SIMULTANEOUS DELAY AND POWER OPTIMIZATION IN GLOBAL PLACEMENT

Mongkol Ekpanyapong, Karthik Balakrishnan, Vidit Nanda, and Sung Kyu Lim
School of Electrical and Computer Engineering, Georgia Institute of Technology
{pop,gte245v,gte272u,limsk}@ece.gatech.edu

ABSTRACT

Delay and power minimization are two important objectives in the current circuit designs. Retiming is a very effective way for delay optimization for sequential circuits. In this paper we propose a framework for multi-level global placement with retiming, targeting simultaneous delay and power optimization. We propose GEO-P for power optimization and GEO-PD algorithm for simultaneous delay and power optimization and provide smooth wirelength, power and delay tradeoff. In GEO-PD, we use retiming based timing analysis and visible power analysis to identify timing and power critical nets and assign proper weights to them to guide the multi-level optimization process. We show an effective way to translate the timing and power analysis results from the original netlist to a coarsened sub-netlist for effective multi-level delay and power optimization. Our GEO-P achieves 27% average power improvement and our GEO-PD provides gains in both delay and power improvement. To the best of our knowledge, this is the first paper addressing simultaneous delay and power optimization in multi-level global placement.

1. INTRODUCTION

Delay minimization and power minimization are two important objectives in the design of the high-performance, portable, and wireless computing and communication systems. Thus, a considerable research effort has been made in trying to find power and delay-efficient solutions to circuit design problems. One such procedure that is applied at the logic level is circuit placement.

The placement problem for a given sequential netlist involves global placement and detailed placement. Global placement identifies the partition block-level location for cells, whereas detailed placement provides complete location information for each cell while preserving the global placement. Recently, global placement has attracted significant attention due to tighter circuit constraints and increasing complexities. There are three major approaches to global placement: min-cut based algorithms, analytical approaches, and simulated annealing techniques. The min-cut based approach uses top-down methods to recursively partition a circuit into smaller sub-netlists. Due to the high flexibility and small runtime of this approach, it has been adopted in many modern state-of-the-art placement algorithms.

In this paper we propose a framework for mincut-based global placement with retiming, simultaneously optimizing delay and power. We first discuss the importance of retiming delay and visible power as opposed to the conventional static delay and total power for sequential circuits. Then we propose GEO-P, the modified version of GEO targeting power optimization. We use visible power analysis to guide the partitioner to group gates such that long wires are not driven by the gates with high switching activity. We also propose GEO-PD algorithm for simultaneous delay and power optimization. In GEO-PD, we use retiming based timing analysis and visible power analysis to identify timing and power critical nets and assign proper weights to them to guide the multi-level optimization process. In general, timing and power analysis are done at the original netlist while a recursive multi-level approach performs partitioning and placement on the sub-netlist as well as its coarsened representations. We show an effective way to translate the timing and power analysis results from the original netlist to a coarsened sub-netlist for effective multi-level delay and power optimization.

The organization of this paper is as follows. Section 2 describes problem formulation. Section 3 is devoted to our algorithm. Section 4 presents our experimental result and analysis. Finally, the last section presents our conclusions.

2. PROBLEM FORMULATION

Given a sequential gate-level netlist $NL(C, N)$, where C is the set of cells representing gates and flip-flops, and N is the set of nets connecting the cells, the purpose of the Performance driven Global Placement with Retiming (PGPR) problem is to assign cells in NL to $m \times n$ ($= K$) blocks while area constraint for each block is satisfied. In other words, the placement region is divided into $m \times n$ tiles, and we perform cell placement at the center of these tiles. Given a PGPR solution B, let $\omega(B)$ and $\phi(B)$ respectively denote the wirelength and retiming delay. The formal definition of PGPR is as follows:

PGPR Problem: the Performance driven Global Placement with Retiming (PGPR) problem under the given area constraints $A = (L_i, U_i)$ has a solution P: $C \rightarrow B$, wherein each cell in C is assigned to a unique block, where $B = \{B_1(x_1,y_1), B_2(x_2,y_2),..., B_K(x_K,y_K)\}$ denotes the set of blocks and (x_i,y_i) represents the geometric location of B_i. B is feasible if it satisfies the following conditions: i) $B_i \subset C$, $1 \le i \le K$, ii) $L_i \le |B_i| \le U_i$, $1 \le i \le K$, iii) $B_1 \cup B_2 \cup ... \cup B_K = C$, iv) $B_i \cap B_j = \varnothing$ for all $i \ne j$. The objective is to minimize $\phi(B)$ while maintaining an acceptable $\omega(B)$.

2.1. Delay Objective

By employing the concept of retiming graph, we model NL using a directed graph $R = (V, E)$. Each vertex v has delay $d(v)$ and each edge $e=(u,v)$ has delay $d(e)$. We assume $d(e)$ is proportional to the Manhattan distance between u and v. The edge weight $w(e)$ of $e=(u,v)$ denotes the number of flip-flops between gate u and v. The path weight can be calculated by $w(p)=\sum_{e \in p} w(e)$. Let $w^r(e)$ denote edge weight after retiming r, i.e. number of flip-flops on the edge after retiming. Then, $w^r(p)=\sum_{e \in p} w^r(e)$. A circuit is retimed to a delay ϕ by a retiming r if the following conditions are satisfies; (i) $w^r(e) \ge 0$ for each e, (ii) $w^r(p) \ge 1$ for each path p such

that $d(p) > \phi$. We define the edge length of $e=(u,v)$ as $l(e)=-\phi w(e)+d(v)+d(e)$, and the path length of p as $l(p)= \sum_{e\in p} l(e)$. The *sequential arrival time* [3] of vertex v, denote $l(v)$, is maximum path length from PIs or FFs to v. If the sequential arrival time of all POs or FFs are less than or equal to ϕ, the target delay ϕ is called *feasible*. Let $q(e)=\phi w(e)-d(u)-d(e)$ be the required edge length of e. The required path length $q(p)= \sum_{e\in p} q(e)$. The *sequential required time* of vertex v, denote $q(v)$ is the minimum required path length from v to POs or FFs, when $q(\text{PO})$ or $q(\text{FF}) = \phi$. Then slack of v is given by $q(v)-l(v)$. Let D_g be the maximum $d(v)$ among all v in V. Then, the *retiming delay* $\phi(B)$ of a PGPR solution B is the minimum feasible $\phi + D_g$.

2.2. Wirelength Objective

We model netlist NL using a hypergraph $H=(V, E_H)$, where the vertex set V represents cells, and the hyperedge set E_H represents nets in NL. Each hyperedge is a non-empty subset of V. The x-span of hyperedge h, denoted h_x, is defined as $h_x = \max_{c\in h}\{x_i|c\in B_i\} - \min_{c\in h}\{x_i|c\in B_i\}$. The y-span, denoted h_y, is calculated using the y-coordinates. The sum of x-span and y-span of each hyperedge h is the half-parameter of the bounding block (HPBB) of h and denoted $HPBB(h)$. The *wirelength* $\omega(B)$ of global placement solution B is the sum of HPBB of all hyperedges in H.

2.3. Power Objective

For power objective, we model NL as hypergraph $H=(V, E_H)$ as discussed in Section 2.2. Let V_{dd} denote the supply voltage, f is the global clock frequency, $C_g(v)$ and $C_w(v)$ represent the gate capacitance and wire capacitance seen by gate v, and $SA(v)$ is switching activity of v. $C_g(v)$ is the sum of the input capacitance of all sink gates driven by v. Let n_v denote the net whose driving gate is v. Let VG be the set of *visible gates* that is defined as $VG=\{v|s(n_v)=1\}$, if n_v is cut. Then, the *visible power consumption* $\pi(B)$ of global placement solution B is calculated as follows: $P_v=(V^2_{dd}f\cdot\Sigma_{v\in VG}(C_g(v)+C_w(v))\cdot SA(v))/2$. The rationale is that the power consumption by the gate driving a long wire is much larger than that of short wire. We note that $C_w(v)=HPBB(n_v)\cdot C_g(v)$, the wire capacitance $C_w(v)$ is the only factor that changes based on a placement solution. Thus, we attempt to minimize the visible power in our algorithms.

3. METHODOLOGY

3.1. Overview of GEO-PD Algorithm

An overview of the GEO-PD algorithm is shown in Figure 1. GEO-PD is a multi-level global placement for simultaneous delay and power optimization. GEO-PD places the given netlist NL into $K=n\times m$ dimension using a top-down recursive bipartitioning approach. `GEO-PD` consists of two subroutines: `GEO-PD-2way` recursively bipartitions NL, whereas `GEO-PD-Kway` refines these partitioning results occasionally. `GEO-PD-2way` is performed on the sub-netlist, whereas `GEO-PD-Kway` is performed on the entire netlist. Initially, the partitioning tree T has only root node R, and all cells in NL are inserted into R. The FIFO (First In First Out) queue Q is used to support the recursive breadth-first cut sequence.

`GEO-PD-2way` first generates the sub-netlist from the given partition tree node and performs multi-level clustering on it. We use ESC clustering algorithm [1] for this purpose. Then we obtain a random initial partitioning B among the clusters at the top level of the hierarchy. The subsequent top-down multi-level refinement is used to improve B in terms of delay and power. We perform retiming based timing analysis RTA [2] to identify timing critical nets. We also perform power analysis [4] to identify power critical nets. Then we compute the delay and power weights for the nets in the sub-netlist. The subsequent iterative improvement through cluster move tries to minimize the weighted cutsize. Finally we project the current solution to the next level coarser netlist for multi-level optimization. At the end of `GEO-PD-2way`, two new children nodes are inserted into T based on B.

`GEO-PD-Kway` refinement is performed when we obtain 2^j partitions ($j > 1$) from `GEO-PD-2way` (4, 8, 16 partitions, etc). We first perform a restricted multi-level clustering, where grouping among cells in different partition is prohibited. This allows the partitioner to preserve the initial partitioning results. Then we again perform multi-level partitioning in the same way as in `GEO-PD-2way` for additional delay and power improvement. `GEO-PD-Kway` is applied onto the global netlist for more global level optimization.

```
==========================================
GEO-PD(NL,K)
insert all cells in NL to root node R in T
insert R into Q (= FIFO queue)
while (leaf nodes in T < K)
     N = remove front element in Q
     GEO-PD-2way(N) (= bipartitioning on N)
     split cells in N into N1 and N2
     insert N1 and N2 into Q and T
     if (2^j leaf nodes exists in T, j>1)
          GEO-PD-Kway(T)
return T
------------------------------------------
GEO-PD-2way(N)
NL' = sub-netlist containing cells in N
ESC(NL') (= multi-level clustering on NL')
h = height of the cluster hierarchy
B = random partitioning for clusters at h
for (i = h downto 0)
     NL'(i) = coarsened NL' at level i
     while (gain)
          DELAY-WEIGHT(NL'(i))
          POWER-WEIGHT(NL'(i))
          net weight = power + delay weight
          while (gain)
               move cells in NL'(i)
               update B
     project B to level i-1
return B
------------------------------------------
GEO-PD-Kway(T)
B = initial partitioning for NL from T
ESC'(NL) (= restricted clustering)
perform multi-level partitioning
update T
==========================================
```

Figure 1. Overview of the GEO-PD algorithm

3.2. Weight Computation

For simultaneous delay and power optimization, we first identify timing and power critical nets and assign proper weights to them to guide the optimization process. A net is *timing critical* if it lies along a critical path and *power critical* if it has high fanout with large wirelength and is driven by a gate with high switching activity. In GEO-PD, retiming delay and visible power are

minimized through retiming based timing analysis [2] and visible power analysis [4]. We use *sequential slack* to compute how much time slack exists before timing violation occurs after retiming. These values are then used to compute the delay weights of the nets for retiming delay minimization. In case of power optimization, we use switching activity and gate/wire capacitance to compute power weights of the nets for visible power minimization. Both delay and power weights are added together, and GEO-PD performs multi-level partitioning to minimize the total weighted wirelength.

We note that the multi-level approach [1] is very effective in minimizing the weighted cutsize and wirelength. However, timing and power analysis is typically done at the original netlist while a recursive multi-level approach performs partitioning and placement on the sub-netlist as well as its coarsened representations. Thus, it is crucial that we have an effective way to translate the timing and power analysis results from the original netlist to a coarsened sub-netlist.

```
=====================================
DELAY-WEIGHT(NL')
set delay of edges in R (= retiming G)
perform RTA(R) (= timing analysis)
compute sequential slack for nodes in R
for each cluster C in NL'
   C(R) = all cells in R grouped into C
   slack(C) = min among cells in C(R)
X = top x% clusters with small slack
for each net N in NL'
   if (all clusters in N are in X)
      compute delay-weight(N) using Eqn1
-------------------------------------
POWER-WEIGHT(NL')
for each net Nv in NL'
   Nv' = corresponding net in NL
   compute HPBB(Nv')
   compute power-weight(Nv) using Eqn2
=====================================
```

Figure 2. Delay and power weight computation in GEO-PD

3.2.1. Delay Weight Computation

Figure 2 shows `DELAY-WEIGHT(NL')` algorithm. Before we perform retiming based timing analysis (RTA), we initialize the edge delay in *R* (= retiming graph) based on the current placement results. We set the delay of edges to their Manhattan distances. Then, a Bellman-Ford variant RTA is performed from a given feasible delay to compute sequential slack. For each cluster *C* from the given coarsened sub-netlist `NL'`, we compute *C*(*R*), the set of all the nodes in *R* that are grouped into *C*. We use the minimum slack among all cells in *C*(*R*) as the slack for *C*. The reason we use the minimum slack value is since the critical path information is preserved regardless of multi-level clustering results (we have also performed experiments using average slack value instead of minimum. But the minimum slack method generated better delay results).

After the cluster slack computation is finished, we sort the clusters in a non-decreasing order of their slack values. We store the top *x*% (we use 3% in our experiment) into a set *X*. For each net that contains *only* the clusters in *X*, we use the following equation to compute the delay weight:

$$dwgt(n) = \alpha\left(1 - \frac{\min\{slack(v) \mid v \in n\}}{\max\{slack(w) \mid w \in NL'\}}\right)^{p1} \quad (1)$$

This equation gives higher weights to the nets that contain smaller minimum cluster slack, thus giving higher priority to the nets containing more timing critical clusters. For those clusters that is not in top x%, we give dwgt(n) = 0 and performing partitioning using only cutsize weight as in ESC [1]. Instead of requiring *all* clusters in a net to be timing critical, we tried another scheme where we give delay weights to the nets with 2 or more timing critical clusters. Our related experiment indicates that this approach produced worse results. Our extensive experiments indicate that α=25, $p1$=1, and x=3% are an excellent empirical choice.

3.2.2. Power Weight Computation

Figure 2 shows `POWER-WEIGHT(NL')`, our power weight calculator. As discussed earlier in Section 2.3, our goal is to minimize visible power consumption i.e. power consumption by the gate driving a long wire (among blocks). Then our goal is to minimize the weighted wirelength. For a net driven by a gate *v*, we use the following equation to assign power weight:

$$pwgt(n_v) = \beta\left(\frac{SA(v)[C_g(v) + C_w(v)]}{\max\{SA(u)[C_g(u) + C_w(u)] \mid u \in V\}}\right)^{p2} \quad (2)$$

where $SA(v)$, $C_g(v)$ and $C_w(v)$ respectively represent the switching activity, gate capacitance and wire capacitance seen by gate *v*. We use $C_w(v) = HPBB(n_v) \cdot Cg(v)$. This equation gives higher weights to the nets that have high fanout, larger wirelength, and source gate with high switching activity. In a multi-level approach, each net in the original netlist *NL* is transformed depending on the given sub-netlist *NL'* and its multi-level clustering information. For example, $n_a = \{a,b,c,d\}$ in *NL* becomes $n_{C1} = \{C1, C2\}$, if *NL'* contains *a* and *b* only and *a* is clustered into *C*1 and *b* into *C*2. In this case, we compute $HPBB(n_a)$ based on the location of *C*1, *C*2, *c*, and *d*, and use $SA(a)$ in our power weight equation. Our extensive experiments indicate that β=25 and $p2$=0.3 are an excellent empirical choice. Since GEO-PD algorithm aims for simultaneously delay and power optimization, by disable retiming analysis and setting delay weight to zero, our algorithm GEO-P can target only for power optimization.

4. EXPERIMENTAL RESULTS

Our algorithms are implemented in C++/STL, compiled with gcc v2.96, and run on Pentium III 746 MHz machine. The benchmark set consists of six big circuits from ISCAS89 [5] and four big circuits from ITC99 [6] suites. We generate random switching activity values for these circuits since such information is not available. The sis package from the university of California at Bekeley can compute the switching activity for sequential circuits, but it takes a prohibited amount of runtime even for a circuit with a few thousand gates. We assume unit delay for all gates in the circuits. Table 1 shows the statistical information of benchmark circuits. We provide the number of gates, PI, PO, and FF for each circuit. Dr and Ds represent the lower bound on retiming delay and static delay, which are calculated by assigning zero delay to all edges and performing retiming and static timing analysis. Gr and Gs represent retiming delay and static delay from our GEO-PD. We note that retiming can improve the delay results significantly. For example, delay can be reduced by 32% from ESC for s38417 with retiming, which makes retiming a very attractive choice for delay optimization. This explains why our

GEO-PD algorithm focuses on retiming delay as opposed to static delay.

Table 1 Benchmark circuit characteristics. Dr and Ds show the lower bound on retiming delay and static delay, and Gr and Gs show the retiming delay and static delay from our GEO-PD.

ckt	gate	FF	Dr	Ds	Gr	Gs
b17o	22854	1414	38	44	61	99
b20o	11979	490	44	74	72	110
b21o	12156	490	43	74	70	113
b22o	17351	703	46	79	76	124
s5378	2828	163	32	33	57	69
s9234	5597	211	39	58	48	95
s13207	8027	669	50	59	91	102
s15850	9786	597	62	82	100	140
s38417	22397	1636	32	47	41	67
s38584	19407	1452	47	56	69	84

We conduct experiments using ESC [1], GEO [2], and our GEO-P and GEO-PD algorithms. ESC is a state-of-the-art cutsize driven multi-level algorithm, and GEO is a state-of-the-art simultaneous cutsize and delay driven multi-level algorithm. GEO-P is obtained by setting delay weights of GEO-PD to zero for power optimization only. Lastly, GEO-PD is a simultaneous power and delay driven multi-level algorithm. We report wirelength, retiming delay, and visible power. Note that the delay and power results are based on block location. We report 8×8 global placement results. We report average improvement ratio normalized comparing with ESC (lower than unity means improvement). We also report the average runtime of each algorithm measured in second.

Table 2 shows the results among ESC, GEO, GEO-P, and GEO-PD. GEO has 10% better retiming delay than ESC at the cost of 16% increase in wirelength. Our GEO-P has 27% better visible power than ESC at the cost of 10% increase in wirelength. Finally, GEO-PD has 5% better retiming delay and 14% better visible power than ESC at the cost of 25% increase in wirelength. GEO-PD improves the retiming delay of s38584 by 21%. The visible power improvement is as much as 31% for s9234. Moreover, the retiming delay and visible power improvement is consistent among all 10 circuits. In overall, GEO-PD reveals a smooth wirelength, delay, and power tradeoff curve and improves both delay and power results of ESC at the cost of increase in wirelength.

5. CONCLUSIONS

To the best of our knowledge, this is the first paper addressing both delay and power optimization in multi-level placement. In addition, we demonstrated the importance of optimizing the retiming delay and visible power as opposed to the conventional static delay and total power. We demonstrated how wirelength has conflicting objectives against power and delay and proposed an effective algorithm GEO-PD for smooth delay, power, and wirelength tradeoff. We also propose GEO-P, which achieve 27% improvement in terms of power.

6. REFERENCE

[1] J. Cong and S. K. Lim, Edge separability based circuit clustering with application to circuit partitioning, *To appear in TCAD*.

[2] J. Cong and S. K. Lim, Physical Planning with Retiming, *In IEEE International Conference in Computer Aided Design*, page 2-7, 2000.

[3] P. Pan, A. K. Karandikar, and C. L. Liu, Optimal clock period clustering for sequential circuits with retiming, *IEEE Trans on Computer-Aided Design, pages 489-498,1998.*

[4] H. Vishnu and M. Pedram, Delay-Optimal Clustering Targeting Low-Power VLSI Circuits., *IEEE Trans on Computer-Aided Design*, page 639-643, 1995.

[5] http://www.cbl.ncsu.edu

[6] http://www.cad.polito.it/tools/9.html

Table 2 Comparison among ESC, GEO, GEO-P, and GEO-PD on 8×8 global placement. Each algorithm reports wirelength, retiming delay, visible power, and runtime.

	ESC			GEO			GEO-P			GEO-PD		
ckt	wire	r-dly	v-pow	wire	r-dly	v-pow	wire	r-dly	v-pow	wire	r-dly	v-pow
b17o	9629	70	5232	10451	63	5697	9982	63	4604	10468	61	4938
b20o	5772	72	3335	6730	79	3660	6450	71	3101	7277	72	3145
b21o	6357	79	3458	6618	65	3468	6703	75	2863	7491	70	3235
b22o	7243	77	4076	7724	69	4473	8570	83	3879	8685	76	4211
s5378	1502	60	384	1462	45	389	1539	57	234	1597	57	269
s9234	1425	50	427	1685	48	476	1510	52	292	1683	48	296
s13207	1525	91	747	1925	77	900	1803	91	536	2367	91	634
s15850	1587	99	584	2085	90	814	1720	96	395	2236	100	517
s38417	2032	41	1158	2695	41	1483	2524	43	963	2819	41	1088
s38584	2973	87	1950	3663	68	2091	3061	79	1619	3546	69	1766
Ratio	1.00	1.00	1.00	1.16	0.90	1.14	1.10	0.98	**0.79**	1.25	**0.95**	**0.88**
Time	104			2231			121			2257		