An optimal algorithm for sizing sequential circuits for industrial library based designs

Sanghamitra Roy, Yu Hen Hu
ECE Department
University of Wisconsin-Madison
{roy1,yhhu}@wisc.edu

Charlie Chung-Ping Chen, Shih-Pin Hung, Tse-Yu Chiang, Jiuan-Guei Tseng
National Taiwan University
{cchen@cc.ee.,r96943078@,r95943082@,d88921003@}ntu.edu.tw

***Abstract*— In this paper, we propose an optimal gate sizing and clock skew optimization algorithm for globally sizing synchronous sequential circuits. The number of constraints and variables in our formulation is linear with respect to the number of circuit components and hence our algorithm can efficiently find the optimal solution for industrial scale designs. To the best of our knowledge our method is the first exact gate sizing algorithm that can handle cyclic sequential circuits. Experimental results on industrial cell libraries demonstrate that our algorithm can yield an average of 12.6% improvement in the optimal clock period by combining clock skew optimization with gate sizing. For identical clock period, our algorithm can achieve an average of 11.3% area savings over a popular commercial synthesis tool.**

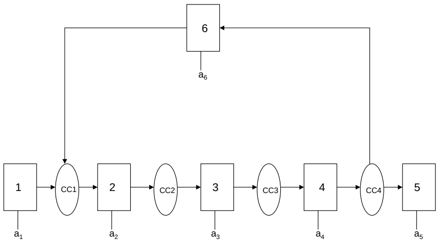

Fig. 1. Feedback loop in datapath

I. Introduction

Synchronous sequential circuits form the core of modern IC designs. A typical integrated circuit consists of several combinational logic blocks lying between flip-flops and often contains feedback paths or cycles. LFSR (linear feedback shift registers), modulo-N counters and data forwarding logic in microprocessor cores are some examples of feedback loops in sequential circuits. Figure 1 shows a typical sequential circuit with a feedback path.

Given a synchronous sequential circuit with a fixed topology, we want to optimize the circuit for delay, power and/or area by varying the sizes of the gates in the circuit. The goal is to find a fast exact optimization technique for globally sizing the whole circuit using accurate delay and power models.

The simplest way to optimize the delay of sequential circuits is to perform gate sizing of each combinational logic block separately and then arrive at a clock frequency for the entire circuit depending on the maximum delay of any combinational logic block. However, this method cannot incorporate clock skew optimization [1] and leads to suboptimal circuit performance. By adding an additional degree of freedom to our optimization using variable clock skews at the flip-flops, we can achieve better power/performance goals for the design [2].

While several algorithms were proposed for gate sizing of combinational circuits, some of which are based on convex optimization techniques [4], [5], [6], [10], only a handful of previous works target sequential circuits. Chuang et al. formulated the sizing problem of *acyclic* sequential circuits as a linear program [2]. However, the piecewise linear models they used have limited accuracy in modern designs since the gate delay is a nonlinear function of the gate size [10]. For larger circuits, this formulation had a prohibitively large number of constraints and they partitioned the circuit to solve the problem, while sacrificing the optimality of the solution. Sathyamurthy et al. formulated the sequential circuit sizing problem as a nonlinear convex program [3]. They used the Elmore delay model which is considered to be highly inaccurate for deep submicron VLSI designs [10]. Again, due to an extremely large number of tunable variables, they could not propose an exact method for solving the problem. None of these techniques can be applied to cyclic sequential circuits with feedback loops, as they rely on DAG (directed acyclic graph) representation of the circuit.

In this paper, we propose an optimal gate sizing and clock skew optimization algorithm for efficiently sizing industrial library based designs. We use nonlinear convex models for accurately representing the gate delays and use the Lagrangian Relaxation technique for solving the formulated optimization problem. We achieve faster convergence of the problem by appropriately choosing the constraints and variables (linear with respect to the circuit components), without sacrificing any accuracy in modeling. Experimental results on industrial cell libraries demonstrate that our algorithm can yield an average of 12.6% improvement in the optimal clock period by combining clock skew optimization with gate sizing. For identical clock period, our algorithm can achieve an average of 11.3% area savings over a popular commercial synthesis tool. To the best of our knowledge, our technique is the first exact gate sizing algorithm that can handle cyclic sequential circuits.

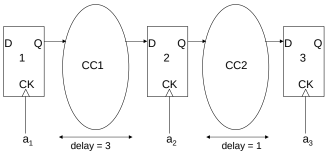

Fig. 2. Motivation: Clock skew optimization

II. Sizing and Clock skew optimization

A. Clock skew optimization

Clock skew optimization [1] is a technique where the various flip-flops in a circuit are intentionally given clock skews to improve the overall performance and reliability of the circuit. Figure 2 shows combinational circuits blocks $CC1$ and $CC2$, having delays of 3 units and 1 unit, respectively. a_1, a_2 and a_3 are the clock skews at flip-flops 1,2 and 3. The circuit can operate at a minimum clock period of 3 units without any skew. But if flip-flop 2 receives a skew of $+1$ unit $(a_1 = 0, a_2 = +1, a_3 = 0)$, then the circuit can operate at a clock period of 2 units. By combining gate sizing with clock skew optimization, we can achieve better power/performance goals in our design.

B. Problem statement

Given a circuit consisting of N gates mapped to a technology library, find the drive strengths of all the combinational gates in the circuit, and the clock skews of all the flip-flops in the circuit, that will minimize the following objective function:

$$\alpha_1 C + \alpha_2 Area + \alpha_3 P_{leakage} + \alpha_4 P_{dynamic}$$

consisting of a linear combination of the clock period and the power and/or area of the circuit, while satisfying all the timing constraints of a synchronous sequential circuit.

III. Illustration with an example

Figure 3 shows a cyclic sequential circuit consisting of combinational gates and flip-flops, with a feedback loop.

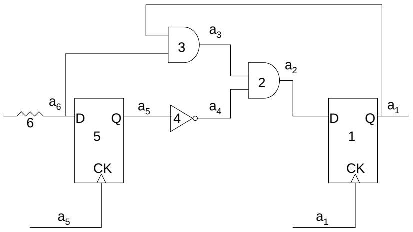

Fig. 3. A cyclic sequential circuit

TABLE I

Notation used in this work

N:	Set of all gates in the circuit, consisting of internal and output gates and excluding inputs
I:	Set of all inputs in the circuit
DFF:	Set of all flip-flops in the circuit
a_i:	Arrival time of the signal at the output of gate i $rise$ and $fall$ superscripts indicate signal rise or fall
C:	Clock period of the circuit
D_{ij}:	Gate delay from output of gate i to output of gate j
t_{setup}, t_{hold}:	Setup and Hold times of flip-flop
$in(j)$:	Set of all input pins of gate j
$(+)un$, $(-)un$:	$positive\_unate$ and $negative\_unate$

A. Notation

Table I gives a list of the notations used in this work. For the circuit in figure 3, $N = \{1, 2, 3, 4, 5\}$, $I = \{6\}$ and $DFF = \{1, 5\}$. The flip-flop clock-to-Q delay has been ignored in this formulation. For flip-flop i the arrival time at the CK pin is a_i which is the clock skew at that flip-flop. The arrival time at the output of flip-flop i is also equal to a_i. All arrival times and skews are relative to a zero skew clock.

B. Convexity of the delay function D_{ij}

Fishburn et al. made a seminal observation that the circuit delay under the Elmore model is a posynomial function of the transistor sizes [5]. Under this model, the gate delay could be represented as a convex function using an exponential transformation [4]. The Elmore model is currently considered inaccurate for modern designs, but many convex models with a high degree of accuracy have been proposed to model the gate delay [10], [11]. For library cells, the gate delay D_{ij} is a function of the gate input slew s_i, the gate load capacitance L_j and the gate input capacitance cap_j and can be modeled as a convex function using some of the existing techniques.

C. Delay constraints for combinational gates

Let us consider the combinational gate numbered 2. If we assume that input (pin) 3 of gate 2 is $positive\_unate$ and input (pin) 4 is $negative\_unate$, then we can write the following constraints.

$$a_3^{rise} + D_{32}^{rise}(s_3^{rise}, L_2, cap_2) \leq a_2^{rise}$$
$$a_4^{fall} + D_{42}^{rise}(s_4^{fall}, L_2, cap_2) \leq a_2^{rise}$$

The above constraints are convex as the D_{ij} functions are convex.

D. Setup time constraints

Let us consider the flip-flop numbered 1. To avoid the long path violations where the signal reaches the flip-flop too late, we need to set the setup constraints as illustrated in figure 4. Thus, we need to satisfy the constraint

$$Max(a_2^{rise} + t_{setup}^{rise}, a_2^{fall} + t_{setup}^{fall}) \leq a_1 + C$$

We can simplify the above constraint by splitting it into two constraints as follows:

$$a_2^{rise} + t_{setup}^{rise} \leq a_1^{rise} + C$$
$$a_2^{fall} + t_{setup}^{fall} \leq a_1^{fall} + C$$

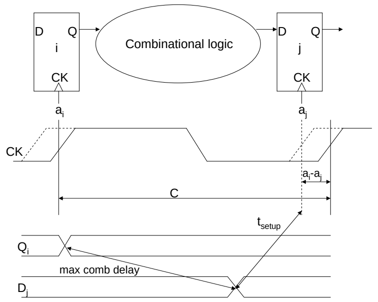

Fig. 4. Setup constraint and clock skew

The above constraints being linear, are convex constraints [7].

E. Hold time constraints

The hold constraints are necessary to avoid the short path violations leading to data racing. But these constraints are non-convex and hence not considered [3]. The constraints are resolved by adding buffers to increase the delay of short paths.

IV. Generalized problem formulation for industrial scale designs

After looking at the different types of constraints in our example, we now provide the generalized problem formulation $\mathcal{PP}$ for circuits synthesized using industrial cell libraries. This formulation is applicable for both cyclic and acyclic synchronous sequential circuits. The minimized objective function is a linear combination of the clock period C and the sum of the input capacitance of all gates, which is an approximation for the area of the circuit [5]. α_1 and α_2 are the corresponding non-negative weights. Note that the objective function being a sum of linear variables is convex. The clock skews at flip-flops as well as the input arrival times are bounded by the minimum and maximum skews $skew_{min}$ and $skew_{max}$, respectively. $\mathcal{PP}$ is a convex optimization problem as the objective function as well as the constraints are convex functions as discussed before. This problem can be solved optimally in polynomial time to give the optimal set of gate sizes and the optimal clock skews of the flip-flops by an algorithm we will discuss in section V.

$$\mathcal{PP}:$$

$$minimize \qquad \alpha_1 C + \alpha_2 \sum_{j \in N} cap_j$$

$$subject\ to$$

$$\begin{array}{l} \forall j{\in}N\wedge \\ j{\notin}DFF \\ \forall i{\in}in(j)\wedge \\ i{\in}(+)un(in(j)) \end{array}$$

$$a_i^{rise} + D_{ij}^{rise}(s_i^{rise}, L_j, cap_j) \leq a_j^{rise}$$

$$a_i^{fall} + D_{ij}^{fall}(s_i^{fall}, L_j, cap_j) \leq a_j^{fall}$$

$$\begin{array}{l} \forall j{\in}N\wedge \\ j{\notin}DFF \\ \forall i{\in}in(j)\wedge \\ i{\in}(-)un(in(j)) \end{array}$$

$$a_i^{fall} + D_{ij}^{rise}(s_i^{fall}, L_j, cap_j) \leq a_j^{rise}$$

$$a_i^{rise} + D_{ij}^{fall}(s_i^{rise}, L_j, cap_j) \leq a_j^{fall}$$

$$\begin{array}{ll} \begin{array}{c} \forall j{\in}N\wedge \\ j{\in}DFF \\ \forall i{\in}in(j) \\ i{\notin}CK \end{array} & a_i^{rise} + t_{setup}^{rise} \leq a_j^{rise} + C \\ & a_i^{fall} + t_{setup}^{fall} \leq a_j^{fall} + C \\ \begin{array}{c} \forall j{\in}N\wedge \\ j{\in}DFF \end{array} & skew_{min} \leq a_j^{rise}, a_j^{fall} \leq skew_{max} \\ \forall i{\in}I & skew_{min} \leq a_i^{rise}, a_i^{fall} \leq skew_{max} \\ \begin{array}{c} \forall j{\in}N\wedge \\ j{\notin}DFF \end{array} & bound_j^{lower} \leq cap_j \leq bound_j^{upper} \end{array}$$

V. Solving the nonlinear convex optimization problem by Lagrangian Relaxation

We optimally solve the sequential circuit sizing problem using the Lagrangian Relaxation technique [6]. We relax the troublesome constraints of $\mathcal{PP}$ by incorporating them into the objective function, using non-negative Lagrange multipliers to form the lagrangian function L_λ. The simple bound constraints are not relaxed. The exact formulation of L_λ has been omitted due to space limitation. The Lagrangian relaxation subproblem $\mathcal{LRS}/\lambda$ associated with a particular value of λ is:

$$\begin{array}{llr} \mathcal{LRS}/\lambda : & & \\ minimize & L_\lambda(\mathbf{a}, \mathbf{cap}, \mathbf{L}, \mathbf{s}, C, \lambda) & \\ subject\ to & skew_{min} \leq a_j^{rise}, a_j^{fall} \leq skew_{max}, & j \in DFF \\ & skew_{min} \leq a_i^{rise}, a_i^{fall} \leq skew_{max}, & i \in I \\ & bound_j^{lower} \leq cap_j \leq bound_j^{upper}, & \begin{array}{c} j{\in}N\wedge \\ j{\notin}DFF \end{array} \end{array}$$

where λ is the vector of all lagrange multipliers, and $\mathbf{a}$, $\mathbf{cap}$, $\mathbf{L}$ and $\mathbf{s}$ are the vectors of arrival time, input capacitance, load capacitance and input slew, respectively.

From the theory of Lagrangian [7], there exists a vector value λ for which the optimal solution of $\mathcal{LRS}/\lambda$ is equal to the optimal solution of the original problem $\mathcal{PP}$. If we can find this λ value, then we can get the optimal solution to the original problem. For a value of λ to be correct, the first order Karush-Kuhn-Tucker (KKT) conditions [7] must hold.

Table II summarizes SEQ_SIZE: our algorithm for gate sizing and clock skew optimization using the Lagrangian relaxation technique. In step 1, we initialize the lagrange multipliers and the tunable variables. We use a bound constrained optimization solver L-BFGS-B [8] to minimize L_λ in step 2. In step 3, we use a modified subgradient approach [9] to update the value of the lagrange multipliers at each iteration (equations not provided for space reasons). The projection of the multipliers to satisfy the KKT conditions (equations not provided) helps in pruning our solution, leading to a considerable runtime reduction. Finally, after getting the optimal gate sizes, we discretize our sizes to choose the nearest available drive strength for each gate from the library.

VI. Experimental results on industrial cell library

We use a $0.13\mu m$ family standard cell library containing 415 generic core cells and 53 I/O cells. Our sequential circuit

```
ALGORITHM SEQ_SIZE:
Output: optimal gate sizes and clock skews
1. $k := 1$
   $\lambda :=$ arbitrary initial vector of lagrange multipliers satisfying
   KKT conditions [7]
   Initialize all tunable variables
2. Solve $\mathcal{LRS}/\lambda$ by minimizing $L_\lambda(\mathbf{a}, \mathbf{cap}, \mathbf{L}, \mathbf{s}, C, \lambda)$ using L-BFGS-B
   Perform static timing analysis to calculate $a_i$s, $s_i$s and C
3. $\lambda_{new} :=$ multiplier after subgradient multiplier adjustment
   Project $\lambda_{new}$ to the nearest point satisfying KKT conditions
4. $k := k + 1$
5. If difference in cost functions of $\mathcal{PP}$ and $\mathcal{LRS}/\lambda$ is greater than
   stopping criteria, go to step 2.
6. Discretize gate sizes to available drive strengths
```

TABLE II

ALGORITHM SUMMARY FOR SEQ_SIZE

TABLE III

OPTIMIZED CLOCK PERIOD

Circuit	Optimized Clock period		Percent Improvement %
	Sizing + skew (ns)	Sizing only (ns)	
s27.v	0.341	0.426	19.95
s298.v	0.866	1.269	31.76
s382.v	0.960	1.219	21.25
s386.v	0.902	0.968	6.82
s444.v	1.070	1.192	10.23
s510.v	1.250	1.371	8.83
s820.v	1.693	1.818	6.88
s5378.v	1.990	2.129	6.53
s9234.v	2.558	2.569	0.43
s13207.v	2.718	3.129	13.14

TABLE IV

AREA OF OPTIMIZED NETLIST

Circuit	Commercial Synthesis tool $(\mu m)^2$	Sizing + skew $(\mu m)^2$	Sizing only $(\mu m)^2$	Percent Improvement %
s27.v	168	99	103	41.07
s298.v	694	653	650	5.91
s382.v	925	853	836	7.78
s386.v	554	499	499	9.93
s444.v	923	836	831	9.43
s510.v	925	825	816	10.81
s820.v	1183	1011	1010	14.54
s5378.v	7773	7688	7688	1.09
s9234.v	10148	9149	9149	9.84
s13207.v	20730	20283	20306	2.16

sizing tool is implemented in C++ using our Lagrangian relaxation based algorithm SEQ_SIZE. We use nonlinear convex posynomial models [10] for modeling the delay and slew look-up tables from the library. We conduct our experiments on ISCAS89 benchmark circuits. All experiments are performed on a PC with 1.40GHz Pentium IV Microprocessor, 1.00 GB RAM and 40 GB hard drive running Windows XP.

Table III shows the optimized clock period obtained after tuning the benchmark circuits. All timing measurements have been performed using our static timing analyzer in-built within our gate sizing tool. The second column shows the clock period obtained by gate sizing coupled with clock skew optimization, while the third column shows the clock period obtained using only gate sizing. We can get an average of 12.6% improvement in the optimal clock period by combining clock skew optimization with gate sizing. Table IV shows the area of the optimized netlists. The second column shows the area obtained by synthesizing the same designs using a popular commercial synthesis tool, by setting the target clock period for the corresponding circuit from the second column in table III. The third column gives the netlist area obtained by combined sizing and clock skew optimization, whereas the fourth column gives the netlist area obtained by gate sizing alone. For the same clock period, our sequential circuit sizer can tune the circuit to achieve an average of 11.3% area savings over the commercial synthesis tool. These results demonstrate that our optimal sizing algorithm can efficiently tune industrial scale designs to arrive at an optimal clock period, while satisfying the timing constraints for a sequential circuit.

VII. CONCLUSION

In this paper, we propose an optimal gate sizing and clock skew optimization algorithm for globally sizing synchronous sequential circuits. To the best of our knowledge our method is the first exact gate sizing algorithm that can handle cyclic sequential circuits. We believe that this algorithm will be useful for efficient chip level optimization of VLSI circuits.

REFERENCES

[1] J.P.Fishburn,”Clock skew optimization”, in *IEEE Transactions on Computers*, Vol. 39, pp. 945-951, July 1990.

[2] W. Chuang, et al.,”Timing and Area Optimization for Standard-Cell VLSI Circuit Design”, in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14, pp. 308 320, March 1995.

[3] H. Sathyamurthy, et al.,”Speeding up Pipelined Circuits through a Combination of Gate Sizing and Clock Skew Optimization”, in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 17, pp. 173182, February 1998.

[4] S.S. Sapatnekar, et al., ”An exact solution to the transistor sizing problem for cmos circuits using convex optimization”, in *IEEE Transaction on Computer-Aided Design*,vol. 12,pp.1621-1634, 1993.

[5] J. Fishburn and A. Dunlop, ”TILOS: A posynomial programming approach to transistor sizing”, in *Proceedings of the IEEE International Conference on Computer-Aided Design*,pp.326-328, 1985.

[6] C.C.P. Chen, et al., ”Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation,” in *IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems*, vol. 18, No. 7, pp. 1014-1025, July 1999.

[7] J. Nocedal and S.J. Wright, ”Numerical Optimization”, Springer, 2000.

[8] R.H. Byrd, et al.,”A limited memory algorithm for bound constrained optimization”, technical report, Northwestern University EECS, 1994.

[9] H. Tennakoon and C. Sechen, ”Gate sizing using lagrangian relaxation combined with a fast gradient-based pre-processing step”, in *International Conference on Computer Aided Design*,pp.395-402, 2002.

[10] K. Kasamsetty, et al., ”A New Class of Convex Functions for Delay Modeling and their Application to the Transistor Sizing Problem”, in *IEEE Journal of Solid-State Circuits*, Vol. 37, pp. 521-525, Apr.2002.

[11] S. Roy, et al.,”ConvexFit: An Optimal Minimum-Error Convex Fitting and Smoothing Algorithm with Application to Gate Sizing”, in *International Conference on Computer Aided Design*, 2005.