Incremental Detailed Placement for VLSI Design

Prabhakar R [1], Sreenivasa Murhthy K E [2], Soundara Rajan K [3]
Assosiate Professor[1], ECE Dept, Dr KVSRIT, Kurnool, A P, India.
Principal[2], SSITS, Rayachoti, Kadapa, A P, India.
Former Rector & Retd. Professor[3], JNTUA, Ananthapuramu, A P, India.

Abstract

The main purpose of VLSI placement is to place the objects into fixed chip such that there should be no overlaps among the objects and some cost metric such as wire length and routability is optimized. Physical synthesis optimizations and changing the placement method typically change the locations of cells, resize cells or add more cells to the design after global placement. But, those changes generally leads to wire length increases; thus another method of optimizations to for further improve wire length, timing and routing congestion characteristics is required. The Incremental Detailed Placement techniques could be useful in this condition. So, we propose a new detailed placement paradigm, which use a set of pin-based timing and electrical constraints in detailed placement to prevent it from degrading timing or violating electrical constraints while reducing wire-length.

Keywords: *Routability, Physical Synthesis, Worst Negative Slack, figure-of-Merit, Routing Congestion.*

I. INTRODUCTION

Global placement is one of the most typical processes in modern physical design. Its task is to determine the overall locations of cells in the design. However, physical synthesis designs such as buffering and gate sizing are applied after component placement to further optimize timing. These methods usually insert new cells or change the size of existing cells. Engineering change order (ECO) is another source of modifications for designs under optimization. It could also introduce new logics, change the physical sizes of objects, or change the locations of existing cells. All these changes may probably result in overlaps among cells. Therefore this design needs additional (extra) legalization to remove those overlaps. Even though many legalization methods have been proposed to minimize the disturbance to the original placement, they usually result in wire length degradation.

The Detailed placement techniques such as Simulated Annealing (SA) based swapping and moving [7], cell interleaving [12], branch-and-bound reordering, branch-and-price reordering, guided local search, global swap and local reordering, and net length constrained SA approach can all reduce the total wire length only. But, reducing total wire length do not necessarily result in timing improvement, particularly after physical design. Therefore an efficient delay model is required for Incremental detailed placement to avoid doing any harm to timing critical paths while improving total wire length.

A. Pin Based Timing And Electrical Constraints

In this section, we will model timing and electrical constraints for Incremental Detailed Placement. The purpose of adding these constraints is to prevent any degradation of timing results (worst negative slack, total negative slacks, etc), Which uses constraints on timing paths and our timing constraints are imposed on individual pins. Even though these constraints may be preserved using pin-based constraint, which would greatly simplify the timing computation during placement because the expensive path propagation computation is not required.

B. Delta Arrival Time Constraints

The arrival time of each pin of a gate is defined as the addition of delay segments from timing start points, i.e. PIs or the output of a sequential logic, to the pin itself. The arrival time of each gate is simply defined as the addition of delay segments on the most critical input pin.

Thus, let *Nm* be a set of gates or PIs connected to the input of gate *m*, and gate $k \in Nm$ connected to input pin *j* of gate *m*. The arrival time of pin will be

$$\mathrm{AT}_{m,j} = \mathrm{AT}_k + d_k + d_{k,m} \qquad (1)$$

The delta arrival time of input pin as the differences between the arrival time of pin itself and the arrival time of the gate. As shown in Fig 1, the delta arrival time of the input j of gate m, $\Delta\mathrm{AT}_{m,j}$ is defined as follows:

$$\Delta\mathrm{AT}_{m,j} = \mathrm{AT}_m - \mathrm{AT}_{m,j} \qquad (2)$$

The delta arrival time for a primary output pin (PO) or a sequential logic input pin is always zero because there is only one pin to compare with. But the delta arrival time is absolutely different from slack. Slack is the difference between arrival time and required arrival time, while delta arrival time is the difference between the arrival time of an input pin to the most critical input pin. Pins with same slack can have different delta arrival time, and pins with same delta arrival time can have different slacks.

$$\mathrm{AT_m} = \mathrm{AT_k} + \mathrm{d_k} + \mathrm{d_{m,j}} + \Delta\mathrm{AT_{m,j}} \qquad (3)$$

If the placement of the cells changes, the gate delay d_m , wire delay $d_{m,k}$ and arrival time on all the gates will also change. Thus, the new arrival time on gate m can be computed as a new formula, which is shown in below:

$$\mathrm{new}_{\mathrm{AT}_m} = \max_{k \in N_m} [\mathrm{new\_AT_k} + \mathrm{d_k} + \mathrm{d_{k,m}} + \Delta\mathrm{d_k} + \Delta\mathrm{d_{k,m}}] \qquad (4)$$

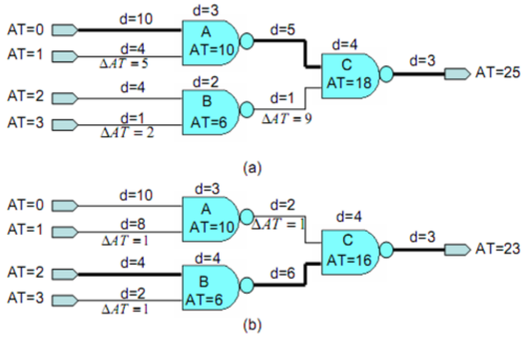

Fig 1: example of Delay Model

Fig 1 shows a small network with 4 PI, 1 PO, 3 gates and 7 nets.

The upper figure 1(a) shown the original gate and wire delays, arrival times on all three gates, PIs and PO, and delta arrival times on noncritical internal pins, while the lower figure 1(b) shown the changed gate/wire delays, and new arrival times. We have observed that the increase on the merged gate delay and wire delay is less than the delta arrival time. For example, the combined gate and wire delay between gate B and C changes from 2+1=3 to 4+6=10. However, the ΔAT on this connection is 9, which is still greater than the amount of delay increase 10-3=7. Even the critical input pin of gate C changes, the arrival time on the input of C is reduced from 18 to 16. We have also verified that the arrival times on all the gates do not increased and so does the PO arrival time. Therefore the slack on PO is not degraded.

II. INCREMENTAL DETAILED PLACEMENT

In this design we used the Incremental Detailed Placement method that convert placement from one legal solution to another legal solution. These methods take a legally placed net list, change locations of cells while still maintaining the legality. Normally these approaches only check whether the movements reduce the total wire length or not.

A. Constraint Formulation

During Incremental Detailed Placement, we can obtain an accurate estimation of the total half perimeter wire length (HPWL). If we can roughly estimate delay or slew based on HPWL, then we can verify whether the constraints are satisfied or not.

To do this we used a differential gate delay and wire delay model, which estimates delay and slew increments of placement change. Conversely, the main difference is the gate delay modeling. Therefore the gate delay and output slew is only determined by output load. The advantage of this is to avoid slew propagation, which is time consuming because which has to propagate the slew all the way down to the end and re-compute the timing on many cells down the way. Using a conservative gate delay modeling actually makes the timing constraints more conservative, which helps protecting of any timing degradation.

Gate delay and output slew can be represented as linear functions of input slew and output load as follows:

$$d_k = A_0 + A_1 c_k + A_2 s_{k,j} \qquad (5)$$

$$s_k = B_0 + B_1 c_k + B_2 s'_{k,j} \qquad (6)$$

Where d_k and s_k are the delay and output slew of gate k; $s_{k,j}$ is the most critical input pin of gate k and $s'_{k,j}$ is the input slew on pin j. $A0$, $A1$, $A2$ and $B0$, $B1$ and $B2$ are constants determined by the standard cell library characterization. Since we have assumed the critical input pin slew is constant, the differential gate delay and output slew can be computed by as follows:

$$\Delta d_k = A_k \Delta c_k \qquad (7)$$

$$\Delta s_k = B_k \Delta c_k \qquad (8)$$

Where Δd_k and Δs_k is the gate delay and output slew increments for gate k, respectively. Δc_k is the total output load increment, which can be computed by

$$\Delta c_k = c \Delta l_i \qquad (9)$$

Where c is the unit wire capacitance; Δl_i is the total Wire length (HPWL) increment for net i, which gate k drives.

B. Objective Function

The objective function of our Incremental Detailed Placement is to reduce both TWL and TNS. Assuming and we can guarantee that slacks do not degraded. We can use weighted total wire length as the optimization objective for Incremental Detailed Placement. Critical nets (nets with negative slacks) are given higher weights than other nets. The weighted wire length objective function is given below.

$$WTWL = \sum w_i l_i \qquad (10)$$

Where WTWL is weighted total wire length. w_i is the net weight for net *i*, and *li* is the HPWL of net *i*.

III. EXPERIMENTAL RESULTS

We have implemented the Incremental Detailed Placement in DSCH and MicroWind Tool. The technology for each design is reported in Table I. The worst negative slack (WNS) and Figure-of-Merit (FOM) of these designs are reported in Table II.

Table I: Design, Technology, TWL and Power of IDP:

Design	Technology	TWL (µm)	Power (µW)
D Flip Flop	90-nm	791.58	0.326
Right Shift Register	90-nm	15796.15	0.938
Dual Port RAM	90-nm	3986.01	1.018
Synchronous Counter	90-nm	71203.59	25.83

Table III shows the total Wire length (TWL) comparison of Existing Placement (Existing) method and those after TDIP, NAIP, DBIP and IDP. Considering that the placement is already optimally placed by a global placer during physical synthesis, the improvements are significant. We have highlighted those cases where TWL did not increase from existing. We have shown that the IDP improves the TWL on all placement methods with an average of 3.127% improvement, while TDIP, NAIP and DBIP improves the TWL a less.

Table II: Design, Cells, Nets, Slack & FOM of IDP:

Design	No.of Cells	No.of Nets	Slack (WNS) (ps)	FOM (Ps)
D Flip Flop	05	17	1.650	28.05
Right Shift Register	56	80	3.083	246.66
Dual Port RAM	12	44	2.740	120.49
Synchronous Counter	178	129	0.842	108.70

Table III: Comparison of Total Wire length of TDIP, NAIP, DBIP & IDP with Existing Methods:

Design	TWL(µm)					Change in WL			
	Existing	TDIP	NAIP	DBIP	IDP	TDIP	NAIP	DBIP	IDP
D Flip Flop	844.23	834.6	825.6	808.92	791.58	1.14%	2.21%	4.18%	6.24%
Right Shift Register	16071.03	16016.05	15924.43	15901.76	15796.15	0.34%	0.91%	1.05%	1.71%
Dual Port RAM	4094.79	4079.25	4040.4	4009.32	3986.01	0.38%	1.33%	2.09%	2.64%
Synchronous Counter	72564.71	71799.08	71628.94	71374.9	71203.59	1.1%	1.2%	1.64%	1.88%
Average reduction in WL						0.74%	1.41%	2.24%	3.12%

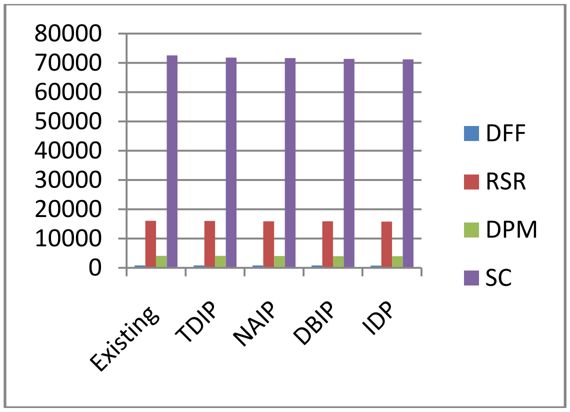

Fig 2: Wire length (µm) comparison Plot of TDIP, NAIP, DBIP & IDP with Existing Method:

Table IV gives the Worst Negative Slack (WNS) comparison among TDIP, NAIP, DBIP and IDP. In terms of Worst Negative Slack the IDP has got better Improvement than among all methods, which is higher 61.15% than TDIP, 36.04% than NAIP, 12.72% higher than DBIP.

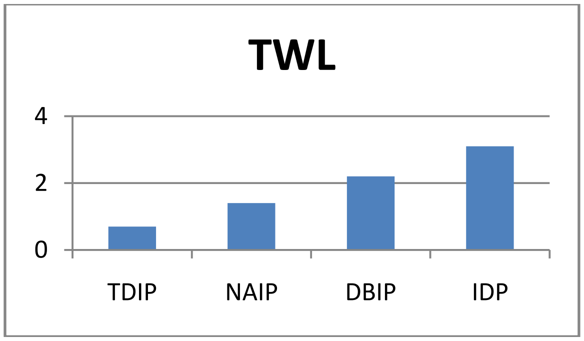

Fig 3: Percentage of Improvement in TWL of TDIP, NAIP, DBIP & IDP with Existing Method:

Table IV: Slack (Ps) of TDIP, NAIP, DBIP & IDP with Existing Method:

Design	Slack(WNS) (Ps)					Improvement (%)			
	Existing	TDIP	NAIP	DBIP	IDP	TDIP	NAIP	DBIP	IDP
D Flip Flop	6.624	4.22	1.787	1.705	1.650	36.3%	73.02 %	74.26 %	75.09 %
Right Shift Register	82.55	63.54	23.96	3.518	3.083	23.03 %	70.98 %	95.74 %	96.27 %
Dual Port RAM	33.18	32.23	31.02	10.79	2.740	2.9%	6.51%	67.48 %	91.74 %
Synchronous Counter	58.98	26.63	19.45	15.73	0.842	54.85 %	67.02 %	73.33 %	98.57 %
Average Improvement in Slack (WNS)						29.27 %	54.38 %	77.70 %	90.42 %

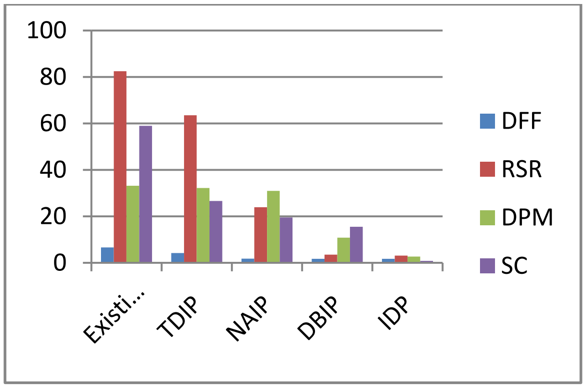

Fig 4: Comparison Plot of Slack:

Table V gives the results of Figure-of-Merit (FOM) of TDIP, NAIP, DBIP & IDP with Existing Methods. The IDP provides better FOM than all placement Methods. Which is 61.16 % better improvement than TDIP, 36.02 % better than NAIP & 12.7 % better than DBIP.

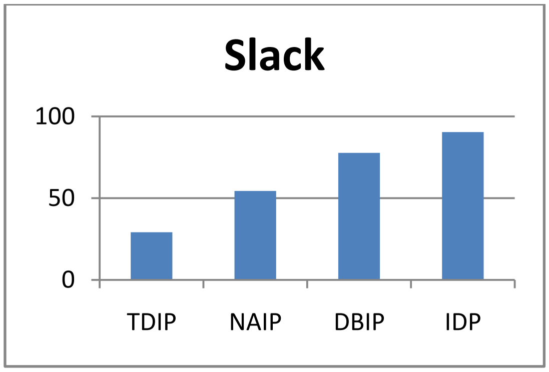

Fig 5: Percentage of Improvement in Slack of TDIP, NAIP, DBIP & IDP with Existing Method:

Table V: FOM (Ps) of TDIP, NAIP, DBIP & IDP with Existing Method:

Design	FOM (Ps)					Improvement (%)			
	Existing	TDIP	NAIP	DBIP	IDP	TDIP	NAIP	DBIBP	IDP
D Flip Flop	112.61	71.81	30.39	28.996	28.05	36.23 %	73.01 %	74.25 %	75.01%
Right Shift Register	6604.2	5083.39	1914.01	281.44	246.66	23.03 %	71.02 %	95.74 %	96.27%
Dual Port RAM	1459.5	1417.8	1364.90	474.79	120.49	2.86%	6.48%	67.47 %	91.74%
Synchronous Counter	7607.3	3435.5	2509.1	2028.96	108.70	54.84 %	67.02 %	73.33 %	98.57%
Average Improvement in FOM						29.24 %	54.38 %	77.7%	90.4 %

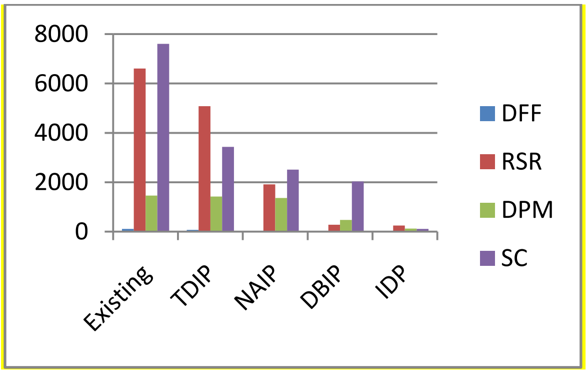

Fig 6 : Comparison of FOM:

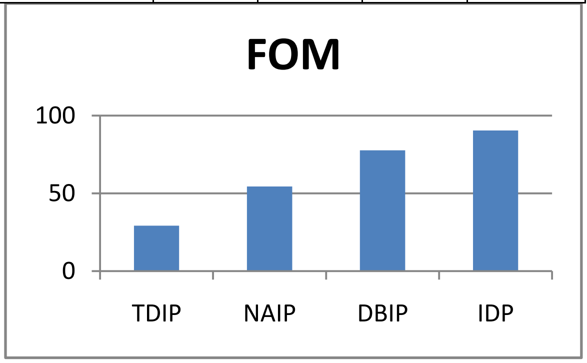

Fig 7: Percentage of Improvement in FOM (ps) of TDIP, NAIP, DBIP & IDP with Existing Method:

Table VI shows the results of Power for all Placement Methods like TDIP, NAIP, DBIP & IDP with existing Methods. The IDP gives better power improvement over other methods. The IDP gives improvement than 31.52 % TDIP, 21.97% than NAIP & 2.87% than DBIP

Table VI: Power (µW) of TDIP, NAIP, DBIP & IDP with Existing Method:

Design	Power(µW)					Improvement (%)			
	Existing	TDIP	NAIP	DBIP	IDP	TDIP	NAIP	DBIP	IDP
D Flip Flop	3.103	1.13	1.258	0.327	0.326	63.58 %	59.46 %	89.46 %	89.49 %
Right Shift Register	18.58	7.75	7.693	1.873	0.938	58.29 %	58.59 %	89.92 %	94.95 %
Dual Port RAM	25.28	10.12	5.063	2.537	1.018	58.97 %	79.97 %	89.96 %	95.97 %
Synchronous Counter	94.82	50.570	30.637	25.831	25.83	46.67 %	67.69 %	72.76 %	73.18 %
Average Improvement in Power						56.88 %	66.43 %	85.53 %	88.4 %

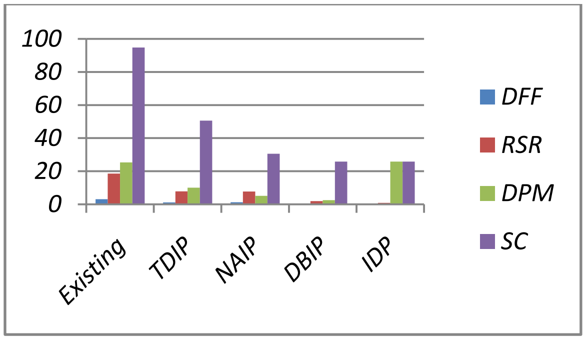

Fig 8: Power (µW) Comparison Plot:

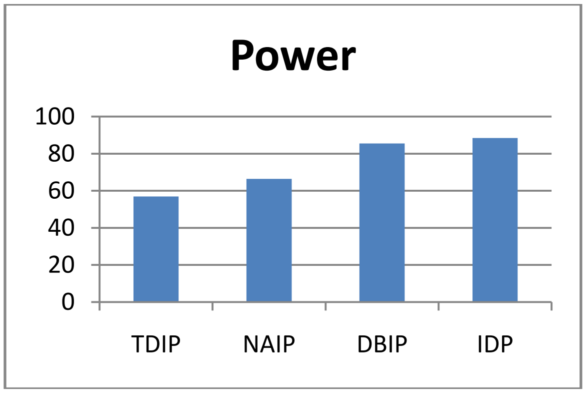

Fig 9: Percentage of Improvement in Power (µW) of TDIP, NAIP, DBIP & IDP with Existing Method:

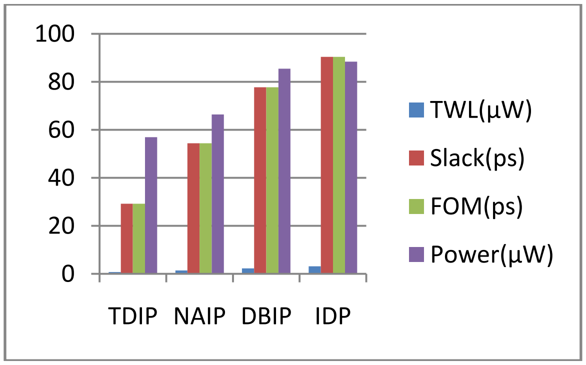

Fig 10: Overall Comparison of Improvement in TWL, Slack, FOM & Power for TDIP, NAIP, DBIP & IDP.

IV. CONCLUSION & FUTURE SCOPE

Incremental Detailed placement not only reduced Total wire length, but also significantly improves timing (WNS & FOM) and Power. These constraints and objective function are simple to implement and can be applied to many detailed placement frameworks. Throughout our work we used minimum sized transistors for the Placement, further if the transistor sizes are still reduced we may get further better improvement in the results.

REFERENCES

[1] A. B. Kahng and Q. Wang, "Implementation and extensibility of an analytic placer," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, pp. 734–747, 2005.

[2] T. Chan, J. Cong, and K. Sze: "Multilevel generalized force-directed method for circuit placement," in Proc. Int. Symp. on Physical Design, pp. 185–192, 2005.

[3] H. Eisenmann and F. M. Johannes: "Generic global placement and floor planning," in Proc. Design Automation Conf., pp. 269–274, 1998.

[4] M. Wang, X. Yang, and M. Sarrafzadeh, "Dragon2000: Standard-cell placement tool for large industry circuits," in Proc. Int. Conf. on Computer Aided Design, pp. 260–263, 2000.

[5] U. Brenner, A. Pauli, and J. Vygen, "Almost optimum placement legalization by minimum cost flow and dynamic programming," in Proc. Int. Symp. on Physical Design, pp. 2–9, 2004.

[6] A. B. Kahng, I. L. Markov, and S. Reda, "On legalization of row-based placements," in Proceedings 14th Great Lakes Symposium on VLSI, pp. 214–219, 2004.

[7] H. Ren, D. Z. Pan, C. J. Alpert, and P. Villarrubia, "Diffusion-based placement migration," in Proc. Design Automation Conf., pp. 515–520, 2005.

[8] L. Tao, H. Ren, C. Alpert, and D. Z. Pan, "Computational geometry based placement migration," in Proc. Int. Conf. on Computer Aided Design, pp. 41–47, 2005.

[9] A. H. Ajami and M. Pedram, "Post-layout timing driven cell placement using an accurate net length model," in Proc. Design Automation Conf., pp. 595–600, 2001.

[10] W. Choi and K. Bazargan, "Incremental placement for timing optimization," in Proc. Int. Conf. on Computer Aided Design, pp. 463–466, 2003.

[11] B. Halpin, N. Sehgal, and C. Y. R. Chen: "Detailed placement with net length constraints," in Proc. of The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, pp. 22–27, 2003.

[12] H. Ren, D. Z. Pan, and D. Kung: "Sensitivity guided net weighting for placement driven synthesis," in Proc. Int. Symp. on Physical Design, pp. 10–17, 2004.