

Ground Plane Partitioning for Current Recycling of Superconducting Circuits

Naveen Kumar Katam, *Member, IEEE*, Bo Zhang and Massoud Pedram, *Fellow, IEEE*

Abstract—Superconducting single flux quantum (SFQ) technology using Josephson junctions (JJs) is an excellent choice for the computing fabrics of the future. Current recycling is a necessary technique for the implementation of large SFQ circuits with energy-efficiency, where circuit partitions with similar bias current requirements are biased serially. Though this technique has been verified for small scale circuits, it has not been implemented for large circuits as there is no trivial way to partition the circuit into circuit blocks with separate ground planes. The major constraints for partitioning are (1) equal bias current and (2) equal area for all the partitions; (3) minimize the connections between adjacent ground planes with high-cost for non-adjacent planes. For the first time, all these constraints are formulated into a cost function and it is minimized with the gradient descent method. The algorithm takes a circuit netlist and the intended number of partitions as inputs and gives the output as groups of cells belonging to separate ground planes. It minimizes the connections among different ground planes and gives a solution on which the current recycling technique can be implemented. The parameters of cost function have been initialized randomly along with minimizing the dimensions to find the solution quickly. On average, 30% of connections are between non-adjacent ground planes for the given benchmark circuits.

Index Terms—SFQ, Energy-efficient computation, Current recycling, Partitioning, Ground planes.

I. INTRODUCTION

As the scaling of CMOS technology has hit a roadblock and facing the problems of high temperatures and high power consumption, there is an intense search for the post-CMOS technologies that perform better in regards to the power and the speed of computation. Superconducting electronics operating at cryogenic temperatures exhibit unique characteristics and excellent performance for various electronic applications to which conventional electronics are not a match [1]. Superconducting single flux quantum (SFQ) technology with its energy-efficient variants appears to be one of the most promising technologies of the future with fast switching (~ 1 ps) and low switching energy per bit (10^{-19} J). One of the earlier demonstrated circuits with rapid SFQ (RSFQ) technology have operating frequencies up to 770 GHz [2].

Over the last three decades, many challenges with design, fabrication, testing, and energy-efficiency of SFQ circuits were solved resulting in a successful demonstration of many important circuits [3], [4]. However, some problems are yet to be solved to successfully fabricate and demonstrate large-scale SFQ circuits that are on par with the state-of-the-art CMOS

circuits. One of those challenges is the transfer of bias current and I/O signals of a (large-scale) SFQ circuit between the cryogenic environment and the room temperature. Focusing on the bias current transfer, it is difficult to carry a large amount of current to the cryogenic system as there will be joule heating in the probes and on the chip [5]. One way to overcome this problem is to use the current recycling technique [6].

II. SINGLE FLUX QUANTUM LOGIC

The two-terminal Josephson Junction (JJ) is the active device in any of the existing superconducting logic circuits [7]. Each JJ is designed to have a specific value of critical current (I_c) which defines the maximum amount of current that can pass through the JJ without its state becoming normal from a superconducting state. When a JJ comes out of its superconducting state, it generates a voltage pulse and the area under the pulse is equal to a single flux quantum whose value at the units are given in (1).

$$\Phi_0 = \int V(t)dt = \frac{h}{2e} = 2.07 \times 10^{-19} \text{V.s} \quad (1)$$

In single flux quantum (SFQ) logic circuits, the presence and absence of an SFQ pulse are treated as logic-1 and logic-0 respectively. Every logic gate in the cell library is designed with JJs of different critical currents and inductances of different values to store, to generate, to pass and to dissipate the SFQ pulse so that the truth table of a logic function can be implemented successfully with the SFQ pulses. To aid the manipulation of SFQ pulses for implementing a logic gate, a specific amount of bias current is given at specific locations of the electrical schematic of the gate.

Some unique characteristics of SFQ technology are: (i) most gates are clocked implying that a circuit is gate-level pipelined [8]; (ii) an active gate called *splitter* [9] need to be employed to take a fanout of two and thus a tree of *splitters* for a larger fanout; (iii) flow clocking is used for distributing the clock pulse to the logic gates since the switching of circuits is very fast and the delay of clock line is comparable to the delay of logic circuit [10]; (iv) use of passive transmission lines for the ballistic propagation of SFQ pulse over longer distances on the chip. [11], [12] provide a more detailed explanation of the technology, current status, and challenges.

Biasing is a very important aspect of SFQ technology as the right amount of current delivery is necessary for the proper functioning of a logic gate and thus a circuit. When resistors are used with a small voltage around 2.5 mV to generate the required bias current for each gate, the circuits are called rapid

Naveen Kumar Katam, Bo Zhang and Massoud Pedram are with the University of Southern California, Los Angeles, CA, 90089, USA.

SFQ (RSFQ) circuits [13]. Since resistors consume power, to eliminate the static power dissipation, Energy-efficient RSFQ (ERSFQ) biasing is proposed recently [14], [15]. Despite the advances in eliminating static power dissipation, there are other challenges to overcome with the biasing current. For large-scale circuits, the required bias current will be in the range of a few tens of Amperes. Providing such high amounts of current to a cryogenic system increases the thermal load of the cables, generates undesirable on-chip magnetic-fields along with inefficiency in overall power supply [16]. These problems can be overcome by employing a circuit technique called current recycling [6] which is the main focus of the paper.

III. CURRENT RECYCLING CONSTRAINTS

A. Current Recycling Concept

In simple terms, the concept of current recycling can be described as partitioning a large circuit into several smaller circuit blocks with an equal bias current requirement and then biasing them serially. Serial biasing implies that the ground return current of a circuit block will be feeding the bias bus of the following circuit block. This requires each circuit block to have its isolated ground plane. There will be a potential difference between each ground plane and between the adjacent ground planes it will be equal to the bias bus voltage which is typically around 2.5 mV. Since the ground planes are isolated as separate islands, the communication between them cannot be done with a direct connection. Communication between any two ground planes (transmission of SFQ pulses) should use differential coupling and can be done either inductively or capacitively [16].

For differential SFQ transfer, there has been significant work on the development of inductive and capacitive coupling circuits. It is well accepted that the inductive (magnetic) coupling circuits can reliably transfer the SFQ pulses between two ground planes. Electrically, the inductive coupling circuit will have a *driver* circuit (which sits on the sending ground plane) that sends an SFQ pulse and a *receiver* circuit (which sits on the receiving ground plane) that receives the SFQ pulse. Note that the connection is one-directional.

B. Partition Constraints

1) *Equal bias current*: All the partitioned circuit blocks should have an equal bias current requirement. Note that the bias current required for one block (Ground plane 1 in Fig. 1) is provided from the outside and it will be reused for the rest of the blocks. If these circuit blocks have an unequal bias current requirement, some blocks may fail because of under-biasing or over-biasing.

Since it is almost impossible to partition a circuit into blocks having an equal bias current (unless it is a regular structure such as memories or FPGA [17]), we propose to use dummy circuit structures to make the bias currents equal for all the blocks after the final partitioning. After the partition, the value of externally supplied bias current should be equal to the bias current requirement for the block which needs maximum

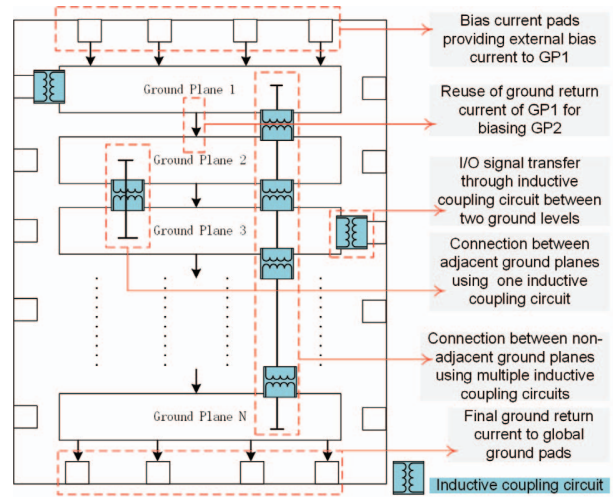


Fig. 1. Illustration of current recycling technique on a superconducting chip

bias current. For the rest of the blocks, dummy structures will be added to pass the additional bias current. The amount of current passing through these dummy structures will be a measure for the quality of partition which will be called bias compensation in the later sections of the paper.

2) *Equal area*: Though this is not a hard requirement, it is preferred to have the same area for all the blocks. This is to avoid the unnecessary white spaces on the chip and resulting in optimal use of the chip area. For every block, we will estimate the free space by subtracting its area from the area of the largest block and use it as the measure of partition quality.

3) *Interconnections*: Though the interconnections are allowed between circuit blocks sitting on any two ground planes (GPs) with inductive coupling circuits, it is physically impossible for non-adjacent blocks to have direct connections as illustrated in Fig. 1. The inductive transfer of an SFQ pulse depends on the mutual inductance implying that the *driver* and *receiver* circuits for inductive transfer have to be laid out side by side for its implementation. Without the loss of generality, the assumption is that all the partitions of the circuit are in parallel and the flow of current is from top block to the bottom block. The chip pads and I/O circuits are on the perimeter of the chip and can share the common ground. From a layout point of view, direct connections are only possible between adjacent GPs and between I/O circuits and a GP. Hence, for a connection between non-adjacent GPs, an SFQ pulse has to pass through all ground planes between that two non-adjacent GPs requiring the usage of several inductive coupling circuits instead of just one. This is undesirable as it consumes more area on the chip and also decreases the operating frequency of the circuit.

Based on the presented knowledge, the partition of circuit blocks should try to (i) minimize interconnections among different ground planes; (ii) where ever the interconnections are required, they should be among the ground planes that are as close as possible in terms of physical distance on the chip.

IV. PARTITIONING ALGORITHM

A. Cost Function formulation

Note that the ground plane partitioning problem with the constraints described in section III-B can not be formulated as a classic K-way partitioning [18] problem. Hence, we proceed to formulate a new cost function that takes all the constraints into account. For a circuit netlist, the cost function should have a mechanism to compute the cost of different gate assignments to a given number of GPs.

Suppose G denotes the total number of gates in the circuit and K denotes the total number of ground planes into which the circuit will be partitioned, the problem at hand is to partition G gates into K ground planes that are to be serially biased. A single gate i can only belong to a single plane k after the partition. To represent this fact in a mathematical formulation and to track the assignment of a gate to a plane, the variable $w_{i,k}$ is used in an indicator function [19] as shown in (2).

$$w_{i,k} = \begin{cases} 1 & \text{if gate } i \text{ is in the plane } k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Equivalently, label l_i is defined to represent the plane to which gate i belongs as shown in Equation (3). Note that the variable $w_{i,k}$ contains two parameters, whereas l_i is a single parameter.

$$l_i = \sum_{k=1}^K k w_{i,k} \quad (3)$$

1) *Interconnections*: As explained before, we would like to keep the connections to gates within a plane as much as possible and minimize the interconnections between the planes. A connection between two adjacent planes is less desirable but acceptable and a connection between two non-adjacent planes not desirable and thus there will be an increased cost for such a connection. If the distance between two connected gates is larger in terms of the number of planes it has to pass through, the larger the cost assignment for that connection. A *distance* parameter, d for two connected gates $i1$ and $i2$ is defined as $d = |l_{i1} - l_{i2}|$ representing the number of planes a connection has to pass through. Note that the connections within a plane will have a distance value of zero and thus no cost. If E is the set of all connections between any two gates $i1$ and $i2$ of the circuit and $|E|$ represents the number of connections, the cost function for the interconnections, $F1$ is formulated as shown in Equation (4). N_1 is normalization constant. Power of 4 is used to model the sharp increment of a connection cost with the increase in distance.

$$F_1 = \frac{1}{N_1} \sum_{(i1,i2) \in E} (|l_{i1} - l_{i2}|)^4 \quad (4)$$

$$N_1 = |E| (K - 1)^4$$

2) *Bias current*: The constraint on bias current is that every partition of the circuit should have (almost) equal bias current. Since it is an almost impossible constraint, the difference in bias current requirement of each block will be made equal by using dummy structures to bypass the extra current. For a mathematical representation of cost function, the difference between the bias current requirement of any two ground planes should be minimized. The variance of the bias current requirement distribution to K planes is a good choice for the cost function of bias current constraint and it is shown in (5) as $F2$. Variables b_i and B_k represents the bias current requirements of the gate i and the plane k , respectively. Note that the value of B_k is equal to the sum of b_i values of the gates assigned to the plane k . \bar{B} is the average bias current of K planes. N_2 is the normalization constant.

$$B_k = \sum_{i=1}^G b_i w_{i,k} \quad \bar{B} = \frac{1}{K} \sum_{k=1}^K B_k \quad (5)$$

$$F_2 = \frac{1}{N_2} \frac{1}{K} \sum_{k=1}^K (B_k - \bar{B})^2 \quad N_2 = (K - 1) \bar{B}^2$$

3) *Block area*: Similar to the bias current distribution, area of each partitioned block should be as close as possible. However, difference in area does not lead to the malfunctioning of the circuit. This constraint enforces an efficient use of area on the chip. Hence, variance of the area distribution over the K-planes is considered as the cost function and is shown in (6) as $F3$. Variables a_i and A_k represent the area of gate i and the block area of all gates in plane k . \bar{A} is the average area over K planes.

$$A_k = \sum_{i=1}^G a_i w_{i,k} \quad \bar{A} = \frac{1}{K} \sum_{k=1}^K A_k \quad (6)$$

$$F_3 = \frac{1}{N_3} \frac{1}{K} \sum_{k=1}^K (A_k - \bar{A})^2 \quad N_3 = (K - 1) \bar{A}^2$$

4) *Problem*: In conclusion, an integer programming problem is modelled with the linear combination of individual cost functions presented in above sections as shown in (7). c_1 , c_2 , and c_3 are constants which can be tuned. This cost function is to be minimized with constraint shown in (7) which contains $G * K$ number of $w_{i,k}$ variables with G equality conditions and $G * K$ indicator functions. Defining sets $\mathbb{G} = \{1, \dots, G\}$ and $\mathbb{K} = \{1, \dots, K\}$.

$$\begin{aligned} & \text{minimize} && F = c_1 F_1 + c_2 F_2 + c_3 F_3 \\ & \text{subject to} && \sum_{k=1}^K w_{i,k} = 1, \forall i \in \mathbb{G} \\ & && w_{i,k} \in \{0, 1\}, \forall i \in \mathbb{G}, \forall k \in \mathbb{K} \end{aligned} \quad (7)$$

B. Final Cost Function

The problem shown in (7) is not solvable in polynomial time for the given constraints. Therefore, the method of Lagrange multiplier will be used to add the constraint into the cost function and convert the original integer programming problem into a linear programming problem. We have modified

the Lagrange multiplier method by formulating another cost function, F_4 which is based on (7) but different from the original constraints. Equation (8) shows the final form of the problem to be solved with the new cost function and the simple linear programming constraints.

$$\begin{aligned} & \text{minimize} && F = c_1 F_1 + c_2 F_2 + c_3 F_3 + c_4 F_4 \\ & \text{subject to} && w_{i,j} \in [0, 1], \forall i \in \mathbb{G}, \forall k \in \mathbb{K} \end{aligned} \quad (8)$$

$$\begin{aligned} \overline{w}_{i,:} &= \frac{1}{K} \sum_{k=1}^K w_{i,k} \\ F_4 &= \sum_{i=1}^G [(K \overline{w}_{i,:} - 1)^2 - \frac{1}{K} \sum_{k=1}^K (w_{i,k} - \overline{w}_{i,:})^2] \\ N_4 &= G(K - 1)^2 \end{aligned} \quad (9)$$

The expression of F_4 shown in (9) combines all the equations of constraints belonging to (7). The first term of F_4 is the squared difference of $\sum_{k=1}^K w_{i,k}$ and 1, enforcing the equality constraint of (7). The second term is the negative of $\frac{1}{K} \sum_{k=1}^K (w_{i,k} - \overline{w}_{i,:})^2$, which forces all $w_{i,k}$ values belonging to a single gate i to have a large variance as the increase in variance results in decrease in cost function value. Combination of both terms in the cost function converges vector $[w_{i,1}, \dots, w_{i,K}] \forall i \in \mathbb{G}$ to have a sum value equal to 1 and a large variance for its $w_{i,k}$ values. For example, assuming that the sum condition is satisfied for $K = 4$, the possibility to have a result vector with values $[0.7, 0.1, 0.1, 0.1]$ is higher than $[0.25, 0.25, 0.25, 0.25]$. The final assignment of gates to the planes will be based on the constituent values of the said vector for each gate i . Since one of the values will be larger than the others, gate i will be assigned to the plane k which has the larger $w_{i,k}$ value.

C. Proposed solution

Cost function presented in (8) is not in the standard form of a linear programming formulation and hence it is difficult to find an optimal solution for $G * K$ number of $w_{i,k}$ variables. Hence, we propose to use the gradient descent algorithm to find a sub-optimal solution for this linear programming formulation as it is differentiable. Gradients are shown in (10). For the gradient descent method, initial values of $w_{i,k}$ will be generated randomly. Since the condition, $\sum_{k=1}^K w_{i,k} = 1$ is already known, G number of K -dimensional vectors $[w_{i,1}, \dots, w_{i,K}]$ with $1 \leq i \leq G$ will be recursively updated as vectors with $K - 1$ dimensions. Starting with the randomly initialized $w_{i,j}$ data and normalizing every vector with the sum of its values will satisfy $\sum_{k=1}^K w_{i,k} = 1$. With the initial $w_{i,j}$ values, $cost_new$ will be computed based on the expression F in (8). Applying the gradient descent method, the value of cost function F will be reduced till the relative cost reduction reaches below the $margin$ value. Finally, we pick the plane corresponding to the index of the largest value among $w_{i,1}, \dots, w_{i,K}$ as the location of gate i . Note that the $margin$ value is chosen such that both the time of computation and the precision of result are not compromised.

$$\begin{aligned} \frac{\partial F_1}{\partial w_{i,k}} &= \frac{4k}{N_1} \left[\sum_{(i,i2) \in E} (|l_i - l_{i2}|)^3 - \sum_{(i1,i) \in E} (|l_{i1} - l_i|)^3 \right] \\ \frac{\partial F_2}{\partial w_{i,k}} &= \frac{2b_i}{KN_2} (B_k - \overline{B}) \\ \frac{\partial F_3}{\partial w_{i,k}} &= \frac{2a_i}{KN_3} (A_k - \overline{A}) \\ \frac{\partial F_4}{\partial w_{i,k}} &= \frac{2}{N_4} \left[\left(K + \frac{1}{K} \right) (\overline{w}_{i,:} - w_{i,k}) + K - 1 \right] \end{aligned} \quad (10)$$

Algorithm 1: Ground Plane Partitioning Algorithm for a given K-number of planes

```

1:  $cost\_old = \infty$ 
2:  $margin = 0.0001$ 
3: for  $i = 1, \dots, G$  do
4:   for  $k = 1, \dots, K$  do
5:     Randomly generate  $w_{i,k}$ 
6:   end for
7:    $w_{i,:} = \sum_{k=1}^K w_{i,k}$ 
8:   for  $k = 1, \dots, K$  do
9:      $w_{i,k} = w_{i,k} / w_{i,:}$ 
10:  end for
11: end for
12: while True do
13:    $cost\_new = c_1 F_1 + c_2 F_2 + c_3 F_3 + c_4 F_4$ 
14:   if  $|cost\_new / cost\_old - 1| \leq margin$  then
15:     break
16:   end if
17:   for  $i = 1, \dots, G, k = 1, \dots, K$  do
18:      $\Delta w_{i,k} = c_1 \frac{\partial F_1}{\partial w_{i,k}} + c_2 \frac{\partial F_2}{\partial w_{i,k}} + c_3 \frac{\partial F_3}{\partial w_{i,k}} + c_4 \frac{\partial F_4}{\partial w_{i,k}}$ 
19:   end for
20:   for  $i = 1, \dots, G, k = 1, \dots, K$  do
21:      $w_{i,k} = w_{i,k} - \Delta w_{i,k}$ 
22:      $w_{i,k} = \min(w_{i,k}, 1)$ 
23:      $w_{i,k} = \max(w_{i,k}, 0)$ 
24:   end for
25:    $cost\_old = cost\_new$ 
26: end while
27: for  $i = 1, \dots, G$  do
28:    $s = \text{argmax}_{1 \leq k \leq K} (w_{i,k})$ 
29:    $l_i = s$ 
30: end for

```

V. RESULTS

SFQ benchmark circuit suite [20], [21] provides a set of post-routing SFQ circuits, such as Kogge-Stone adders (KSA), multipliers (MULT), integer dividers (ID), and some ISCAS benchmark circuits in DEF format [22]. The algorithm is implemented in *Python3* programming language, including the parser for DEF-format circuits. The gradient descent method is an optimization algorithm based on the first derivative of the cost function. Advanced algorithms such as the Newton method are based on the second derivative and requires the

TABLE I
PARTITION RESULTS OF BENCHMARK CIRCUITS WITH $K = 5$

Circuits	# Gates	# Connections	$d \leq 1$	$d \leq 2$	B_{cir} (mA)	B_{max} (mA)	I_{comp} (%)	A_{cir} (mm^2)	A_{max} (mm^2)	A_{FS} (%)
KSA4	93	118	74.6%	97.5%	80.089	17.50	9.24	0.4512	0.0972	7.71
KSA8	252	320	70.3%	94.4%	216.72	45.27	4.43	1.2192	0.2520	3.35
KSA16	650	826	66.5%	88.7%	557.66	118.09	5.88	3.1392	0.6600	5.12
KSA32	1592	2029	64.4%	85.9%	1362.55	304.07	11.58	7.6800	1.7028	10.86
MULT4	254	310	73.2%	93.2%	222.03	47.70	7.42	1.2192	0.2616	7.28
MULT8	1374	1678	63.6%	85.6%	1201.32	256.85	6.90	6.5952	1.4004	6.17
ID4	553	678	71.1%	91.4%	467.00	100.29	6.69	2.6796	0.5700	6.36
ID8	3209	3705	58.2%	81.6%	2783.89	622.39	11.78	15.5400	3.4860	12.16
C432	1216	1434	65.0%	87.5%	1045.17	222.31	6.35	5.9448	1.2792	7.59
C499	991	1318	63.5%	86.3%	834.92	178.17	6.70	4.8060	1.0212	6.24
C1355	1046	1367	61.8%	85.4%	883.35	192.41	8.97	5.0808	1.1076	9.00
C1908	1695	2095	60.0%	85.0%	1447.03	328.53	13.52	8.2536	1.8804	13.91
C3540	3792	4927	54.0%	77.7%	3193.23	670.01	4.91	18.5556	3.8784	4.51

TABLE II
PARTITION RESULTS OF KSA4 NETLIST FOR DIFFERENT K VALUES

K	$d \leq 1$	$d \leq \lfloor \frac{K}{2} \rfloor$	B_{max} (mA)	I_{comp} (%)	A_{max} (mm^2)	A_{FS} (%)
5	74.6%	97.5%	17.50	9.24	0.0972	7.71
6	64.4%	94.9%	14.40	7.88	0.0840	11.70
7	53.4%	89.8%	12.45	8.79	0.0696	7.98
8	45.8%	95.8%	11.16	11.49	0.0648	14.89
9	38.1%	83.9%	10.24	15.12	0.0576	14.89
10	38.1%	90.7%	9.69	21.64	0.0552	22.34

calculation of the Hessian matrix, which is computationally expensive. Based on the experimental results, the gradient descent method provides a good estimation for the result within an acceptable time window.

These circuit netlists are run through the algorithm for a K value of 5 and the partitioning results are shown in Table I. It shows the properties of each netlist, namely, number of gates, number of gate-to-gate connections, total bias current requirement (B_{cir}) and the area needed for all the gates (A_{cir}). For showing the performance of the partition in terms of interconnections, *distance* parameter (d) is used. On average, the percentage of the number of connections with *distance* less than 1 and 2 are 65.1% and 87.7%, respectively. This implies that the connections with *distance* value of 3 and 4 have been minimized to avoid the high connection costs between two gates in non-adjacent planes. For the performance of algorithm in terms of bias current distribution, results are given as parameters B_{max} (bias current of the partitioned block that has maximum bias current requirement) and I_{comp} (bias current passing through the dummy structures to make the partitioned blocks to have an equal bias current requirement) which are defined in (11). Similar to the bias current distribution results for parameters, A_{max} (the largest area consumed by gates for a partition) and A_{FS} (free space due to the imbalance in the area of partitions) are presented. With the proposed solution, the average I_{comp} and the average A_{FS} on-chip for the benchmark circuit suite are only 8.0% and 7.7%, respectively.

$$B_{max} = \max_{1 \leq k \leq K} B_k \quad I_{comp} = \sum_{k=1}^K (B_{max} - B_k) \quad (11)$$

$$\% \text{ of } I_{comp} = \frac{I_{comp}}{B_{cir}} * 100$$

To present the performance of the partition algorithm with the increase in the number of ground planes, results of KSA4 netlist partition are presented in Table II. As the number of partitions increases, the cost also increases and thus the penalty in terms of the number of connections. For the specific partition problem we are solving, the increase in penalty is even more significant since the connections among farther planes cost higher than the closer planes in terms of *distance*. It can be seen that the percentage of connections with $d \leq 1$ decreases with an increase in K value. This significant decrease is because of the additional constraints posed by the equal bias current and the equal area. For a better comparison, the result for percentage of connections within *distance* $d \leq \lfloor \frac{K}{2} \rfloor$ are given which covers at least 75% of all connections. On average, 92.1% connections have *distance* less than half the number of ground planes. As expected, with the increase in K-value, both B_{max} and A_{max} values decrease, and I_{comp} and A_{FS} increase indicating the extra penalty.

One of the physical constraints for the operation of a superconducting circuit is the maximum amount of bias current that can be provided from the room temperature to the superconducting temperature. Based on this physical constraint, a limit has to be placed on the B_{max} value of the partitions. We have provided the results for the partition of the same benchmark circuits of Table I with a limit of $B_{max} = 100$ mA. This limit corresponds to the maximum current that a bias pad can sustain in a typical superconducting chip [23]. Table III presents the result of number of partitions (K_{res}) along with the lower bound on it (K_{LB}) which is calculated as $\lfloor \frac{B_{cir}}{B_{max}} \rfloor$. With the increase in the complexity of the circuit, the difference between K_{res} and K_{LB} also increases. With a large number of partitions, the values of I_{comp} and A_{FS} are also larger

which can be seen as in the results of circuits ID8 and C3540. [23] uses 31 bias lines to provide 2.5 A of current to the demonstrated chip. If we use the presented partition results for the same chip, we can save 30 bias lines at the expense of an unused area of approximately A_{FS} for the circuits in the benchmark suite.

TABLE III
PARTITION RESULTS FOR 100 MA OF MAXIMUM SUPPLIED CURRENT

Circuits	$\frac{K_{LB}}{K_{res}}$	$d \leq \lfloor \frac{K}{2} \rfloor$	B_{max} (mA)	I_{comp} (%)	A_{max} (mm^2)	A_{FS} (%)
KSA8	3 / 3	95.9%	78.31	8.40	0.4476	10.14
KSA16	6 / 7	84.9%	93.37	17.20	0.5208	16.13
KSA32	14 / 17	77.4%	99.98	24.74	0.5628	24.58
MULT4	3 / 3	91.0%	79.34	7.20	0.4404	8.37
MULT8	13 / 15	77.5%	96.78	20.87	0.5340	21.45
ID4	5 / 6	92.6%	87.38	11.55	0.4944	10.70
ID8	28 / 40	75.3%	99.65	43.17	0.5580	43.63
C432	11 / 14	83.0%	87.15	16.73	0.5040	18.69
C499	9 / 11	79.6%	91.42	20.44	0.5340	22.22
C1355	9 / 11	80.7%	96.77	20.51	0.5628	21.85
C1908	15 / 17	78.2%	97.78	14.88	0.5628	15.92
C3540	32 / 50	77.1%	92.61	45.01	0.5400	45.51

VI. CONCLUSIONS

This work presents the first-ever attempt at partitioning a superconducting circuit into different ground planes for implementing the current recycling technique. All the requirements and conditions for a ground plane partitioning have been identified. All the constraints were successfully converted into a mathematical formulation and used a cost function that is a linear combination of all the constraints. We have also successfully converted a large number of equality conditions and indicator functions into one cost function so that the method of Lagrange multipliers can be applied along with converting an integer programming problem into a linear programming formulation. Gradient descent algorithm is used on our final cost function to successfully partition any given circuit into a given number of ground planes. The results are presented for different benchmark circuits showing the performance of the algorithm in terms of the number of interconnections between different planes, the required bias current compensation and the estimated free space on-chip after the partition.

ACKNOWLEDGEMENTS

This work was supported by the Office of the Director of National Intelligence (ODNI), the Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office Grant W911NF-17-1-0120.

REFERENCES

[1] A. I. Braginski, "Superconductor electronics: status and outlook," *Journal of Superconductivity and Novel Magnetism*, vol. 32, no. 1, pp. 23–44, 2019.

[2] W. Chen, A. Rylyakov, V. Patel, J. Lukens, and K. Likharev, "Rapid single flux quantum T-flip flop operating up to 770 GHz," *IEEE Transactions on Applied Superconductivity*, vol. 9, no. 2, pp. 3212–3215, 1999.

[3] T. V. Filippov, A. Sahu, A. F. Kirichenko, I. V. Vernik, M. Doronjevs, C. L. Ayala, and O. A. Mukhanov, "20GHz Operation of an Asynchronous Wave-Pipelined RSFQ Arithmetic-Logic Unit," *Physics Procedia*, vol. 36, pp. 59–65, 2012.

[4] R. Sato, Y. Hatanaka, Y. Ando, M. Tanaka, A. Fujimaki, K. Takagi, and N. Takagi, "High-Speed Operation of Random-Access-Memory-Embedded Microprocessor With Minimal Instruction Set Architecture Based on Rapid Single-Flux-Quantum Logic," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–5, 2017.

[5] A. M. Kadin, R. J. Webber, and D. Gupta, "Current leads and optimized thermal packaging for superconducting systems on multistage cryocoolers," *IEEE Transactions on Applied Superconductivity*, vol. 17, no. 2, pp. 975–978, 2007.

[6] J. Kang and S. Kaplan, "Current recycling and SFQ signal transfer in large scale RSFQ circuits," *IEEE transactions on applied superconductivity*, vol. 13, no. 2, pp. 547–550, 2003.

[7] H. Hayakawa, N. Yoshikawa, S. Yorozu, and A. Fujimaki, "Superconducting digital electronics," *Proceedings of the IEEE*, vol. 92, no. 10, pp. 1549–1563, 2004.

[8] N. K. Katam and M. Pedram, "Logic optimization, complex cell design, and retiming of single flux quantum circuits," *IEEE Transactions on Applied Superconductivity*, vol. 28, no. 7, pp. 1–9, 2018.

[9] N. Katam, A. Shafaei, and M. Pedram, "Design of multiple fanout clock distribution network for rapid single flux quantum technology," in *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*. IEEE, 2017, pp. 384–389.

[10] K. Gaj, E. G. Friedman, and M. J. Feldman, "Timing of multi-gigahertz rapid single flux quantum digital circuits," in *High Performance Clock Distribution Networks*. Springer, 1997, pp. 135–164.

[11] P. Seidel, *Applied Superconductivity: Handbook on Devices and Applications*. John Wiley & Sons, 2015.

[12] N. K. Katam, J. Kawa, and M. Pedram, "Challenges and the status of superconducting single flux quantum technology," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1781–1787.

[13] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," *IEEE Transactions on Applied Superconductivity*, vol. 1, no. 1, pp. 3–28, 1991.

[14] O. A. Mukhanov, "Energy-efficient single flux quantum technology," *IEEE Transactions on Applied Superconductivity*, vol. 21, no. 3, pp. 760–769, 2011.

[15] N. K. Katam, O. Mukhanov, and M. Pedram, "Simulation analysis and energy-saving techniques for ERSFQ circuits," *IEEE Transactions on Applied Superconductivity*, vol. 29, no. 5, pp. 1–7, 2019.

[16] M. W. Johnson, Q. P. Herr, D. J. Durand, and L. A. Abelson, "Differential SFQ transmission using either inductive or capacitive coupling," *IEEE transactions on applied superconductivity*, vol. 13, no. 2, pp. 507–510, 2003.

[17] N. K. Katam, O. A. Mukhanov, and M. Pedram, "Superconducting magnetic field programmable gate array," *IEEE Transactions on Applied Superconductivity*, vol. 28, no. 2, Art. No. 1300212, 2018.

[18] G. Karypis and V. Kumar, "Multilevelk-way partitioning scheme for irregular graphs," *Journal of Parallel and Distributed computing*, vol. 48, no. 1, pp. 96–129, 1998.

[19] I. Mondal, A. Neogi, P. Chaporkar, and A. Karandikar, "Bipartite graph based proportional fair resource allocation for d2d communication," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2017, pp. 1–6.

[20] N. Katam, S. N. Shahsavani, T.-R. Lin, G. Pasandi, A. Shafaei, and M. Pedram, "Sport lab SFQ logic circuit benchmark suite," *Univ. Southern California, Los Angeles, CA, USA, Tech. Rep.*, 2017.

[21] M. A. et al., "Progress towards an open-source front-end cad flow for dc-biased sfq logic circuits," in *Superconductive Electronics Conference (ISEC), 2019 17th International*. IEEE, 2019.

[22] Cadence, "LEF/DEF language reference - international symposium on physical design," <http://www.ispd.cc/contests/18/lefdefref.pdf>, Nov 2009.

[23] T. Ono, H. Suzuki, Y. Yamanashi, and N. Yoshikawa, "Design and implementation of an SFQ-based single-chip FFT processor," *IEEE Transactions on Applied Superconductivity*, vol. 27, no. 4, pp. 1–5, 2017.