Multi-Objective Design Optimization for Flexible Hybrid Electronics

Ganapati Bhat, Ujjwal Gupta, Nicholas Tran, Jaehyun Park, Sule Ozev, Umit Y. Ogras
School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ
{gmbhat, ujjwal, ntran, jpark244, sule.ozev, umit}@asu.edu

ABSTRACT

Flexible systems that can conform to any shape are desirable for wearable applications. Over the past decade, there have been tremendous advances in the domain of flexible electronics which enabled printing of devices, such as sensors on a flexible substrate. Despite these advances, pure flexible electronics systems are limited by poor performance and large feature sizes. Flexible hybrid electronics (FHE) is an emerging technology which addresses these issues by integrating high performance rigid integrated circuits and flexible devices. Yet, there are no system-level design flows and algorithms for the design of FHE systems. To this end, this paper presents a multi-objective design algorithm to implement a target application optimally using a library of rigid and flexible components. Our algorithm produces a set of Pareto frontiers that optimize the physical flexibility, energy per operation and area metrics. Simulation studies show a 32× range in area and 4× range in flexibility across the set of Pareto-optimal design points.

1. INTRODUCTION

Wearable devices are heralded as the new frontier in the consumer electronics market. Market growth is expected to exponentially increase in the next decade with sales reaching \$70B by the year 2025 [25]. Flexible electronics technology offers a great potential to develop light weight wearable systems that can conform to any shape, and be manufactured at a low cost. A variety of sensors, processing circuitry and analog to digital converters have been demonstrated using flexible electronics [1,26]. These devices can enable internet-of-things (IoT) nodes for medical applications, motion processing and gesture recognition [20].

Despite their great potential, flexible electronics devices are quite limited in terms of performance and computational capabilities [22, 28]. For instance, feature sizes of thin film transistors (TFT) are in the range of μm [10], and operating frequencies barely exceed $10MHz$ [13]. In contrast, CMOS technology offers transistors with nm feature sizes and faster than $1GHz$ switching frequency. Therefore, in the foreseeable future, it is unrealistic to design complex systems capable of sensing, signal conditioning, processing, and radio frequency (RF) communication using only flexible electronics.

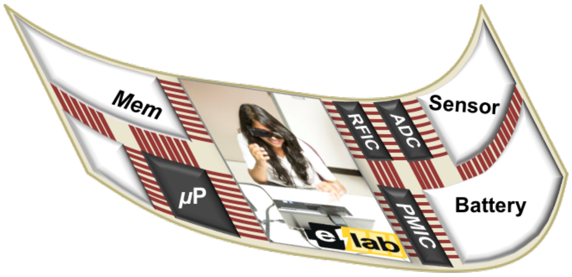

Figure 1: An illustration of flexible hybrid electronics system with flexible circuits, display, battery and rigid CMOS ICs.

Flexible *hybrid* electronics (FHE) is an emerging technology that integrates rigid high performance integrated circuits (ICs) with flexible devices onto a stretchable or flexible substrate [4, 27]. The FHE technology combines the small area and high performance advantages of rigid ICs, with the physical advantages of flexible electronics. Hence, it can bridge the gap between today's complex systems and flexible electronics by judiciously coalescing rigid and flexible resources. Despite the potential offered by the FHE technology, there are no known systematic techniques for designing a complete system capable of sensing, processing, and communicating. Filling this gap is one of the principle goals of this paper.

This paper presents a multi-objective design optimization algorithm for flexible hybrid systems. Designing FHE systems differs from traditional hardware design, since physical flexibility emerges as a new metric, which adds an uncharted dimension to the design space. This new dimension exhibits intricate design trade-offs with the traditional performance, power consumption and area metrics. The focal point of this optimization problem is in the partitioning of the target application between rigid and flexible resources. On the one hand, better energy efficiency and performance require a larger number of rigid ICs, since flexible electronics exhibit poor performance and scalability [11,22]. On the other hand, a large number of rigid ICs would undermine the advantages of flexibility. The optimum design point depends critically on the rigid and flexible resources that can be employed in the design, and the target application. Therefore, we store the resources available to the designer in a component library. This library specifies the component parameters, similar to the standard cell library used in technology mapping [14]. Likewise, the target application is modeled by a data flow graph, whose nodes describe the tasks that make

ICCAD '16, November 07-10, 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4466-1/16/11...\$15.00
DOI: http://dx.doi.org/10.1145/2966986.2967057

Table 1: The component library format (a few sample components from our library)

Component	Supported Tasks	Height (mm)	Width (mm)	Average Power (mW)	Max Throughput (MB/s)	Flexibility Index
Comp - 1	Temperature	4	4	2.31	0.000054	0
Comp - 2	Gyroscope Accelerometer	3	3	8 1.08	0.048 0.024	0
Comp - 3	CPU	19.6	17.2	0.92	0.32	1
Comp - 4	CPU Bluetooth LE (BLE)	10	10	2.5 0.98	160 0.12	1
Comp - 5	CPU BLE Zigbee	10	5	8.14 18.49 8.33	192 0.12 0.031	0

up the application. Since each task can be implemented by more than one component in the library, our algorithm finds the optimal subset of the components, and the mapping of the tasks into these components. The optimality of each feasible solution is evaluated in terms of flexibility, area, and energy per operation that reflects both the power consumption and performance. We evaluated this multi-objective optimization algorithm using four health monitoring applications and three realistic component libraries. Our simulation studies show 32× range in area and 4× range in flexibility across the set of Pareto-optimal design points.

Towards this end, our major contributions are as follows:

- A multi-objective optimization algorithm that implements a given target application using a library of rigid and flexible components,
- Introducing flexibility as a new dimension of a multi-objective optimization problem,
- An extensive trade-off analysis between flexibility, area and energy per operation.

The rest of the paper is organized as follows. We provide background on FHE, and discuss related work in Section 2. Section 3 presents the proposed multi-objective optimization algorithm, Section 4 discusses the experimental results, and finally Section 5 concludes the paper.

2. RELATED RESEARCH

Flexible electronics refer to integrated circuits implemented on substrates that can bend, roll, conform, and stretch. These electronics are lighter, thinner, and inexpensive to manufacture [10, 20]. Flexible electronics are composed of a substrate (e.g., thin glass, plastic film, metal foil) [23], backplane electronics (e.g., silicon thin-film transistors (TFT), organic TFT, transparent TFT) [21], and front plane (e.g., liquid crystal display, organic light-emitting display and sensors) [18]. Devices such as flexible displays, sensors, photovoltaic cells, and electronic paper have been demonstrated successfully [3, 7, 24].

Design automation challenges of flexible electronics are presented in [11], while a flexibility-aware placement optimization of flexible TFT Circuits is addressed in [17]. The authors in [17] used indirect measures for flexibility and temperature by approximating their impact through mobility variations. This approach is not suitable for system level design, where the flexibility affects many parameters other than mobility. Flexible hybrid electronics has been proposed to address the performance and scalability problems of purely flexible systems [27]. FHE integrates traditional rigid ICs and printed electronics on a stretchable or flexible substrate [4] to address the performance and scalability problems of flexible circuits [10, 13]. The work presented in [8] presents an approach to place rigid ICs to flexible substrates to minimize the loss in flexibility. However, this approach is limited to the placement problem. It does not consider the target application and its mapping to actual devices. In general, there are no systematic approaches for designing hybrid flexible systems. Our approach mitigates this shortcoming by directly addressing the design optimization of hybrid flexible systems.

The proposed optimization approach is conceptually similar to the technology mapping problem [14]. The major difference is the lack of logic decomposition using universal gates and the placement step, required for computing the flexibility and an accurate area estimate. While placement increases the complexity, smaller problem sizes make an exhaustive design space exploration amenable, as illustrated in Section 4. We note that ideas from technology mapping can be applied to improve the scalability of our solution, which is one of the first steps towards a complete FHE design methodology.

3. MULTI-OBJECTIVE OPTIMIZATION FOR FHE

3.1 Preliminaries and Notation

Component Library: The off-the-shelf rigid and flexible components that can be employed in the FHE design are stored in a *component library*, illustrated in Table 1. We denote the total number of components in the library by N, and the total number of supported tasks by M. The library also lists the component height, width, the average power consumption for each supported task, the maximum supported throughput, and a flexibility index. A flexibility index of 0 means a rigid component, while a flexibility index of 1 means that the corresponding component has a flexibility larger than or equal to that of the substrate. The formal definition of the physical flexibility will be given later in Equation 1.

Definition 1. *Coverage matrix* ($C \in \mathbb{R}^{M \times N}$) is an $M \times N$ matrix where M is the total number of tasks supported and N is the number of components in the library. For task i and component j:

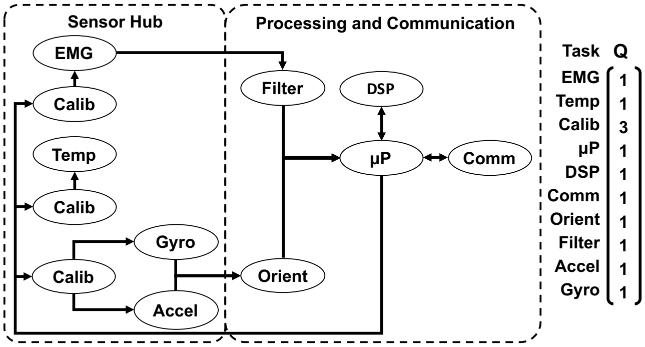

Figure 2: Task flow graph of a simple illustrative health care application. Accel: Accelerometer, Gyro: Gyroscope, Temp: Temperature, EMG: Electromyography, Orient: Orientation, Calib: Calibration, Comm: Communication.

- $c(i, j) = 1$, if the task i is supported by component j,
- $c(i, j) = 0$, otherwise.

Definition 2. *Power consumption matrix* ($P \in \mathbb{R}^{M\times N}$) is an $M \times N$ matrix. For task i and component j:

- $p(i, j)$ = average power consumption of component j running task i
- $p(i, j) = \infty$, otherwise.

Definition 3. *Throughput matrix* ($T \in \mathbb{R}^{M\times N}$) is an $M \times N$ matrix. For a task i, and component j:

- $t(i, j)$ = throughput of component j running task i
- $t(i, j) = 0$, otherwise.

Target Application: In order to automate the FHE system design, we model the target application as a task flow, as illustrated in Figure 2.

Definition 4. *Application Graph*: The target application is represented by the directed graph $G_A(V, E)$,

- The vertices V represent the set of tasks that constitute the application.
- The edges E describe the data flow between the tasks. Each edge $e_i \in E$ has an attribute bw_i, which specifies the data bandwidth.
- The *quantity vector* Q is an $M \times 1$ vector that stores the required tasks for the target application. The entries $Q(i)$ are natural numbers indicating how many instances of the same task are required. For example, $Q(3) = 3$ in Figure 2 means that three instances of calibration task are needed by the target application.

Definition 5. *Quality Metrics*: We use the following primary metrics to evaluate the quality of design:

- The *total area* (A) occupied by the rigid and flexible components on the substrate,
- The target throughput $\mathcal{T}_h$ (Ops/s) required by the application,
- The total average power consumption P (W) at the target throughput $\mathcal{T}_h$,
- The energy per operation E_{op}, which is defined as the ratio of the power consumption to throughput, $E_{op} = P/\mathcal{T}_h$ (J/Ops)
- The physical flexibility δ, which is measured by the maximum deflection of the substrate under a force F.

We compute the maximum deflection in a cantilever beam due to a uniformly distributed force F applied at one end as: $\delta_{\max}(L) = \frac{L^3F}{3EI}$ [2], where E is the modulus of elasticity, and I is the moment of area. Then, the maximum bending along x and y directions are defined as $\delta_{\max}(L_x)$ and $\delta_{\max}(L_y)$, where L_x and L_y are the longest flexible portions left along x and y directions after the rigid components are placed [8]. Hence, we compute the physical flexibility as:

$$\delta = \delta_{\max}(L_x) + \delta_{\max}(L_y) \tag{1}$$

Since we target wearable applications, the total energy consumed per operation is more important than the power consumption. Furthermore, there is a minimum performance requirement set by the application. Therefore, we first eliminate the solutions that do not satisfy the minimum performance requirement. We then use the total area, energy per operation, and flexibility to find the Pareto-optimal design points.

3.2 Optimization Flow Overview

The proposed flow takes the target application and component library, as shown in Figure 3. The central portion of the proposed flow is the optimization loop, where we perform the design space exploration. This loop first finds a mapping of the application tasks to a subset of the components in the library, as shown in Figure 3.

A potential solution that is to be explored is specified by a Selection vector S^p.

Definition 6. *Selection vector (S^p)* is an $N \times 1$ vector where

- $S^p(i) = 1$, if component i is selected in the potential solution
- $S^p(i) = 0$, otherwise.

Each potential solution is first evaluated for its feasibility of implementing the target FHE system by checking if it covers all the necessary tasks of the application. A potential solution S^p is a feasible solution if:

$$S^p \times C \geq Q \tag{2}$$

This constraint ensures that the current selection can support the number of required instances specified by the quantity vector Q.

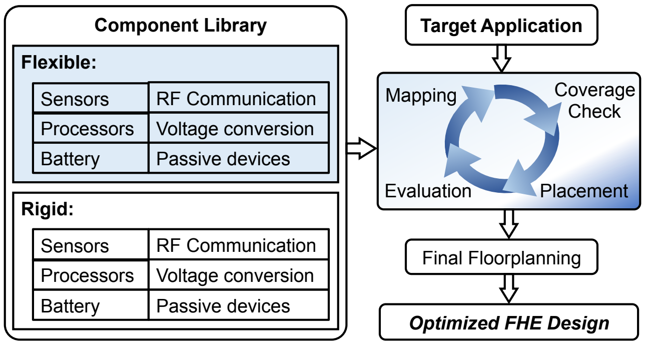

Figure 3: The overview of the proposed multi-objective optimization flow.

For each feasible solution, we need to compute the quality metrics, as given in Definition 5. The power consumption is computed as the sum of the average power of each component when running the assigned task, while the throughput is computed by finding the worst-case throughput using data flow graph analysis [16]. Since the area and flexibility computations require the locations of the components in the floorplan, we also run a placement algorithm at this stage, as detailed in Section 3.3.

The FHE optimization problem is a multi-objective (minimize energy per operation and area, maximize flexibility) and multi-parameter (selection, placement, and assignment of components) optimization problem. Many iterative search-based solutions, to such problems, can be found in the literature. The most commonly used algorithm for such problems is the genetic algorithm [6]. However, random-walk algorithms, such as the genetic algorithm, are not suitable for our problem. Since each task may map to only a small subset of components, the number of feasible solutions is far smaller than the number of infeasible solutions. Random-walk algorithms can also result in the incomplete exploration of the design space, since no hill climbing is possible when starting from the infeasible solutions. Hence, we need an algorithm that systemically explores the design space and takes the specialization of each component into account to eliminate the infeasible solutions up front. We present our proposed solution in the next subsection.

3.3 Optimization Algorithm

The proposed multi-objective optimization algorithm is depicted in Figure 4. The root of the search tree is an empty list without any flexible or rigid components. We start the design space exploration by expanding the root with the first component in the library. We then traverse the tree by adding new library components, following a depth-first search.

Coverage: After visiting each node, we first check whether the current set of components covers all of the tasks required by the target application using Equation 2. For example, if the target application requires wireless connectivity, at least one of the components has to support this functionality. If the coverage fails we call the current solution infeasible, and we continue traversing the tree.

Evaluation: Upon reaching a feasible solution, we *evaluate the quality metrics*. Since evaluating the flexibility and area requires the position of the components, we also place the components during this stage. We can use any floorplanner or placer tool to generate a legitimate placement. To avoid using a computationally expensive algorithm, we employed a bin packing algorithm with O($NlogN$) complexity [12]. Our placement tool gives the area, while flexibility is computed using Equation 1. The quality metrics of the candidate solution are then compared to the best solution found so far, and the search will continue until the feasible design space is exhaustively explored. We note that a more detailed floorplan algorithm can be employed after identifying the most promising design points.

If the new solution improves the Pareto-frontier, *i.e.*, any one of the flexibility, area, and energy per operation metrics, then the new solution and Pareto optimal set are recorded. Note that even after a feasible solution is found, a better solution may be possible by adding more components. For instance, a feasible solution may assign the computation, communication, and calibration tasks, to a microprocessor chip. However, by adding a small dedicated calibration component, and shifting the calibration task to the dedicated component, the power consumption may be reduced. Therefore, the search continues to evaluate solutions, and reports the Pareto-optimal solutions at the end.

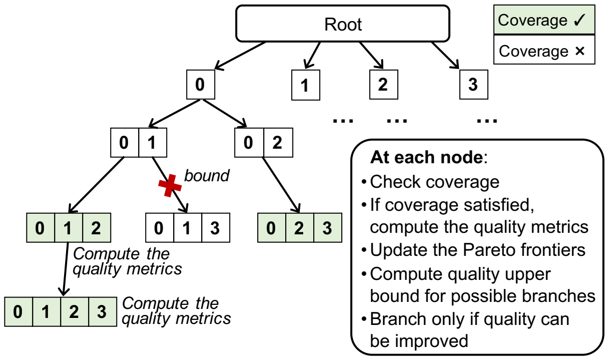
Root
Coverage ✓
Coverage ×
0
1
2
3
...
...
...
0 1
0 2
bound
0 1 2
0 1 3
0 2 3
Compute the quality metrics
0 1 2 3
Compute the quality metrics
At each node:
• Check coverage
• If coverage satisfied, compute the quality metrics
• Update the Pareto frontiers
• Compute quality upper bound for possible branches
• Branch only if quality can be improved

Figure 4: The branch and bound algorithm proposed for optimizing the FHE design.

Bounding: The exploration continues with branching, only if expanding the current solution with more components from the library can improve the current solution. Otherwise, we bound the search and trace back. Bounding is performed by finding an upper bound for flexibility, and lower bounds for the power consumption and area.

Let the height and width of the flexible substrate be H_s and W_s, respectively. Suppose the dimension of the smallest bounding box for the current set of rigid ICs is $H_{cur} \times W_{cur}$. We compute an upper bound on the flexibility using the height (H_i) and width (W_i) of the next component as:

$$\delta_{UB} = \frac{F}{3EI}[(H_s-(H_{cur}+H_i))^3+(W_s-(W_{cur}+W_i))^3] \tag{3}$$

This equations assumes that the new component can be placed without introducing any white space either in the x or the y direction. Hence, it serves as an upper bound. Similarly, the lower bound on the area can be found using the same assumption as:

$$A_{LB} = A_{current} + (H_i \times W_i) \tag{4}$$

We finally compute a lower bound for the power consumption as:

$$P_{LB} = P_{current} + \min_{1\leq j \leq M} \; P(j,i) \tag{5}$$

where the minimum average power for component i is computed over the tasks that it supports. We use these three bounds since they can be computed very efficiently.

Runtime versus Optimality Trade-off: We note that the proposed branch and bound algorithm guarantees an exhaustive design space exploration. This is desirable when the search space is small, but the runtime grows combinatorially. Therefore, we have also introduced a pruning ratio, to bound more aggressively at the expense of loss of optimality. More precisely, let r_δ, r_A and $r_P \in (0,1]$ be pruning ratios for flexibility, area, and power consumption, respectively. We revise the condition to bound as:

Branch *only if*

$$\begin{aligned} &r_\delta \times \delta_{UB} \geq \delta_{max} \quad OR \\ &A_{LB} \leq r_A \times A_{min} \quad OR \quad P_{LB} \leq r_P \times P_{min} \end{aligned} \tag{6}$$

Note that a pruning ratio of one means that there is no runtime optimization, and the search is exhaustive. As the pruning ratio approaches zero, the algorithm will branch only if one of the quality metrics has a potential to improve significantly. Hence, both the runtime and the percentage of the design space that is explored will reduce with the pruning ratios.

4. EXPERIMENTAL EVALUATION

4.1 Experimental Setup

We have developed an in-house tool using C++ to implement the proposed optimization framework, and assess its quality. Our tool reads in a library file and the target application described as a data flow graph. Then, it finds the Pareto-optimal results and corresponding quality vectors using the proposed algorithm.

Component Library: We have tested the proposed approach primarily using three kinds of libraries:

1. *Rigid only* library contains only rigid ICs that are integrated on a flexible substrate. We constructed this library using the data sheets of currently available off-the-shelf ICs. This library represents the most practical scenario available today.
2. *Contemporary flexible* library encompasses the rigid library, and has five additional flexible resources. The identifier *contemporary* is used to indicate the flexible resources that are feasible today. This data is also collected from the existing products [1] and a number of publications [9, 15, 19, 29].
3. *Futuristic flexible* library encompasses the *contemporary flexible* library and adds five more flexible resources. More precisely, we added two copies of the flexible MCU in the *contemporary flexible* library: One that increases the performance by 2× at the same power and the other which reduces the power by 3×. We also included a new flexible MCU which can support Bluetooth Low Energy (BLE) in addition to the processing capability. In addition, we included flexible versions of an accelerometer and a gyroscope as a lot of research on flexible devices is taking place in the area of sensors.

Target Applications: Our target application domain is health monitoring using wearable systems. To provide a complete solution, we employed the following four target applications all of which support integrated sensing, processing, and wireless communication.

1. *Application-1* is a toy example that monitors the user's body temperature and transmits it using ZigBee, after simple processing.
2. *Application-2* monitors the user motion and temperature using gyroscope, accelerometer, and temperature sensors. It processes the motion data to recognize user gestures, and transmits the inferred gesture using BLE.
3. *Application-3* adds electromyography (EMG) electrodes and processing circuitry to *Application-2*.
4. *Application-4* collects user emotion and body motion during an exercise, using an electroencephalogram (EEG), EMG, gyroscope, accelerometer, and temperature sensors. It processes the collected data based on the context, and transmits the inferred intent using BLE.

Public Release: To facilitate research in this emerging area, we released our optimization tool, libraries, and benchmark applications, under GNU General Public License. The source code is available for download at eLab webpage [5].

4.2 Optimization Results

All the solutions reported by our framework satisfy the minimum performance requirements by definition. Therefore, we focus on the Pareto-optimal design points in terms of area, flexibility and energy per operation. Our simulation studies confirm that the FHE design optimization problem exhibits a vast amount of design trade-off. More precisely, we observed 32× range in the area, and 4× range in flexibility *across the Pareto-optimal design points.* Furthermore, the energy per operation, ranged from 5.2×10^{-4} mJ/Ops to 4.095 mJ/Ops, showing the importance of optimizing the design decision. In what follows, we analyze the trade-off between different quality metrics in more detail.

4.2.1 Trade-off Analysis using Pareto Optimal Points

We analyzed the Pareto optimal design points in terms of area, flexibility and energy per operation. For the clarity of visualization, we first present pairwise trade-off analysis for *Application-2* in Figures 5–7 instead of plotting in three dimensions simultaneously. In these plots, the Pareto optimal points for area versus flexibility are represented by red $\diamond$ markers, energy per operation versus flexibility are represented by black $*$ markers, and area versus energy per operation are represented by green $\circ$ markers. In addition, the flexibility is normalized with respect to the flexibility of the substrate.

When we implement the target application using the rigid library, smaller area implies better flexibility. This relation is evident in Figure 5(a), which shows that the design point with the minimum area results in the largest flexibility (the red $\diamond$ marker). The rest of the points on this plot confirm that indeed smaller area leads to better flexibility. When we employ the contemporary library, *i.e.*, there are flexible resources at our disposal, we start to see a trade-off, as depicted in Figure 6(a). The additional Pareto points show that we can use the flexible components to improve the flexibility at the expense of a larger area. However, the improvement in the flexibility is marginal, when we are limited to the flexible components available today. In contrast, Figure 7(a) shows that the futuristic flexible library offers a much wider and more interesting trade-off. We start finding solutions with significantly larger flexibilities, when we add futuristic flexible components. In particular, there are solutions with more than 50% better flexibility than achievable with the rigid library.

We also observe that higher degree of flexibility is achieved at the expense of larger energy per operation, as depicted in Figure 5(b). Detailed inspection of the component library reveals that smaller rigid ICs improve flexibility, but they have significantly lower performance. As a result, the energy per operation increases as smaller rigid ICs are chosen to maximize the flexibility. This observation is also reiterated by the Pareto points on the area versus energy per operation plane in Figure 5(c), which shows that smaller area implies larger energy. Moving from rigid library to the contemporary flexible library shows the same trends, as shown in Figure 6(b) and Figure 6(c). That is, flexibility improves at the expense of the energy per operation. However, the

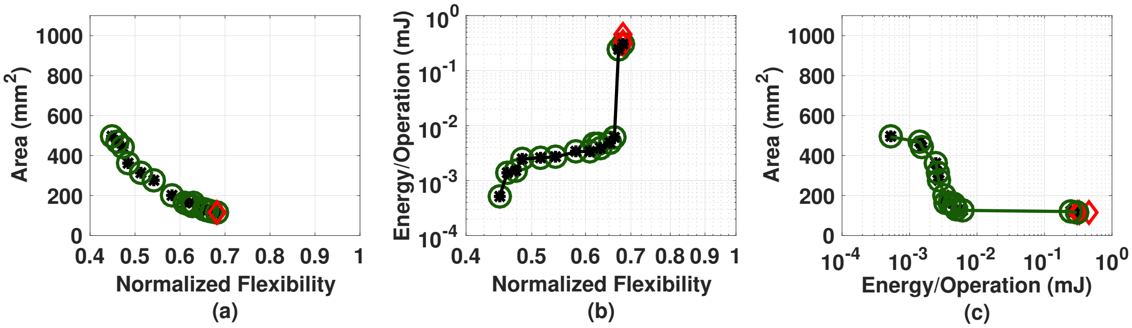

Figure 5: The Pareto-optimal points obtained with the *rigid library*.
A rigid solution would be a line at normalized flexibility = 0.

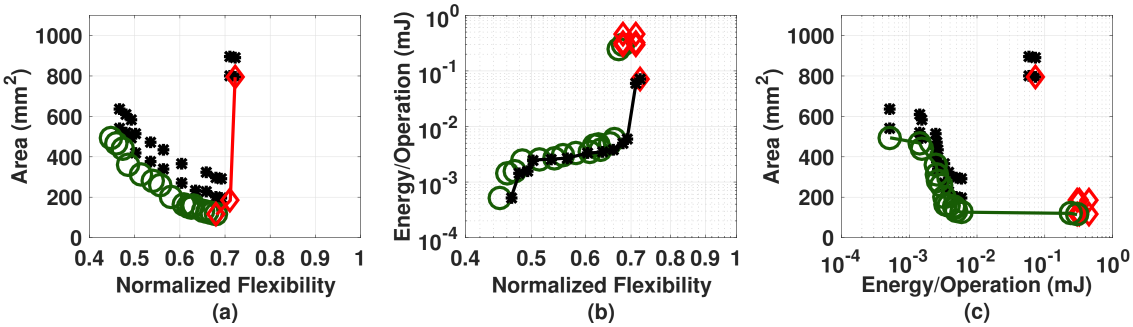

Figure 6: The Pareto-optimal points obtained with the *contemporary flexible library*.

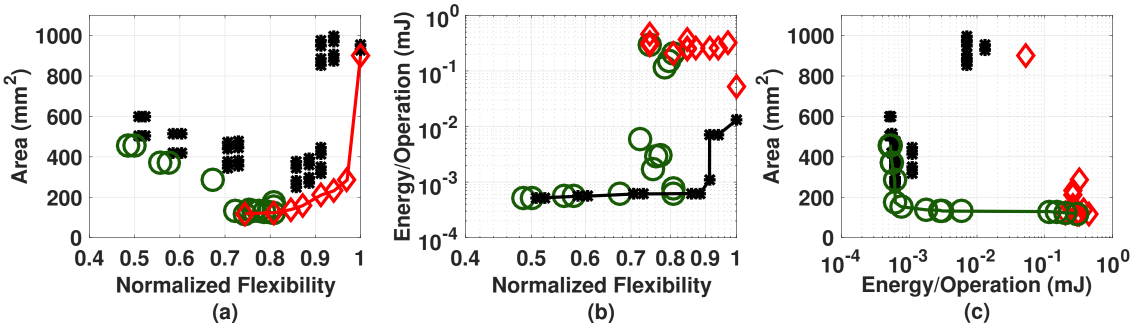

Figure 7: The Pareto-optimal points obtained with the *ideal flexible library*.

newly added flexible components enable a design point with 4× less energy per operation and slightly better flexibility. We observe even larger benefits when we add the futuristic flexible components. In particular, Figure 7(b) shows that we can obtain 6.25×10^{-4} mJ/ops — which is within 20% of the minimum achievable with the other libraries — with a normalized flexibility of 0.88. Furthermore, the design point with the largest flexibility has 35× less energy per operation than the most flexible design that can be obtained using only rigid ICs. Finally, comparing Figure 6(c) and Figure 7(c) reveals that area versus energy per operation also improves considerably when using the futuristic library.

In summary, the flexible components available today improve the quality of the solutions marginally in terms of flexibility. However, the quality of the design and range in flexibility increase significantly, when we add flexible components that have lower power consumption and better performance. The most pronounced impact is observed in Figure 7(b), which shows that, a significant increase in flexibility becomes possible with negligible increase in the energy per operation.

4.2.2 *Trade-off Analysis using Spider Chart*

After the Pareto-optimal solutions are found, we can compare different design points in more detail to shed light on how an ideal solution would look like. To illustrate this process, we considered the design optimization of *Application-3* using the contemporary library. We chose the design points with the highest flexibility (*Solution 1*), the minimum area (*Solution 2*), and the minimum energy per operation (*Solution 3*). We represent these design points using three different planes in the design space, as shown in Figure 8. Note that, the flexibility dimension is represented using the transformation (*1 - flexibility*) for the clarity of the illustration. *Solution 1* has the best flexibility, but it also exhibits the largest area among these three solutions. *Solution 2* significantly improves the area at a small decrease in flexibility, but a large penalty in the energy per operation. In contrast, *Solution 3* achieves significantly better energy per operation, but with the worst flexibility. The ideal quality metric, which is not achievable with the contemporary library, is represented by the shaded solid plane in Figure 8.

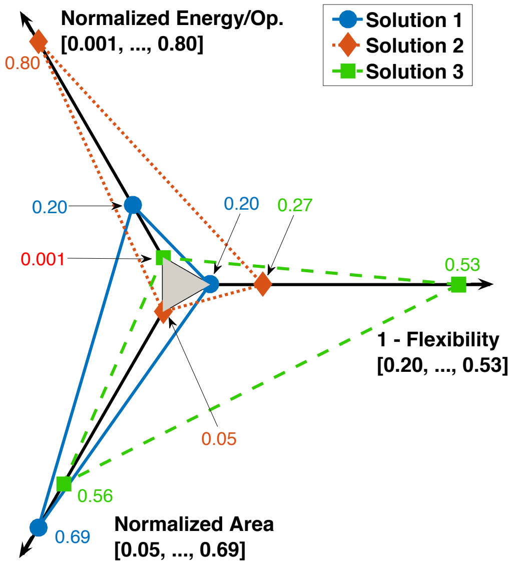

Figure 8: Spider chart depicting the trade-off between quality metrics. The shaded triangle denotes the ideal solution space

This point can be used as a target to optimize the design either by adding new components to the library or by making changes to the application. The optimization tool presented in this paper can be used to guide this optimization process by regenerating the Pareto-optimal points.

4.2.3 *Selecting the Frontier Design Points*

The ultimate goal of the proposed optimization framework is to provide the designer with the most promising solutions. A handful of frontier solutions can later be evaluated using more detailed energy/performance and finite element simulations. To achieve this goal, we use the 3D scatter plot depicted in Figure 9, which has normalized axes similar to Figure 8. In this plot, the distance to the origin along each axes is a measure of quality. Therefore, the marker sizes are inversely proportional to distance to the origin. Furthermore, we highlight the solutions that satisfy a specific criteria given by the designer with ◇ markers. For example,

$$\{\text{Frontiers} \mid (A_i \leq A_{\max}) \cap (E_{\text{ops},i} \leq E_{\text{ops},max}) \cap (\delta_i \geq \delta_{\min})\}$$

As a result, the Pareto-optimal solutions that *also satisfy* this criteria are visualized and selected automatically.

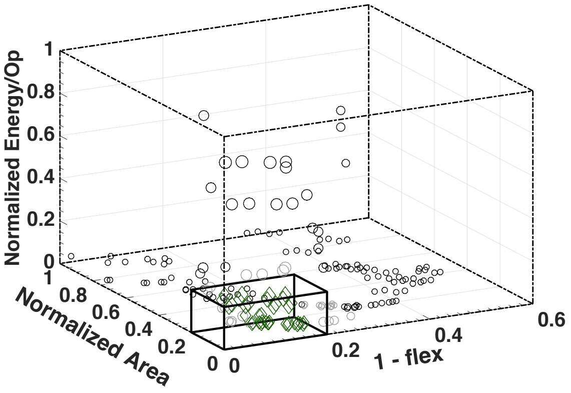

Figure 9: 3D scatter plot showing all the feasible solutions. The solutions inside the smaller cube meet a particular specification

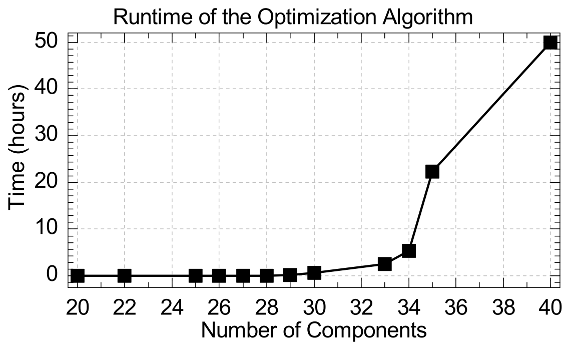

Figure 10: Runtime versus library size.

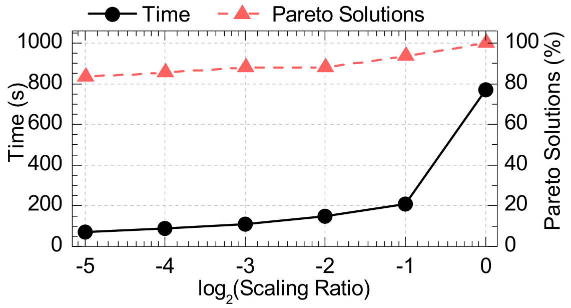

Figure 11: Runtime and percentage of Pareto-optimal points discovered versus scaling ratio.

4.3 Runtime Analysis and Optimization

We performed runtime analysis using *Application-4* and the biggest library with 40 components. Figure 10 shows that the exhaustive search completes for libraries with as many as 40 components. Since the default operation mode is exhaustive, the computational complexity is combinatorial. However, a significant speed-up can be achieved by relaxing the exhaustive nature of the current implementation, as described in Section 3.3. More precisely, sweeping the pruning ratios (r_P, r_δ, r_A) defined in Section 3.3 reduces the number of evaluations at the expense of missing some Pareto-optimal points. To investigate the impact on runtime and solution quality, we ran *Application-2* with different pruning ratios and a library (*Contemporary Library*) with 30 components. We employed a small library to be able to analyze the loss in optimality, and used equal pruning ratios ($r_P = r_\delta = r_A$) for illustration purposes. Figure 11 shows that the reduction in in runtime is almost on order of magnitude, while the percentage of Pareto-optimal points discovered reduces marginally. Finally, we ran the proposed algorithm for the same application with a component library of size 51 components. While the exhaustive optimization does not complete in more than two days, a pruning ratio of 1/1024 resulted in 65 minutes of runtime. This simple trade-off shows that a probabilistic lower bound can enable us to scale the proposed technique significantly.

5. CONCLUSION

The use of flexible electronics has a great potential for designing wearable systems. However, the poor performance and large feature sizes make them impractical to implement systems that are on par with the current CMOS technology. Flexible hybrid electronics technology has emerged to address the limitations of pure flexible electronics. FHE technology can enable physically bendable and stretchable systems whose energy efficiency and performance approach that of traditional ICs. This paper presents a multi-objective design algorithm that can realize this potential. The proposed algorithm utilizes a component library to implement a given application to optimize flexibility, area and energy per operation. The proposed approach can be used for two purposes. First, it can find the optimum design possible with a given library. Secondly, it can be used to explore the properties of the flexible components that can most improve the library.

6. REFERENCES

[1] American Semiconductor Inc. FleX-MCU$^{\mathrm{TM}}$, FleX-RFIC$^{\mathrm{TM}}$, FleX-ADC$^{\mathrm{TM}}$. http://www.americansemi.com [Online; accessed 9-July-2016].

[2] M. F. Ashby. Materials Selection in Mechanical Design (Fourth Edition). Butterworth-Heinemann, 2011.

[3] S. Bai, W. Wu, Y. Qin, N. Cui, D. J. Bayerl, and X. Wang. High-Performance Integrated ZnO Nanowire UV Sensors on Rigid and Flexible Substrates. *Advanced Functional Materials*, 21(23):4464–4469, 2011.

[4] R. L. Chaney et al. FleX$^{\mathrm{TM}}$ Silicon-on-Polymer$^{\mathrm{TM}}$: Flexible (Pliable) ICs from Commercial Foundry Processes. In *Proc. of Government Microcircuit Applications and Critical Technology) Conf.*, March 2013.

[5] eLab. SoPtimizer Design and Optimization Framework for Systems-on-Polymer. [Online; accessed 17 July 2016], http://elab.engineering.asu.edu/public-release/.

[6] C. M. Fonseca and P. J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary computation*, 3(1):1–16, 1995.

[7] S. Gowrisanker et al. A Novel Low Temperature Integration of Hybrid CMOS Devices on Flexible Substrates. *Organic Electronics*, 10(7):1217–1222, 2009.

[8] U. Gupta, S. Jain, and U. Y. Ogras. Can Systems Extended to Polymer? SoP Architecture Design and Challenges. In *Proc. of Intl. System-on-Chip Conf.*, pages 203–208, 2015.

[9] H. Huang. Flexible Wireless Antenna Sensor: A Review. *IEEE Sensors Journal*, 13(10):3865–3872, 2013.

[10] T.-C. Huang, J.-L. Huang, and K.-T. Cheng. Robust Circuit Design for Flexible Electronics. *IEEE Design and Test of Computers*, 28(6):8–15, 2011.

[11] T.-C. J. Huang et al. Design, Automation, and Test for Low-Power and Reliable Flexible Electronics. *Found. and Trends in EDA*, 9(2):99–210, 2015.

[12] J. Jylänki. A Thousand Ways to Pack the Bin - a Practical Approach to Two-Dimensional Rectangle Bin Packing, 2010. http://clb.demon.fi/files/RectangleBinPack.pdf [Online; accessed 18-July-2016].

[13] H. E. Katz and J. Huang. Thin-Film Organic Electronic Devices. *Annual Review of Materials Research*, 39:71–92, 2009.

[14] K. Keutzer. DAGON: Technology Binding and Local Optimization by DAG Matching. In *Papers on Twenty-five Years of Electronic Design Automation*, pages 617–624, 1988.

[15] C.-Y. Lee, S.-J. Lee, and G.-W. Wu. Fabrication of Micro Temperature Sensor on the Flexible Substrate. In *7th IEEE Conference on Nanotechnology*, pages 1050–1053, 2007.

[16] E. A. Lee and D. G. Messerschmitt. Synchronous Data Flow. *Proceedings of the IEEE*, 75(9):1235–1245, 1987.

[17] J.-L. Lin, P.-H. Wu, and T.-Y. Ho. Placement Optimization of Flexible TFT Circuits with Mechanical Strain and Temperature Consideration. *ACM Journal on Emerg. Tech. in Comput. Sys.*, 11(1):1, 2014.

[18] P. Lin and F. Yan. Organic Thin-Film Transistors for Chemical and Biological Sensing. *Advanced Materials*, 24(1):34–51, 2012.

[19] K. Myny, E. Van Veenendaal, G. H. Gelinck, J. Genoe, W. Dehaene, and P. Heremans. An 8-bit, 40-instructions-per-second organic microprocessor on plastic foil. *IEEE Journal of Solid-State Circuits*, 47(1):284–291, 2012.

[20] A. Nathan et al. Flexible Electronics: The Next Ubiquitous Platform. *Proc. of IEEE*, 100:1486–1517, 2012.

[21] T. N. Ng, W. S. Wong, M. L. Chabinyc, S. Sambandan, and R. A. Street. Flexible Image Sensor Array with Bulk Heterojunction Organic Photodiode. *Applied Physics Letters*, 92(21):213303, 2008.

[22] J. Noh, M. Jung, Y. Jung, C. Yeom, M. Pyo, and G. Cho. Key Issues With Printed Flexible Thin Film Transistors and Their Application in Disposable RF Sensors. *Proc. of the IEEE*, 103(4):554–566, 2015.

[23] K. Nomura et al. Room-Temperature Fabrication of Transparent Flexible Thin-Film Transistors using Amorphous Oxide Semiconductors. *Nature*, 432(7016):488–492, 2004.

[24] J.-S. Park et al. Flexible Full Color Organic Light-Emitting Diode Display on Polyimide Plastic Substrate Driven by Amorphous Indium Gallium Zinc Oxide Thin-Film Transistors. *Applied Physics Letters*, 95(1):013503, 2009.

[25] R. D. G. H. Peter Harrop, James Hayward. Wearable Technology 2015-2025: Technologies, Markets, Forecasts. http://www.idtechex.com/research/reports/wearable-technology-2016-2026-000483.asp. [Online; accessed 9-July-2016].

[26] T. Sekitani, U. Zschieschang, H. Klauk, and T. Someya. Flexible organic transistors and circuits with extreme bending stability. *Nature Materials*, 9(12):1015–1022, 2010.

[27] J. Sheats. Truly wearable electronics. In *Presented at the FlexTech Alliance Workshop: Flexible Hybrid Electronics - Challenges and Solutions*, July 2014.

[28] R. Street, T. Ng, D. Schwartz, G. Whiting, J. Lu, R. Bringans, and J. Veres. From printed transistors to printed smart systems. *Proceedings of the IEEE*, 103(4):607–618, 2015.

[29] W. Xiong, U. Zschieschang, H. Klauk, and B. Murmann. A 3v 6b successive-approximation adc using complementary organic thin-film transistors on glass. In *2010 IEEE Intl. Solid-State Circuits Conf. Digest of Technical Papers*, pages 134–135, 2010.