An Explorative Tile-based Technique for Automated Constraint Transformation, Placement and Routing of High Frequency Analog Filters

Hui Zhang, Preethi Karthik, Hua Tang, Alex Doboli
Department of Electrical and Computer Engineering
State University of New York at Stony Brook
Stony Brook, NY 11794-2350, U. S. A.
Email: {huizhang,preethik,htang,adoboli}@ece.sunysb.edu

***Abstract*— This paper presents an original methodology and algorithms for designing high-frequency analog filters. Filters are realized to meet AC performance specifications as closely as possible. Addressed design steps include parameter sizing, placement and routing. The explorative design method, based on tabu search, takes into account layout parasitics of the circuit. The synthesis strategy alternates between (1) a uniform sampling of the solution space and (2) a focused search of attractive regions. We also discuss the usage of a layout representation based on symbolic tiles for analog system synthesis. Experiments on filters demonstrate the quality of our solution.**

I. Introduction

Currently, there is a shortage of efficient CAD tools for sizing and layout design of high performance analog circuits and systems [7]. With analog layout tools, the goal is not merely to design the circuit within a given chip area but also to keep *minimal* the performance degradation due to layout parasitics [3] [6] [9]. Most of the analog layout techniques follow an optimization-based place-and-route approach, where layout is generated by an optimization process e.g. simulated annealing, genetic algorithms etc driven by a cost functions expressing criteria such as performance degradation, wire length, area etc [1] [3] [6] [9]. However, it is possible that the fixed circuit parameters do not leave enough performance margins to accommodate the layout-induced performance degradations. In this case, the final design would be incorrect. Typical examples include high-frequency filters that incorporate capacitors of the same order of magnitude (tens/hundreds of fF) as interconnect parasitics. Parasitic capacitors are an integral part of the signal processing performed by the passive elements. Costly re-iterations through circuit sizing and layout generation are needed to produce a constraint satisfying design. *The solution is to combine the parameter sizing process with the layout design step to improve design quality and convergence of the CAD algorithms*.

A second specifics of the current layout design methods is that placement and routing are distinct and subsequent steps [3] [9] [10] [11]. This works well for medium range frequencies. *At high frequencies, however, placement must be tightly integrated with routing to allow accurate evaluation of performance degradation due to layout parasitics*. In spite of the merits of existing analog layout tools, new approaches are compulsory for sizing, placement and routing of high frequency analog circuits.

This paper proposes an original synthesis algorithm for high-frequency filter design. The method combines (1) sizing of individual components (constraint transformation), (2) placement and (3) routing based on an (4) improved circuit representation. The addressed problem is difficult as it involves three combined design steps, each having a large complexity. For example, a 6th order filter (one of the examples in Section 4) requires about 190 design variables for combined sizing, placement and routing. Similar to other work [5], our experiments with traditional optimization algorithms such as simulated annealing showed poor convergence and long execution times for a significant number of applications. Another difficulty originates in the nature of the optimization problem. For example, the best design point for a 100-400 MHz bandpass filter is located in a narrow solution region, for which the neighboring points present poor values of the cost function. Thus, adjacent regions might not predict the existence of high quality solution points. This results in gradient-based search not being effective. We observed for analog-to-digital converters(ADC) a similar characteristic of the solution space. This suggests that the exploration strategy for synthesis must perform (1) a uniform sampling of the solution space with (2) a step of focused search of attractive regions. This strategy is difficult to implement in typical random search algorithms e.g. simulated annealing, genetic algorithms, which significantly rely on random moves.

This paper discusses a tabu search algorithm [8] for analog filter synthesis through combined constraint transformation, placement and routing. The tabu search algorithm is used to methodically explore attractive regions of the solution space to find their local optima. This is possible because of the deterministic nature of the technique [8]. Solution space sampling is achieved by a component that executes a number of random moves until the solution point is significantly perturbed. This is different from *aspiration* and *diversification* steps [8] of tabu search. For example, diversification attempts to explore parameters which rarely change. For analog synthesis, however, this is not an efficient strategy because these parameters might have secondary effects on the system performance. Experiments show that superior quality filters are designed with the proposed synthesis methodology than with traditional approaches, which separate sizing, placement and routing. In fact for filters operating at $n \times 100$ MHz or above, the combined methodology is compulsory as separate sizing, placement and routing result in very poor solutions.

The synthesis algorithm uses *symbolic tiles* [12] as a compact representation of the layout. Though this approach has been used before [3] [9], our representation is much simpler and more compact, thus increases the efficiency of the design algorithms. A smaller number of tiles offers the advantage of faster methods for tile swapping and tile moving, which have a linear complexity with the number of tiles [12]. In Section 3, we offer examples to support this conclusion. Constraints on the relative position of tiles are used to express symmetries and matching. Another difference as compared to previous work [3] [9] is that our tiles are *soft* (their sizes and aspect ratio can change). This is a consequence of constraint transformation being part of the synthesis loop. For keeping the complexity of tile managing methods low, complete knowledge about left, right, top

0-7803-8834-8/05/$20.00 ©2005 IEEE.

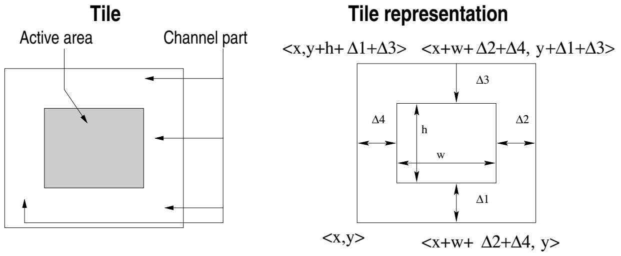

Fig. 1. **Tile definition**

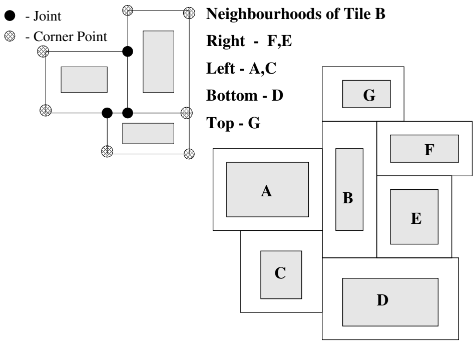

Fig. 2. **Tile relations**

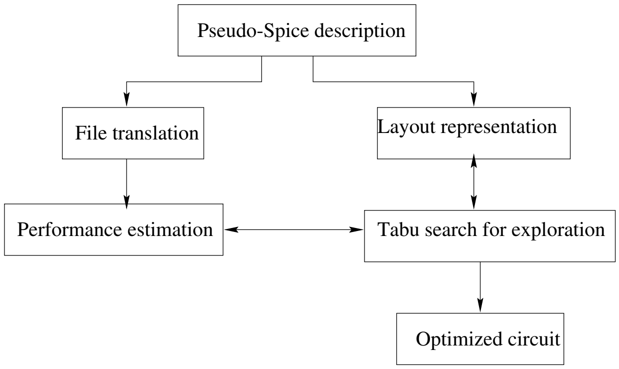

Fig. 3. **Flowchart of the filter design method**

and bottom neighbors of each tile must be *explicitly* available. We decided to store the neighborhood information as distinct O trees [13] for each of the four directions. Other representations such as sequence-pairs [1] or B* trees [2] are not efficient for our problem as they implicitly offer the neighborhood information.

This paper is organized as follows. Section 2 details the basic representation adopted in our approach. Section 3 discusses the design methodology. Section 4 presents our experiments. Finally, Section 5 provides conclusions and plans for future work.

II. Tile Based Layout Representation

A tile [12] based representation is adopted for our layout. The tile is the basic building block for the layout. Figure 1 depicts the tile representation. It represents both active blocks and channels. The active part of the tile is the actual component, and the channel part is the portion of the channels surrounding the active region. The widths of the channel part is denoted by Δ_i, i=1,4. A layout is a collection of tiles. The used definition reduces the number of tiles for a layout as it decreases the number of tiles needed to express empty spaces.

Tiles can be of three types: (1) active tiles, (2) empty tiles, (3) margin tiles The active tile is a tile which represents an electrical component such as a resistor, a capacitor, an op-amp etc. A tile which does not represent an electrical component is called an empty tile. The tiles at the four borders of the layout are called margin tiles. The layout size is determined from the number of tiles present.

A tile is defined by its four *corner-points*. Figure 2 shows corner-points as gray bubbles. A corner-point is the pair $(x, y) \in R^2, 0 < x < w_{max}, 0 < y < h_{max}, (x, y) \in T$. T is the set of all tiles. A corner-point belongs to at least one tile. The set of all corner-points is denoted by CP. A *joint* of a tile is a corner-point which meets an adjoining tile at a point other than its corner-point. Or it can be alternately defined as a point which is a corner-point of two or more tiles and is not the corner-point of at least one tile. A joint is defined as the pair $(x, y) \in CP, (x, y) \subseteq T \& \exists t \in T$ for which (x,y) is not a member but one of its corner-points has its x- or y coordinate equalt to x or y. The left part of Figure 2 illustrates joints as black bubbles. A *sequence* is an ordered list of corner-points. Sequences are used during the routing stage. The relative positions of the neighboring tiles is defined by neighborhood relationships. Four neighborhood relations exist for each tile, as shown in Figure 2. The width of the Δ-s is determined during the routing phase by the number of nets which are to be routed through the channels. These relationships are stored as distinct O trees [13] for each of the four directions. This helps in immediate retrieval of the neighbors of a tile.

Advantages of the proposed tile representation

Since the layout dimensions are flexible, the resizing of blocks can be easily handled by a move in the optimization algorithm known as domino move, which will be introduced in the next section. This representation is better than earlier tile based representations as the same circuit representation can be used for both the placement and routing stages.

Wiring the circuits is also simplified as depending on the path of the wire, the Δ-s of the tiles through which the wire is routed is correspondingly increased. In the case that there are tiles with no wires running through their channel the tile space is decreased, thus allowing optimization in area. Because of the Δ-s, we need not at every stage ensure that there is sufficient space in the layout for routing.

The resizing of the components is a major advantage of our solution. Resizing helps us manipulate the specification according to our requirements. Device merging cannot be handled by our software as the devices are represented by black boxes. A modeling of the thermal and substrate effects can be easily added to our current solution to further improvise it.

III. Combined Sizing, Placement and Routing

A. Exploration using Tabu Search

Figure 3 presents the flowchart of the method for combined parameter sizing, placement and routing for design of high-frequency filters. The input to the placement tool is a SPICE-like format description of the circuit. This input file is parsed and a data structure is created which holds the information of the blocks and their connections. At the same time, tiles are also created corresponding to the various blocks.

The design flow is done by the tabu search algorithm [8] where in the tiles are (1) moved, (2) routed or (3) their dimensions changed depending on their functionality and the requirement to match the system specifications. Figure 4 presents the pseudocode for the tabu-search exploration method. A single iteration of the tabu search

```
PROCEDURE system_synthesis IS
  while the final iteration number is not reached do
    if initial routing is not done then
      do initial routing for layout L;
    for (D = each direction of LEFT, DOWN, RIGHT, UP) do
      save current design L;
      resize_tile T by amount AMT in direction D;
      fix_routing ();
      Cost = calculate cost function ();
      save current design L;
      resize_tile T by amount -AMT in direction D;
      fix_routing ();
      Cost = calculate cost function ();
    for any possible pair of tiles T1 $<>$T2 do
      save current design L;
      swap_tiles T1 and T2;
      fix_routing ();
      Cost = calculate cost function ();
    find the min. cost, and change the design L to get this min. cost;
    fix_routing ();
    if cost is within bounded range for a fixed # of moves then
      execute random moves on the important parameters;
END PROCEDURE
```

Fig. 4. **Tabu-search based exploration method**

```
PROCEDURE domino_move IS INPUT: T - tile id in the layout which will
be resized, moved or swapped
D - the direction of change
CHG_AMT - the quantity by which the tile is to be changed
BEGIN
  make L_COPY as the copy of the layout L;
  if CHG_AMT ¿ 0
    add all neighbors of tile T on direction D to set N,
    add all neighbors of every tile in set N to N;
    set all members of N to UNCHANGED;
    add the tiles which are not the members of N but are the
    neighbors of some members of N on reverse direction of D;
    add T to set M;
    change T on direction D by amount CHG_AMT;
    while N $<>$ $\emptyset$ do
      NT = a tile removed from N;
      if NOT all neighbors of NT on reverse direction
       of D are CHANGED then
        add NT back to N;
      else
        move NT on direction D by the minimum distance
         to make sure NT does not overlap with any tiles of M;
        if NT $<>$ $\emptyset$ then add NT to M;
END PROCEDURE
```

Fig. 5. **Domino_move algorithm**

involves increasing or decreasing a tile's dimensions in all the four directions and swapping the tiles to obtain a solution which has the least cost. The tabu search algorithm is used as it avoids being trapped in a local minima [8]. Traditionally simulated annealing (SA) has been used in analog layout tools [3]. Simulated annealing is a weaker search heuristic. Repeatedly annealing is very slow, especially if the cost function is expensive to compute. The subjective nature of choosing SA parameters (temperatures & step size) and the complicated tuning restricts the value of the algorithm. Tabu search on the other hand is much quicker, and does not have the limitations of SA.

The evaluation of the effectiveness of each move and how it affects the layout is determined by the cost function:

$$\text{Cost} = \alpha \times area + \beta \times freq\_resp\_error,$$

where α and β are weighting factors. The cost function considers layout area as well as the error of the obtained transfer function with respect to the desired AC response at different frequencies (term $freq\_resp\_error$).

Each tile is assigned a set width and height depending on the component it is assigned to. At the initial stage each block is empty i.e. the models for the resistors, capacitors, op-amps etc. in the block have not been included at this stage. These empty tiles will be replaced by macro-models at a later stage.

All the moves are tried on a copy of the layout, called the shadow layout. The moves are replicated on the original layout only if they are successful.

Given that the number of nets is small, routing is done using the Dijkstra's shortest path algorithm [4].

The tabu search algorithm was expanded with a solution space sampling step (last if statement in Figure 4). If the cost function remains within a fixed bounded range for a number of iterations then the algorithm attempts to move to a different solution region. This is done by performing a number of random moves, predominantly resizings of the important parameters of the design. Important parameters have a dominant impact on the filter performance. For example, the first gm and C of the third order elliptic filter in Section 5 are more important than the rest of the parameters. Knowledge on the importance of parameters is provided as input. This strategy prevents from modifying less important parameters, which keeps exploration basically in the same region.

We also experimented with traditional strategies for escaping from local optima such as aspiration and diversification [8]. Aspiration attempts to move parameters in a tabu status, if they result in the best solution ever reached until that step. Diversification defines a penalty that encourages the changing of parameters, which are rarely modified. None of the two strategies improved the convergence of tabu search. Diversification actually offered worse results as a significant number of moves was wasted to modify unimportant parameters of the design.

B. Tile Based Placement

The tile based placement functions are move_tile, swap_tiles and resize_tile. They rely on a basic function called the domino_move function.

Domino Move: In the domino effect, to accommodate an expansion in a tile in the layout, its neighbors will have to be moved to avoid overlapping. This movement of tiles moves towards the edges of the layout stopping either when the move is completely absorbed by the empty tiles or at the border is called domino effect. The domino move function which implements this phenomenon is an important function in placement as all other routines will rely on this function. Figure 5 introduces the pseudocode for this step.

IV. Experiments

Experiments were developed to observe the practicality of our tool in designing high-frequency filters. The results were analyzed based on the quality of AC response, the layout and its compactness. The accuracy of the parasitics generation by the program is also

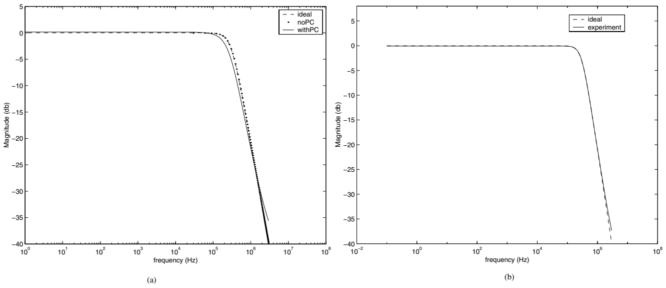

Fig. 6. 2^{nd} order Filter Results (a) without parasitics (b) with parasitics

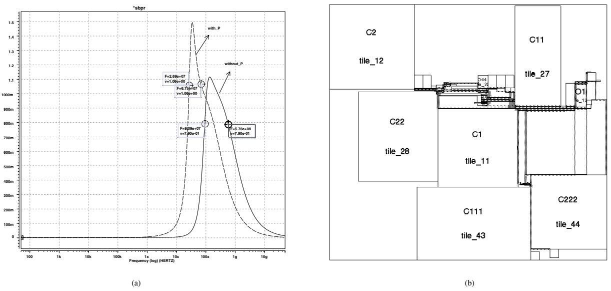

Fig. 7. (a)Response of the 6th order elliptic filter with and without considering parasitic (b)Layout

considered. Experiments were carried out on filters of various orders. Second order, third order elliptic and sixth order low-pass and bandpass filters were used for different bandwidth requirements. The filters are presented in [14]. The filters were optimized for the 3-dB points and frequency response by the resizing the components of the layout. The experiments were conducted on a SUN 80 workstation. The optimization for the 2-nd and 3-rd order filters involved about 30,000 iterations and ran for about 36 hours. The 6-th order filter required about 30,000 iterations and 70 hours.

Figure 6(a) presents the frequency responses for a 300 KHz second order filter when parasitics and coupling capacitances are not considered during the optimization. We see from these results that the ideal filter response and the response of the filter synthesized without considering the capacitances are almost identical. For the layout obtained by this method, when capacitances are considered, we obtain the graph of with_P&C which we can see is clearly different from the required response. The 3dB point is shifted by about 100kHz. This indicates the impact of layout parasitics on the performance of the design and also the need for the inclusion of parasitics and coupling capacitances during the synthesis stage. Figure 6(b) shows the filter characteristics when the layout parasitics were considered during synthesis. Results are much better in this case. Similar results were obtained for the 6-th order filter, also. Figure 7(a) presents the frequency response of the sixth order bandpass filter with the bandwidth of 100-400 MHz. Note that layout parasitics has a significant impact on the filter performance. These results motivate that constraint transformation, placement and routing must be integrated for synthesis of high frequency filters. Traditional approaches (which separate the three steps) will result in very poor designs.

Figure 7(b) shows the layout obtained for the 6-th order filter, the larger example. The largest blocks are the capacitors and the smaller blocks represent resistors and op-amps. The placement shows that the most communicating devices are placed close together thus also reducing wire length and improving routing. The results can be further improved by varying the macromodels currently being used. Reduction of wire length also improves performance by reducing the parasitics in the circuit.

V. Conclusions and Future Work

In this paper we have presented a novel algorithm for combined parameter sizing, placement and routing for the design of analog filters. The optimization based on tabu search takes into account parasitics of the circuit. The synthesis strategy alternates between (1) a uniform sampling of the solution space and (2) a focused search of attractive regions. We use a layout representation based on symbolic tiles. As compared to other tile representations, the described layouts have a smaller complexity, which improves the efficiency of the CAD algorithms. The representation is used for uniform expression of sizing, placement and routing. The effectiveness of our tool is presented by experimenting with three high-frequency filters.

Future work includes extension of the tool to other applications. To adapt this tool for layout of converters and other circuits, some more input parameters are required. These changes can be incorporated without major variations in the program. Low-power, low-voltage circuits may also designed by varying the models and adopting low power op-amps and OTAs. Faster algorithms have to be used to decrease the time for the design.

Acknoledgement

This work was funded in part by Defense Advanced Research Projects Agency (DARPA) and managed by the Sensors Directorate of the Air Force Research Laboratory, USAF, Wright-Patterson AFB, OH 45433-6543.

References

[1] F. Balasa *et al*, “Module Placement for Analog Layout Using the Sequence-Pair Representation”, *Proc. of 36th DAC*, pp. 274-279, 1999.

[2] Y. Chang *et al*, “B* Trees: - A New Representation for Non-Slicing Floorplans”, *Proc. of DAC*, 2000.

[3] J. Cohn *et al*, “Analog Device-Level Layout Automation”, *Kluwer*, 1994.

[4] T. H. Cormen *et al*, ”Introduction to algorithms”, *McGraw Hill*, 1990, pp.527-531.

[5] N. Dhanwada, R. Vemuri, “Hierarchical Constraint Transformation using Directed Interval Search for Analog System Synthesis”, *Proceedings of DATE*, 1999, pp. 328-335.

[6] Sree Ganesan *et al*, “FAAR: A Router for Field-Programmable Analog Arrays”, *Proc. of 12th Intl. Conf. on VLSI Design* , pp.556-563, January 1999.

[7] G. Gielen, R. Rutenbar, “Computer Aided Design of Analog and Mixed-signal Integrated Circuits”, *Proc. of IEEE*, Vol. 88, No 12, Dec 2000, pp. 1825-1852.

[8] F. Glover, ”Tabu search - part II”, *ORSA Journal of Computing”, pp.4-32*

[9] K. Lampaert, G. Gielen, W. Sansen, “Analog Layout Generation for Performance and Manufacturability”, *Kluwer*, 1999.

[10] E. Malavasi *et al*, “Area Routing for Analog Layout”, *IEEE Trans on CAD*, Vol.12, No.8, August 1993, pp.1186-1197.

[11] E. Malavasi *et al*, “Automation of IC Layout with Analog Constraints”, *IEEE Transactions CAD*, Vol. CAD-15, no. 8, pp. 923-942, August 1996.

[12] J. Osterhout, “Corner Stitching: A Data-Structuring Technique for VLSI Layout Tools”, *IEEE Trans. on CAD*, Vol. CAD-3, No. 1, pp. 87-100, January 1984.

[13] Y.Pang *et al*, “Block Placement with Symmetry Constraints based on the O-tree Non Slicing Representation”, *Proc. of 37th DAC*, pp. 464-467, 2000.

[14] B. N. Ray *et al*, “Efficient Synthesis of OTA Network for Linear Analog Functions”, *IEEE Trans. on CAD*, Vol. 21, No. 5, May 2002, pp. 517-533.