Walking Pads: Fast Power-Supply Pad-Placement Optimization

Ke Wang*, Brett H. Meyer†, Runjie Zhang*, Kevin Skadron*, and Mircea Stan‡

*Dept. of Computer Science
University of Virginia
Charlottesville, VA, 22904, USA
{kewang, runjie}@virginia.edu,
skadron@cs.virginia.edu

†Dept. of Elec. and Comp. Engineering
McGill University
Montréal, Québec, H3A 0E9, Canada
brett.meyer@mcgill.ca

‡Dept. of Elec. and Comp. Engineering
University of Virginia
Charlottesville, VA, 22904, USA
mircea@virginia.edu

Abstract— We propose a novel C4 pad placement optimization framework for 2D power delivery grids: Walking Pads (WP). WP optimizes pad locations by moving pads according to the "virtual forces" exerted on them by other pads and current sources in the system. WP algorithms achieve the same IR drop as state-of-the-art techniques, but are up to 634X faster. We further propose an analytical model relating pad count and IR drop for determining the optimal pad count for a given IR drop budget.

I. Introduction

In modern system-on-chip design, supply-voltage-noise induced reliability issues are becoming increasingly challenging due to increasing current density [1]. Among the various sources of voltage noise, IR drop refers to the resistive drop across metal wires in the power delivery network (PDN). Typical design rules tolerate an IR drop ratio no more than 5% of supply voltage; violations can lead to timing errors.

In a flip-chip design, because the underlying silicon chip has a non-uniform power dissipation, the number and locations of controlled-collapse-chip-connection (C4) pads connecting to the on-chip PDN have a large impact on IR drop. Thus optimizing both the number and location of power supply C4 pads becomes critical to guarantee the desired IR drop target. Moreover, given the fact that both power supply and signal I/O share the same physical interface—C4 pads—determining the minimum number of power pads required for a given chip design through such optimization can help a designer to determine the available I/O bandwidth, or even perform tradeoffs between I/O bandwidth and the IR drop target.

Previous works have addressed pad placement optimization for the purpose of minimizing IR drop [2, 3, 4]. However, their approaches have scalability limitations, and as a result are not suitable for the large pad placement design space of modern systems. Some other works provide analytical methods to estimate max IR drop when pad number and pad locations are given [5, 6]. To the best of our knowledge, no existing work investigates the minimum number of C4 pads required to satisfy a target IR drop in a 2D PDN grid.

In this paper, we propose a fast method to obtain the minimum pad number for a target IR drop and corresponding optimized pad locations. First, we introduce a new method of power pad placement optimization, *Walking Pads* (WP). The key idea behind WP is to convert a global optimization problem, the placement of n pads given m candidate locations, into a local balance problem, the placement of individual pads (current sources) with respect to various nearby current demands. Treating pads as "mobile positive charges" and the on-chip PDN grid as a 2D electrostatic voltage field, WP optimizes pad location by letting pads "walk" in the direction of the total virtual force exerted upon them to achieve local force balance.

WP achieves significant speedup over existing methods in the literature because it has two significant advantages:

1. WP leverages the underlying voltage gradients to quickly identify promising pad locations.
2. WP allows all pads to step toward their balanced positions simultaneously, reducing algorithm complexity significantly as a function of target pad count.

Second, we derive an analytical formula to describe the relationship between IR drop and pad number based on optimized pad locations. While not a closed-form model, our analytical formula only requires that three coefficients be fit to a curve, and can identify the optimal pad count to within two pads for systems with 128-1024 pads. When combined with WP, our analytical formula can quickly and accurately predict the minimum required pad count.

This paper makes two principal contributions:

1. We propose WP and demonstrate that it achieves at least 100X speedup with respect to the classical simulated annealing (SA) methods in the literature, while sacrificing no more than 0.1% VDD in steady-state IR drop.
2. We propose an analytical formula that describes the relationship between the number of pads and the expected maximum IR drop assuming optimized pad locations.

Together, the analytical model and WP algorithm are positioned to significantly accelerate the optimization of power pad count and placement, and therefore create new opportunities for joint optimization.

II. Related Work

Sato et al. proposed the Successive Pad Assignment (SPA) method of power pad location optimization for pad ring allocation [3]. Zhao et al. provided a solution of mixed integer linear program (MILP) for pad ring allocation [2]. The computational complexities of both SPA and MILP grow quickly as problem size increases. As a result, they are not tractable for large scale 2D C4 arrays. Zhong and Wong proposed a fast power pad placement optimization algorithm within the framework of simulated annealing (SA) [4]. This method localizes the ef-

fect of pad movement using a node-based iterative method and therefore improves the performance of each SA iteration. However, the localization is based on the hypothesis that the voltages of pad-PDN connection points cannot affect each other. This is not true when the package circuit and pad resistance are considered. Furthermore, their approach sacrifices accuracy when accelerating calculations [4], and cannot work with other efficient numerical methods like preconditioned Krylov subspace methods [7].

Shakeri proposed a theoretical method of accurate IR drop estimation for uniform power consumption floorplans with uniformly distributed pads [5]. Rius extended this work to a closed-form expression for non-uniform power consumption floorplans with arbitrary pad counts and locations [6]. However, Rius' work is based on the assumption that power pads are uniformly distributed on a rectangular 2D array. As shown in Section VII, IR drop is systematically overestimated in this case relative to the expected IR drop of optimally placed pads.

Walking Pads and the analytical model we have developed, unlike any prior work, enable designers to efficiently determine the relationship between pad count and IR drop, and therefore optimal pad allocation. Such an approach is critical for pre-RTL design, as the number of pads required for power delivery affects the number of pads available for I/O, and therefore has implications for system architecture and microarchitecture.

III. Problem Formulation

A. Power Delivery Network Model

The typical regularity of the on-chip PDN's physical structure makes compact PDN modeling feasible. A well accepted methodology models the multi-layer metal stack as a 2D resistor mesh [8]. C4 pads are modeled as individual resistors attached to on-chip grid nodes, and the relative locations of those connection points in the grid represent the actual locations of the C4 pads on the silicon die. Ideal current sources are used to model the load (*i.e.* switching transistors). Off-chip components like the package or printed circuit board (PCB) are lumped into single resistors. To the best of our knowledge, lumped package models are adopted in most current related work. We adopt this methodology and build the model skeleton as in Fig. 1 [9]. We assume the PCB represents an ideal power supply and simultaneously model lumped package resistors, pad resistors and on-chip 2D resistor mesh; the steady state equations we solve therefore capture not only the on-chip 2D resistor mesh, but the package and pad resistors as well, with the latter elements changing as pads move from one candidate location to another.

To solve for voltage and current values in the model circuit, we employ sparse LU decomposition with pivoting, using SuperLU [10]. A direct solver with pivoting is generally considered a numerically stable and accurate method, and protects optimization quality from numerical errors. When implemented using advanced reordering techniques [11], sparse LU reduces memory usage significantly and achieves adequate performance for use in our experiments. It is worth noting that the proposed Walking Pad algorithm framework is a high level optimization framework, is thus not restricted to a particular numerical method, and can therefore take advantage of ongoing advances in numerical methods [7, 12].

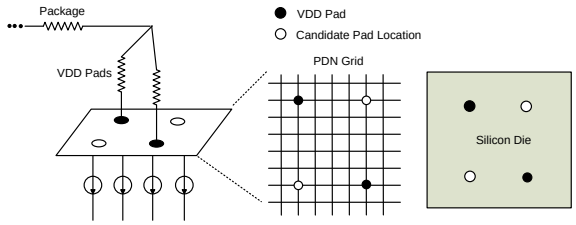


Fig. 1. Model of 2D PDN.

B. Power Pad Location Optimization

Given the system floorplan, the number of power pads to place, and system power trace, the objective of power pad location optimization is to identify grid locations at which to place pads in order to minimize the maximum observed IR drop. The size of C4 bumps restricts the locations where they may be placed. We assume that power pads can be allocated on a coarse pad grid that depends on the ratio of pad pitch and metal pitch. Each possible allocation of power pad to grid locations is called a *configuration*. The total number of configurations is the binomial coefficient of the number of pad locations and number of pads, and is larger than 10^{200} in the case studies considered in this paper (and larger than 10^{1400} for a scaled system in Section VI.B). In this context, effective and computationally efficient search techniques are needed to rapidly identify pad allocations that achieve near-optimal IR drop.

IV. Walking Pads

The key idea behind WP is to convert a global optimization problem, the placement of n pads given m candidate locations, into a local balance problem, the placement of individual pads (current sources) with respect to various nearby current demands. To find the proper virtual force for local balance, we first observe that there is a similarity between the 2D PDN on-chip voltage field and a 2D electrostatic voltage field. The steady state equation of a voltage field can be regarded as the finite-difference version of the 2D Poisson equation [5, 13]:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = I_{xy} R, \tag{1}$$

where V is the on-chip voltage field, I_{xy} is the workload current density at point (x, y) and R is the resistance per unit length in the x and y directions. Gauss's law of electrostatic systems can be similarly described [14]:

$$\frac{\partial^2 \tilde{V}}{\partial x^2} + \frac{\partial^2 \tilde{V}}{\partial y^2} = \frac{\rho_{xy}}{\varepsilon_{xy}}, \tag{2}$$

where $\tilde{V}$ is the electrostatic field, and ρ_{xy} and ε_{xy} are the charge density and permittivity at point (x, y). Note that in this paper we only consider the case where R is the same in the x and y directions; WP algorithms are also suitable if on-chip resistance is anisotropic.

When viewing pad placement as a 2D electrostatic voltage field problem, the current in the PDN is analogous to the electric flux lines in an electrostatic system, which are proportional to the voltage gradient. In this way, power pads can be regarded as "positive point charges" that source currents, and the underlying architectural blocks in the processor system can be regarded as "negative surface charges" that sink currents. Like charges repel each other, while unlike charges attract each other. We therefore define the voltage gradient at a pad location as the virtual force to direct pad movement.

In this context, Walking Pads allows pads to move in reaction to the forces exerted on them by current sources and other pads in the PDN; the pads "walk," toward the locations where these forces balance. No matter where the pads are placed, the total current through all pads is invariant. However, when pads reach their balanced positions, the gradient of the voltage field (directly proportional to the current) in each direction is equalized and reduced. Therefore, IR drop (the integral of voltage gradients) is minimized.

WP also minimizes max on-chip current density and PDN metal power dissipation at the same time. On-chip max current always occurs in those wires directly connected to a pad; max on-chip current density is therefore also minimized by WP because WP minimizes the current through these wires. PDN metal power dissipation is an analogue to the total energy of the electrostatic system. Therefore, the PDN metal power dissipation is also reduced when pads move under virtual forces, and is minimized when all forces on surface charges are balanced.

A. Walking Pads Algorithm Framework

An iteration of a Walking Pads algorithm uses three steps to incrementally move all pads toward their balanced positions:

1. Solve steady state equations.
2. Calculate virtual forces and decide the direction and distance of movement for each pad based on total forces.
3. Move pads.

Grid voltage and current values are determined in step 1. In step 2, current values are used to guide pad movement. Step 3 moves all pads simultaneously. WP achieves a significant performance improvement over SA by employing a deterministic approach to the selection of pad movement direction and distance in step 2 and allowing all pads to move simultaneously in step 3. As more optimization is achieved with each iteration, fewer iterations are needed.

B. Efficient Total Force Calculation

Once steady state current and voltages have been calculated for each node in the PDN, WP must determine in which direction to move each pad by computing virtual forces.

A intuitive way to determine the total virtual force on each pad is to apply the law of superposition and sum the contributions of virtual force from all other pads and current sources together. Some previous work uses this approach [15]. However, such methods are inherently inefficient due to their complexity. Using Gauss's Law, the force on a pad in one direction can be calculated from the voltage gradient in that direction. In the case of 2D PDN, one pad connects to four lines in the east, north, west and south directions. The resultant force is the vector summation of these four currents.

C. Walking Pads Algorithm Variants

We propose three variants of Walking Pads. The first, Walking Pads - Neighbor (WP-N), only allows the pads to move to neighboring locations based on a comparison of the strength of vertical and horizontal forces: the stronger force determines the direction the pad moves, either up/down or left/right. Because all pads move at the same time and traverse a constant distance—one pad candidate location in the direction of motion—this algorithm results in the oscillation of pad locations around balanced positions. In practice, WP-N regards oscillation as convergence: when oscillation is detected, the algorithm terminates. As a result, WP-N does not perform well, but remains useful for quick, but low-quality, optimization.

The second variant, Walking Pads - Freezing (WP-F), is shown in Algorithm 1. WP-F allows pads to move in an arbitrary direction defined by the normalized virtual force $\vec{F} / \left\| \vec{F} \right\|$. Large move distances are also adopted in early iterations. To force pads to stop at approximately balanced positions, we introduce a freezing process which gradually decreases the move distance of each pad. The distance a pad moves D_i decreases with the constant freezing rate γ. WP-F terminates when pads no longer move. The large-step stage of WP-F helps pads to jump out of local minima, while the small-step stage helps pads gradually freeze in their balanced positions.

Set: initial move distance D_0, freezing rate γ
repeat
  Solve steady state;
  foreach *pad* **do**
    $\vec{F} = (I_{north} - I_{south}, I_{east} - I_{west})$
    $\vec{Disp} = \vec{F} / \left\| \vec{F} \right\| * D_i$
  end
  $D_{i+1} = D_i * \gamma$
until *check_converge()* == *True*;

Algorithm 1: Walking Pads - Freezing (WP-F) algorithm.

Walking Pads - Refined (WP-R), is shown in Algorithm 2. The first two versions of WP take advantage of the simultaneous movements of all pads. Simultaneous movements reduce the quality of the solution to some extent, however, because the forces on one pad may change when other pads move. To address this, WP-R performs a greedy search: it moves pads one by one and only accepts movements that decreases the max IR drop. For a 2D grid, we assume that moving pads near the location of max IR drop has greater effect than moving distant ones. To improve efficiency, WP-R sorts the pads by their distances to the max IR drop location and lets nearby pads move first. When the location or the value of maximum IR drop changes, WP-R re-sorts the pads and continues. The algorithm terminates when no pad movement improves IR drop. Because of its algorithm complexity, WP-R is used to supplement WP-F or WP-N to further improve the results when high optimization quality is required.

Set: $D_0 = PadPitch$, initial $maxIRDrop$
repeat
  Sort pads by distance to max IR place → PadList;
  foreach *pad in PadList* **do**
    $\vec{F} = (I_{north} - I_{south}, I_{east} - I_{west})$
    $\vec{Disp} = \vec{F} / \left\|\vec{F}\right\| * D_0$
    Solve steady state and get $new\_maxIRDrop$;
    if $new\_maxIRDrop < maxIRRrop$ **then**
      accept the movement;
      $maxIRRrop = new\_maxIRDrop$; break;
    else
      reject the movement;
    end
  end
until *check_converge() == True*;

Algorithm 2: Walking Pads - Refine (WP-R) algorithm.

D. Algorithm Complexity Analysis

The worst-case complexity of WP algorithms occurs when a pad must move from an initial position in one corner of the chip (e.g., the left-top corner) to a balanced position in the opposite corner (e.g., the right-bottom). In this case, WP-N requires $\#grid_{row} + \#grid_{column} - 2$ iterations to converge. For the practical cases of randomly initialized pad positions, the average number of iterations required is on the order of $B_0(\#grid_{row} + \#grid_{column} - 2)/\#pad$. B_0 is larger than 1 for the case that a pad does not move directly from its initial to the balanced position (i.e., it takes a *detour*).

For WP-F, the convergence speed is controlled by freezing rate γ. The approximate traveling distance of one pad before being frozen is $(D_0 - 0.5pad\_pitch)/(1-\gamma)$, where D_0 is the initial move distance. Again, to beat the worst case, D_0 and γ are chosen to make the travel distance of each pad larger than the diagonal length of the grid. In our experiments, starting from roughly uniform pad locations results in much faster convergence than this theoretical upper bound. Detours are also possible in WP-F. In practice, we add a safety coefficient C_0 in the range of $2.0 \sim 4.0$ to balance the effect of detours and the speedup due to uniform initial positions and get:

$$\frac{D_0 - 0.5pad\_pitch}{1-\gamma} = C_0 * \sqrt{\#grid_{row}^2 + \#grid_{column}^2}. \quad (3)$$

We choose an initial move distance $D_0 = 3 * pad\_pitch$ and freezing rate $\gamma = 0.99$ for our case studies; this results in 180 WP-F iterations. The total number of iterations required is independent of the number of pads to be placed.

V. Experimental Setup

To evaluate our WP algorithms, we compare their convergence speed and solution quality with the simulated annealing (SA) algorithm proposed by Zhong and Wong [4]. For SA, we evaluate two cooling rates, 0.98 (practical cooling speed, SA-P) and 0.999 (very slow cooling speed, SA-S) for efficiency and quality comparison respectively; we have observed that the cooling rate of 0.85 proposed by Zhong and Wong is too fast to produce high-quality results. In our SA implementation, we maximize the square of the worst node voltage and implement the movement window shrinking strategy proposed in the literature [4]. The SA algorithm is considered converged when the movement window is too small for pads to move.

We begin by comparing SA with WP-N and WP-F, and compare SA with WP-F+WP-R then. To compare WP-R and SA, we need to terminate WP-R iteration to get results of similar quality as those from SA, and then compare the speedup. To compare with SA-P and SA-S respectively, WP-F+WP-R-T1, terminates after #pad/2 iterations of WP-R, and WP-F+WP-R-T2 terminates after #pad*8 iterations of WP-R. These cutoffs were determined heuristically to yield similar quality.

We select a 24-core, Intel Penryn-like multiprocessor at 16nm technology as the platform to evaluate the above optimization algorithms. To estimate the power consumption for each functional block, we use McPAT, an architecture-level power model [16]. To model the worst-case power dissipation in the system, we assume that each architectural unit dissipates 85% of its max power [17]. We assume a supply voltage of 0.7V; architectural floorplans were generated using an architecture-level tool, ArchFP [18]. We assume that the top metal pitch is $30\mu m$ top layer metal pitch, and that wires in this layer are $6\mu m$ wide and $4\mu m$ thick; this results in a PDN model consisting of a 236 by 296 resistor grid, where each resistor has a resistance of $41m\Omega$. We assume that the C4 pad pitch is 285 μm, resulting in a grid with 2880 pad candidate locations for our 24-core system. According to ITRS projections, C4 pad density will be held constant in the foreseeable future [19]; we adopt the ITRS projection for pad density in our experiments. All our experiments are conducted on an Intel Xeon E5-1650 3.20 GHz CPU with 32 GB memory.

VI. Results

A. WP Speedup and Result Quality

We first compare two basic WP algorithms, WP-N and WP-F, with SA-P; the results of this comparison are illustrated in Fig. 2. Fig. 2 plots algorithm convergence and solution quality for WP-F (dotted line), WP-N (dashed line), and SA-P (solid line) with respect to IR drop, max current density and power consumed in PDN metal; iteration count is plotted on the x axis. We use iteration counts alone to compare the efficiency of each approach because solving for steady state voltage and current values—required by, and equivalent in, each approach—requires over 99.9% of the total time to complete a single iteration in each case. SA, WP-N, WP-F and WP-R have about the same runtime *per iteration* and memory usage (approximately 0.3s and 220MB for the case of 512 pads on 24-core floorplan).

In Fig. 2, VDD pads are initially allocated uniformly to every fourth pad candidate location in the vertical and horizontal directions, representing 180 pads among 2880 pad candidate locations. We summarize the IR drop (IR), max on-chip current density (J), metal power dissipation (P) and required iteration (Iter) for each pad allocation method in Table I.

We observe that uniform pad allocation does not produce good results: SA reduces IR drop by 45% with respect to that from uniform pad location. Furthermore, we observe that all three algorithms jointly optimize all three metrics, if at different rates, and with differing effectiveness. WP-N converges the

TABLE I
COMPARISON OF DIFFERENT ALLOCATION METHODS

Method	IR (% VDD)	J ($10^{10} A/m^2$)	P (W)	Iter
Uniform	12.5	2.246	10.11	–
WP-N	10.2	1.903	8.752	36
WP-F	7.5	1.543	8.365	180
SA-P	6.9	1.530	8.571	28,261

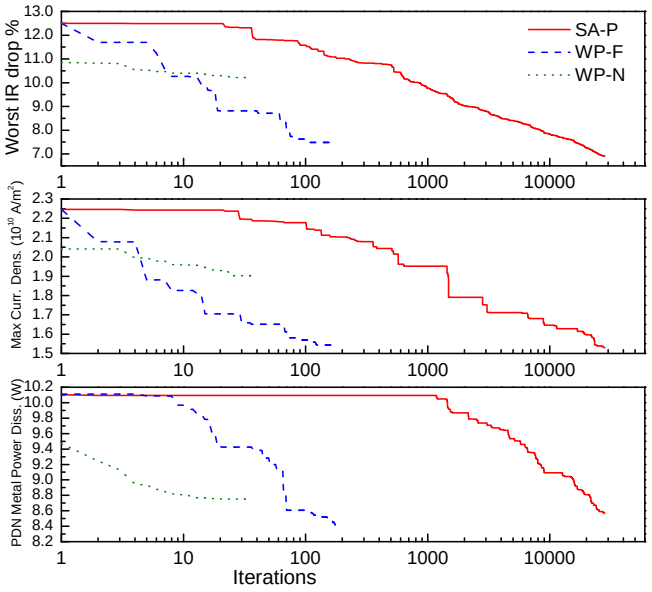


Fig. 2. WP-F (dotted), WP-N (dashed) and SA (solid) all jointly optimize IR drop, max current density and power dissipated in on-chip PDN metal, but at different rates and with different effectiveness. In practice, the techniques above do not monotonically improve each figure of merit; for clarity, we plot the results for the best explored configuration so far at a given iteration count.

fastest, finishing in 20% of the time required for WP-F; however, WP-N converges too quickly to get high-quality results, resulting in an IR drop 48% higher than that produced by SA. WP-F only sacrifices 0.6% VDD in IR drop, but obtains a 157X speedup when compared with SA.

We next evaluate the effect of combining WP-F and WP-R to achieve better optimization quality. Fig. 3 plots the IR drop gap and convergence efficiency of WP-F, WP-F+WP-R-T1 (terminates after #pad/2), and SA-P for varying pad counts, relative to the results from SA-S. The pad allocations selected by SA-S are considered the global optimal and are used to evaluate the result quality of other methods. SA-S, which cools at a rate of 0.999 instead of 0.98, needs $3176 \times \#pad$ iterations to converge while SA-P needs $157 \times \#pad$ to converge.

In Table II we summarize the quality and speedup on a 24-core floorplan with 128 to 1024 pads. Four different WP strategies (WP Str.) - WP-F, WP-F+WP-R-T1 (F+R-T1, WP-R-T1 terminates at $\#pad/2$), WP-F+WP-R-T2 (F+R-T2, WP-R-T2 terminates at $\#pad * 8$), and WP-F+WP-R (F+R, no early termination), are investigated. WP-F achieves up to 893X speedup with respect to SA-P, but sacrifices too much quality (0.54 %VDD). When refined with WP-R, WP-F+WP-R-T1

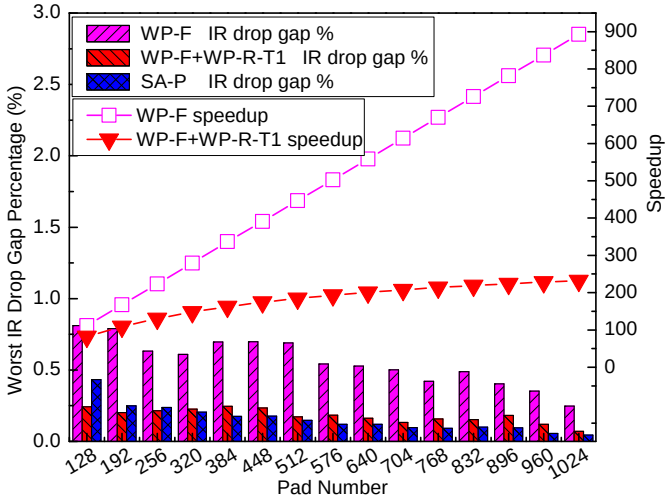


Fig. 3. Comparision of Walking Pads and simulated annealing: differences in worst IR drop and speedup. WP-R-T1 terminates after $\#pad/2$ iterations.

TABLE II
COMPARISON OF DIFFERENT WALKING PADS ALGORITHMS

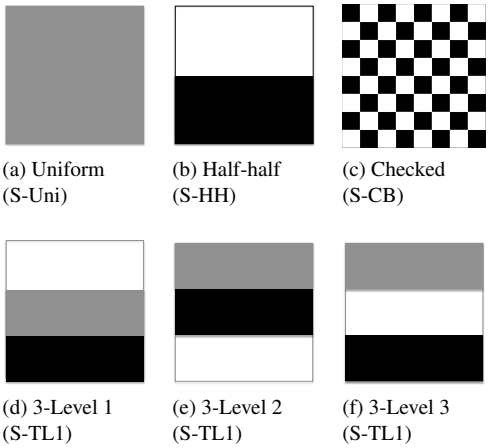
WP Str.	Speedup (X)		Max Gap in %VDD	
	vs SA-P	vs SA-S	vs SA-P	vs SA-S
WP-F	112-893	–	0.54	0.81
F+R-T1	82-232	–	0.09	0.25
F+R-T2	–	337-388	–	0.12
F+R	–	20-220	–	0.10

achieves up to 232X speedup with respect to SA-P, but produces results matching those from SA-P with a gap less than 0.1% VDD. We therefore think WP-F+WP-R-T1 can replace SA-P to obtain optimized pad locations with practical quality. In the case of 832 pads, WP-F+WP-R-T1 requires less than four minutes to achieve results of comparable quality to SA-P after 15 hours. For the same reason, we think WP-F+WP-R-T2 can replace SA-S to obtain intensively optimized pad locations with a speedup in the range of 337-388X. We have not compared WP-F and WP-R-T1 with SA-S and WP-R-T2 and WP-R with SA-P.

B. Synthetic and Scaled System Benchmarks

To demonstrate that WP performs well under a variety of scenarios, we developed a series of benchmarks including (a) six synthetic floorplans (Fig. 4) and (b) three variants of the 24-core system with 16, 32, and 48 cores. Our results are summarized in Table III. For each benchmark (Bench.), we report the number of pads allocated (# pads), the number of candidate locations (# loc), and the corresponding speedup (Speedup) of WP-F (F), WP-F+WP-R-T1 (R-T1) and the IR drop gap (% Gap) of WP-F (F), WP-F+WP-R-T1 (R-T1) and WP-F+WP-R (R), each relative to SA-P. The IR drop gap between SA-P and WP is calculated as $(IR_{WP} - IR_{SA\text{-}P})/VDD$. A negative gap means WP outperforms SA-P.

For the synthetic benchmarks, we observe that WP-F and



(a) Uniform (S-Uni) (b) Half-half (S-HH) (c) Checked (S-CB)

(d) 3-Level 1 (S-TL1) (e) 3-Level 2 (S-TL1) (f) 3-Level 3 (S-TL1)

Fig. 4. The floorplan of each synthetic model is 20 x 20 mm^2. 512 pads are allocated to deliver a total of 150 W. In (b), the power density ratio of black to white is 4:1. In (d), (e) and (f) the power density ratio of black, gray and white is 3:2:1.

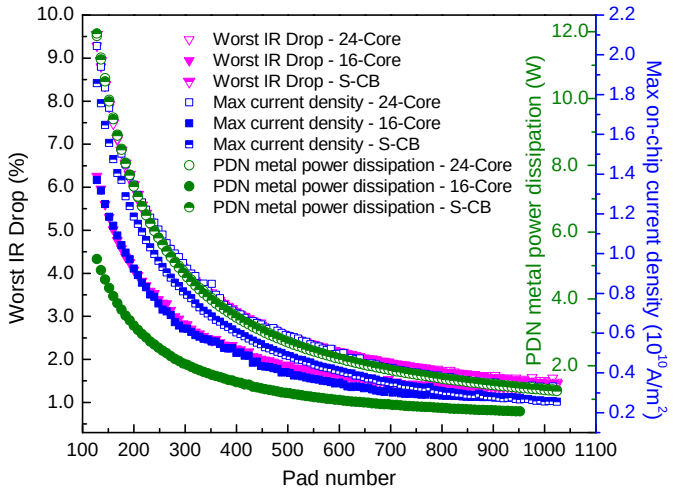


Fig. 5. Pad number effect on IR drop, max current density and PDN metal power dissipation based on optimized pad locations. Optimization uses WP-F+WP-R (no early termination) and starts from randomly allocated pads. Points are plotted at a interval of 8 pads in this figure.

TABLE III
WP RESULTS FOR SYNTHETIC AND MULTI-CORE MODELS

Bench.	# pads	# loc	Speedup		% Gap		
			F	R-T1	F	R-T1	R
S-Uni	512	4900	498	206	0.18	-0.03	-0.11
S-HH	512	4900	498	206	0.23	-0.03	-0.11
S-CB	512	4900	498	206	0.20	-0.06	-0.12
S-TL1	512	4900	498	206	0.15	-0.03	-0.10
S-TL2	512	4900	498	206	0.24	0.01	-0.12
S-TL3	512	4900	498	206	0.19	-0.03	-0.11
16-Core	512	1914	375	155	0.42	0.16	-0.07
24-Core	768	2880	670	277	0.33	0.071	-0.04
32-Core	1024	3844	961	397	0.41	0.070	-0.07
48-Core	1536	5776	1536	634	0.39	0.055	-0.09

WP-R-T1 achieve a speedup of 498 and 206X relative to SA-P. WP-F and WP-R-T1 further achieve IR drops within 0.25% and 0.01% of SA-P. For the Penryn-like variants, the speedup advantage of WP-F and WP-R-T1 increases as the chip grows, up to 634X, and the IR drop gap for WP-R-T1 shrinks marginally; the IR drop gap for WP-F is relatively constant across chip sizes.

VII. ANALYTICAL MODEL

While the above results show that WP efficiently places a given number of pads, naively determining the appropriate pad count to meet a given IR drop budget requires many WP executions, one for each pad count. We therefore developed an analytical model capable of predicting the appropriate pad count, significantly reducing the number of required WP executions.

Fig. 5 illustrates the relationship between pad count, IR drop, max current density and PDN metal power when pad locations are optimized with WP-R. As the pad count increases, each of the three metrics decreases in a similar way.

To model the relationship between pad count and IR drop, we begin with several simplifying assumptions:

1. The load current density ρ is uniform.
2. All pad currents are equal.
3. Each pad serves a circular area around it with radius r_0.

From Gauss's Law we have:

$$\frac{\partial V}{\partial r} = \frac{\pi r_0^2 \rho - \pi r^2 \rho}{2\pi r} * R. \quad (4)$$

Integrating V from (r_ε) to r_0, the IR drop at r_0 is:

$$V|_{r_0} = \frac{\rho R r_0^2}{2} ln\frac{r_0}{r_\varepsilon} - \frac{\rho R}{4}(r_0^2 - r_\varepsilon^2) + \frac{I_0 R_p}{N_p} + V_{packagedrop}. \quad (5)$$

where r_ε is the effective radius of pad, and R is the resistance per unit length of on-chip resistor grid. Substituting $r_0 = \sqrt{\frac{I_0}{\pi \rho N_p}}$ and substituting for the constant coefficients with a, b, and c, we have:

$$V_{drop} = a\frac{1}{N_p} log(\frac{1}{N_p}) + b\frac{1}{N_p} + c. \quad (6)$$

To validate Eq. (6), we performed curve fitting against the IR drop data in Fig. 5, and find that $R^2 = 0.998$ and 0.9998 for the 16-core and 24-core models respectively. Furthermore, when used to derive max on-chip current density and PDN metal power, fitting Eq. (6) results in $R^2 = 0.998$ and 0.9999 respectively for the 16-core model, and $R^2 = 0.9995$ and 0.99997 respectively for the 24-core model. Eq. (6) clearly is effective at predicting each metric as a function of pad count.

To explore the predictive power of our analytical model, we select four different IR drop budgets for the 24-core system, use Eq. (6) to estimate the appropriate number of pads, and compare this with the minimum pad count satisfying the budget. The parameters of Eq. (6) are fitted using three randomly selected pad counts: 200, 520, and 840. The results of this experiment are summarized in Table IV. We observe that the predicted pad count (Pred.) is within two of the optimal pad count (Optimal) in each case. It is worth noting that even if all

TABLE IV
PREDICTED AND OPTIMAL PAD COUNT FOR 24-CORE MODEL

IR Drop Budget	Pred.	Optimal	Actual IR Drop
5%, 35mV	240	238	34.63mV
4%, 28mV	304	306	27.97mV
3%, 21mV	416	418	20.77mV
2%, 14mV	673	672	13.99mV

pad counts in $\{128, 136, ..., 1024\}$ are used for curve fitting, the predicted number of pads does not change.

While validating our analytical model, we noticed that there is a significant difference between the worst-case IR drop experienced under uniform pad distribution and that experienced when pad locations are optimized. For example, the worst IR drops with uniform pads allocations on a rectangular 2D array are 12.0%, 7.0% and 3.3% for the cases of 180, 320 and 720 VDD pads in our 24-core model. The corresponding worst IR drops with WP-optimized pad allocations are 6.6%, 3.8% and 1.9% respectively. This suggests that previous analytical models based on uniform pad allocations (e.g., [6]) systematically overestimate worst-case IR drop.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper we describe a fast method for determining the minimum number of pads required to satisfy an IR drop constraint and their corresponding optimized locations. We introduce a novel pad placement optimization framework for 2D grids: Walking Pads (WP). Three algorithms are proposed in the WP framework to meet the conflicting requirements of results quality and optimization time. The experimental results show that combining the Walking Pads - Freezing (WP-F) and Walking Pads - Refined (WP-R) algorithms achieves up to 634X speedup when compared with simulated annealing (SA), without sacrificing more than 0.1% VDD in IR drop. Our scalability test also shows that speedup and result quality of WP increase as the chip grows. We also propose an analytical model to describe the relationship between the number of allocated, optimized pads and resulting IR drop. This model matches WP results well and leads to fast minimum-pad-number determination when working with WP algorithms.

In this paper we take the first step of demonstrating the viability of the WP paradigm. There are several directions for future research using the WP framework: (1) The joint optimization of VDD and GND pad placement should be considered to make further IR drop optimization across both the VDD and GND layers; (2) Spatial constraints in the 2D pad candidate location grid should be considered in WP for the placement of signal pads; (3) WP could be used to support IR-drop-aware floorplanning, by moving 'negative charges' (functional units or standard cells) instead of 'positive changes' (power pads); (4) WP algorithms could be simply extended for through-silicon via (TSV) placement in 3D IC; (5) WP algorithms can be easily extended to temperature-aware placement by replacing the voltage field with a temperature field.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants CNS-0916908, CCF-0903471 and CCF-1116673, and C-FAR, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

REFERENCES

[1] Mikhail Popovich, Andrey V. Mezhiba, and Eby G. Friedman. *Power distribution networks with on-chip decoupling capacitors*. Springer, New York; London, 2008.

[2] Min Zhao, Yuhong Fu, Vladimir Zolotov, Savithri Sundareswaran, and Rajendran Panda. Optimal placement of power supply pads and pins. *Proc. DAC '04*, pp. 165–170, New York, NY, USA, 2004. ACM.

[3] T. Sato, Hidetoshi Onodera, and M. Hashimoto. Successive pad assignment algorithm to optimize number and location of power supply pad using incremental matrix inversion. *Proc. ASP-DAC '05*, vol. 2, pp. 723–728, 2005.

[4] Yu Zhong and Martin D. F. Wong. Fast placement optimization of power supply pads. *Proc. ASP-DAC '07*, pp. 763–767, Washington, DC, USA, 2007.

[5] K. Shakeri and J.D. Meindl. Compact physical IR-drop models for chip/package co-design of gigascale integration (GSI). *IEEE Transactions on Electron Devices*, vol. 52(6), pp. 1087–1096, 2005.

[6] J. Rius. IR-Drop in on-chip power distribution networks of ICs with nonuniform power consumption. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21(3), pp. 512–522, 2013.

[7] Jianlei Yang, Zuowei Li, Yici Cai, and Qiang Zhou. PowerRush: a linear simulator for power grid. In *ICCAD '11*, pp. 482–487, 2011.

[8] Meeta S. Gupta, Jarod L. Oatley, Russ Joseph, Gu-Yeon Wei, and David M. Brooks. Understanding voltage variations in chip multiprocessors using a distributed power-delivery network. In *Proc. DATE '07*, pp. 624–629, San Jose, CA, USA, 2007.

[9] Runjie Zhang, Brett H. Meyer, Wei Huang, Kevin Skadron, and Mircea R. Stan. Some limits of power delivery in the multicore era. *WEED*, Oregon, USA, 2012.

[10] Xiaoye S. Li. An overview of SuperLU: algorithms, implementation, and user interface. *ACM Trans. Math. Softw.*, vol. 31(3), pp. 302–325, 2005.

[11] Joseph W. H. Liu. Modification of the minimum-degree algorithm by multiple elimination. *ACM Trans. Math. Softw.*, vol. 11(2), pp. 141–153, 1985.

[12] Zhuo Li, Raju Balasubramanian, Frank Liu, and Sani Nassif. 2011 TAU power grid simulation contest: benchmark suite and results. In *Proc. ICCAD '11*, pp. 478–481, Piscataway, NJ, USA, 2011.

[13] Andrew B. Kahng, Bao Liu, and Qinke Wang. Stochastic power/ground supply voltage prediction and optimization via analytical placement. *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 15(8), pp. 904–912, 2007.

[14] David J. Griffiths. *Introduction to Electrodynamics*. Addison-Wesley, 4 edition, October 2012.

[15] Yi-Lin Chuang, Po-Wei Lee, and Yao-Wen Chang. Voltage-drop aware analytical placement by global power spreading for mixed-size circuit designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(11):1649–1662, 2011.

[16] Sheng Li, Jung-Ho Ahn, R.D. Strong, J.B. Brockman, D.M. Tullsen, and N.P. Jouppi. McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. *MICRO-42*, pp. 469–480, 2009.

[17] A.M. Joshi, L. Eeckhout, L.K. John, and C. Isen. Automated microprocessor stressmark generation. *HPCA 2008*, pp. 229–239, 2008.

[18] Gregory G. Faust, Runjie Zhang, Kevin Skadron, Mircea R. Stan, and Brett H. Meyer. ArchFP: rapid prototyping of pre-RTL floorplans. *VLSI-SoC*, pp. 183–188. IEEE, 2012.

[19] ITRS, 2011. http://www.itrs.net.