Timing-Driven Placement Based on Monotone Cell Ordering Constraints

Chanseok Hwang and Massoud Pedram

University of Southern California
Department of Electrical Engineering
Los Angeles CA, USA

Outline

- Timing-Driven Placement
  - Problems & General Methods
- Our Approach
  - Motivations
  - Preferred Signal Directions
  - PSDP Algorithm
- Experimental Results
- Conclusion

Timing-Driven Placement (TDP)

- Goal

  To minimize the circuit delay while obtaining a legal placement solution

- Challenges
  - Increasing dominance of Interconnect delay (50-70% of the longest path delay)
  - Increasing circuit size (>10M gates)

Solution Techniques for TDP

- Path-based Methods
  - Consider input-output paths during the problem formulation
    - Monitoring critical and near-critical paths
  - Maintain accurate timing information during the optimization
  - Suffer from high complexity and low scalability since the number of near-critical paths can become exponentially large

Solution Techniques (Cont'd)

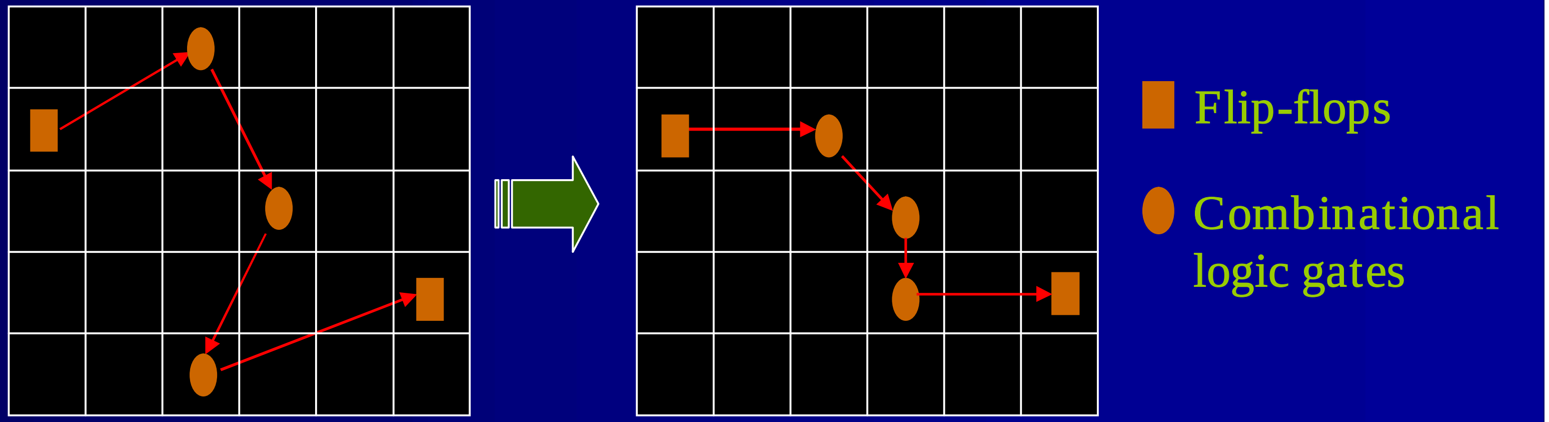
- Net-based Methods
  - Run STA at intermediate steps of the placement process
  - Assign weights (or net length bounds) to timing-critical nets according to their criticalities
  - Convert the TDP problem to a weighted wire length minimization (or bounded wire length) problem
  - Suffer from the difficulty of identifying the proper net weights and tend to exhibit poor convergence for net re-weighting (or result in over-constraining for net length bounding)

PSDP: Preferred Signal Direction Driven Placement

- Starting from an initial placement solution, relying on a move-based optimization strategy, we assign a preferred signal direction to each critical path in the circuit, which in turn encourages the timing-critical cells on that path to move in a direction that would maximize the monotonic behavior of the path in the 2-D placement solution.
- This is based on our observation that most of paths causing timing problems in a circuit meander outside the minimum bounding box of the start and end nodes of the path.

Monotone Cell Ordering

- Cells on a target path do not zigzag or crisscross when the physical path from input to output is traced.
  - – Previously used for wire planning during synthesis and for net list partitioning.



Non-monotone cell ordering

Monotone cell ordering

Related Work on Signal Directions and Monotone Paths

- S. Iman, M. Pedram, C. Fabian, and J. Cong, “Finding uni-directional cuts based on physical partitioning and logic restructuring”, IWPD, 1993.
- W. Gosti, A. Narayan, R. K. Brayton and A. L. Sangivanni-Vincentelli, “Wire planning in Logic Synthesis”, ICCAD, 1998.
- Cong and Lim, “Performance Driven Multi-way Partitioning”, ASP-DAC, 2000.
- A. B. Kahng and X. Xu, “Local Unidirectional Bias for Smooth Cutsize-Delay Tradeoff in Performance-driven bipartitioning.” ISPD, 2003.
- C. Hwang and M. Pedram, “PMP: Performance-driven multilevel partitioning by aggregating the preferred signal directions of I/O conduits”, ASP-DAC, 2005.

Path Grouping and I/O Conduits

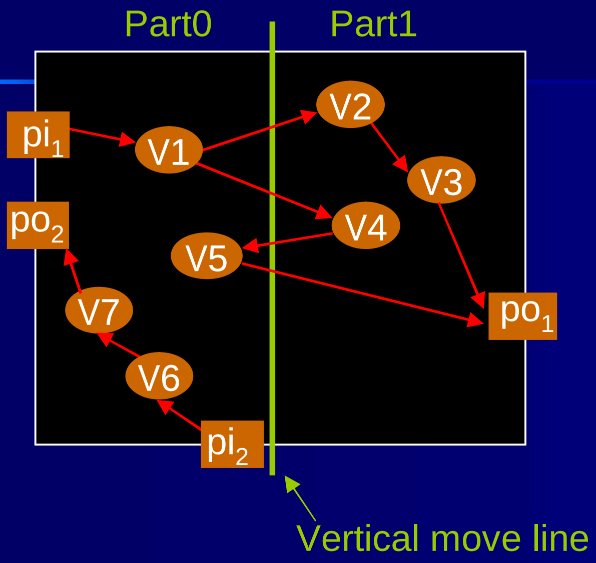
- To make the direction assignment tractable, we implicitly group all circuit paths into a set of input-output conduits and assign a unique preferred direction to each such conduit.
- Definition
  - I/O conduit σ : the set of all paths from some PI (or FF) to some PO (or FF)

$$N_{I/O\ conduits} = (N_{PI} + N_{FF}) * (N_{PO} + N_{FF})$$

Preferred Signal Directions of I/O Conduits

- Definition
  - Preferred signal direction of σ SD(σ): one of the following directions, LL, LR, RL and RR, depending on the locations of PI and PO of σ.
- All paths in σ satisfy the monotone cell ordering property (resulting in minimum wire delay), if the preferred signal direction is satisfied for all edges in the I/O conduit.

Signal Direction Constraints



σ_1 : $pi_1 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow po_1$

$e_1(pi_1,v_1)$, $e_2(v_1,v_2)$, $e_3(v_2,v_3)$, $e_4(v_3,po_1)$

$pi_1 \rightarrow v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow po_1$

$e_1(pi_1,v_1)$, $e_5(v_1,v_2)$, $e_6(v_2,v_3)$, $e_7(v_3,po_1)$

σ_2 : $pi_2 \rightarrow v_6 \rightarrow v_7 \rightarrow po_2$

$e_8(pi_2,v_6)$, $e_9(v_6,v_7)$, $e_{10}(v_7,po_2)$

Signal direction constraints for the vertical move line:

$P(s(e_i)) \leq P(t(e_i))$, $1 \leq i \leq 7$ for σ_1 // $SD(\sigma_1) = LR$

$P(s(e_i)) = P(t(e_i)) = 0$, $8 \leq i \leq 10$ for σ_2 // $SD(\sigma_2) = LL$

$s(e_i)$: Source node of e_i
$t(e_i)$: Target node of e_i
$P(v_i)$: Part number (0 or 1) of v_i

Signal Direction Constraint (Cont'd)

Signal direction constraint for a VERT (HORZ) move line

SDC^1: if $SD(\sigma)$ = LL (BB), $\forall e_i \in \sigma$, $P(s(e_i)) = P(t(e_i)) = 0$
SDC^2: if $SD(\sigma)$ = RR (TT), $\forall e_i \in \sigma$, $P(s(e_i)) = P(t(e_i)) = 1$
SDC^3: if $SD(\sigma)$ = LR (BT), $\forall e_i \in \sigma$, $P(s(e_i)) \leq P(t(e_i))$
SDC^4: if $SD(\sigma)$ = RL (TB), $\forall e_i \in \sigma$, $P(s(e_i)) \geq P(t(e_i))$

- Difficulty
  - No solution that satisfies SDCs of all I/O conduits exists.
  - Increases the total wire length.
- Solution
  - SDC's need to be relaxed.

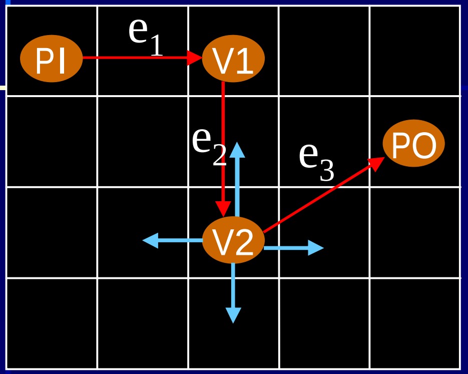
SD Violation Count as the Timing Gain Function for a Cell Move

- We treat delay as an optimization objective instead of a hard constraint to be satisfied, and use the violation count of SDC's.
  - Timing gain function, TG(v_i), is defined to quantitatively evaluate the desirability of moving v_i from part_0 to part_1. It is calculated as:

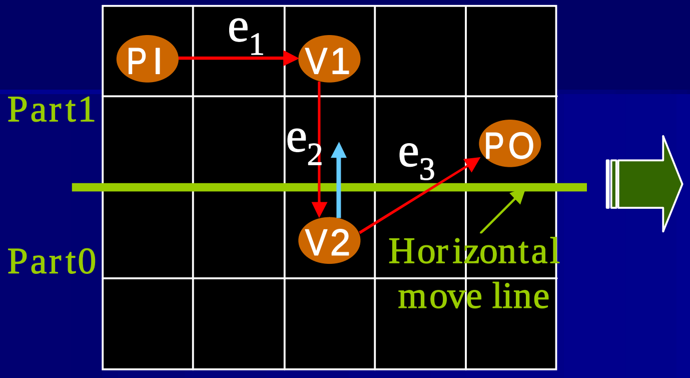
$$TG(v_i) = VC(v_i \mid P(v_i)=0) - VC(v_i \mid P(v_i)=1)$$

VC(v_i | P(v_i)): violation counts of SDC when P(v_i) is 0 or 1

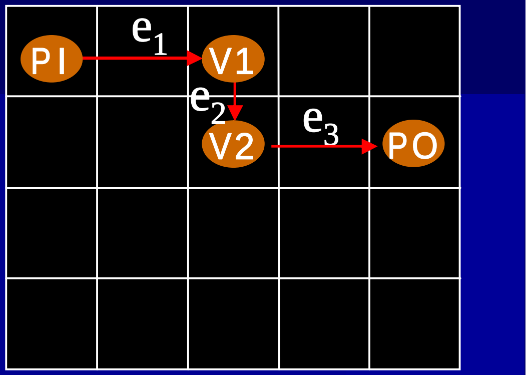
Computation of the Timing Gain



Possible move-directions for v_2



Computing TG of v_2 for a move



V_2 moves to the upper cell

Computing TG(v_2) for moving in the upper direction:

σ : pi→v1→v2→po, edges: e1(pi,v1), e2(v1,v2), e3(v2,po)

SDC^2: $SD(\sigma)=TT$, $\forall\ e_i \in \sigma$, $P(s(e_i)) = P(t(e_i)) = 1$

$SDC^2\text{-count}(e_2) = 1$, $SDC^2\text{-count}(e_3) = 1$

$VC(V_2 \mid P(v_2)=0) = 2$ // SDC violations before v_2-move

$\Rightarrow$ SDC^2 violated for e2 and e3.

$VC(V_2 \mid P(v_2)=1) = 0$ // SDC violations after v_2-move

$\Rightarrow$ SDC^2 violations for e_2 and e_3 are eliminated.

$\therefore TG(v_2) = VC(V_2{:}P(v_2)=0) - VC(V_2{:}P(v_2)=1) = 2$

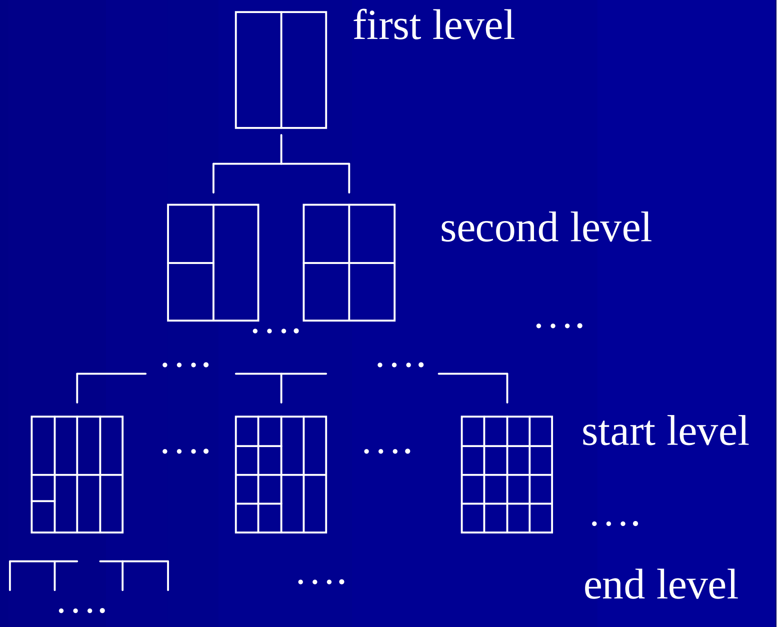
For other directions: $TG(v_2)_{LEFT} = -2$, $TG(v_2)_{RIGHT} = 0$, $TG(v_2)_{BOTTOM} = -2$

Aggregating the SD Violation Counts

- The timing gain of a node v_i w.r.t. a target move direction is obtained by summing the number of SDV's of each edge e_j connected to node v_i .
- This calculation is done by considering all I/O conduits (with given preferred signal directions) that go through the edge e_j .
- We thus aggregate preferred signal directions for all critical paths that pass through an edge, which in turn enables us to maximize the monotonic behavior of the critical paths.

Our TDP Algorithm (PSDP) : Preferred Signal Direction Driven Placement

- We integrate the proposed timing optimization process into a general recursive bipartitioning-based placement framework.
- We adopt hMetis as the bipartitioner.
- We perform timing optimization only once per hierarchical level after an initial global placement is generated.
- We legalize the obtained global placement solution when it reaches the “end” level.



PSDP Algorithm

```
PSD_Placement (G, T)
G : A directed graph representing a sequential circuit
T : Timing constraints
1. Calculate the start and end levels of timing-driven global placement;
2. Do initial wirelength-driven global placement from level one to start level;
3. While (start_level ≤ i ≤ end_level)
        • While (j=0; j < number of sub_regions in level i; j++)
                • Generate a bipartitioning-based placement $P_{i,j}$ of sub_region j;
        • Do Timing_Optimization_PSD($P_i$ ,T);
4. Do the legalization;
```

PSDP Algorithm (Cont’)

```
Timing _Optimization_PSD (P,T)
P : An initial hierarchical placement solution with J regions
T : Timing constraints
1. Perform static timing analysis;
2. Find critical nodes, edges and I/O conduits;
3. Compute initial timing gains for all critical nodes;
4. Put all critical nodes into a timing gain heap;
5. While (heap != empty)
        • Extract root node $v_i$ from the heap and move it in its
          preferred direction to a neighbor region in  P;
        • If the region capacity is violated, select a non-critical node
          in the region and move it back to the parent region of $v_i$;
        • Update timing gains and restructure the heap as needed;
6. Find a sequence of moves that produces max_total_gain;
7. Undo moves that are not in the selected sequence;
8. If max_total_gain > 0 then goto step 3;
9. Else exit;
```

Experimental Setup

- 6 test cases; four (matrix, vp2, mac1 and mac2) are obtained from ISPD 2001 benchmarks while the other two (indust1 and indust2) are from a partner ASIC company.
- The delay models are based on TSMC 0.18um technology.
- PSDP is compared with Capo-boost and a leading industry placement tool (called QuadP) in terms of wire length and worst negative slack.
- We use Cadence WarpRoute and Pearl to report the experimental results.

Circuit Benchmark Data

Circuits	#Cells	#Nets	#IOs
indust1	5931	5969	179
indust2	20193	21699	351
matrix	3,083	3,200	117
vp2	8,714	8,789	321
mac1	8,902	9,115	211
mac2	25,616	26,017	415

Experimental Results

- Comparisons of TNS (total negative slack of all timing endpoints) between wirelength-driven and timing-driven modes of PSDP

Benchmark circuits	Clock cycle	Wirelength-driven mode	Timing-driven mode	% Improvement
indust1	5.54	-38.2	-24.4	36.1%
indust2	8.75	-204.5	-93.1	54.5%
matrix	3.23	-5.8	-4.3	25.9%
vp2	3.67	-68.3	-25.1	63.3%
mac1	2.07	-21.4	-13.5	36.9%
mac2	2.35	-125.4	-62.7	50.2%
Average				44.5%

Experimental Results (Cont’d)

- Comparisons in terms of HPWL (wirelength after placement), RWL (wirelength after routing) and WNS (worst negative slack after routing)
  - PSDP runs 48% slower than QuadP in non-timing mode, but 58% faster than QuadP in timing-driven mode.

Circuits	QuadP (wirelength-driven mode)			QuadP (timing-driven mode)			Capo-boost			PSDP (timing-driven mode)		
	HPWL	RWL	WNS	HPWL	RWL	WNS	HPWL	RWL	WNS	HPWL	RWL	WNS
indust1	3.50	4.62	-1.23	3.59	4.65	-1.22	3.54	4.72	-1.85	3.58	4.80	-0.89
indust2	15.73	27.55	-4.31	15.67	28.10	-3.81	16.39	28.66	-3.52	16.07	29.07	-3.17
matrix	1.05	1.17	-2.20	1.08	1.20	-2.06	1.05	1.16	-2.04	1.12	1.23	-2.01
vp2	3.71	4.51	-3.02	3.77	4.53	-3.21	3.65	4.83	-3.21	3.81	4.89	-2.95
mac1	4.44	5.07	-0.56	4.45	5.09	-0.49	4.77	5.24	-0.41	4.81	5.25	-0.30
mac2	22.48	32.44	-14.46	22.49	32.97	-3.63	23.55	29.49	-1.01	24.08	31.23	-3.73
Ratio	1.00	1.00	1.00	1.01	1.01	0.83	1.03	1.01	0.85	1.05	1.04	0.69

Conclusions

- We introduced a new timing-gain function model based on preferred signal directions for the timing-driven placement context.
- The advantage of the new methodology has been confirmed by experimental results: on average 31% improvement on WNS. compared to a leading industry placer at the expense of wirelength increase, on average, by 5%.