BEST: Behavioral Area-Delay Estimator

Kayhan Kucukcakar and Alice C. Parker

CENG Technical Report 93-24

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213)740-4476

May 1993

BEST: Behavioral Area-Delay ESTimator

Kayhan Küçükçakar
Motorola Inc.
2100 E. Elliot Road
Tempe, AZ 85284

Alice C. Parker
Department of Electrical Engineering – Systems
University of Southern California
Los Angeles, CA 90089-2562

October 20, 1992

This work was supported in part by the Defense Advanced Research Projects Agency and monitored by the Federal Bureau of Investigation under contract No. JFBI90092. The views and conclusions considered in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Defense Advanced Research Projects Agency or the U.S. Government.

BEST: Behavioral Area-Delay ESTimator

Abstract

This paper presents a comprehensive, probabilistic, efficient, and integrated area-delay estimation approach which can be used to support system-level design and high-level synthesis activities and describes results obtained using the tool. BEST (Behavioral Area-Delay ESTimator) is the first estimator prototype, to our knowledge, to address the high-level estimation problem in its totality. BEST starts with a behavioral description of a design, a library of RTL (Register-Transfer Level) components, a tentative data path clock cycle time and design constraints, and estimates the area-delay characteristics of designs which can be generated within the framework of a synthesis system. The estimation model used in BEST includes functional, register, multiplexing, control and wiring area and delay estimations. By quickly searching the design space, BEST is intended to guide the designer during system-level synthesis, behavioral partitioning, and high-level synthesis and to reduce the number of iterations in the design cycle. It provides a mechanism for the evaluation of tentative design decisions (e.g. high-level transformations, control/data path tradeoffs and module selection), and the evaluation of already synthesized designs. BEST is currently being used by a system-level partitioning tool in the ADAM Synthesis System.

1 Introduction

Even though automated synthesis can speed up the design process by orders of magnitude, the growing complexity of designs together with the multitude of tradeoffs possible during synthesis make it impossible to completely search the design space, even with synthesis tools which span behavior to layout. In order to improve the quality of generated designs, the designer has to iterate several times by modifying the functional specification (high-level transformations), design constraints and design decisions. Fast estimation tools which estimate the impact of tentative design decisions while taking into account physical design effects can be extremely beneficial in reducing the iteration time.

The organization of the paper is as follows. Section 2 summarizes some related work. Section 3 introduces BEST and discusses features and limitations of the tool. Section 4 gives the experimental results for validation of the estimation techniques and concluding remarks are given in Section 5.

2 Related Work

A few high-level synthesis systems use estimations. There are generally some assumptions (implicit or explicit) in the techniques used on what constitutes a first order effect on the design characteristics. Some synthesis systems use estimation techniques for wire delays and/or scheduling difficulties (data dependency constraints) to complement their decision making process. The complexity of these techniques can be as high as the complexity of synthesis algorithms.

To date there have been many estimation techniques developed with different goals. Layout area estimation for RTL designs [Zim88, KR91, WCG91] has been studied comprehensively. Hagerman's work to estimate the functional unit allocation from behavioral specification is another example [Hag91]. The HYPER system uses several estimation techniques to bound the min-max search involved in their synthesis process [RCHP91].

BEST inherits many existing estimation techniques previously developed at USC (see Section 3 for references).

3 The estimator

BEST is a comprehensive and integrated area-delay estimation tool to support high-level synthesis. The estimation model used in BEST includes functional, register, multiplexing, control, and wiring area and delay estimation. The estimation mechanism, in general terms, mimics an overall synthesis process (currently tuned for the ADAM Synthesis System [JKMP89]) without making detailed decisions. BEST follows the same set of synthesis tasks and ordering of these tasks, as in the ADAM Synthesis System: design style selection (pipelined or non-pipelined), module selection (library configuration selection), scheduling, operator allocation, register allocation, interconnect allocation, control generation, and layout generation. Although the general flow of the tasks in BEST is taken from the ADAM System, the estimations are more general and not designed for use exclusively by ADAM.

The input for BEST consists of the tentative clock cycle time, a directed acyclic dataflow graph, an operator library (preferably with more than one operator style to implement every operation type) and constraints for total area, initiation interval and total circuit delay. The estimation results are in the form of a set of area-delay pairs for non-pipelined designs and area-delay-initiation interval triples for pipelined designs. Each pair or triple forms a point (estimated design) in the estimated design space.

BEST generates estimated designs for all possible meaningful implementations of the design. It simply enumerates all possible library configurations (each library configuration contains one operator per operation type). For each library configuration, pipelined and non-pipelined estimations are generated for each possible value of initiation interval and number of stages (in terms of clock cycles). BEST can be instructed to produce only non-pipelined data path estimations. The selection of the best feasible estimated design (satisfying design constraints) from the pool of estimated designs also solves the problem of design style selection and module selection.

The overall behavioral synthesis process (down to the layout) is modelled as a design decision tree (Figure 1), the leaves of the decision tree being the potential designs which can be generated. Each estimation task contributes some estimated characteristics of each potential design to the data pool and uses the original input data as well as estimation results prior to its execution.

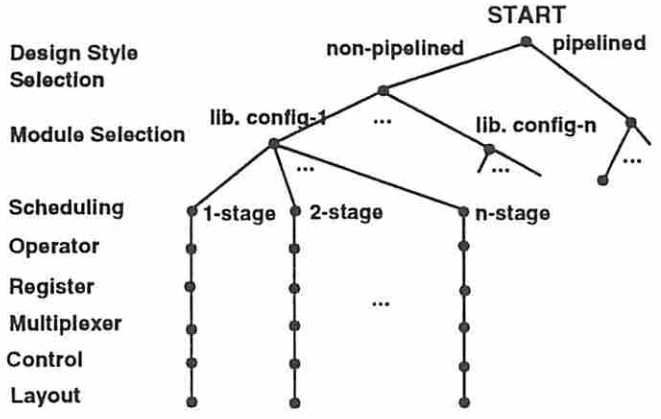

Figure 1: The current estimation generation in BEST

BEST is also capable of deleting estimated designs which are inferior or cannot meet specified designer requirements. In this way, estimated designs which would eventually be too large or too slow can be eliminated automatically. The tool optionally handles multi-cycle operations. The enumeration phase for generating possible library configurations automatically screens out selection of library operators which would be too slow for the given clock cycle.

Many estimation techniques developed assume that there exists a partial design at the point where the estimation technique is used and that detailed characteristics are available to the tool. For example, the RTL area and the equivalent number of 2-point nets are inputs to PLEST [KP89], while other methods [Zim88, KR91] assume that the complete RTL design exists. When such techniques are used in an estimation tool which spans from behavior

to layout, many additional entities characterizing the design must be estimated (some are trivial) in order to integrate the estimation techniques. BEST works with 22 such entities, 9 of which are estimated with techniques inherited as is, with modification, improvement, or simplification from previous USC research. These entities include the following: maximum fine grain parallelism [PP88] across conditional operations (e.g., maximum 8 out of 12 conditional operations can execute in parallel), critical path delay (calculation with an approximate model – no hardware chaining), serial-parallel tradeoff range (min and max) consisting of the range for the number of time steps (non-pipelined design style) and the range for the performance (pipelined design style), number of time steps as a function of the performance (pipelined design style), number of operators of each type [Jai89], dataflow graph width, number of registers as a function of parallelism [Mli91], register delays, average operator chaining as a function of parallelism, complexity of operator multiplexing as a function of parallelism, complexity of register multiplexing as a function of parallelism, total multiplexer area, multiplexer delays introduced into the clock cycle, abstract controller parameters (product terms, inputs, and outputs) [Mli91], PLA area for the controller [Mli91] (imported as is), PLA delay for the controller [Gup90] (imported as is), wiring complexity (number of nets and fanout), average wire length [KP89] (imported as is), effect of wire delays into the clock cycle as a function of individual wire delays [Gup90] (imported as is) and number of wires in sequence within time step boundaries, and finally routing area [KP89].

3.1 Integration of estimation techniques

Using estimations in behavioral synthesis requires special care because of the discrete nature of the problem being solved. In addition, there are many types of estimation tools, each dealing with a different aspect of a design. The fact that it is not always possible to have estimations of the same nature complicates the problem even further. Some estimation results are in the form of lower-bound values while others might be expected values or upper-bound values. There is usually more data available than just a single expected value or a bound and it is desirable to make use of as much data as possible. Most estimation techniques incorporated into BEST originally produced results in the form of lower bounds, expected values or upper bounds.

BEST is a fast, practical tool which combines former theoretical and heuristic estimation research done at USC along with new additions, using a unified representation. Statistical properties (average and variance) for each design characteristic are derived from estimations of the lower bound, upper bound, and most likely values of the design characteristic and the overall characteristics are approximated by a normal distribution.

The PERT modeling technique [LK66] has been chosen for the prototype software to approximate estimated design characteristics. In the PERT model every estimated point consists of three values (for each design characteristic): a lower bound, a most likely value and an upper bound. The average and the variance are approximated from these three values using the PERT technique, and standard statistical methods [Rob85] can then be applied.

For example, the PERT modeling technique is used for the area characteristic. *area.lb*, *area.ml*, *area.ub* are the lower-bound, most likely and upper-bound values of the characteristic, respectively. Based on these three values, *area.avg* and *area.var*, the average value for area and variance of area in the PERT model are approximated as follows:

$$area.avg = \frac{area.lb + 4 \times area.ml + area.ub}{6}$$

$$area.var = \frac{(area.ub - area.lb)^2}{36}$$

Average total area is computed by summing individual area averages. Total area variance is computed by summing individual area variances.

After the total of the estimation results for a particular design characteristic has been computed, infeasible estimated designs are eliminated as follows. The PERT model assumes that aggregate characteristics can be approximated with normal distributions. The accuracy of these approximations depends on the data pool size and how well the PERT model fits the application. The probability of satisfying the hard area constraint A_{max} for the estimated aggregate area A is computed by Equation 1 [Rob85] (also see Figure 2).

$$Pr[A \leq A_{max}] = \int_{-\infty}^{\frac{A_{max} - A.avg}{\sqrt{A.var}}} \frac{e^{\frac{-x^2}{2}}}{\sqrt{2\pi}} dx \tag{1}$$

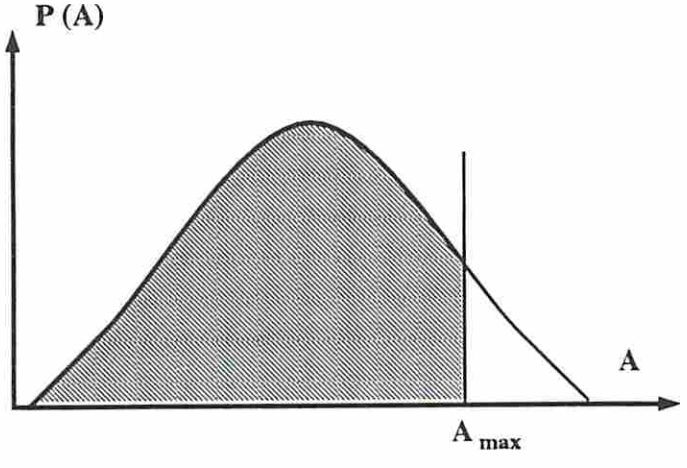

Figure 2: The probability of feasibility as the area under a probability density curve

Any estimated design whose probability of satisfying a design constraint is lower than the user-specified minimum probability of feasibility for the same design constraint is eliminated. These user-specified probability figures indicate not only the severity of the constraint but also the confidence of the designer in the estimation tool. If a constraint (e.g the total circuit delay) is not very critical, then it is better to include some of designs which are estimated to violate the given constraint, but are still close to the acceptable limits.

Another problem is that the error in estimators may not be consistent. Therefore, it is important not to eliminate estimated designs which violate some constraints but still stay close to the acceptable limits. The user-specified probability figure for each constraint is used as a measure of how strictly that constraint should be enforced. The right hand

side of Equation 1 is readily available in a tabular form which makes it suitable for fast execution [Rob85].

Using the PERT modeling technique to address uncertainties in estimation results is generally not sufficient. Tight bounds should be used in PERT. If very loose bounds are used, the average values calculated by PERT tend to shift away from the most likely value, resulting in inaccurate estimations.

The current integration method in BEST allows easy integration of new techniques to replace existing techniques in the tool or to estimate other design characteristics which are not currently estimated, e.g. register file area/delay, power consumption, and custom block layout area (instead of or in addition to standard-cell layout area) estimations.

3.2 Estimation Techniques

BEST enumerates all possible[1] cases of design style selection and module selection for constrained designs (branches in Figure 1) within its framework and proceeds with estimation generation across all branches of the decision tree.

Scheduling is a design phase which explores serial-parallel tradeoffs and decides on the parallelism (specific assignments of operations to time-steps are not important for estimations). During this phase BEST estimates the range (min and max) for serial-parallel tradeoffs. This tradeoff is controlled by the number of time steps (for the non-pipelined style) and the initiation interval (for the pipelined style) in our framework. BEST enumerates all cases between these estimated bounds to explore the design space and generates estimations across all branches. The front-to-back delays are also estimated for the pipelined design style.

Operator allocation estimations are based on Jain's work [Jai89]. Given the design style, module library, and the parallelism (from the scheduling phase) lower-bound operator allocation is estimated. We extended Jain's techniques to include multi-cycle operations and to estimate most likely and high values for operator allocation based on operator utilization [KP90a].

The estimation of register area is performed as generally proposed by Mlinar [Mli91]. Mlinar first finds the exact width of the data flow graph but we currently use a simple heuristic instead. This is followed by the estimation of the number of registers as a function of the parallelism, the graph width and other graph characteristics.

The effects of multiplexing is handled by first estimating the complexity of operator and register multiplexing as a function of allocation. This is followed by the estimation of multiplexer area and delays assuming that 2-to-1 multiplexers are used in the final implementation. The total multiplexer area estimation is also based on the empirical data.

The control estimator assumes a PLA implementation. First abstract PLA characteristics (the number of PLA inputs and outputs, and the number of product terms) are estimated in a similar way to that proposed by Mlinar [Mli91]. Then Mlinar's [Mli91] and Gupta's [Gup90] techniques are used to estimate the area and the delay of the PLA implementation, respectively.

Wiring estimations are currently based on the standard-cell layout methodology. The

[1] See Section 3.5 for limitations of the assumed module selection method.

wire area estimation is taken from Kurdahi [KP89]. A series of experiments showed that it was possible to interpolate the wiring area using previous results generated by Kurdahi's technique. Since run-time was very critical for us, we adopted a table look-up technique based on previous results. Wire delays are estimated as a function of the multiplexer complexities for operators and registers, the average fanout, average operator chaining, and average wire length.

3.3 Design Space Exploration

A major difficulty in exploring the design space (by estimations or by synthesis) is the fact that the set of all potential designs is quite large and not all potential designs are interesting. Therefore, it is essential to use a mechanism which does not miss good designs during design space exploration but does not report too many inferior potential designs, either. In short, we would like to cover all potential designs without searching the complete design space. BEST performs controlled enumeration at the scheduling phase to explore serial-parallel tradeoffs. This controlled enumeration is performed between estimated bounds which are tuned to be loose enough that the important parts of the design space are covered (no good designs are missed), but not too loose that the number of potential designs explored explodes. As a result, some uninteresting potential designs are explored and some of these uninteresting designs are eliminated later by feasibility analysis and by the concept of inferiority[2].

3.4 Additional Features of BEST

BEST can perform a feasibility analysis to find if potential implementations satisfy the global design constraints on the area, the performance and the front-to-back delay. In addition, BEST can be asked to eliminate inferior designs automatically. Using these two features it is very easy to converge on the best possible implementation which meets the design constraints.

Guidance on how to synthesize a design is also provided whenever BEST reports a design. This is accomplished by keeping the complete trace of the estimations and decisions on the design decision tree shown in Figure 1 except on pruned branches. Therefore, it is quite easy to drive the synthesis tools to reach the designs estimated by BEST. However, designs can be produced differently than BEST suggests and in some cases these may be better.

3.5 Limitations of BEST

The current model used in BEST assumes that only one type of operator will be used to implement only one type of operation. Since we are trying to address the constrained synthesis problem, non-deterministic delays are not supported. Loops with known or unknown iteration counts can be handled in a bottom up fashion during synthesis as well as estimation. Estimation techniques in our current pipelining model do not handle designs with internal feedback loops.

[2]The current elimination criterion is based on absolute global characteristics. Instead a more elaborate weighted cost criterion can be used.

Like most high-level systems, our estimation tool currently does not take into account the effects of hardware chaining on the clock cycle time, where the total delay through a sequence of chained operations can be less than the sum of the individual operation delays. Hardware chaining becomes more important for highly parallel non-pipelined implementations in which operation chaining is extensively used. In order to model hardware chaining a more accurate delay model (describing pin-to-pin delay characteristics of RTL modules) is planned to be used by both the estimation and the synthesis tools.

Scheduling infeasibilities are not modelled completely in BEST. However, we feel that our estimations are fairly accurate over a wide range of designs (except highly regular designs) based on the data path tradeoff study we have done in the past [KP90b]. We assume that a synthesis system with a global view of the design process is used (non exists yet, but the use of human interaction is equally effective). We would like to improve our model to include more elaborate consideration of scheduling infeasibilities without increasing the run-time complexity of the estimator. A more detailed discussion of this can be found in [Küç91].

4 Experimental Results

Two groups of validation efforts will be reported here. The first experiment was performed when technology-dependent estimation techniques (PLA, wire delays, routing area, etc.) were tuned for a 3-micron CMOS technology.

The second experiment was performed after the technology of the estimation tool was upgraded to 1.2 micron CMOS. This technology upgrade involved modifying feature-size-related parameters to 1.2 micron technology and modifying Rent's exponent to reflect the layout tool's style as close as possible. The layouts for synthesized designs were produced by the Seattle Silicon (now Cascade) Chipcrafter System.

In both set of experiments, the ADAM Synthesis System was used to produce the RTL designs. Some schedules were modified manually to obtain better optimized results than we would get from our synthesis tools, but not all synthesized designs were screened for better optimization opportunities. For experiments with the 3-micron technology, PLAs were generated using ESPRESSO and MKPLA, and for the 1.2 micron technology the Seattle Silicon Chipcrafter tools were used. In both set of experiments, we have not tried to generate a layout/design corresponding to all estimated designs from BEST due to the amount of work needed.

4.1 Pre-layout Comparisons - Experiment 1

Since we didn't have the capability to generate complete layouts using the 3-micron CMOS technology, we will only report the comparisons of the functional (RTL + PLA) area and delay characteristics for the first experiment. Since estimating pre-layout characteristics of potential designs is also quite important, comparisons of pre-layout results for the fifth order Elliptic Wave Filter from [KWK85] and the FIR Filter from [PP88] will be reported. These examples have 34 and 23 operations consisting additions and multiplications, respectively.

The library used in this set of experiments was similar to the library used in [JKMP89] and partial contents are given in Table 1. The pre-layout (RTL + PLA) comparisons are

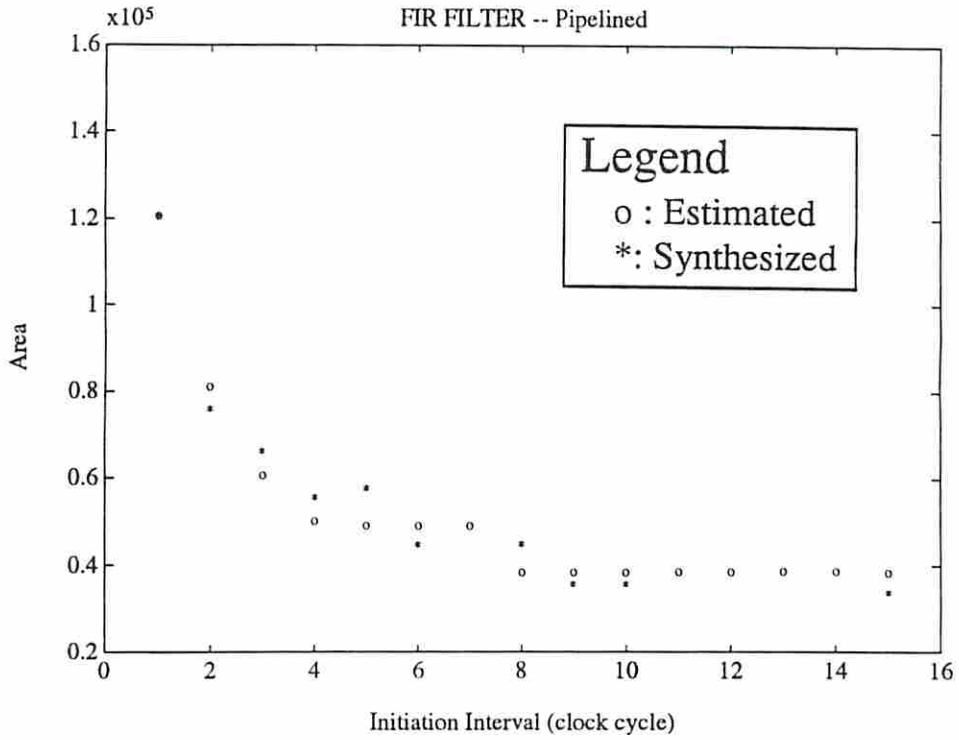

Figure 3: RTL + PLA area for the pipelined FIR Filter (3-micron)

shown in Figures 3 through 6 for pipelined and non-pipelined implementations of FIR and Elliptic Wave Filters. The Elliptic Wave Filter has an outer feedback loop, but we ignored the feedback loop for pipelined designs in order to present results for this well-known dataflow graph structure.

Module Name	Bit Width	Area (mil^2)	Delay (ns)
Adder	16	1200	151
Multiplier	16	9800	2950
Register	1	31	5
2:1 Multiplexer	1	18	4

Table 1: The 3-micron library used in the experiments

4.2 Layout Comparisons - Experiment 2

It is important to note that there are differences between the design style assumed by BEST to perform estimations and the design style of the actual layouts. Layout estimations in BEST assume a standard-cell implementation in contrast to the custom block implementation used by the Chipcrafter System[3]. BEST assumes that only 2-to-1 multiplexers are used while larger size multiplexers (3-to-1, 4-to-1, 10-to-1, etc.) which were better optimized for

[3]Chipcrafter was the only tool available to us to produce layouts.

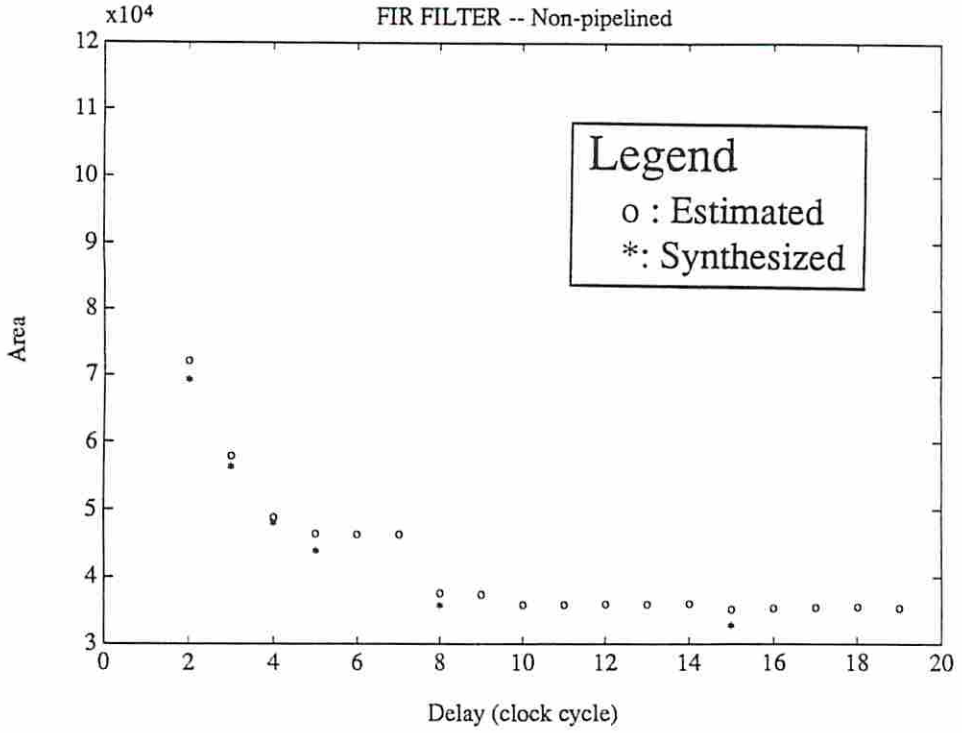

Figure 4: RTL + PLA area for the non-pipelined FIR Filter (3-micron)

area and speed were used in the layouts. These differences make us believe that good layouts from the Chipcrafter System would generally be smaller and faster than the estimated counterparts.

In this experiment, only the AR Filter from [JKMP89] was used. The AR Filter has 28 operations consisting of multiplications and additions. The layouts were created by pre-buffering signals for fanout. This reduced the clock cycle times by 25-50+ percent over non-buffered layouts. The bottleneck was found to be control signals with high fan-outs. The library used in the experiment is shown in Table 2. Comparisons of estimated characteristics and layout characteristics are shown in Figures 7 through 10. The timing information reported for layouts in Figures 9 and 10 was obtained from the timing analysis utility of the Chipcrafter System.

Figure 9 compares the estimated clock cycle time from BEST and the Chipcrafter System. BEST's estimated clock cycle times were higher than the clock cycle times obtained from the Chipcrafter System. This was expected due to limitations of the current delay model used by BEST (see hardware chaining in Section 3.5. For 1 and 4 time-step designs, BEST's clock cycle time estimates were unacceptably high[4]. The effect of hardware chaining becomes less for more serial designs as it can also be seen from the figure. Although the basic know-how exists to model hardware chaining in our estimation techniques, it was left out of the prototype to simplify its implementation.

In Figure 10, estimated designs which do not have any corresponding implementations are not shown in order to simplify the plot. It is easy to see that the A-T (Area-Time) curve

[4]Manual experiments with a simple hardware chaining model showed acceptable results.

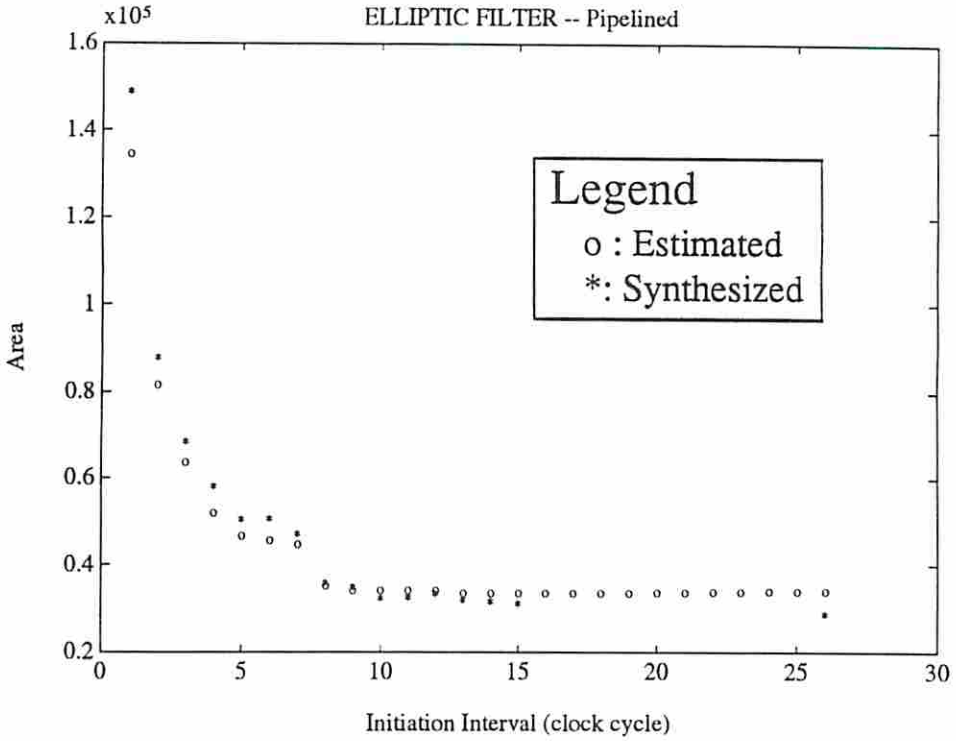

Figure 5: RTL + PLA area for the pipelined Elliptic Filter (3-micron)

of layouts in Figure 10 roughly tracks the A-T curve of the estimated designs. The shift in the layout A-T curve towards the origin was expected due to reasons described at the beginning of this section.

In this experiment two layouts for pipelined design style were also created. The layout area(clock cycle time) differences between the synthesized and estimated pipelined designs were 20%(8%) and 13%(4%) respectively.

Module Name	Bit Width	Area (mil^2)	Delay (ns)
Adder	16	126.0	35.0
Multiplier	16	3718.0	54.5
Register	1	9.3	3.0
2:1 Multiplexer	1	3.0	2.6

Table 2: The 1.2-micron library used in the experiments

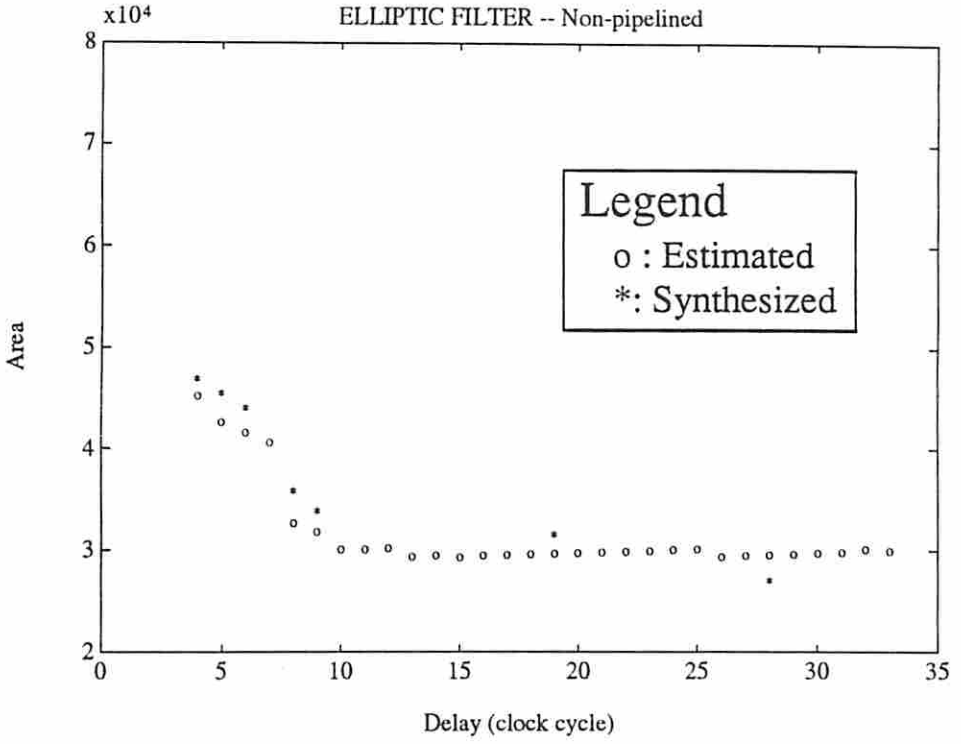

Figure 6: RTL + PLA area for the non-pipelined Elliptic Filter (3-micron)

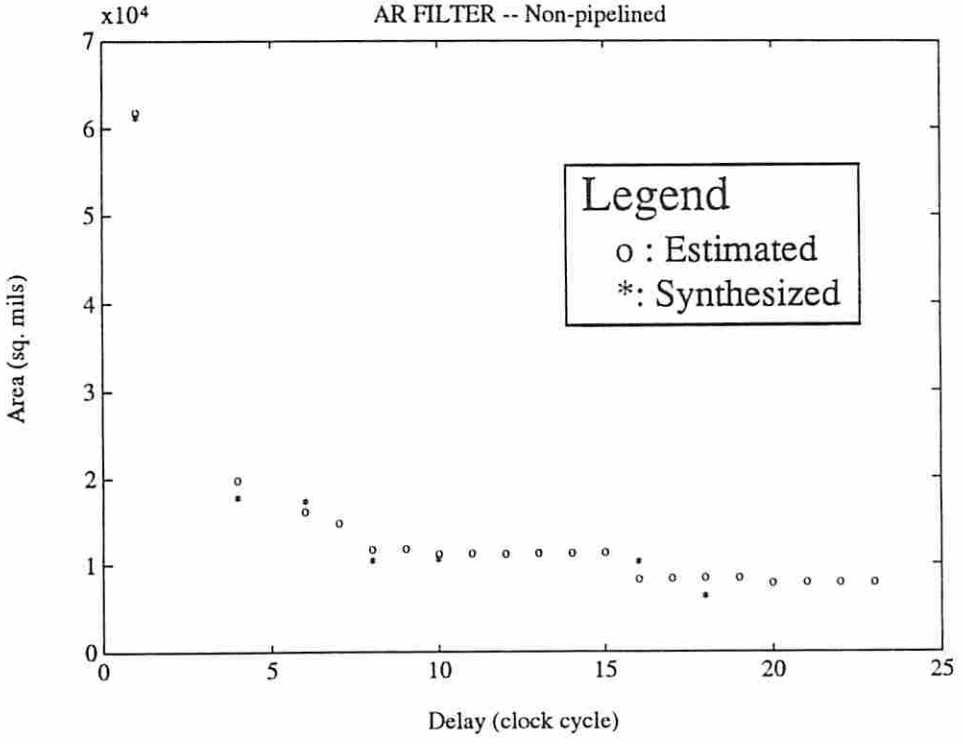

Figure 7: RTL area for the non-pipelined AR Filter (1.2-micron)

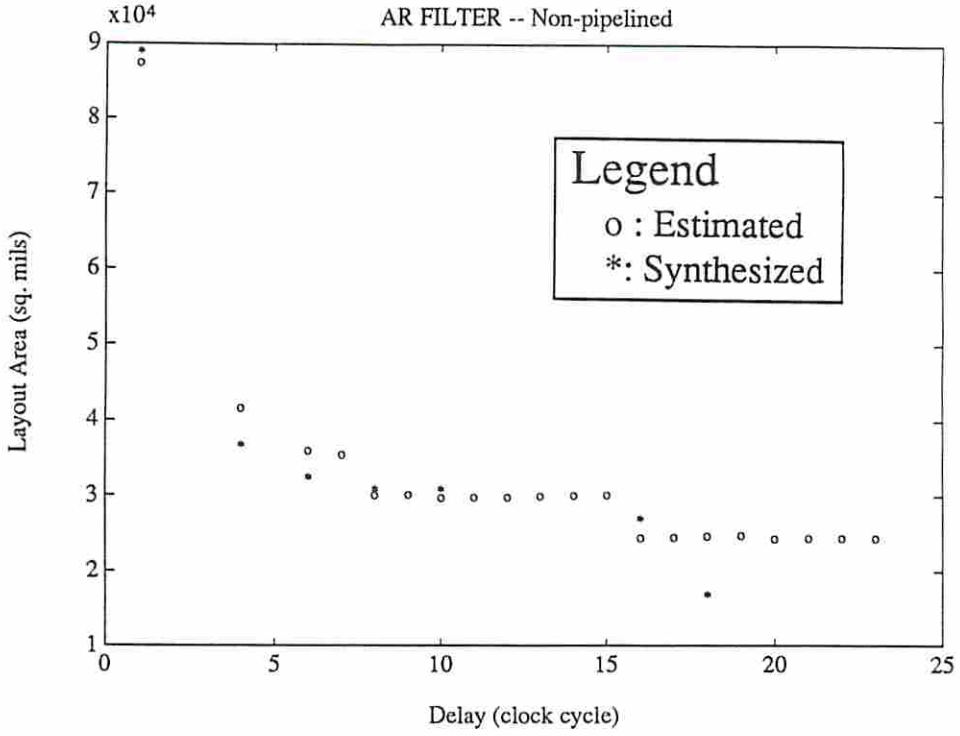

Figure 8: Layout area for the non-pipelined AR Filter (1.2-micron)

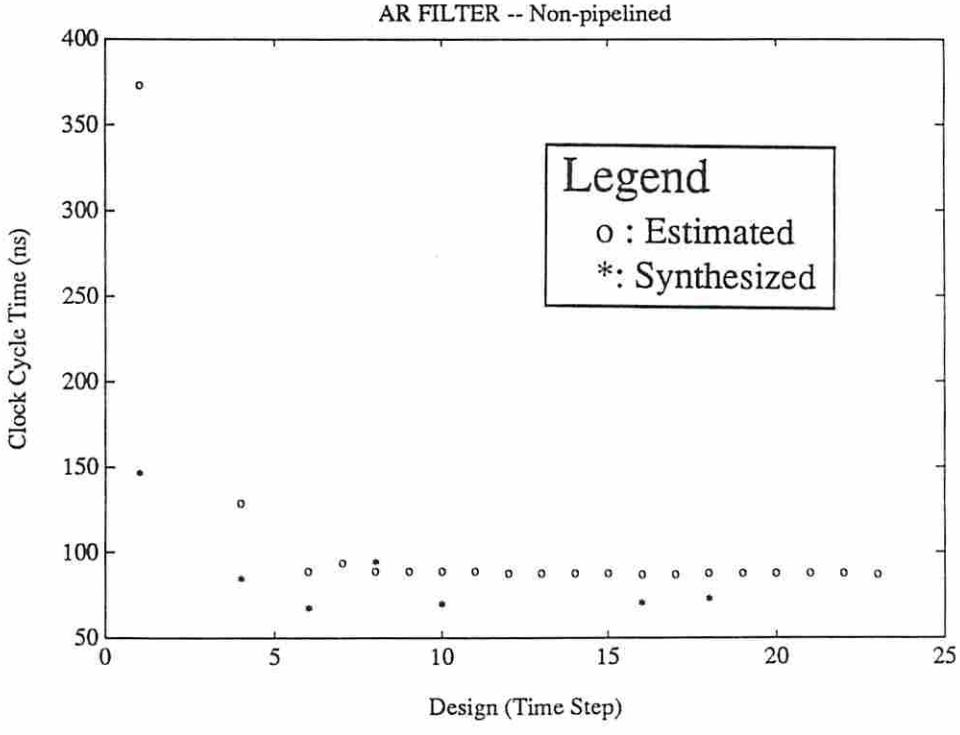

Figure 9: Clock cycle time for the non-pipelined AR Filter (1.2-micron)

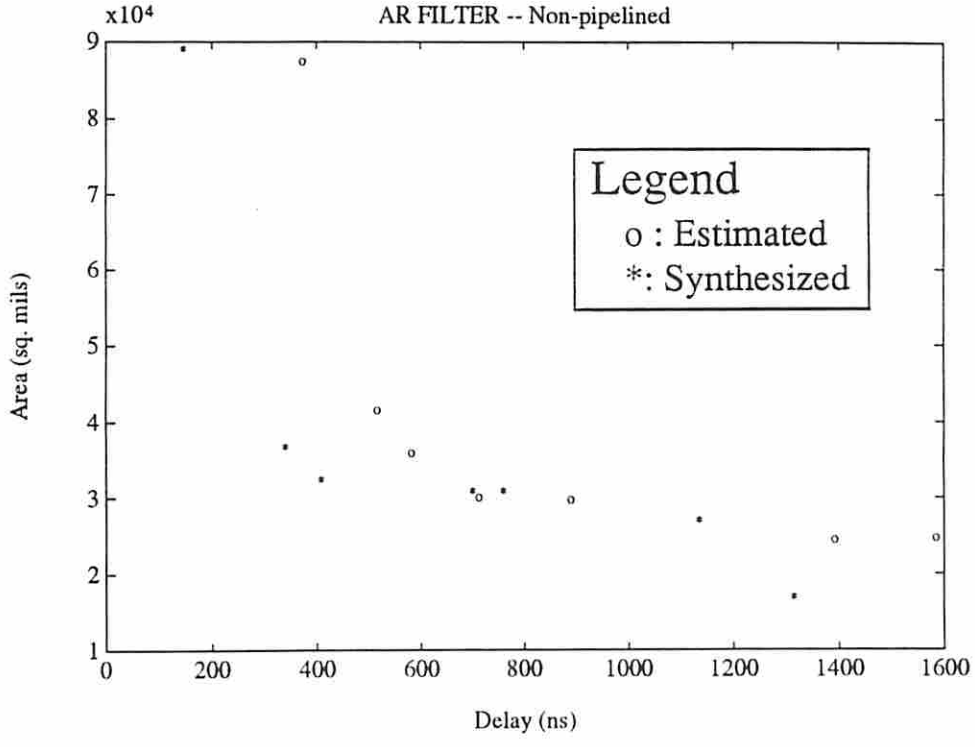

Figure 10: Area-Time plot for the non-pipelined AR Filter (1.2-micron)

5 Conclusions

In spite of the differences of the layout styles of BEST and the layout tool used, the experimental results show behavioral estimations of final design (layout) characteristics without performing any synthesis is possible. BEST is the first estimator to address the high-level estimation problem comprehensively. Experiments strengthen our belief that estimations used in any design environment have to follow the specific synthesis/design methodology used in the same design environment.

The run-time complexity of BEST is $O(n^2m)$ where n is the number of operations in the behavioral specification and m is number of possible library configurations in our restricted model. Currently, BEST tries $m = \Pi_i m_i$ library configurations where i varies over the operation types and m_i is the number of operators of type i. This run-time complexity is for the generation of all possible estimation points which is $O(nm)$. It is always possible to generate an almost infinite number of estimation points by choosing the clock cycle time arbitrarily small compared to operator delays since BEST supports multi-cycle operations. In the complexity calculations it was assumed that the ratio of the clock cycle time and operator delays are bounded by a constant. In order to reduce the run-time, BEST prunes any branch shown in Figure 1 as soon as the branch is detected to lead to a infeasible design. The run-time of BEST averages about 0.5 msec of CPU time per estimated design on a Sun Sparc 4/460 (180 msecs for 346 estimated designs).

BEST uniquely provides the means for module selection by enumerating possible library configurations, the means for evaluating control/data path and serial/parallel tradeoffs, and the means for performing design space search by exploring alternative paths before synthesis. An important capability BEST provides, which was not intended during the development phase, is to be able to evaluate the quality of already synthesized designs. During the validation of estimation techniques, we discovered that performance of some of the synthesized designs severely degraded due to fanout delays and this fact was not found until we tried to compare synthesis results to estimation results.

The estimation techniques described in this paper is currently being used to guide the search mechanism in a behavioral partitioning package [KP91].

6 Acknowledgements

The authors would like to thank Pravil Gupta and Atul Ahuja for their help in synthesizing some of the designs and for producing the layouts.

References

[Gup90] P. Gupta. PLA and Wire Delay Analysis. Department of Electrical Engineering, University of Southern California, 1990. Internal Report.

[Hag91] J. W. Hagerman. A Fast and Accurate Technique for Function Unit Allocation Estimation. Technical Report CMUCAD-91-28, Carnegie Mellon University, April 1991.

[Jai89] R. Jain. *High-Level Area-Delay Prediction with Application to Behavioral Synthesis.* PhD thesis, Department of Electrical Engineering, University of Southern California, 1989.

[JKMP89] R. Jain, K. Küçükçakar, M. J. Mlinar, and A. C. Parker. Experience with the ADAM Synthesis System. In *Proc. 26th Design Automation Conf.*, pages 55–61. ACM/IEEE, June 1989.

[KP89] F. J. Kurdahi and A. C. Parker. Techniques for Area Estimation of VLSI Layouts. *IEEE Trans. on Computer-Aided Design*, 8(1):81–92, January 1989.

[KP90a] K. Küçükçakar and A. C. Parker. BAD: Behavioral Area-Delay Predictor. Technical Report CENG-90-31, Department of Electrical Engineering, University of Southern California, November 1990.

[KP90b] K. Küçükçakar and A. C. Parker. Data Path Tradeoffs using MABAL. In *Proc. 27th Design Automation Conf.* ACM/IEEE, June 1990.

[KP91] K. Küçükçakar and A. C. Parker. CHOP: A Constraint-Driven System-Level Partitioner. In *Proc. 28th Design Automation Conf.*, pages 514–519. IEEE/ACM, June 1991.

[KR91] F. J. Kurdahi and C. Ramachandran. LAST: Layout Area and Shape Function EsTimator. In *Proc. 1st European Design Automation Conf.*, pages 351–355. ACM/IEEE, February 1991.

[Küç91] K. Küçükçakar. *System-Level Synthesis Techniques With Emphasis On Partitioning And Design Planning.* PhD thesis, Department of Electrical Engineering, University of Southern California, October 1991.

[KWK85] S. Y. Kung, H. J. Whitehouse, and T. Kaliath. *VLSI and Modern Signal Processing.* Prentice-Hall, 1985.

[LK66] R. I. Levin and C. A. Kirkpatrick. *Planning and Control with PERT/CPM.* McGraw Hill, 1966.

[Mli91] M. J. Mlinar. *System Level Tradeoffs in VLSI Design.* PhD thesis, Department of Electrical Engineering, University of Southern California, May 1991.

[PP88] N. Park and A. C. Parker. Sehwa: A Software Package for Synthesis of Pipelines from Behavioral Specifications. *IEEE Trans. on Computer-Aided Design*, 7(3):356–370, March 1988.

[RCHP91] J. Rabaey, C. Chu, P. Hoang, and M. Potkonjak. Fast Prototyping of Datapath-Intensive Architectures. *IEEE Design and Test of Computers*, pages 40–51, June 1991.

[Rob85] E. A. Robinson. *Probability Theory and Applications.* Int'l Human Resources Development Corporation, 1985.

[WCG91] A. C-H Wu, V. Chaiyakul, and D. D. Gajksi. Layout-Area Models for High-Level Synthesis. In *Proc. Int'l Conf. on Computer-Aided Design.* IEEE Computer Society, November 1991.

[Zim88] G. Zimmerman. A New Area and Shape Function Estimation Technique for VLSI Layouts. In *Proc. 25th Design Automation Conf.*, pages 60–65. ACM/IEEE, June 1988.