

ゲートサイジングによる回路の最適化

1H-10

田宮 豊
(株)富士通研究所

1 はじめに

同じ機能で駆動力が異なるセルが数種類ずつ用意されているライブラリを用いた回路設計では、ゲートの駆動力の選択により、タイミング、面積および消費電力などの項目について色々なヴァリエーションを持った回路を作ることができる。

例えば、クリティカルパス上のゲートには駆動力が大きくて速いセルを使い、クリティカルでないパス上のゲートではタイミング制約が許す範囲内で面積が小さく消費電力が少ないセルを使うことによって、動作速度はそのまま、面積および消費電力最小の回路が得られる。

しかし、セル交換に伴ってクリティカルパスが変動するため、全体のタイミング制約を満たす最適セル選択を見つけることは困難である。

本手法は、LP(Linear Programming)を用いて、大規模回路に対して、大局的観点から最適セル選択を行なう。まず、回路の最長遅延時刻、最短遅延時刻、面積、消費電力に関する制約および目的関数をゲートの駆動力を変数として線形近似する。そしてこれをLPで解いて、各ゲートの最適駆動力を求める。最後に各ゲートについて、最適ゲート駆動力に合ったセルをテクノロジライブラリから選んで割り付ける。

本手法は、従来のゲートサイジング[1]と異なり、最長遅延だけでなく最短遅延を扱うことにより、set up 制約だけでなく hold 制約を考慮できる。

2 回路制約の線形化

各端子 i において、最長遅延時刻(最も遅い信号遷移が到達する時刻) $T(i)$ 、最短遅延時刻(最も早い信号遷移が到達する時刻) $t(i)$ 、および slew(信号波形の鈍り) $S(i)$ をそれぞれ定義する。各端子において、時刻 $t(i)$ で前データが無効になり、時刻 $T(i)$ で新しいデータが有効になる(図1の信号 i)。

パスの認識は、ネットによるゲート同士の接続関係だけから考慮する[2]。ゲートの論理動作まで考慮していないため、最長遅延時刻は実際より遅めに、最短遅延時刻は早めに評価される。

本手法では、rise と fall の信号遷移を区別するが、複雑さを避けるために以下の表記では区別しない。

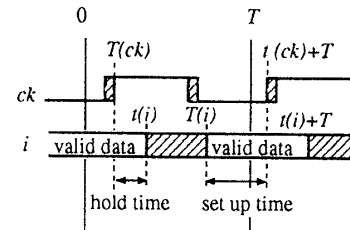


図 1: FF の遅延制約

各ゲートの、入力端子 i と出力端子 j との遅延関係は、以下の通りである。

$$T(j) \geq T(i) + \text{GateDelay}_{ij} \quad \forall i \in \text{fanin}(j)$$

$$t(j) \leq t(i) + \text{GateDelay}_{ij} \quad \forall i \in \text{fanin}(j)$$

ネットの出力端子から fanout 先への遅延関係は fanout 毎の配線遅延差を考慮して、以下の式で表される。

$$T(j) = T(i) + \text{NetDelay}_{ij} \quad \forall j \in \text{fanout}(i)$$

$$t(j) = t(i) + \text{NetDelay}_{ij} \quad \forall j \in \text{fanout}(i)$$

遅延制約の境界条件は、外部入出力端子の到達時刻と Flip Flop またはラッチ(以後まとめて FF と記す)の set up および hold 制約に分類される。

外部入力端子 PI および外部出力端子 PO における境界条件は以下の通りである。4 番めの式で出力端子への信号到達は T (クロックサイクル)以前でなくてはならないことを指定する。

$$T(i) = T_{\text{start}}(i) \quad i \in PI$$

$$t(i) = t_{\text{start}}(i) \quad i \in PI$$

$$S(i) = S_{\text{start}}(i) \quad i \in PI$$

$$T(i) \leq T \quad i \in PO$$

$$t(i) \geq t_{\text{end}}(i) \quad i \in PO$$

上式において、 $T_{\text{start}}(i)$ 、 $t_{\text{start}}(i)$ 、 $S_{\text{start}}(i)$ は外部入力端子 i における到着時刻と slew を、 $t_{\text{end}}(i)$ は外部出力端子 i に与えられている必要時刻を表す定数である。

FF におけるデータ信号 i のクロック信号 ck に対する set up、hold 時間は図1のように定義される。これより、set up および hold 制約は以下の式となる。

$$t(i) - T(ck) \geq W_{\text{hold}}(i, ck)$$

$$(t(ck) + T) - T(i) \geq W_{\text{setup}}(i, ck)$$

表 1: MCNC ベンチマーク回路の結果

circuit	method	delay[ps]	area[bc]	run	circuit	method	delay[ps]	area[bc]	run
C17 8 nodes 13 nets	orig	820.1	36.6	-	S27 21 nodes 26 nets 3 FF's	orig	3,014.7	91.5	-
	S.A.	796.7	36.4	3s		S.A.	2,438.2	91.5	4s
	round	814.3	36.8	13s		round	2,361.1	91.6	15s
	ceil	790.7	37.2			ceil	2,054.7	92.8	
	floor	1,264.6	34.4			floor	2,713.7	90.0	
C432 415 nodes 451 nets	orig	15,215.6	1,647.5	-	S208 156 nodes 167 nets 8 FF's	orig	7,982.0	639.4	-
	S.A.	12,606.7	1,647.3	10m05s		S.A.	4,891.0	639.4	1m26s
	round	11,925.0	1,648.8	50m33s		round	4,297.9	639.1	3m07s
	ceil	11,055.0	1,693.9			ceil	3,985.5	652.7	
	floor	16,399.1	1,621.1			floor	5,410.8	632.5	
C880 657 nodes 717 nets	orig	15,310.6	2,568.4	-	S400 326 nodes 330 nets 21 FF's	orig	1,123.1	1,340.5	-
	S.A.	12,587.4	2,567.5	27m25s		S.A.	804.0	1,340.7	6m16s
	round	10,332.1	2,568.3	2h11m22s		round	821.2	1,340.1	11m29s
	ceil	8,904.2	2,635.6			ceil	821.2	1,350.7	
	floor	13,753.7	2,548.1			floor	781.6	1,326.9	

上式において、 W_{hold} , W_{setup} はそれぞれ最小許容 hold 時間、最小許容 set up 時間を表す定数である。

富士通のゲートアレイ用ライブラリでは、ゲート遅延 $GateDelay_{ij}$ は、駆動力と出力負荷そして入力 slew の関数である。この遅延式は線形でないため、出力負荷一定と仮定して、駆動力と入力 slew について線形近似した。この近似では、実際のライブラリから得られた遅延の値をサンプリングした結果を最小二乗法を用いて、駆動力と入力 slew に関して線形近似した。同様に、ゲートの出力 slew、ゲート面積およびゲート消費電力についても、駆動力に対して線形近似を行なった。

ゲート g の遅延、出力 slew、面積および消費電力を駆動力変数 $drive_g$ で近似して得られる制約式をまとめると以下の通りになる。

$$GateDelay_{ij} = a1_{ij} \cdot drive_g + a2_{ij} \cdot S(i) + a3_{ij} \quad \forall i, j: \text{ゲート内部パス } i \rightarrow j$$

$$S(j) = b1_j \cdot drive_g + b2_j \quad \forall j: \text{ゲートの出力端子}$$

$$GateArea_g \geq c1_k \cdot drive_g + c2_k \quad k = 1 \dots \text{面積制約個数}$$

$$GatePower_g \geq d1_k \cdot drive_g + d2_k \quad k = 1 \dots \text{消費電力制約個数}$$

上式において、 $a1_{ij}$, $a2_{ij}$, $a3_{ij}$, $b1_j$, $b2_j$, $c1_k$, $c2_k$, $d1_k$, $d2_k$ は、最小二乗法で求められた線形近似定数である。

許容最大/最小サイクルタイムと、許容最大総面積/総消費電力は、定数 T_{max} , T_{min} , A_{max} , P_{max} を用いて以下のように表される。

$$T \leq T_{max}$$

$$T \geq T_{min}$$

$$A = \sum_{g \in \text{全ゲート}} GateArea_g \leq A_{max}$$

$$P = \sum_{g \in \text{全ゲート}} GatePower_g \leq P_{max}$$

最後に目的関数は、0 以上のユーザ指定の定数 a , b , c を用いて、サイクルタイム、総面積および総消費電力の線形結合で表す。LP によって、以上に挙げた制約を満たし、かつ、この目的関数の最小化する各ゲートの最適駆動力を求める。

$$Obj = a \cdot T + b \cdot A + c \cdot P \rightarrow \min$$

3 実験結果

面積増加 0、消費電力条件無し、外部出力端子での hold 制約無し、の制約下で、サイクルタイムを最小にする実験を行なった。使用したセルライブラリは、INV、2 入力 NAND、2 入力 NOR および 1 ビット D-FF で、それぞれ、駆動力が異なるセルが 3-5 種類用意されている。LP パッケージには、AMPS (単体法 [3]) を用いた。

LP で求めたゲートの最適駆動力に対するセルの割り付け方法として、一番近いセルを選ぶ (round)、大きめのセルを選ぶ (ceil) および小さめのセルを選ぶ (floor) の 3 通りを試みた。これらの結果と Simulated Annealing 法との比較を表 1 に示す。

C17 を除けば、全ての回路で、round または floor (S400 に対して) が面積制約をほぼ満たしながら、S.A. 法よりも短いサイクルタイムを得ている。これより、本手法が大規模回路では、大局的観点から最適処理ができ、S.A. 法より優っていることが分かる。

参考文献

- [1] Michel R. C. M. Berkelarr, "Area-Power-Delay-Trade-off in Logic Synthesis," *CIP-GEVEENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG*, pp.91-104, 1992.
- [2] Thomas G. Szymanski, "Computing Optimal Clock Schedules," *Proc. of 29th Design Automation Conference*, pp. 399-404, 1992.
- [3] "FUJITSU S ファミリー AMPS 説明書"