A Module Area Estimator for VLSI Layout*

Xinghao Chen and Michael L. Bushnell
CAIP Research Center

Department of Electrical and Computer Engineering
Rutgers University P.O. Box 909
Piscataway, New Jersey 08855-0909

Abstract

An efficient *Module Area Estimator* for VLSI chip layout has been developed to reduce the number of design iterations required to develop a chip floor plan. Module area is estimated for Standard-Cell and Full-Custom layout methodologies. We discuss the structure of the estimator and its algorithms. The layout area estimates are very close to those of manually laid out modules.

1. Introduction

We present a module area estimator for both the *Standard-Cell* and the *Full-Custom* layout methodologies. Among many different VLSI chip layout methodologies, only these three are popular: Full-Custom, Standard-Cell, and Gate Array methodologies. The Full-Custom methodology allows arbitrary device shapes and placements in the VLSI layout. The Standard-Cell methodology assumes that all devices have the same height, but varying widths. Devices may be arbitrarily placed in rows in the layout; routing channels between the rows allow wires to connect to the tops and bottoms of devices. The remaining methodologies and Gate Arrays are not covered here. Gerveshi [1] verified that for PLA's, the module area has a simple linear relationship to the number of basic logic functions and the number of devices in the chip.

Chip floor planning is a time-consuming, iterative process for all VLSI design methodologies. First, the chip is partitioned into large modules which are laid out independently. Next, an experienced designer must estimate areas and aspect ratios of modules. This information governs the activities of a chip floor planner [2, 3] as it floor plans the entire chip. Inaccurate aspect ratio estimates may lead to an unacceptable floor plan, requiring another design iteration. More accurate module aspect ratio estimates will significantly reduce the number of floor planning iterations. Unfortunately, module layout area estimation is difficult, even for senior designers.

Designers may mix different layout methodologies within a large VLSI chip. Standard-Cells are widely used because the design effort and time are significantly reduced by using a Standard-Cell layout library and running automatic place and route tools. The Full-Custom methodology leads to reasonable wire length and small chip area. Unfortunately, not all module layouts exist at floor planning time. Designers want to know how big a chip will be before they expend a large amount of effort to lay out the chip. Accurate module area estimators and floor planners allow the generation of trial floor plans for comparing the various different layout methodologies or mixtures of them. The designer can then intelligently choose the most appropriate methodology and floor plan for his chip.

A CAD tool is needed to automatically estimate realistic module aspect ratios for floor planning in order to reduce not only the design cost, but also the designer's effort to complete a chip layout. The module area estimator presented here estimates area(s) of small to moderate-sized modules for both the Standard-Cell and the Full-Custom layout methodologies in a modest amount of computer time. The estimator deals with different chip fabrication technologies (e.g., CMOS and nMOS) and can easily be adjusted to cope with new chip fabrication processes.

We focus mainly on wire area for module area estimation, since active cell area for a module is easily computed from the number and types of devices in the module. The module area estimation problem therefore becomes the problem of estimating module wiring (routing) area. In Standard-Cell area estimation, the *expectation* value of the total number of wiring tracks is computed and the central row is found to have the highest probability of having the largest number of feed-throughs. In Full-Custom area estimation, we calculate the minimum interconnection area for each net, instead of each wire, because we cannot compute exact wire lengths. Therefore, the total estimated wiring area is the sum of the net interconnection areas.

Comparisons are made between automatically estimated module areas and manually-created layouts and Standard-Cell layouts generated by a CAD tool. The estimated module areas are very close to the exact areas for the Full-Custom layouts. The estimation algorithm produces an upper bound estimate of Standard-Cell area.

2. Prior Work

Several existing tools perform area estimation, but they need either heuristics from the designer or descriptions of previous designs with "similar blocks" -- either similar in complexity but with different functionality or similar in functionality but for a different fabrication process [4, 5, 6].

Kurdahi developed PLEST [4] -- an area estimation CAD

*The research reported here was made possible through the Center for Computer Aids for Industrial Productivity (CAIP) and its supporters: the New Jersey Commission on Science and Technology, Rutgers University, and the CAIP industrial members.

tool for Standard-Cell VLSI chips. PLEST must know the local wiring density in order to estimate layout area. This local wiring density is known only when physical layout is done, but area estimation and floor planning occur before detailed physical layout in the design process.

The ALPHA design system uses an area estimator, CHAMP [5]. During floor planning, CHAMP estimates the areas of Standard-Cell blocks by using empirical formulas obtained by running numerous layout experiments.

AMBER, a knowledge-based area estimation assistant [7], is an iterative intelligent notebook that allows designers to consult various expert programs (consisting of estimation heuristics) as they estimate module area. AMBER breaks modules down to a level where at least one of the experts or the designer can estimate the submodule area.

No published results for area estimation for Full-Custom modules currently exist.

3. The Structure of the Module Area Estimator

The module area estimator needs these two input files for the estimation task: the circuit schematic expressed in a standard hardware description language and the fabrication technique or process data base for the particular technology used to fabricate the chip. Multiple process data bases can be stored in the computer system to describe various VLSI technologies. The process data includes the areas of different types of devices, the height of the Standard-Cell rows, and the value of λ, the maximum allowable masque misalignment. The circuit schematic is translated into a mathematical representation for numerical analysis.

For each layout methodology (Standard-Cell and Full-Custom), the module area and one or two module aspect ratios are estimated. These results are stored in a data base, which also contains the global module descriptions and global interconnections for the whole chip. This data base is input to the floor planner. Figure 1 shows the structure of the module area estimator and its input/output interface, which performs the data format translations.

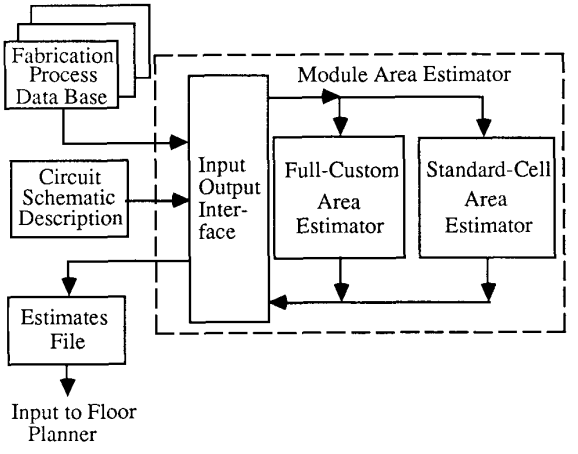

Figure 1: Structure of the Module Area Estimator

4. Algorithms for Area Estimation

We now discuss the algorithms for module area estimation for both the Standard-Cell and the Full-Custom layout methodologies. The following parameters are used throughout this paper:

- N: the number of devices
- n: the number of rows
- D: the number of components in a net
- H: the number of nets
- M: the number of feed-throughs
- W_i : the width of each distinct type of device
- X_i : the number of devices of the same type which have the same physical width
- Y_i : the number of nets which have the same number of components.

4.1 The Standard-Cell Layout Methodology

Estimation of Standard-Cell layout area focuses mainly on the routing problem, since routing area usually occupies more space than active-cell devices, even for a mid-size chip. The size of the routing area strongly depends on the interconnection strength among devices.

In the past, both *one-row* and *n-row* models have been used with the Standard-Cell layout methodology. In the one-row model, all devices are simply placed in one row. A one-row model can be converted into a n-row model by *folding* [5] the single row into n equal-length rows.

There are three unknowns during module area estimation:

- The locations of specific devices
- The number of tracks in each routing channel
- The number of wires sharing each routing track.

The number of Standard-Cell rows may be adjusted during the estimation process, in order to obtain acceptable aspect ratios; the number of routing channels will be naturally determined by the number of rows. Rows consist of two portions: active cells and feed-throughs. If we assume that each row has the same number of active cells, then we can estimate the number of feed-throughs. This allows a calculation of the *expectation value* of the total number of tracks. Finally, the module area, which consists of active cell area and wiring area, can be estimated.

The inputs to the estimation task are N (the number of devices), W_i (individual device widths), and H (the number of nets). A scan of the circuit schematic (which consists of logic gates and transistors) will produce these values. The common assumption for the Standard-Cell layout methodology is that *cells have the same height, but different widths.* The authors of PLEST also assumed that *cells also have (roughly) the same width.* We now make this simplification more precise and reasonable by defining W_{av} to represent the average cell width:

$$W_{av} = \frac{1}{N} \times \sum_{i=1}^{k} X_i \times W_i \qquad (1)$$

N, W_i and X_i are defined above and k indicates the number of different device types. W_{av} is the average device width in the module, and is used for area estimation.

In the Standard-Cell layout methodology, the total routing area is the sum of each routing channel area, calculated by multiplying the height and length of each channel. All routing channels and rows have the same length. The Standard-Cell placements are unknown, so we must make these assumptions:

1. Assume that the rows are long enough to be occupied by the net which connects the largest number of devices. This assumption allows us to consider each cell placement independently and holds for real fabrication technologies, because the number of components in a net is normally far less than the number of devices in a row.

2. Assume that feed-throughs are straight lines crossing one or more Standard-Cell rows. This simplification is necessary, because the number of rows a feed-through crosses is unknown.

3. Assume that each routing track only contains one signal net. This assumption leads to an upper bound estimate for total routing area.

One net needs at least one track in a routing channel, because even when all Standard-Cells attached to a net are placed in one row, they are usually wired through a routing channel. An i-component net usually needs at most i routing tracks (in i different channels) for its wiring.

The exact number of routing tracks needed for a net cannot be calculated. However, we can obtain the probability of placing the D components of a net in different numbers of rows:

$$P_{rows}(i) = (\frac{1}{n})^k \times \binom{n}{i} \times b[i]; \qquad 1 \le i \le k \tag{2}$$

$$k = minimum\ (n, D)$$

$$b[1] = 1$$

$$b[i] = i^D - (\sum_{j=1}^{i-1} \binom{i}{j} \times b[j])$$

In Equation 2, i indicates the assumed number of rows in which the components of a net are placed. The probability of each component being placed in one specific row out of n rows is $1/n$. The first factor gives the probability that all components of a net are placed in i rows. The exponent, k, is the minimum of n and D because the number of rows, n, may be less than the number of components of a net, D, and there are only n ($n<D$) components which are placed in rows with the probability of $1/n$. The remaining components are placed in any row. The second factor is the combination of choosing i rows from the total of n rows. Finally, $b[i]$ is the number of ways of placing D components in exactly i rows and $b[1]=1$. For higher i values, the first term of $b[i]$ gives the number of ways to place D components in i rows and the second term subtracts the number of ways to place D components in fewer than i rows (summed from 1 to $i-1$).

Applying a mathematical expectation function to P_{rows}, we calculate the expectation value of the number of rows containing the net by solving the following equation:

$$E(i) = \lceil \sum_{i=1}^{D} i \times P_{rows}(i) \rceil \tag{3}$$

$E(i)$ should be rounded up to the next higher integer. The product of $E(i)$ and Y_i, the number of nets having i components, gives the number of tracks needed for all nets having i components. An expectation value for the total number of tracks for the module is determined by applying Equations 2 and 3 to all nets.

The main part of row length will be the total width of cells which occupy the row, and is given by:

$$Row\_Length_{cell\_portion} = W_{av} \times \frac{N}{n}$$

where N is the total number of devices and n is the total number of rows. Each row is assumed to have this total cell width. The rest of the row length is occupied by feed-throughs. We will find an expectation value for the number of feed-throughs in any row, which allows us to estimate row length and module length.

To simplify the algorithm for estimating the feed-through contribution to row length, one must find the row most likely to contain the feed-through. A net with any number of components can contribute only one feed-through in any cell row because it only needs one wire through the row to connect the two separated parts of the net. A feed-through in the ith row means that at least two components of the net are separately placed in two rows. One of these two rows is above the ith row and the other one is below. For any net with D components, suppose that no components in the net are placed in the ith row. Of the D components in the net, j are placed in rows above the ith row and the remaining $D-j$ components are placed in the rows below the ith row. The probability that the net contributes one feed-through to the ith row (rows are numbered from top to bottom, starting from 1) is as follows:

$$P_{feed-th}(i) = \sum_{j=1}^{D-1} (\frac{i-1}{n})^j \times (\frac{n-i}{n})^{D-j} \times \binom{D}{j} \tag{4}$$

where $1 \le i \le n$, $(i-1)/n$ is the probability that one component is placed in the rows above the ith row, $(n-i)/n$ is the probability that another component is placed in the rows below the ith row, and the last factor is the number of ways that j components can be chosen from a network of D components to be placed above the ith row. When we assume that the net has l components placed in the ith row, we replace D by $D-l$ in Equation 4 and expand it into the general form below:

$$P_{feed-th}(i) = \sum_{l=0}^{D-2} \binom{D}{l} \times Z[l] \tag{5}$$

where $1 \le i \le n$ and

$$Z[l] = \sum_{j=1}^{D-l-1} (\frac{i-1}{n})^j \times (\frac{n-i}{n})^{D-j-l} \times \binom{D-l}{j}$$

The first factor in Equation 5 is the number of ways l components can be chosen from a total of D components in the net to be placed in the ith row. For a net causing a

feed-through in the ith row, l ranges from 0 to $D-2$ because at least one component is placed in each of the top and bottom parts. Thus, the total probability of a feed-through caused by a net in the ith row is the sum of l from 0 to $D-2$. The other factors appear as in Equation 4, but with the upper bound D changed to $D-l$. Next, we take the partial derivative of Equation 5 with respect to i :

$$\frac{\partial}{\partial i}P_{feed-th}(i)=\sum_{l=0}^{D-2}\binom{D}{l}\times T[l] \tag{6}$$

where $1\le i\le n$ and

$$T[l]=\sum_{j=1}^{D-l-1}\binom{D-l}{j}\times\frac{\partial}{\partial i}[(\frac{i-1}{n})^{j}\times(\frac{n-i}{n})^{D-j-l}]$$

We find a maximum by setting the partial to zero:

$$\frac{\partial}{\partial i}[(i-1)^{j}\times(n-i)^{D-j-l}]=0 \tag{7}$$

We find that the solutions:

$$i=1,\quad n,\quad or\quad \frac{D-l+j\times(n-1)}{D-l}$$

will satisfy Equation 7, but only the last one is useful because generally neither the top row nor the bottom row have feed-throughs. Assuming that l=0, D=2 and j=1, we find that i=$(n+1)/2$. That is, for a net with two components, neither of which is in the ith row, the central row has the largest probability of containing the feed-through caused by the net. Numerical simulation results show that in the intervals $0\le l\le D-2$ and $1\le j\le D-l-1$, the central row always has the largest probability of containing a feed-through. Another way to illustrate this conclusion is to assume that the area of one row is small compared to the whole module area and divide the module area into two parts (top and bottom) by the ith row. Then, the probabilities of each component being placed in the top or bottom parts are, respectively, the ratio of the top part area to the whole module area and the ratio of the bottom part area to the whole module area. The probability of a feed-through is proportional to the product of these two ratios. Thus, equal-sized top and bottom areas will maximize this product. Now, by replacing i with $(n+1)/2$ in Equation 5, we obtain Equation 8, which gives the probability of a feed-through in the central row:

$$P_{feed-th}=\sum_{l=0}^{D-2}\{\binom{D}{l}\times\sum_{j=1}^{D-l-1}(\frac{n-1}{2n})^{D-l}\times\binom{D-l}{j}\} \tag{8}$$

Further numerical simulation results also show that the central row always has the largest probability of containing a feed-through, regardless of the value of D. Thus, we can use a two-component-net as a model instead of a D-component-net. Therefore, Equation 8 can be further simplified, as follows, because there are only two ways to choose a component to place in the top part from the total of two components.

$$P_{feed-th}=2\times(\frac{n-1}{2n})^{2} \tag{9}$$

$$P_{max-feed-th}=\lim_{n\to\infty}P_{feed-th}=0.5$$

Applying Equation 9 to M feed-throughs among a total of H nets, we find that the probability of M feed-throughs occurring on the i=$(n+1)/2$ th row is:

$$P_{M}=P_{feed-th}{}^{M}\times(1-P_{feed-th})^{H-M}\times\binom{H}{M} \tag{10}$$

where $1\le M\le H$. The first term in Equation 10 gives the total probability of M nets having feed-throughs in the ith row, the second term gives the probability of $H-M$ nets having no feed-throughs in the ith row, and the last term is the number of ways choosing M nets from a total of H nets. Again, we obtain the expectation value of M (rounded up to an integer):

$$E(M)=\lceil\sum_{M=1}^{H}M\times P_{M}\rceil \tag{11}$$

Finally, the estimated module layout area is the product of the total height of n rows plus all routing tracks, multiplied by the row length including feed-throughs:

$$A_{std\_cell}=[n\times r\_h+\sum_{i=1}^{D}Y_{i}\times E(i)]\times[W_{av}\times\frac{N}{n}+E(M)\times f\_w]$$

where f_w is the feed-through width and r_h is the row height.

4.2 The Full-Custom Layout Methodology

The following standards are used in practice for the Full-Custom layout methodology:

- *Minimum area* or *maximum area utilization*
- *Minimum interconnection length*
- *Minimum length critical path.*

As for Standard-Cell layouts, the estimation of Full-Custom layout area consists of estimated device (cell) area and estimated interconnection (wiring) area. Because the device area can be computed directly from the circuit description, we only need to estimate the interconnection area.

We can estimate the interconnection area of a module by assuming that the transistors connected to the same net are placed into two rows of equal length, with a one-track routing channel between them. In other words, a two-row Standard-Cell model with a one-track routing channel is used to calculate Full-Custom layout interconnection area for each net, and individual transistor layouts are used as Standard-Cells instead of typical Standard-Cell devices (logic gates and flip-flops). For any multiple-component net, the width of the net interconnection area is the width of a one-track routing channel and the module length is half of the device row length (half of the net components), rounded up to an integer. This is the minimum interconnection area, assuming that all components in the net are strongly connected. Each component may connect to more than one net. When the multiple nets are are not placed closely together, the above estimation method may understate the interconnection area and area estimation becomes more difficult.

The estimated area can now be calculated from the total area of devices and nets. Additionally, this estimation

method is easily mapped into the maximum area utilization standard. Because active-cell area is constrained, reducing the module area only reduces the connection area. Therefore, this minimum area estimation method provides a lower bound, according to the minimum connection length standard. This minimum area estimation is first performed using exact device areas and again performed using the average device area. The estimated areas are:

$$Area_{full\_cust\_ex} = \sum_{i=1}^{k} X_i \times W_i \times h_i + \sum_{j=1}^{H} A_j \quad (12)$$

and

$$Area_{full\_cust\_av} = N \times W_{av} \times h_{av} + \sum_{j=1}^{H} A_j \quad (13)$$

where k is the number of different device types, h_i is the height of each device, A_j is the interconnection area estimate for each net, h_{av} is the average device height, and H is the total number of nets.

5. Aspect Ratio Estimation

Currently, we estimate the module aspect ratio by dividing the estimated module area by the length along a module side in which all input and output ports can be fitted. The estimator may iterate several times to produce acceptable aspect ratios. We use the control criterion that all input and output ports must fit along any one of the four layout edges or at least along one of the longer edges.

The Standard-Cell aspect ratio estimator requires an estimate of the number of rows. The initial number of rows is obtained by dividing the square root of total active cell area by twice the height of a Standard-Cell row. If the initial row length, not counting feed-throughs, is greater than the length of all input and output ports for the module, the estimator will first estimate module area and then estimate the aspect ratio. Otherwise, the estimator will adjust the initial number of rows until the row length fits this requirement. The following algorithm chooses an initial value of n (the number of Standard-Cell rows):

1. Let $i = 2$
2. Let $n = \lceil \sqrt{\frac{active\_cell\_area}{i \times row\_height}} \rceil$
3. Let $row\_length = \frac{active\_cell\_area}{n \times row\_height}$
4. If all input and output ports can fit within this row_length, the current value of n is accepted. Otherwise, increment i and go to *Step 2.*

Therefore, the module aspect ratio estimate is:

$$Aspect_{std\_cell} = \frac{W_{av} \times \frac{N}{n} + E(M) \times fd\_thru\_w}{n \times row\_height + \sum_{i=1}^{D} Y_i \times E(i)} \quad (14)$$

For the Full-Custom layout methodology, the module aspect ratio estimation algorithm is as follows:

1. Assume an aspect ratio of *1:1* by using the square root of the estimated module area as both module width and height.
2. Compare this module width with the total length of all the module input and output ports.
   a. Edge length $\neq$ total port length. Divide the estimated module area by the length of the module input and output ports to obtain the aspect ratio.
   b. Edge length = total port length. Use an aspect ratio of *1:1.*

6. Experimental Results

Experiments for the Full-Custom layout methodology with exact device areas (see Table 1) show that the estimated areas for small and moderate-sized modules are very close to the areas of manually-created layouts for the same circuits. Comparisons are made against Newkirk and Mathews' [8] Full-Custom layout examples for nMOS technology with λ=2.5μm using the Mead-Conway design rules. In Experiment 5 using exact device areas, the estimated area was only 2.6 percent greater then the actual area. In Experiment 2 using average device areas, the estimate was only 1.5 percent greater than the actual area. Area estimates ranged from a 17% underestimate to a 26% overestimate. The average estimation error was 12%. The estimator computed for less than 1.5 CPU seconds on a Sun 3/50 computer for all examples.

Table 1: Full-Custom Module Layout Area Estimates

Experiment #	1	2	3	4	5	6	7
# Devices	43	23	13	31	6	31	30
# Nets	31	16	9	23	6	24	22
# Ports	14	13	5	14	6	11	7
Device Area (λ^2)	8960	5136	3120	6832	2496	9414	9254
Estimated Wire Area (λ^2) Using These Device Areas:							
Exact areas	3203	1579	847	2202	0**	2100	2016
Ave. areas	5659	3034	1627	3949	857	4758	4058
Total Estimated Area (λ^2) Using These Device Areas:							
Exact areas	12163	6715	3967	9034	2496	11514	11270
Ave. areas	14619	8170	4747	10781	3353	14172	13312
Real Area (λ^2)	11430	8052	3569	7174	2432	12402	12402
Estimated Aspect Ratio Using These Device Areas:							
Exact areas	0.58	0.91	0.23	0.78	0.52	1.00	1.00
Ave. areas	0.48	0.75	0.12	0.65	0.39	1.00	1.00
Real Aspect Ratio	0.14	0.24	0.52	0.16	0.42	0.49	0.49

**All nets in this module were two-component nets, and therefore contributed nothing to wire area.

Two experiments to estimate Standard-Cell layout area are shown in Table 2. The estimates are compared with Standard-Cell layouts for the same circuits created by the TimberWolf [9] Standard-Cell placement and routing package, using NMOS Standard-Cell layouts created at Rutgers. We show three estimates associated with three different initial values of n, the number of rows, for Experiment 1 and two for Experiment 2. The area estimates were related to the number of Standard-Cell rows in both examples because the area estimate decreased as the number of rows increased. Area estimates ranged from a 42% overestimate to a 70% overestimate. We believe that these overestimates occur because the estimator ignores track sharing in routing channels, which is especially significant in larger designs. The estimator also fails to consider that inter-module wires in some technologies are routed through Standard-Cells, as well as through channels. The estimator computed for less than three CPU seconds on a Sun 3/50 for each Standard-Cell example.

Table 2: Standard-Cell Module Layout Area Estimates

Experiment #	1	1	1	2	2
# Rows	4	5	6	4	5
# Devices	41	41	41	43	43
# External Ports	11	11	11	20	20
Est. Module Height (λ)	342	370	408	385	418
Est. Module Width (λ)	129	108	94	97.8	85.7
# Tracks Estimated	46	46	48	49	50
Real	18	19	21	13	13
Total Est. Area (λ^2)	44118	40034	38488	37653	35808
Real Area (λ^2)	25856	24675	27027	22960	21436
Est. Aspect Ratio	0.38	0.3	0.23	0.25	0.2
Real Aspect Ratio	0.63	0.45	0.36	0.55	0.4

The estimated module aspect ratios are hard to match with exact ones because we cannot know the number of input and output ports on each module edge (since floor planners move ports from edge to edge). Even if port locations are known, there are still many ways to satisfy the control criterion since a short length of module input and output ports always fits along a longer module edge. Generally, the estimator chooses an initial aspect ratio in the range from *1:4* to *1:1*, since aspect ratios of most manually laid out modules fall in this range.

The estimator is written in the C language, and runs on a Sun model 3/50 work station under Berkeley Unix***.

***Trademark for an operating system developed at AT&T Bell Labs.

7. Conclusions

The module area estimator makes two contributions to automatic VLSI chip layout. First, it generates realistic Full-Custom module area estimates for floor planning; no published results exist for Full-Custom module area estimation. Module area and aspect ratio estimates are provided for both the Full-Custom and Standard-Cell layout methodologies. Secondly, it significantly reduces the floor planning time and design cost since more accurate initial area estimates reduce the number of floor planning and layout iterations required to complete the whole chip. The estimator works well for small and moderate-sized modules, but is not intended for area estimation of entire chips.

In the future, additional experiments will be run, the estimator will be changed to account for routing channel track sharing in Standard-Cell layouts, and the estimator will be changed to output four or five aspect ratio estimates to allow chip floor planners more flexibility in choosing module shapes. Also, we will determine the reduction in floor planning iterations due to the estimator by testing it on larger designs.

References

[1] C.M. Gerveshi, "Comparisons of CMOS PLA and Polycell Representations of Control Logic", *Proceedings of the 23rd DAC*, June 1986, pp. 638-642.

[2] D.P. LaPotin and S.W. Director, "Mason: A Global Floorplanning Approach for VLSI Design", *Transactions on CAD*, Vol. CAD-5, No. 4, October 1986, pp. 477-489.

[3] M.L. Bushnell, *Ulysses -- An Expert-System Based VLSI Design Environment,* PhD dissertation, Carnegie Mellon University, 1987.

[4] F.J. Kurdahi and A.C. Parker, "Plest: A Program for Area Estimation of VLSI Intergrated Circuits", *Proceedings of the 23rd DAC*, June 1986, pp. 467-473.

[5] K. Ueda, H. Kitazawa and I. Harada, "CHAMP: Chip Floor Plan for Hierarchical VLSI Layout Design", *IEEE Transactions on CAD*, Vol. CAD-4, No. 1, January 1985, pp. 12-22.

[6] M.M. How and B.Y.M. Pan, "Amber: A Knowledge-Based Area Estimation Assistant", *Proceedings of ICCD*, Oct. 1986, pp. 180-183.

[7] B.Y.M. Pan, M. How and A.J. Kuchinsky, "Knowledge-Assisted Design and the Area Estimation Assistant", *Hewlett-Packard Journal*, Vol. 37, No. 6, June 1986, pp. 8-9.

[8] J. Newkirk and R. Mathews, *The VLSI Designer's Library,* Addison-Wesley, 1983.

[9] C. Sechan, University of California, Berkeley, *The TimberWolf 3.2 Standard Cell Placement and Global Routing Program User's Guide,* 1986.