y} = \widetilde{\mathbf{x}} \tag{4}$$

Letting $\mathbf{C} = \widetilde{\mathbf{\Phi}}^T\widetilde{\mathbf{\Phi}}$, inverse of the matrix $\mathbf{C}$ can be found by using the CD method. The matrix $\mathbf{C}$ can be factorized into two matrices; an upper triangular matrix $\mathbf{L}$ and a diagonal matrix $\mathbf{D}$.

$$\mathbf{C} = \mathbf{L}\mathbf{D}\mathbf{L}^T \tag{5}$$

The matrices $\mathbf{L}$ and $\mathbf{D}$ are computed using the following equations:

$$L_{ij} = \frac{1}{D_{jj}}(C_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}D_{kk}),\ for\ i > j \tag{6a}$$

$$D_{ii} = C_{ii} - \sum_{k=1}^{i-1} L_{ik}^2 D_{kk} \tag{6b}$$

The estimated signal is then computed by using the CD of the matrix $\mathbf{C}$ to obtain $\mathbf{C}^{-1} = (\mathbf{L}^{-1})^T\mathbf{D}^{-1}\mathbf{L}^{-1}$ and inserting it into Equation 4. The inverse of $\mathbf{D}$ is computed by taking its reciprocal whereas the inverse of $\mathbf{L}$ is computed by using the formula:

$$L_{ij}^{-1} = -\sum_{k=1}^{i-1} L_{ik}L_{kj}^{-1},\ for\ i \neq j \tag{7}$$

CD was implemented by instantiating a set of individual processing elements (PEs) to perform the necessary arithmetic operations. The matrix $\mathbf{C}$ is stored in K parallel RAM blocks each holding a separate column. At each iteration, L columns corresponding to the L most correlated indices are loaded to these RAM blocks for the LS operation. The PEs for the decomposition matrices $\mathbf{L}$ and $\mathbf{D}$ contain 2 multipliers while the PEs for the matrix $\mathbf{L}^{-1}$ uses 1 multiplier. These PEs compute the decomposition matrices according to Equations 6 and 7. The structure of the LS unit is scalable and reconfigurable to support varying level of sparsity.

The matrix $\mathbf{C^{-1}}$ is computed from $\mathbf{L^{-1}}$ and $\mathbf{D^{-1}}$ similarly by utilizing parallel PEs that use 2 multipliers. The LS unit is activated by the main FSM as soon as the matrix $\mathbf{C}$ is stored in the RAM. It can compute the inverse of $\mathbf{C}$ whose size can range from 1×1 to $L \times L$ depending on the configuration. The high level diagram of the LS unit is shown in Figure 3.

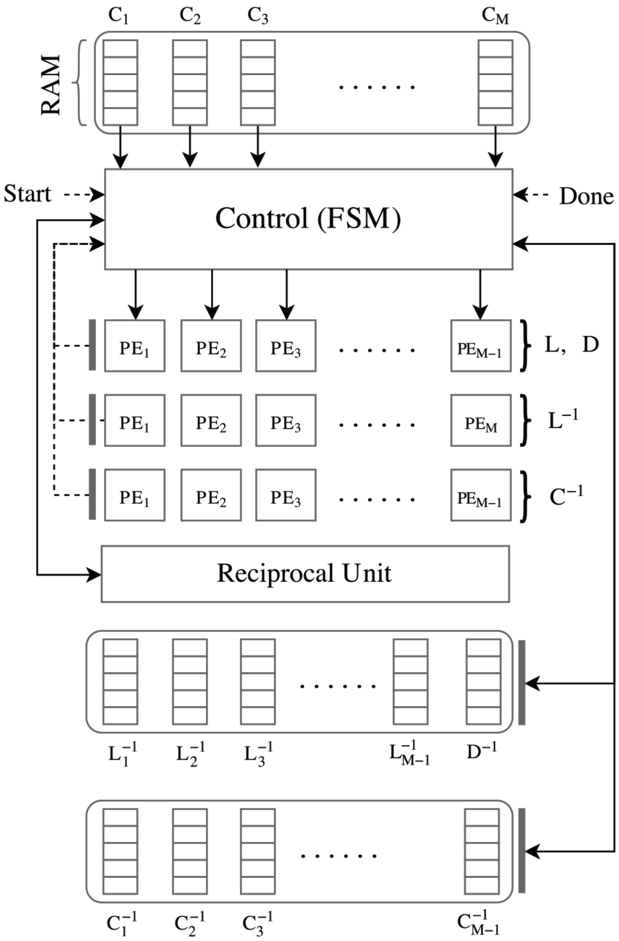


Figure 3. The LS unit.

The estimated signal $\widetilde{\mathbf{x}}$ is computed in three steps using Equation 4. Firstly, the dot product between the augmented matrix $\widetilde{\mathbf{\Phi}}$ and the measurement vector $\mathbf{y}$ is done using the vector multiplier. The obtained result is a column vector whose length is $L \times i$ where i is the iteration number. The second step is to multiply each row of $\mathbf{C^{-1}}$ with this column vector. Since $\mathbf{C^{-1}}$ is a symmetric matrix, only the lower triangular part is stored in memory to avoid repeated computation. Therefore, each column of $\mathbf{C^{-1}}$ is stored in a separate RAM block such that each one contains elements from the lower part in addition to the diagonal ones. The first element of every RAM block is multiplied with the first element of the column vector ($\widetilde{\mathbf{\Phi}}\mathbf{y}$) and added together to obtain the first element of $\widetilde{\mathbf{x}}$. Similarly, other elements of $\mathbf{C^{-1}}$ are multiplied with the corresponding part of the column vector and summed up to obtain other parts of the estimated signal. A third step is needed to compute the multiplication between the upper part of the matrix $\mathbf{C^{-1}}$ and the column vector. Thus, RAM blocks are accessed two times to complete the operation. Finally, the results of the matrix-vector multiplications are added together using an adder tree similar to the one used in index selection.

After the computation of $\widetilde{\mathbf{x}}$ is finished, the main FSM checks the current iteration number and, depending on the value, it either switches to residual computation or goes to inverse wavelet stage where the final ECG signal is computed. A flowchart is given in Figure 4 which shows the state transitions of the main FSM and operations performed at these stages.

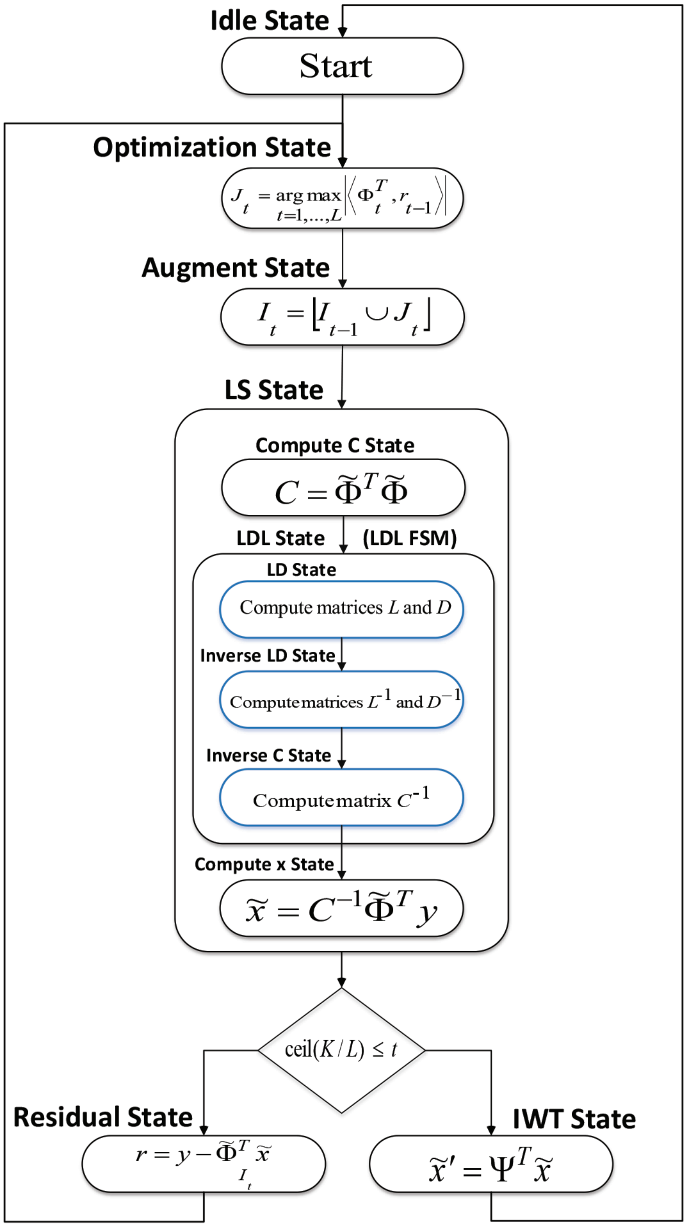

Idle State
Start
Optimization State
$J_t = \arg\max_{t=1,\dots,L} \left| \langle \Phi_t^T, r_{t-1} \rangle \right|$
Augment State
$I_t = \lfloor I_{t-1} \cup J_t \rfloor$
LS State
Compute C State
$C = \widetilde{\Phi}^T \widetilde{\Phi}$
LDL State
(LDL FSM)
LD State
Compute matrices L and D
Inverse LD State
Compute matrices L^{-1} and D^{-1}
Inverse C State
Compute matrix C^{-1}
Compute x State
$\widetilde{x} = C^{-1} \widetilde{\Phi}^T y$
$\text{ceil}(K/L) \le t$
Residual State
$r = y - \widetilde{\Phi}_{I_t}^T \widetilde{x}$
IWT State
$\widetilde{x}' = \Psi^T \widetilde{x}$

Figure 4. Flowchart of the main FSM.

3.3. Inverse wavelet transform

The last step of the ECG reconstruction is to convert the estimated signal in frequency domain into the time domain. The CS theory dictates that a signal $\widetilde{\mathbf{x}}$ that is sparse in a domain can be reconstructed using the randomly created sampling matrix $\mathbf{\Phi}$ (Equation (1)). The measurement vector $\mathbf{y}$ is created by multiplying a random matrix (for sampling) with a basis matrix (for sparse representation) with the sampled signal such that:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}\widetilde{\mathbf{x}}^{'} \tag{8}$$

$$\widetilde{\mathbf{x}} = \mathbf{\Psi}\widetilde{\mathbf{x}}^{'} \tag{9}$$

where $\mathbf{\Psi}$ is the basis matrix for wavelet transform. Here, the signal $\widetilde{\mathbf{x}}$ is obtained as output from the LSD-OMP algorithm. Similar to the matrix $\mathbf{\Phi}$, each row of the wavelet matrix $\mathbf{\Psi^T}$ has to be stored in a separate memory block. Discrete wavelet transform (DWT) is performed on the original ECG signal by using the orthogonal symlet wavelets. Since the wavelet matrix is orthogonal, its transpose can be used as the inverse wavelet matrix to obtain the time domain ECG signal $\widetilde{\mathbf{x}}^{'}$. Thus, IWT is performed by multiplying the inverse wavelet matrix with the estimated signal such that:

$$\widetilde{\mathbf{x}}^{'} = \mathbf{\Psi^T}\widetilde{\mathbf{x}} \tag{10}$$

Dimensions of the matrix $\mathbf{\Psi^T}$ is $N \times N$. This requires N^3 multiplications to complete the matrix multiplication in Equation 10. There are two limitations that arise in this process. The first limitation is that the number of parallel multipliers required for the operation exceeds the number of available multipliers (M). Secondly, at least N clock cycles are required to multiply every column of $\mathbf{\Psi}$ with the estimated signal, which will increase the latency. Rather than increasing the number of multipliers by utilizing more DSP slices, the first limitation was solved by storing more than one row of the matrix $\mathbf{\Psi^T}$ in each memory block. The number of rows to be stored in each block was determined by the compression ratio (CR):

$$CR = N/M \tag{11}$$

In this structure, by storing more than one row of the matrix $\mathbf{\Psi^T}$ in every memory block, each column of $\mathbf{\Psi^T}$ is divided into segments of M elements and each segment is multiplied separately with the corresponding element of the estimated signal and added together to obtain the ECG signal.

Since $K \ll N$, majority of the elements of the estimated signal is zero. Thus, only the nonzero elements are multiplied with the corresponding columns of $\mathbf{\Psi^T}$. This way, the latency is reduced to $CR \times K$ clock cycles instead of $CR \times N$. The computation of IWT of the estimated signal is visualized in Figure 5.

4. Performance evaluation and implementation results

In this section, the implementation results of the proposed LSD-OMP circuit are given together with the performance analysis for different metrics. Comparisons with the existing studies are also described with respect to the synthesis results of the proposed design. The circuit was designed and verified by using the Verilog HDL for Xilinx Virtex-5, Virtex-6, and Virtex-7 FPGA devices. The data format was determined as fixed point 18-bit due to the specifications of the DSP slices available in these devices and also for an efficient hardware implementation.

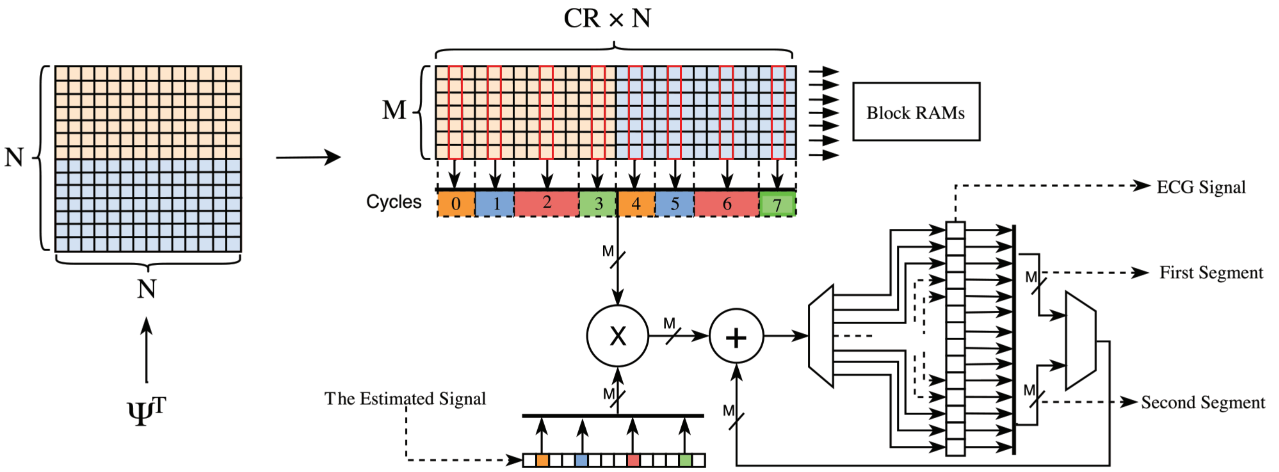


Figure 5. IWT Process

4.1. Timing analysis

The latency of the proposed design is dependent on the number of iterations performed (depending on the parameters) and the size of the sampling matrix $\mathbf{\Phi}$. The least support parameter L and the sparsity value K determines the number of iterations required to complete the algorithm. It was assumed that the algorithm will run for K/L iterations (rounded to nearest greater integer), picking L columns at each iteration to simplify the hardware implementation. Index selection is completed in $N+p+2$ clock cycles at each iteration where p is the delay introduced by the adder tree. The LS step takes $6i+29$ clock cycles for the ith index used to form the matrix $\mathbf{C}$. Total latency of the LS step, therefore, depends on the sparsity value K. Additional latency introduced by the sorting circuit and L is $5K$ clock cycles for $L > 1$. IWT takes $CR \times K$ clock cycles to reconstruct the ECG signal. The latency is therefore:

$$T = ceil(K/L) * (N+p+2) + K * (34+CR) + \sum_{i=1}^{K} 6i \tag{12}$$

Although higher L values indicate lower iteration count due to higher number of columns selected, they also increase the area utilization of the proposed circuit due to increase in the matrix size of $\mathbf{C}$ in the LS unit.

4.2. Reconstruction efficiency

The reconstruction performance of the proposed design was evaluated with fixed point simulations that were repeated for 100 times for different values of K and L. The ECG signal taken from the MIT-BIH Arrhythmia Database (MIT-BIH 117) [18, 19] was converted into frequency domain using the wavelet matrix $\mathbf{\Psi}$ and sampled with the random matrix $\mathbf{\Phi}$. Note that the original ECG signal can be acquired by using the matrix $\mathbf{\Theta}$ such that:

$$\mathbf{\Theta} = \mathbf{\Phi\Psi} \tag{13}$$

by providing the matrix $\mathbf{\Theta}$ as the sampling matrix. Thus, the ECG signal acquisition can be performed without separately multiplying the basis matrix $\mathbf{\Psi}$ for domain conversion. The resultant measurement vector y is passed into the LSD-OMP algorithm via the wireless channel for the signal reconstruction. In the proposed design, the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are readily available in the block memory. Therefore, signal reconstruction using the LSD-OMP algorithm is first performed by the proposed hardware to obtain the frequency domain signal $\widetilde{x}$ and then this signal is multiplied with the basis matrix $\mathbf{\Psi^T}$ as described in the Section 3.3 to obtain time domain ECG signal $\widetilde{\mathbf{x}}'$. The ECG sampling and reconstruction using the CS theory and the LSD-OMP algorithm is shown in Figure 6.

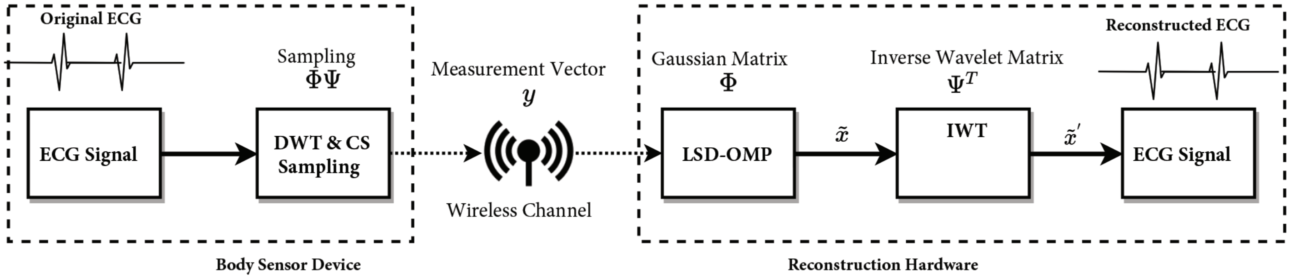


Figure 6. The ECG sampling and reconstruction process for WBAN application using CS and LSD-OMP.

In the literature, there are many different metrics to evaluate the signal quality and accuracy of the ECG signals. An extensive study was performed in [20] for the assessment of the ECG signals. After detailed testing of different metrics for ECG evaluation, the authors of [20] recommended some of them for the assessment of the compressed ECG signals. These metrics are divided into two main groups: subjective methods and objective methods. Subjective methods include an expert in the area such as a cardiologist to evaluate the signal accuracy with visual inspection. Objective methods are computed according to mathematical formulas and they are categorized into two subgroups: metrics with diagnostic information and metrics without diagnostic information. They also determined the thresholds for these metrics which correspond to the quality level of the evaluated ECG signal. Among the recommended metrics, three methods without diagnostic information and one method with diagnostic information were chosen for the assessment. Percentage root mean square difference normalized (PRDN1), signal to noise ratio (SNR1), and mean square error (MSE) are the objective methods without diagnostic information whereas wavelet-energy based diagnostic distortion using stationary wavelet transform (WEDD SWT) is the chosen objective method with diagnostic information. PRD and its variants are quite popular for the assessment of the ECG signals and widely used in the literature. The normalized version, PRDN, was used since it is recommended over the basic PRD method [20]. PRDN is computed as:

$$PRDN = \sqrt{\frac{\sum_{i=1}^{n}[\mathbf{x(i)} - \widetilde{\mathbf{x}}\mathbf{(i)}]^2}{\sum_{i=1}^{n}[\mathbf{x(i)} - \overline{\mathbf{x}}^2]}} \tag{14}$$

where $\mathbf{x}$ is the original signal, $\widetilde{\mathbf{x}}$ is the reconstructed signal, and $\overline{\mathbf{x}}$ is the mean of the original signal. SNR measures the error between the original and the reconstructed signal. According to [20], it is more accurate than PRD methods when compared with subjective methods where cardiologists or researchers evaluate the ECG signal visually and the mean score is calculated. MSE was chosen as another metric to evaluate the error between the original and reconstructed signal. SNR and MSE are given in Equations (15) and (16), respectively.

$$SNR = 10log_{10}(\frac{\sum_{i=1}^{n}[\mathbf{x(i)} - \overline{\mathbf{x}}]^2}{\sum_{i=1}^{n}[\mathbf{x(i)} - \widetilde{\mathbf{x}}\mathbf{(i)}]^2}) \tag{15}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[\mathbf{x(i)} - \widetilde{\mathbf{x}}\mathbf{(i)}]^2 \tag{16}$$

WEDD SWT is a wavelet-based technique where the ECG signal is divided into frequency bands and a weight score is computed for that band depending on its diagnostic quality. WEDD SWT is then calculated by multiplying the weight for each band (Equation (17)) with the PRD of that particular band (Equations (18) and (19)) [21]. WEDD SWT is sensitive to any distortion related to diagnostic quality of the ECG signals and SWT version is recommended for the ECG signal evaluation [20].

$$w_j = \frac{\sum_{i=1}^{n=j}|c_j(i)|}{\sum_{j=1}^{j=N_B}\sum_{i=1}^{n=j}|c_j(i)|} \tag{17}$$

$$WPRD_j = \sqrt{\frac{\sum_{i=1}^{n=j}(c_i - \widetilde{c}_i)^2}{\sum_{i=1}^{n=j}c_i^2}} \times 100 \tag{18}$$

$$WEDD\ SWT = \sum_{j=0}^{j=N_B} w_j WPRD_j \tag{19}$$

c_j and $\widetilde{c}_j$ in Equations (17)–(19) are the wavelet coefficients for the jth band of original and reconstructed signals respectively.

The simulation results are shown in Figures 7a–7c for varying values of L and K and for a CR of 4 (M=64, N=256). Figures 8a–8c present the same analysis for a CR of 2 with M = 128 and N = 256. In the study presented in [20], each algorithm has a recommended threshold value which was determined by two experts by visually evaluating the reconstructed ECG signals. Each figure is highlighted based on the recommended quality levels to better analyze the reconstruction efficiency. The green and yellow areas indicate very good and good quality levels, respectively. The blue area represents the acceptable quality and beyond that the signal quality is very bad that it is deemed unacceptable. Note that there are two different recommended levels by two experts in [20] and the values determined by expert 1, which is stricter than the other, were used in this paper. It was observed that the best performance is obtained for $L = 2$ though the difference decreases with the increased sparsity value. This quality difference is more apparent for higher CR of 4, compared to the CR value of 2. The reconstructed signal qualities reach ”good” level for all three metrics (PRDN, SNR, and MSE) with the increased sparsity level for CR=4 (Figures 7a–7c). As expected, for a CR value of 2 (Figures 8a–8c), the signal qualities reach ”very good” level for all three metrics since the CR value is decreased.

The same analysis was performed for varying values of CR and sparsity with N = 256 in Figures 9a–9c. All CR levels excluding CR = 4 reach ”very good” signal quality with the increased sparsity level. The WEDD SWT evaluation is shown in Figure 9d, which is a method sensitive to diagnostic information. The results of the WEDD SWT are consistent with the other metrics where the signal qualities reach ”very good” level with decreased CR and increased sparsity. Another observation was that the optimum value is obtained

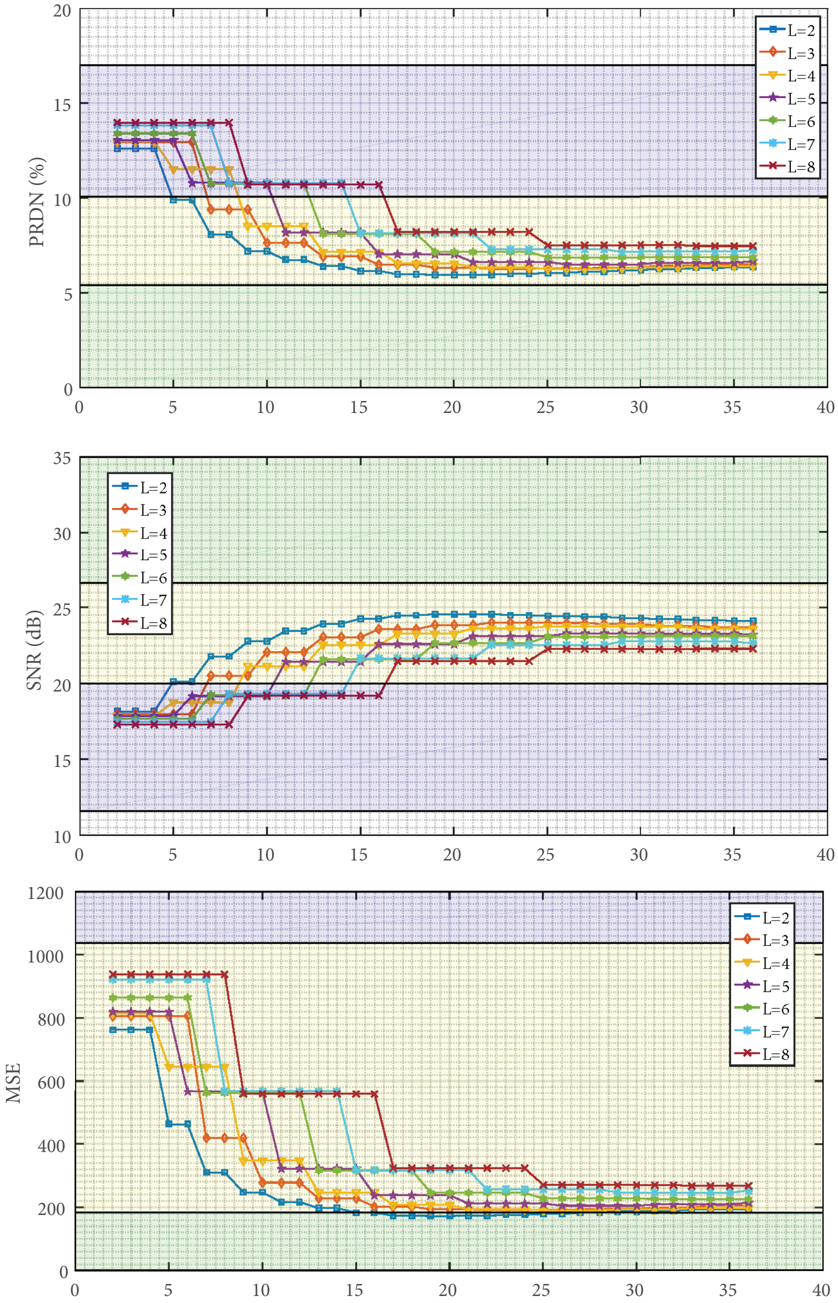


Figure 7. The signal quality versus sparsity for various values of the parameter L, CR = 4 (a) PRDN versus sparsity for M = 64, N = 256, CR = 4 (lower is better), (b) SNR versus sparsity for M = 64, N = 256 (CR = 4), (c) MSE versus sparsity for M = 64, N = 256, CR = 4 (lower is better).

with a sparsity level of 20–25 for all CR levels. Further increasing the sparsity level usually provides negligible performance gain in terms of the signal quality.

The LSD-OMP algorithm normally checks the signal quality before stopping the algorithm from further iterating. Without this check, the iteration count is fixed to $ceil(K/L)$ and the signal quality may decrease

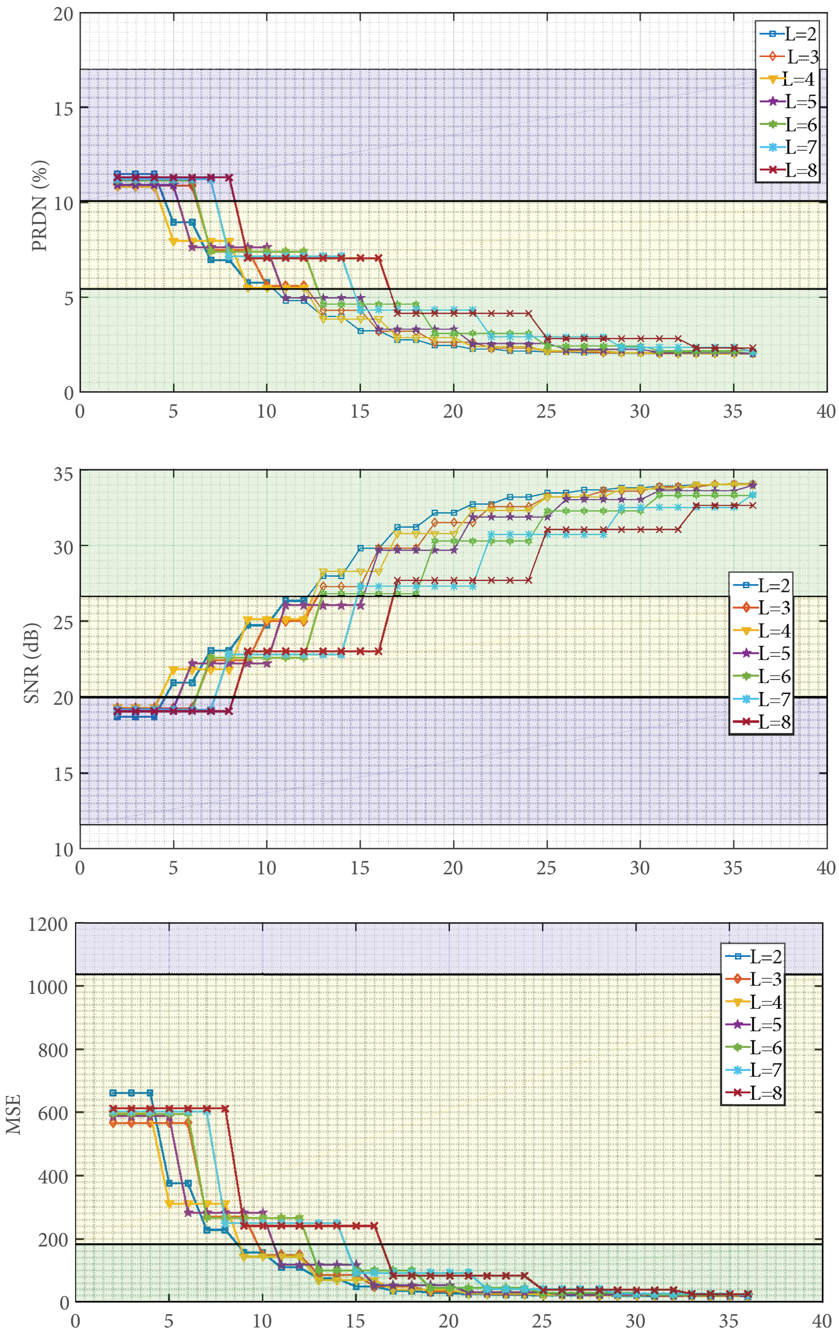


Figure 8. The signal quality versus sparsity for various values of the parameter L, CR = 2 (a) PRDN versus sparsity for M = 128, N = 256, CR = 2 (lower is better), (b) SNR versus sparsity for M = 128, N = 256, CR = 2, (c) MSE versus sparsity for M = 128, N = 256, CR = 2 (lower is better).

or converge to a value as shown in 7a–7c, 8a–8c. On the other hand, incorporating quality check causes the algorithm to stop at uncertain number of iterations. From a hardware viewpoint, it must be ensured that the proposed system supports the maximum number of possible indices that can be chosen. This greatly increases the resource usage and timing performance of the circuit due to increased complexity. In addition, slight decrease in the signal quality is negligible compared to performance gain in hardware. Because of that, the proposed design assumes $ceil(K/L)$ iterations and L indices at each iteration.

The same ECG signal was read into a memory array in the testbench file for the simulation of the proposed design. The output of the circuit was read into a text file from the testbench and plotted for comparison. In Figure 10a, the ECG signal was reconstructed using a 256×1024 sampling matrix and compared with the original signal for the sparsity value of 36 and least support parameter of 2. In Figure 10b, the same process was performed for a matrix size of 64×256. In this case, the ECG signal was divided into 4 segments and each segment was reconstructed separately. There is a trade-off that must be taken into account for both approaches. The first approach produces better signal quality at the cost of increased hardware complexity. While the second method produces lower quality signal, it is area-efficient and faster than the first one, and also convenient for parallel reconstruction of all the segments by using more than one core. The visual quality of the reconstructed signal also depends on CR where the increased CR value corresponds to the decreased signal quality. This is further demonstrated in Figure 10c in which the same ECG signal was reconstructed with CR = 2 and $N = 256$ and visually better fits the original signal compared to Figure 10b. Since the proposed design is reconfigurable, the synthesis results for different matrix sizes and sparsity levels are given and compared with the existing literature in the following section.

4.3. Synthesis results

The proposed design was synthesized for various problem sizes and sparsity levels to compare its performance with the existing implementations. The synthesis results obtained after the place and route stage are given in Table 1 for the least support parameter $L = 2$. Note that all the results shown in Table 1 for the proposed design include the overhead of the integrated IWT module. Synthesize options were configured to achieve the fastest reconstruction speeds. Most of the compared studies are implementations of the base OMP algorithm whereas the work presented in [12] uses POMP algorithm where it picks more than one index initially and later picks the most suitable index among those. For a fair comparison, the same sampling matrix size and sparsity value were used for the proposed design. Two results are presented in each case, one using the same FPGA device family for the comparison and the other using Virtex-7 FPGA to assess the performance of the proposed design on a resourceful and faster device. Due to reduced number of clock cycles required to complete the reconstruction operation, the proposed LSD-OMP implementation achieves better execution time than the existing studies for the corresponding problem size. Power consumption of the proposed design was also analyzed by using the Xilinx Power Analyser tool. As shown in Table 1, dynamic power consumption of the proposed design is lower than those of the existing FPGA-based designs.

Area utilization of the proposed design is presented in Table 2. The number of occupied slices, DSP slices, and block RAM usage are listed. Comparisons are made with the data reported for the existing studies. Although the slice utilization is slightly higher than those of the existing studies, the DSP and block RAM utilization is on comparable levels while also maintaining the high speed. Note that the RAM utilization also includes the extra wavelet matrix stored in the device, which is not included in the existing studies.

5. Conclusion

In this study, FPGA implementation of the LSD-OMP algorithm for real-time ECG reconstruction is presented. The architecture is reconfigurable and can be synthesized for different values of $M \times N$ (the size of the sampling matrix), K (the sparsity value), and L (the least support parameter). The IWT process was integrated into

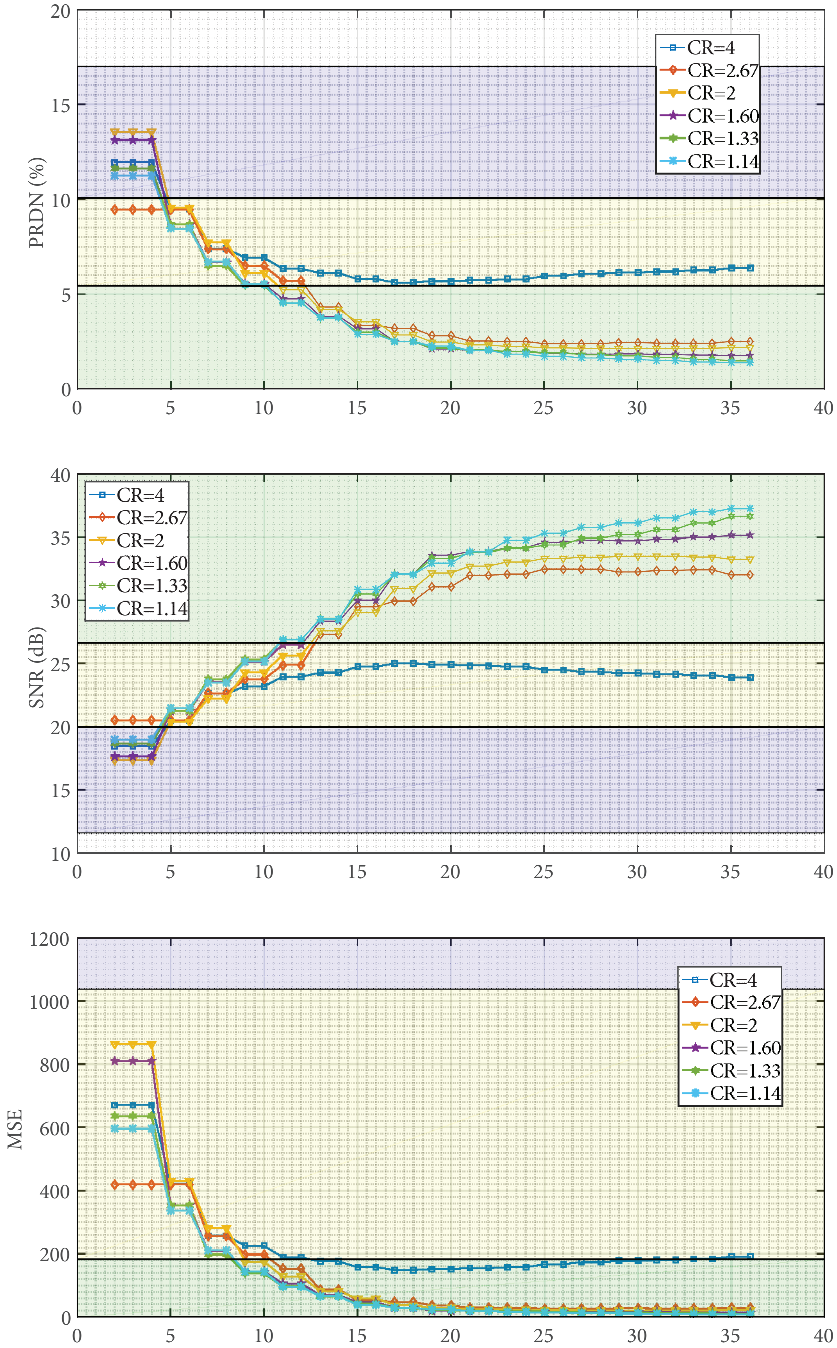
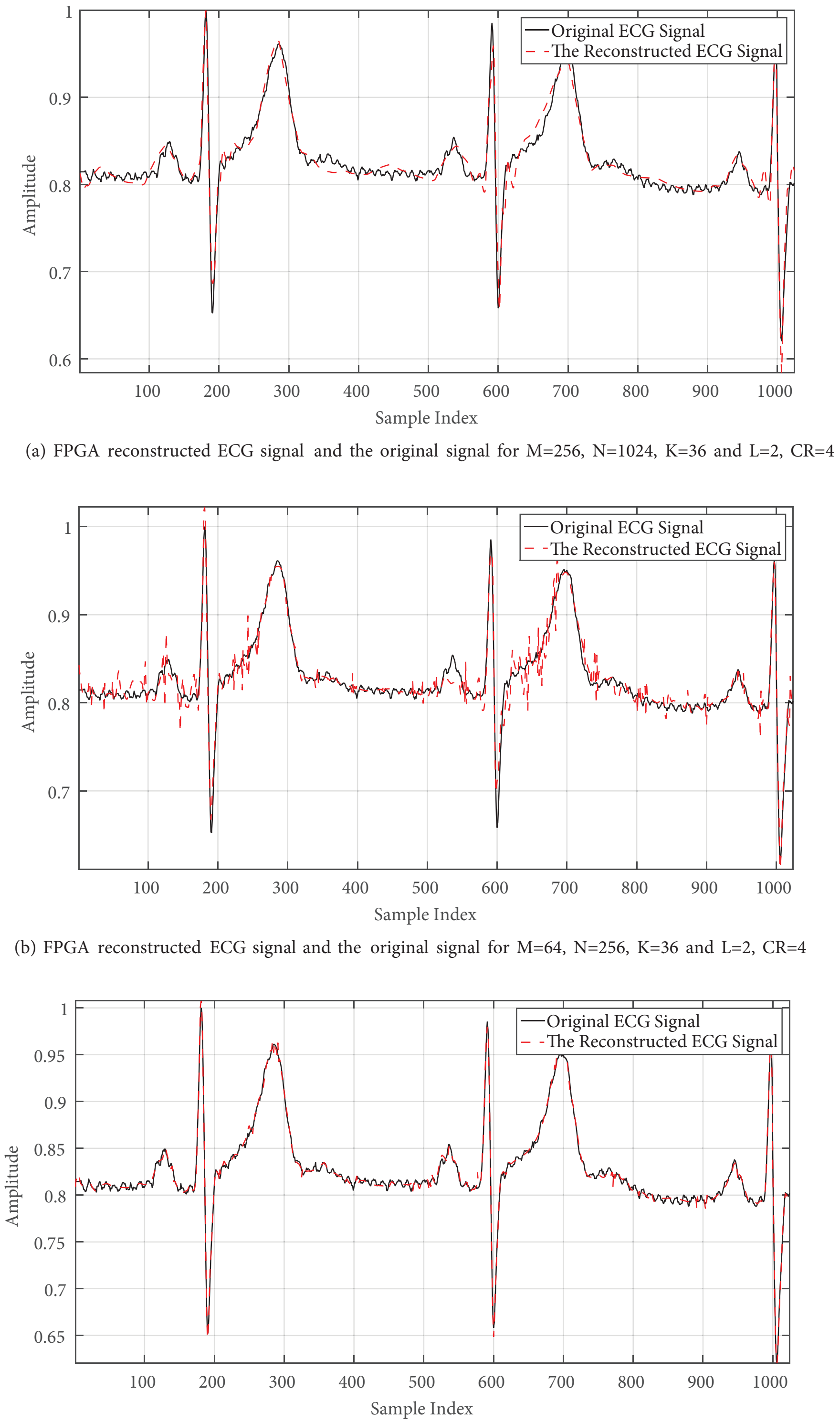


Figure 9. The signal quality versus sparsity for various CR values (a) PRDN versus sparsity N = 256 with varying CR (lower is better), (b) SNR versus sparsity N = 256 with varying CR, (c) MSE versus sparsity N = 256 with varying CR (lower is better), (d) WEDD SWT versus Sparsity N = 256 with varying CR (lower is better).

the proposed hardware in an efficient manner to provide full ECG reconstruction ability in one chip. The reconstruction efficiency is assessed with four recommended metrics (PRDN, SNR, MSE, WEDD SWT) for the varying values of L and CR. The signal quality of the reconstructed ECG signals easily reaches good



(a) FPGA reconstructed ECG signal and the original signal for M=256, N=1024, K=36 and L=2, CR=4

(b) FPGA reconstructed ECG signal and the original signal for M=64, N=256, K=36 and L=2, CR=4

(c) FPGA reconstructed ECG signal and the original signal for M=128, N=256, K=36 and L=2, CR=2

Figure 10. FPGA reconstructed ECG signal (a) CR = 4, N = 1024, (b) CR = 4, N = 256, (c) CR = 2, N = 256.

Table 1. Comparison of latency and power with the existing works (L = 2 for the proposed design).

Implementation	$\mathbf{\Phi}$ size	Sparsity	Technology	Clock cycles	Max. frequency	Processing time	Dynamic power (Watt)
FPGA [17]	32×128	5	Virtex-5	1680	107.33 MHz	15.65 μs	NA
FPGA [8]			Virtex-5	820	85/69 MHz	10 μs	NA
FPGA [10]			Virtex-5	NA	NA	16 μs	NA
Proposed			Virtex-5	688	121 MHz	5.69 μs	0.456
Proposed			Virtex-7	688	171 MHz	4.02 μs	0.505
FPGA [8]	64×256	8	Virtex-5	2260	85/69 MHz	27.14 μs	NA
Proposed			Virtex-5	1584	114 MHz	13.89 μs	0.737
Proposed			Virtex-7	1584	160 MHz	9.90 μs	0.867
ASIC [12]	64×256	12	90nm CMOS	8995	141 MHz	63.8 μs	0.0825†
Proposed			Virtex-5	2520	105 MHz	24 μs	0.799
Proposed			Virtex-7	2520	150 MHz	16.8 μs	0.895
FPGA [10]	256×1024	36	Virtex-6	63000	100 MHz	630 μs	NA
FPGA [11]			Virtex-6	40788	120 MHz	340 μs	3.233
Proposed			Virtex-6	23976	100 MHz	239.76 μs	2.107
Proposed			Virtex-7	23976	130 MHz	184.43 μs	1.986

†: Reported average power consumption of the ASIC design.

Table 2. Comparison of resource usage (L = 2 for the proposed design).

	$M = 32$, $N = 128$, $K = 5$			$M = 64$, $N = 256$, $K = 8$	$M = 256$, $N = 1024$, $K = 36$		
Implementation	[17]	[8]	Proposed	Proposed	[10]	[11]	Proposed
FPGA	XC5VSX50T	XC5VL110T	XC5VLX155T	XC5VLX155T	XC6VLX240T	XC6VLX240T	XC6VLX240T
Occupied slices	NA	NA	3407	4065	32010	6208	20191
DSP slices	134	51	63	105	516	589	437
BRAM/FIFO	42	NA	96	170	261	576	694

levels despite the CR values of as low as 4. Diagnostic distortion is also investigated with the WEDD SWT method and the results are consistent with the other three metrics. Combined with its reduced latency as a consequence of the multiple index selection and its reconfigurable features, the proposed design demonstrates faster reconstruction times and lower power consumption which makes it suitable for real-time ECG signal reconstruction tasks in CS applications where it can especially be utilized in WBAN applications as a back-end unit.

Acknowledgment

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), project grant no: 215E142.

References

[1] Boudoulas H, Schaal SF, Lewis RP, Robinson JL, Goodwin RS. Superiority of 24-hour outpatient monitoring over multi-stage exercise testing for the evaluation of syncope. Journal of Electrocardiology 1979; 12 (1): 103-108. doi:

10.1016/S0022-0736(79)80052-7

[2] Donoho D. Compressed sensing. IEEE Transactions on Information Theory 2006; 52 (4): 1289-1306. doi: 10.1109/TIT.2006.871582

[3] Craven D, McGinley B, Kilmartin L, Glavin M, Jones E. Compressed Sensing for Bioelectric Signals: A Review. IEEE Journal of Biomedical and Health Informatics 2015; 19 (2): 529-540. doi: 10.1109/JBHI.2014.2327194

[4] Mamaghanian H, Khaled N, Atienza D, Vandergheynst P. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. IEEE Transactions on Biomedical Engineering 2011. 58 (9): 2456-2466. doi: 10.1109/TBME.2011.2156795

[5] Craven D, McGinley B, Kilmartin L, Glavin M, Jones E. Impact of compressed sensing on clinically relevant metrics for ambulatory ECG monitoring. Electronics Letters 2015. 51 (4): 323-325. doi: 10.1049/el.2014.4188

[6] Da Poian G, Liu C, Bernardini R, Rinaldo R, Clifford GD. Atrial fibrillation detection on compressed sensed ECG. Physiological Measurement 2017; 38 (7): 1405-1425. doi: 10.1088/1361-6579/aa7652

[7] Septimus A, Steinberg R. Compressive sampling hardware reconstruction. In: IEEE International Symposium on Circuits and Systems; Paris, France; 2010. pp. 3316-3319. doi: 10.1109/ISCAS.2010.5537976

[8] Stanislaus J, Mohsenin T. High performance compressive sensing reconstruction hardware with QRD process. In: IEEE International Symposium on Circuits and Systems; Seoul, Korea; 2012. pp. 29-32. doi: 10.1109/ISCAS.2012.6271921

[9] Stanislaus J, Mohsenin T. Low-complexity FPGA implementation of compressive sensing reconstruction. In: IEEE International Symposium on Circuits and Systems; Beijing, China; 2013. pp. 671-675. doi: 10.1109/ICCNC.2013.6504167

[10] Bai L, Maechler P, Muehlberghuber M, Kaeslin H. High-speed compressed sensing reconstruction on FPGA using OMP and AMP. In: 19th IEEE International Conference on Electronics Circuits and Systems; Seville, Spain; 2012. pp. 53-56. doi: 10.1109/ICECS.2012.6463559

[11] Rabah H, Amira A, Mohanty BK, Almaadeed S, Meher PK. FPGA Implementation of orthogonal matching pursuit for compressive sensing reconstruction. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2015; 23 (10): 2209-2220. doi: 10.1109/TVLSI.2014.2358716

[12] Jhang JW, Huang YH. A high-SNR projection-based atom selection OMP processor for compressive sensing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2016; 24 (12): 3477-3488. doi: 10.1109/TVLSI.2016.2554401

[13] Tawfic I, Kayhan SK. Strong recovery conditions for least support orthogonal matching pursuit in noisy case. Electronics Letters 2015; 51 (17): 1368-1370. doi: 10.1049/el.2015.0222

[14] Tawfic I, Kayhan SK. Compressed sensing of ECG signal for wireless system with new fast iterative method. Computer Methods and Programs in Biomedicine 2015; 122 (3): 437-449. doi: 10.1016/j.cmpb.2015.09.010

[15] Tropp J, Gilbert A. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on Information Theory 2007; 53 (12): 4655-4666. doi: 10.1109/TIT.2007.909108

[16] Miranker G, Tang L, Wong CK. A zero-time VLSI sorter. IBM Journal of Research and Development 1983; 27 (2): 140-148. doi: 10.1147/rd.272.0140

[17] Blache P, Rabah H, Amira A. High level prototyping and FPGA implementation of the orthogonal matching pursuit algorithm. In: 11th International Conference on Information Science, Signal Processing and their Applications; Montreal, QC, Canada; 2012. pp. 1336-1340. doi: 10.1109/ISSPA.2012.6310501

[18] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC et al. PhysioBank, PhysioToolkit, and PhysioNet components of a new research resource for complex physiologic signals. Circulation 2000; 101 (23): e215-e220. doi: 10.1161/01.CIR.101.23.e215

[19] Moody G, Mark R. The impact of the MIT-BIH Arrhythmia Database. IEEE Engineering in Medicine and Biology Magazine 2001; 20 (3): 45-50. doi: 10.1109/51.932724

[20] Němcová A, Smíšek R, Maršánová L, Smital L, Vítek M. A comparative analysis of methods for evaluation of ECG signal quality after compression. BioMed Research International 2018; 2018: 1-26. doi: 10.1155/2018/1868519

[21] Al-Fahoum AS. Quality assessment of ECG compression techniques using a wavelet-based diagnostic measure. IEEE Transactions on Information Technology in Biomedicine 2006; 10 (1): 182-191. doi: 10.1109/TITB.2005.855554