Predicting the Number of Vias and Dimensions of Full-custom Circuits Using Neural Networks Techniques

Marwan A. Jabri & Xiaoquan Li

Systems Engineering and Design Automation Laboratory

Sydney University Electrical Engineering

NSW 2006 Australia

SEDAL Technical Report 1991-1-6

Abstract

Block layout dimension prediction is an important activity in many VLSI design tasks (structural synthesis, floorplanning and physical synthesis). Block layout *dimension* prediction is harder than block *area* prediction and has been previously considered to be intractable [6]. In this paper we present a solution to this problem using a neural network machine learning paradigm. Our method uses a neural network to predict first the number of vias and then another neural network that uses this prediction and other circuit features to predict the width and the height of the layout of the circuit. Our approach has produced much better results than those published, *dimension* (aspect ratio) prediction average error of less than 18% with corresponding *area* prediction average error of less than 15%. Furthermore, our technique predicts the number of vias in a circuit with less than 4% error on average.

I. Introduction

This paper is concerned with the prediction of the layout dimensions (width and height and not simply the area) of rectangular full-custom blocks. Figure 1 shows the input and output of the prediction process. The inputs consist of a circuit netlist (gate or transistor level), transistor sizes (or average transistor area), and information about the floorplan of the higher level block (where the present circuit layout will fit). The floorplan information consists of the number and type of ports (data, clock, control, Vdd and

Vss). The outputs of the prediction process are the estimate of the width and height of the circuit layout.

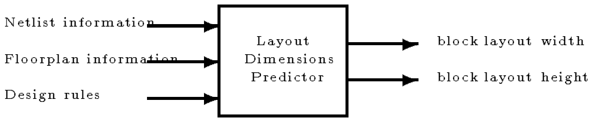

Figure 1: Input and output of the block layout dimension prediction process.

Block layout dimension prediction is an important activity in the structural synthesis, floorplanning, and (physical) layout synthesis tasks of a VLSI design process. The availability of block dimension predictions based on circuit netlist (gate or transistor level, not at the mask level), average transistor size, and ports (terminals) distribution on block periphery, would give a valuable information for inclusion in a cost function that may be used for speed and area optimisation in these tasks. For example in structural synthesis, block dimension is important when the target technology makes use of cell synthesis and automatic layout generation. In floorplanning, block dimension prediction is important for an iterative generation of floorplans [5,4], and in cell synthesis, it is important for the partitioning and floorplanning of complex cells [1]. In all of these activities, if block dimension predictions are produced at a low computing costs they may be used in the optimisation cost function with little overheads.

Block layout dimension prediction is also useful as a separate stand-alone tool that designers may consult to explore block shapes and evaluate their effects on chip speed early in the design process.

The layout dimensions prediction problem has been previously approached as an extension to area prediction [6,2,10]. Furthermore, full-custom block layout dimension prediction has been considered intractable previously [6] because of the randomness of full-custom circuits.

In this paper we describe a novel solution to this problem using a neural network based machine learning paradigm. Our method uses a neural network to predict first the number of vias and then another neural network that uses this prediction and other circuit features to predict the width and the height of the layout.

We have collected over 54 circuits and their full-custom layout from two different sources and split these layouts into two sets (34 for training and 20 for testing). The circuits were chosen as not to bias the network towards any design style or block types (ALUs, registers, buffers, random blocks, *etc*...). Features are extracted from the training set and used to train two multi-layer perceptrons using gradient descent. Training was performed until all training patterns produce block dimensions that fall within a margin. Once the networks are trained, they were able to generalise with an average error on aspect ratio that is less than 18% and an average area error that is less than 15%. Furthermore, our method predicts the number of contacts (vias) in a block with an average accuracy of less than 4%.

The paper is structured as follow. Section II. reviews previous work in this area. In Section III. the layout dimension prediction problem is briefly studied and key characteristics features outlined. Then Section IV. presents our methodology and describes the the structures of the neural networks and their operations. In Section V., we discuss our experiments and present testing results. Finally, in Section VI. we draw some conclusions and present directions for future work.

II. Previous Work

Many researchers have addressed the problem of area prediction and dimension prediction as a by-product.

Reis [8] has reported an empirical study for estimating VLSI topologies. How and Pan [3] have used a knowledge-based system methodology to allow VLSI designers to interact with a number of area prediction "experts" and to explore block areas. Beside that their technique did not address the problem of dimension prediction for full-custom blocks, a knowledge-based technique is unpractical with respect to "knowledge growth", that is, the expansion of the knowledge-base is a tedious task and requires the intervention of the program designer. Knowledge acquisition is known to be problematic to expert systems growth and although machine learning is advancing as field, robust knowledge acquisition tools is still a challenging problem. Kurdahi and Parker[6] have developed a probabilistic model the prediction of wiring space for standard cell circuits. Aspect ratio prediction is performed by the selection of an optimal number of columns and rows. Zimmerman [10] has reported a technique for area and shape prediction with an average error for standard circuits ranging between 5 and 10%. His technique was successfully applied for circuit with an aspect ratio ranging from 1:5 to 5:1 and is based on shape function. No published results were provided for full-custom circuits. Although shape functions represent an interesting concept, the building of these functions for general full-custom blocks is tedious. Chen and Bushnell [2] reported an empirical model for area and aspect ratio prediction for standard cells and full-custom. Although the area prediction results were encouraging, the aspect ratio results they have reported were excessive (with an average over 50%).

III. The Block Layout Dimension Prediction Problem

A. Block Layout Features

To predict the layout dimensions of full-custom circuits one needs to identify key features that characterise layout. The key features commonly identified and used are:

- number of transistors,
- area of transistors, and
- number of netlists (multi-point or point-to-point netlists).

All of the features above give an indication on block area, but none however characterises the dimensions (width or height) as such. Understandably, we cannot expect to obtain good dimension predictions on the basis of the features above only.

Our research has indicated that (not surprisingly) in addition to the features above, the following features have a direct effect on the prediction of block dimensions:

- number of ports on each side and their type (data, clock, control).
- number of Vdd and Vss ports on each side.
- number of vias in block.

Clearly, although designers are able to specify the number of ports (data, control, clock, Vdd and Vss), they do not possess any clues on the number of vias. Therefore, this number of vias has to be accounted for either directly using prediction techniques or indirectly by treating it as a *hidden* feature. We explain how to predict accurately it later in the paper. In addition, our study of a very large number layouts indicates that most of them include blank areas resulting from the process of pitch matching of ports and therefore ambiguity would result when evaluating the performance of block dimension prediction systems. As a result an additional feature/parameter is needed, to characterise the packing density of the layouts.

B. Hidden, Complex Layout Features and Machine Learning

The features we have described above are by no means independent of each other, nor they completely characterise layout dimensions. The number of vias is related to the number of netlists and the number and type of ports and so on. Therefore, it is necessary that dimension prediction procedures consider the inter-relations between features and extract the hidden or complex relationships that exist between them.

The extraction of hidden and complex relationships between features may either be done manually or automatically. A manual extraction would be tedious and of high complexity. The automatic extraction corresponds to the commonly known process of *machine learning*. However, not any machine learning algorithm can be used in the block prediction task. As we are dealing with a *prediction* problem, *machine learning* algorithms such as *ID3* [7] are not appropriate as they are more targeted towards *classification* problems.

A more suitable machine learning technique that can extract the hidden relationships and produce a system that makes use of all features is backpropagation [9], operating on an appropriate neural network with the known features presented at its input and it output representing the block width and heigh. The next section presents this approach and a system that implements it.

IV. A Neural Network Machine Learning Approach to the Prediction of Number of Vias and Full-Custom Layout Dimensions

Our layout dimension prediction technique is based on machine learning using neural networks. It makes use of the capability of neural networks to generalise once they are trained, that is, to be able to predict the dimensions of blocks that they have not seen previously using the information they have learned from the training data. Our system has two neural networks in cascade as components: The number of vias prediction sub-system (NVPS) and the block layout dimension prediction sub-system (LDPS). NVPS predicts the number of vias, which is fed together with other features to the LDPS to produce the prediction of layout block width and height. A block diagram of the system is shown in Figure 2.

Both the NVPS and LDPS are feedforward multi-layer perceptrons. Two types of “neuron” transfer functions are used: sigmoidal and hyperbolic tangents. In both sub-systems all neurons except those in the output layer are sigmoidal. The neurons of the output layer are hyperbolic tangents. The range of the sigmoidal function is between 0 and 1, with a common non-linear behavior. The sigmoidal functions are responsible for the non-linear mapping of features. The choice of hyperbolic tangents for the output layers, however, was based on the need of a differentiable mapping function with a good linear behavior over all the range of prediction (0 to 1). The transfer function of the output neurons has the form:

$$O_j = Gain * tanh(\sum_{i=0}^{i=N-1} O_i * w_{ij})$$

Where N is the number of neurons in the hidden layer feeding into the output layer, j is the index of the output neuron (one output for the NVPS and two outputs corresponding to the width and height in the case of the LDPS), w_{ij} is the weight from neuron i to neuron j, and finally $Gain$ is a constant selected to push the saturation region of the hyperbolic tangent far above 1. Several values of the Gain constants have been evaluated with the best results obtained for a value of 2.2.

The input to the number of vias prediction sub-system are the following:

N_{trans}: Total number of transistors. If gates are involved then the number of transistors in these gates is used.

$T_{average\ area}$: Average transistor area. This input is calculated automatically from transistor sizes and number.

N_{nodes}: Total number of netlists in the block (Spice-like nodes).

$Ports_{side}$: Total number of ports on each of the four sides.

$Data_{side}$: Total number of data ports on each of the four sides.

Ctl_{side}: Total number of control ports on each of the four sides.

Clk_{side}: Total number of clock ports on each of the four sides.

$Power_{side}$: Total number of Vdd and Vss ports on each of the four sides.

$Pack$: This is a parameter between 0 and 1 aimed at controlling the expected density of the final layout. A default value is supplied if user is unable to specify this parameter.

The output of the NVPS is a floating value between 0 and 1.

The inputs to the LDPS are both the inputs and the output of the NVPS, and its outputs are two floating values between 0 and 1 representing the predicted width and height of the block layout.

A. *Training*

As mentioned earlier a backpropagation training algorithm [9] was used to train separately the NVPS and LDPS. The "on-line" version of the backpropagation is being used in that weight update is performed after the presentation of each pattern. The convergence criterion is satisfied when either an error criterion is met or when all patterns produce an output which is within an error percentage margin. The least mean square rule is used to express the errors at the output.

V. Experiments

A large number of experiments have been conducted to evaluate:

- Role and effects of layout features,
- Network architectures (number and size of hidden layers),
- Error percentage margin for good training and generalisation,

Before we summarise our results we describe in the following paragraphs our training and testing data and the pre-processing procedure.

A. *Training and Testing Data*

Our training and testing data was collected from two sources (in contrast with other published techniques which make use of single source data like in [2]). The first source is the VLSI Designer's Library which data is used in [2]. The layout from the second source are full-custom blocks developed in our laboratory. Altogether 54 layouts have been assembled, all are digital and correspond to a 5μ nMOS process, and represent circuits of a variety of functions. Layouts that correspond to regular structures like PLA, RAM, ROM have been excluded. The reason is that these blocks are commonly produced by "module generators" and their dimensions are predictable and exact solutions often exist for their predictions.

Note that although our training and testing data are technology dependent, the actual approach is not.

The collection of data for such a task is not easy as most companies that develop full-custom methodologies treat their circuits as a matter of secrecy.

B. Data Pre-Processing

The only pre-processing step performed on the data is normalisation. This step is performed independently on each input and output. Final values are between zero and one. Note this process will impose a limit on the maximal size of blocks. This limitation is not desirable, but may be overcome by normalising the data to very large values of features.

C. Simulations and Results

Training and testing have been performed for the NVPS and LDPS separately to evaluate their performance on their specific tasks, and combined to evaluate the performance of the overall layout dimension prediction system.

Testing results of the NVPS are shown in Table 1. Only 16 test circuits are shown here to illustrate the system performance. The average error of the NVPS on the overall testing data (20 samples) is less 4%. Many network architectures have been evaluated with the best performance being achieved by a 24-24-10-10-10-8-6-1 architecture.

It is important to note that much better generalisation performance has been recorded when the number of layers is increased while the size of the lower layers (number of units) is decreased. This is not a surprise as it is known (and it can be proven) [1] that for smaller training set sizes, networks generalisation performance improves with increasing number of layers.

Testing results of the LDPS using actual number of contacts (not the predicted ones) are shown in Table 2. All actual width, length and area of blocks are in λ and in λ^2 respectively. The predicted ones are in the same units except that floating point output values of the networks are shown for the sake of consistency with the percentage errors (last two columns of Table 2). The average error of the LDPS with actual number of vias is less than 12% on area and less than 18% on aspect ratio. An interesting prediction is that of the block Dpreg (a Dual Ported Register) where the error on the aspect ratio is about 78% and that of the area is about 28%. An investigation of the actual layout indicates that the high error on the aspect ratio in this case may be due to the fact the LDPS has produced dimension predictions of a "rotated" version of the layout. Many architectures for the LDPS have been investigated with a 25-25-10-10-10-8-6-2 having achieved the best results given the size of our training set.

[1] Private communication from Sara Solla, November 1990

Testing results of the combined NVPS and LDPS are shown in Table 3 which also shows a comparison between the performance of the network on the actual number of vias and on that of the predicted ones. Understandably the performance of the overall system drops slightly on average at the area prediction level with average error rising from about 12% to 15%. However, the performance of the overall system on average with respect to the aspect ratio when the predicted number of vias is used has not changed noticeably as compared to that on the actual number of vias with an average error still under 18%. This suggest that the LDPS has on average reacted evenly on both dimensions. Interestingly, the error on the area for the Dpreg block has dropped to just under 6% (from about 28% with the actual number of vias) while the aspect ratio error has risen from about 78% to about 83%. The drop in the predicted area is explained by the lower predicted number of vias for this block.

A comparison of the performance with actual and predicted number of vias fed in the LDPS is shown in Table 4. The right most block of this table shows the percentage errors for the two cases.

A summary of the performance of our layout dimensions prediction system is shown in Table 5. We have excluded here the results obtained for other architectures, but it is interesting to note that as expected the performance of all architectures (for NVPS and LDPS) improves quickly as the number of layers increases from three to seven and then remains almost stable . As expected the smaller the error percentage margin is set during training the more "specialised" the network becomes. Setting this margin to less than 20% leads to networks with poor generalisation performance.

VI. Conclusions

We have presented in this paper a novel approach to the layout dimensions prediction problem. Our approach makes use of neural network based machine learning paradigm and operates by predicting first the number of vias that a circuit would need and then uses this prediction together with other circuit features to predict the dimensions of the layout. Our method has produced much better dimension prediction results than those published. The average error of our prediction system is less than 15% on area and 18% on aspect ratio. Furthermore, to our best knowledge this has been the first attempt on estimating the number of vias that a circuit would need for its layout. Our results need consolidation by further testing on a larger test set. The simplicity of the machine learning paradigm that we have used makes possible the inclusion of such a technique in a CAD system to provide on-line dimension estimations. Furthermore, the system is capable of continuously learning by adding new data to the existing data set and retraining. This may require a more automated architectural evaluation system with probably a cross-testing facility. Our future work will implement this technique for full-custom CMOS technology and will investigate its applicability to the prediction of wiring and cell areas in standard cell based layout.

VII. Acknowledgments

This work was partially funded by the Key Centre For Design Quality (University of Sydney). One of us (XL) acknowledges the support of an Overseas Postgraduate Research Award.

References

[1] Donald George Baltus. *Generating Efficient Layouts from optmized MOS circuit schematics.* Master's thesis, MIT, 1988.

[2] X. Chen and M.L. Bushnell. A module area estimator for vlsi layout. In *Proceedings of 25th ACM/IEEE Design Automation Conference*, pages 54–59, 1988.

[3] M.M How and B.Y.M Pan. Amber: a knowledge based area estimation assistant. In *Proceedings of ICCD*, pages 180–183, 1986.

[4] M.A. Jabri and D. Skellern. PIAF — Efficient IC Floorplanning. *IEEE Expert*, 4(2):33,45, 1989.

[5] M.A. Jabri and D.J. Skellern. PIAF: A Knowledge-Based/Algorithmic top-down floorplanning system. In *Proceeding of the 26th ACM/IEEE Design Automation Conference*, pages 582–585, Las Vegas,USA, 1989.

[6] F.J. Kurdahi and A.C. Parker. Techniques for area estimattion of vlsi layouts. *IEEE Transactions on Computer-Aided Design*, 8(1):81–92, January 1989.

[7] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[8] R.A. Reis. A topological evaluator as the first step in VLSI design. In *Proc. Microelectronics 82 Conference*, pages 22–26, Adelaide, Australia, 1982.

[9] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing: Explorations in the microstructure of cognition.* Volume I, MIT Press, Cambridge, Massachusetts, 1986.

[10] Gerhard Zimmerman. A new area and shape fucntion estimation technique for vlsi layouts. In *Proceedings of 25th ACM/IEEE Design Automation Conference*, pages 60–65, 1988.

List of Figures

List of Tables

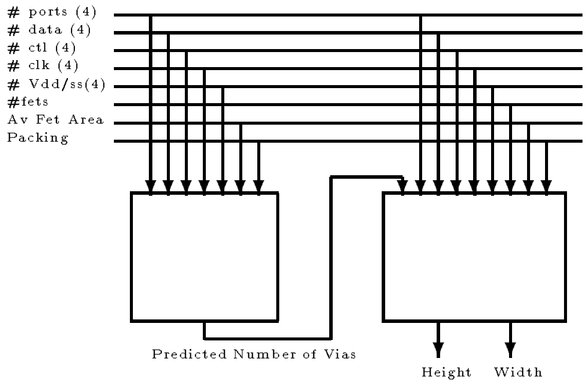

Figure 2: Block digram of the layout dimension prediction system.

Table 1: Test results of the NVPS sub-system. Via units are expressed in terms of number of $4\lambda x 4\lambda$ contacts.

Cell Name	Contacts		
	Actual	Predicted	Error (%)
CntUDRestore	46	46	0.00000%
SannClkIn	15.5	15	3.22580%
ScanIn	10.5	9	14.28570%
StScanUnClkIn	17	17	0.00000%
StScanJoin	21	20	4.76190%
S_AddSub	71.5	72	0.69931%
TimingDelay	62.5	50	20.00000%
Addcorebuf1	33.5	33	1.49253%
Addernobbuf	22	18	18.18180%
Adderlatch	15	10.5	30.00000%
Dpreg	18	9	50.00000%
StaticAdder	44.5	47	5.61797%
PlaClockout	10	10	0.00000%
PlaIn	6	7.5	25.00000%
PlaNorOut	5.5	6.5	18.18180%
ScanUnClkIn	14.5	13	10.34480%

Table 2: Testing results of the LDPS using actual (not predicted number of vias. Widths and heights are expressed in λ and area in λ^2 . The aspect ratios are simply the width divided by the height.

Cell Name	Actual				Predicted				Error (%)	
	Width	Height	Area	Asp. Ratio	Width	Height	Area	Asp. Ratio	Area	Asp. Ratio
CntUDRestore	44	183	8052	0.2404371	49.35	186.87	9222.09	0.2640873	14.537426%	9.841100%
SannClkIn	16	104	1664	0.1538461	17.50	109.53	1916.76	0.1597735	15.698788%	4.312601%
ScanIn	16	104	1664	0.1538461	15.52	108.09	1677.56	0.1435840	0.841321%	6.641101%
StScanUnClkIn	19	111	2109	0.1711711	19.37	113.61	2200.47	0.1704955	4.355929%	0.369802%
StScanJoin	28	112	3136	0.2500000	25.83	115.57	2985.22	0.2235009	4.814823%	10.607388%
S_AddSub	78	159	12402	0.4905660	81.39	173.30	14104.92	0.4696480	13.730093%	4.265085%
TimingDelay	57	200	11400	0.2850000	61.26	204.12	12504.39	0.3001175	9.681199%	5.302803%
Addcorebuf1	124	32	3968	3.8750000	120.40	30.24	3641.77	3.9814815	8.364053%	2.613697%
Addernobbuf	98	32	3136	3.0625000	95.83	31.55	3023.44	3.0374010	4.205879%	0.184541%
Adderlatch	46	32	1472	1.4375000	47.33	29.42	1392.45	1.6087695	5.411569%	11.906848%
Dpreg	66	32	2112	2.0625000	34.74	77.78	2702.08	0.4466443	27.934293%	78.347630%
StaticAdder	34	211	7174	0.1611374	34.16	213.23	7284.10	0.1602026	1.546801%	0.570367%
PlaClockout	16	57	912	0.2807017	12.49	66.52	830.81	0.1877630	8.925773%	33.124918%
PlaIn	16	41	656	0.3902439	17.49	38.24	668.90	0.4573744	1.979536%	17.204086%
PlaNorOut	16	45	720	0.3555555	18.33	43.39	795.33	0.4224475	10.491445%	18.845811%
ScanUnClkIn	16	104	1664	0.1538461	16.45	110.66	1820.31	0.1486535	9.390279%	3.376178%

Table 3: Testing results using the predicted number of vias (overall layout dimension prediction system.

Cell Name	Actual				Predicted				Error (%)	
	Width	Height	Area	Asp. Ratio	Width	Height	Area	Asp. Ratio	Area	Asp. Ratio
CntUDRestore	44	183	8052	0.2404371	49.06	186.18	9133.99	0.2635084	13.439244%	9.591604%
SannClkIn	16	104	1664	0.1538461	17.00	108.53	1845.08	0.1566387	10.888415%	1.817457%
ScanIn	16	104	1664	0.1538461	14.58	106.66	1555.17	0.1366960	6.573277%	11.176521%
StScanUnClkIn	19	111	2109	0.1711711	19.10	113.16	2161.30	0.1687875	2.497550%	1.372130%
StScanJoin	28	112	3136	0.2500000	25.13	114.22	2870.35	0.2200140	8.486510%	12.009279%
S_AddSub	78	159	12402	0.4905660	81.24	174.67	14190.21	0.4651056	14.417760%	5.190910%
TimingDelay	57	200	11400	0.2850000	55.12	167.26	9219.14	0.3295468	19.126510%	15.638768%
Addcorebuf1	124	32	3968	3.8750000	117.94	30.20	3561.54	3.9052980	10.242175%	0.790852%
Addernobbuf	98	32	3136	3.0625000	82.57	30.70	2534.53	2.6895765	19.180283%	12.165052%
Adderlatch	46	32	1472	1.4375000	42.26	29.36	1240.76	1.4393733	15.722563%	0.138722%
Dpreg	66	32	2112	2.0625000	26.26	75.66	1986.83	0.3470790	5.922910%	83.172354%
StaticAdder	34	211	7174	0.1611374	36.24	217.40	7878.58	0.1666973	9.815186%	3.437452%
PlaClockout	16	57	912	0.2807017	12.60	65.49	825.17	0.1923957	9.540050%	31.472398%
PlaIn	16	41	656	0.3902439	17.14	40.49	694.00	0.4233144	5.797664%	8.480812%
PlaNorOut	16	45	720	0.3555555	17.56	46.17	810.75	0.3803335	12.626467%	6.995661%
ScanUnClkIn	16	104	1664	0.1538461	15.57	109.12	1699.00	0.1426869	2.120091%	7.238522%

Table 4: Comparison of test results when the overall system with actual and predicted number of vias.

Cell Name	Contacts		Predicted Area Using		Predicted Asp. Ratio Using		Error (%)			
	Actual	Predicted	Actual Contacts	Predicted Contacts	Actual Contacts	Predicted Contacts	Using Actual Contacts		Using Predicted Contacts	
							Area	Asp. Ratio	Area	Asp. Ratio
CntUDRestore	46	46	9222.09	9133.99	0.2640873	0.2635084	14.537426%	9.841100%	13.439244%	9.591604%
SannClkIn	15.5	15	1916.76	1845.08	0.1597735	0.1566387	15.698788%	4.312601%	10.888415%	1.817457%
ScanIn	10.5	9	1677.56	1555.17	0.1435840	0.1366960	0.841321%	6.641101%	6.573277%	11.176521%
StScanUnClkIn	17	17	2200.47	2161.30	0.1704955	0.1687875	4.355929%	0.369802%	2.497550%	1.372130%
StScanJoin	21	20	2985.22	2870.35	0.2235009	0.2200140	4.814823%	10.607388%	8.486510%	12.009279%
S_AddSub	71.5	72	14104.92	14190.21	0.4696480	0.4651056	13.730093%	4.265085%	14.417760%	5.190910%
TimingDelay	62.5	50	12504.39	9219.14	0.3001175	0.3295468	9.681199%	5.302803%	19.126510%	15.638768%
Addcorebuf1	33.5	33	3641.77	3561.54	3.9814815	3.9052980	8.364053%	2.613697%	10.242175%	0.790852%
Addernobbuf	22	18	3023.44	2534.53	3.0374010	2.6895765	4.205879%	0.184541%	19.180283%	12.165052%
Adderlatch	15	10.5	1392.45	1240.76	1.6087695	1.4393733	5.411569%	11.906848%	15.722563%	0.138722%
Dpreg	18	9	2702.08	1986.83	0.4466443	0.3470790	27.934293%	78.347630%	5.922910%	83.172354%
StaticAdder	44.5	47	7284.10	7878.58	0.1602026	0.1666973	1.546801%	0.570367%	9.815186%	3.437452%
PlaClockout	10	10	830.81	825.17	0.1877630	0.1923957	8.925773%	33.124918%	9.540050%	31.472398%
PlaIn	6	7.5	668.90	694.00	0.4573744	0.4233144	1.979536%	17.204086%	5.797664%	8.480812%
PlaNorOut	5.5	6.5	795.33	810.75	0.4224475	0.3803335	10.491445%	18.845811%	12.626467%	6.995661%
ScanUnClkIn	14.5	13	1820.31	1699.00	0.1486535	0.1426869	9.390279%	3.376178%	2.120091%	7.238522%

Table 5: Summary of performance results of the overall system on 20 test circuits.

Conta. Av. Err(%)	Area Av. Err(%)	Asp. Ratio Av. Err.(%)
3.686737%	14.459908%	17.844106%