2009-SLDM-138 (10)
2009 / 1 / 29

時間多重I/Oを考慮した回路分割手法

磯村 達樹† 稲木 雅人†† 高島 康裕† 中村 祐一†††

† 北九州市立大学 国際環境工学部 情報メディア工学科 〒808–0135 北九州市若松区ひびきの 1–1
†† 広島市立大学 情報科学研究科 情報工学専攻 〒731–3194 広島市安佐南区大塚東三丁目 4 番 1 号
††† NEC システム IP コア研究所 〒211–8666 神奈川県川崎市中原区下沼部 1753
E-mail: †{tatsuki.isomura@is.,takasima@}env.kitakyu-u.ac.jp,
††inagi@hiroshima-cu.ac.jp, †††yuichi@az.jp.nec.com

あらまし 本稿では，大規模システムのプロトタイプに必要である時間多重 I/O の利用を考慮した回路分割手法を検討する．近年の大規模システムのプロトタイプにおいては，複数 FPGA 実装が必要となっている．そこでは，複数の FPGA 間に跨がる信号の通信に利用される FPGA のピン数が信号線数よりも少ないという問題が，深刻となっている．その有効な解決法の一つとして，1 本のピンを時間で区分し，複数信号で共有する時間多重 I/O が提案されている．この時間多重 I/O の利用にあたっては，回路分割と時間多重化する信号線の選択がプロトタイプシステムの動作速度を決定する．ここで，FlipFlop(FF) 間の通過段数が多い信号線を多重化として選択すると動作速度の低下が起こりやすいとの観測から，そのような信号線の重みを大きくするような最小重み分割手法を提案する．そして，その提案手法を計算機上に実装し，実験によりその性能を確認した．

キーワード 時間多重 I/O，複数 FPGA 実装，カット重み最小分割，FF 間段数

Circuit Partition Method with Time-mutiplexed I/O

Tatsuki ISOMURA†, Masato INAGI††, Yasuhiro TAKASHIMA†, and Yuichi NAKAMURA†††

† Faculty of Environmental Engineering
Department of Information and Media Scieneces
The University of Kitakyushu
Hibikono 1–1,Wakamatsu,Kitakyushu,808–0135 Japan
†† Graduate School of Information Sciences
Department of Computer and Network Engineering
Hiroshima City University
3–4–1 Ozuka-Higashi, Asa-Minami-Ku, Hiroshima, 731–3194, Japan
††† System IP Core Research Labs., NEC Corp.
1753, Shimonumabe, Nkahara-ku, Kawasaki, Kanagawa, 211–8666, Japan
E-mail: †{tatsuki.isomura@is.,takasima@}env.kitakyu-u.ac.jp,
††inagi@hiroshima-cu.ac.jp, †††yuichi@az.jp.nec.com

Abstract We propose a partition method to prototype a large scaled system with time-multiplexed I/Os. Recent prototyping of a large scaled system needs several FPGAs. In the prototyping, there is a severe issue that the number of I/Os on one FPGA is less than the number of signals to connect the multiple FPGAs. To solve the issue, a time-multiplexed I/O has been proposed. The time-multiplexed I/O is that a pin is divided into several time slots and shared by several signals. When the time-multiplexed I/O is utilized, the circuit partition and the signal selection to be time-multiplexed determine the speed of the prototype system. In the selection, signal lines on the long path between FFs tend to detoriate the system speed. In this paper, we propose a min-weight partition method in which each signal line on the longer path has large weight. We implement the proposed method on computer and confirm its performance.

Key words Time-mutiplexed I/O, Multi-FPGA implementation, Min-weight partition, FF-FF Stage-number

1. まえがき

近年，設計及び製造技術の進歩により，1チップ上に数百万から数千万のゲートが実装可能となってきている．一方で，システムの大規模化に伴ない，システムの検証が問題となっている．これは対象とする規模が大きすぎるため，既存のソフトウェアによる検証が実用上不可能となるからである．そこで，対処法として，FPGAを用いたプロトタイピングが広く利用されている．しかし，FPGAで実装できる回路規模も制限があるため，対象システムが非常に大規模である場合，複数のFPGAに実装せざるを得ない状況が頻繁に起きている．この複数FPGA実装においては，システムを適切な規模に分割することが必要な処理である．この分割においては，カット数最小分割を行なうことが一般的である [1–6]．しかし，このような分割を利用しても，近年のシステムでは，複数のFPGAに跨がって伝搬する信号線数が，FPGAのピン数よりも多いという問題が起きている [7,8]．このピン数制約の解決策として，1個のピンを時間で区分し，各区分を別の信号に割り当てることにより，信号の伝搬を実現する時間多重I/Oが提案されている [7,8]．

時間多重I/Oの利用では，1) 回路分割，2) 時間多重化する信号線の選択，の最適化が必要である．ここで，後者に関しては，[9] において，分割が与えられたとき，動作周波数を最大化するような最適な多重化信号線選択手法が提案されている．一方，前者に関しては，[10] において，時間多重I/Oの利用を考慮した分割手法が提案されているが，計算時間が大きく，大規模システムへの適用は事実上不可能である．そこで，本稿では，時間多重I/Oを考慮した分割手法を提案する．ここでは，[9] の手法を解析し，FlipFlop(FF) 間のゲート段数が大きいパス上の信号線を多重化するとシステム動作周波数が低減することを観測した．そのため，各信号線を通るパスの最大段数を計算し，その段数に従って信号線に重みを付けて，カット重み最小を求める手法として定式化する．そして，提案手法を計算機上に実装し，既存のカット数最小分割と比較し，性能向上を実験により確認した．

以降，本稿の構成は以下の通りである．2. 節で時間多重I/Oを記述し，3. 節において，多重化信号線選択手法について説明する．そして，4. 節において，時間多重I/Oを考慮した分割手法を提案する．実験結果を5. 節に示し，6. 節でまとめる．

2. 時間多重I/O

時間多重I/Oとは，1個のピンを時間で区分し，各区分を別の信号に割り当てることにより，信号の伝搬を実現するI/Oである．詳細を図1に示す．まず，信号の送信側では，1個のピンに割り当てられた全信号がparallel/serial converterに接続される．そして，parallel/serial converterでは，時間区分毎に割り当てられた信号のデータをI/O clockを用いて転送する．受信側のFPGAでは，送信されて来た信号をserial/parallel converterを用いて各信号に分配する．

この時間多重I/Oを用いたシステムの動作速度は，多重化段数と多重化度の2個の指標を用いて，評価することが可能である．多重化段数とはFF間のパス中で多重化された信号線数を表わし，多重化度とは，1個のピンで転送される信号線数を表わす．ここで，多重化段数が大きいと，通過する時間多重I/O数が増加し，時間多重I/Oでの待ち時間の合計が増大するため，FF間のパス遅延が増大する．また，多重化度が増大すると，1個の時間多重I/Oでの待ち時間が増大するため，FF間のパス遅延が増大する．そして，システム全体の動作速度は，FF間のパス遅延の最大値で決定されるため，これらが増大すると，動作速度は低下する．そこで，[9] では，FF間のパス遅延では，この時間多重I/Oに起因するものが支配的であるとの仮定より，多重化段数を N，多重化度を M としたとき，パス遅延を $N \times M$ で評価する．本稿でも動作速度を $N \times M$ で評価する．

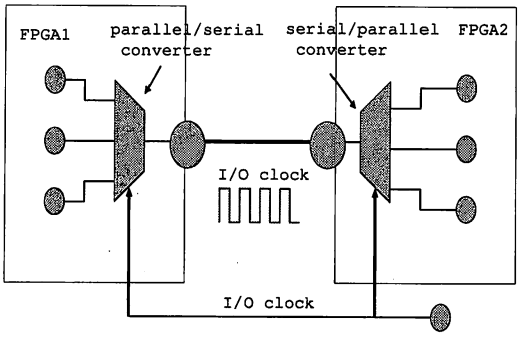

図1 時間多重I/Oのモデル

この時間多重I/Oを利用したシステムでの実装では，以下の2点での最適化が必要である [9]．

- 時間多重I/Oを考慮した分割
- 時間多重する信号線の選択

後者の選択問題においては，[9] において，分割が与えられたときに，線形計画法を用いて最大速度を実現するような多重化する信号選択が決定できる．こちらについては，3. 節において説明する．一方，前者の分割問題に対しては，[10] において，分割手法が提案されているが，実行に時間が必要であるため，大規模回路での利用は困難である．そこで，本稿では，時間多重I/Oの利用を考慮した高速な分割手法を提案する．詳細は，4. 節に記述する．

3. 多重化される信号の選択法

本節では，[9] の信号選択手法を説明する．[9] の手法では，考慮する問題は，

入力： 回路とその2分割結果，及び，最大多重化段数 N^*
出力： 多重化される信号線
制約： 全てのFF間のパスの多重化段数が N^* 以下
目的： 多重化される信号線数の最大化

である．ここで，多重化される信号線数を最大にすることは，必要な多重化度の減少につながり，その結果，動作速度が向上することになる．

提案手法の流れは以下の通りである．まず，与えられた回路をsignal I/Oグラフに変換する．このsignal I/Oグラフとは，分

割された回路中のカットされた信号を頂点に，カット間の接続関係を有向枝で表現したグラフである．例えば，図2のような分割回路に対して，signal I/O グラフは図3のようになる．

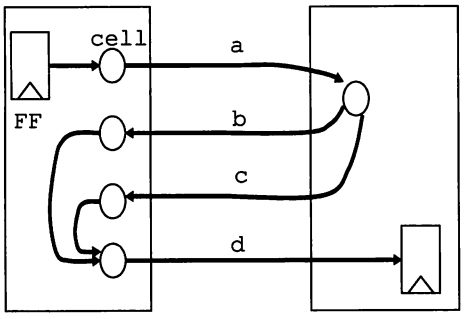

図2 分割の例

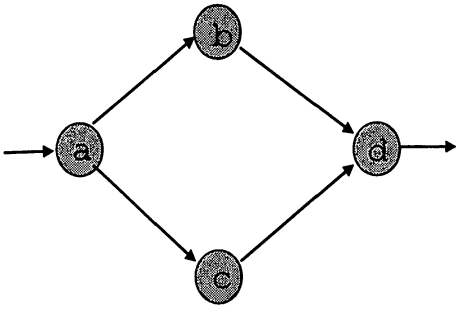

図3 Signal I/O グラフの例

次に，signal I/O グラフに対し，整数線形計画法を用いて定式化する．ここで変数として，signal I/O グラフの各頂点に対応するカットされた信号に対し，その信号が多重化されるかを示す0-1整数変数 A と，その信号を通るパスのその信号までの最大多重化段数を示す整数変数 S を定義する．ここで，2個のカットされている信号 i 及び j において，信号 i が信号 j の直接先行する信号であるとする．すると，式(1)が成立する．

$$S_j \geqq S_i + A_j \tag{1}$$

また，変数 S に関しては，非負であり，かつ，入力で与えられた最大多重化段数 N を下回らなければならない．つまり，式(2)のように表わされる．

$$0 \leqq S_i \leqq N \tag{2}$$

また，目的が多重化される信号線数の最大化より，式(3)で示される．

$$\max \sum_{i \in \text{cutnet}} A_i \tag{3}$$

以上の定式化を図3に適用する．ここで，最大多重化段数 N を1とする．制約は，

$$A_a, A_b, A_c, A_d : 0-1 \text{整数変数}$$

$$S_a, S_b, S_c, S_d : \text{整数変数}$$

$$0 \leqq S_a, S_b, S_c, S_d \leqq 1$$

$$A_a \leqq S_a$$

$$S_a + A_b \leqq S_b$$

$$S_a + A_c \leqq S_c$$

$$S_b + A_d \leqq S_d$$

$$S_c + A_d \leqq S_d$$

である．また，目的は，

$$\max A_a + A_b + A_c + A_d$$

となる．この例の場合，$A_a = A_d = 0$, $A_b = A_c = 1$ が最適解である．

次に，この結果を用いて，多重化度を計算する．式(4)が多重化度と多重化される信号線数との関係式である．

$$(C - X) + \frac{X}{M} \leqq I \tag{4}$$

ここで，

C : カット数

X : 多重化する信号線数

M : 多重化度

I : FPGA のピン数

である．そして，式(4)を M について解くと式(5)が得られる．

$$M \geqq \frac{-X}{C - I - X} \tag{5}$$

多重化段数 N は入力で定まっているため，このときのシステム遅延は $M \times N$ で評価される．

4. 時間多重 I/O を考慮した回路分割手法

本稿で考慮する分割問題の定式化を図4に示す．

入力：	回路
出力：	2分割
制約：	分割後の部分回路のサイズの差
目的：	システムの動作速度最大

図4 分割問題

先にも述べたように，システム遅延は $M \times N$ で評価される．式(5)より，カット数 C が小さくなれば，M も小さくなることは明らかである．このことより，C を小さくすることはシステムの動作速度向上につながる．

しかし，カット数最小化だけでは，不十分である．図5と図6は，それぞれ，最小カット分割と多重化段数を考慮した分割の例を示す．ここで，図5ではカット数が5であり，図6では

カット数が6と大きくなっている．しかし，図5のカットでは，一つのFF間パス上の信号線がカットとなっているため，多重化段数が大きくなってしまう．例えば，FPGAのピン数が3であるとすれば，図5では多重化段数は3必要であるのに対し，図6では多重化段数が1で十分である．その結果，多重化度は，図5では3，図6では2となり，システム遅延が図5では9，図6では2となる．

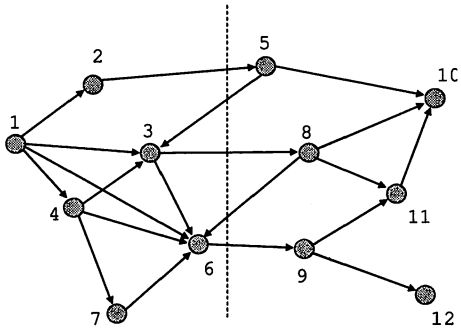

図5　カット数最小分割の例

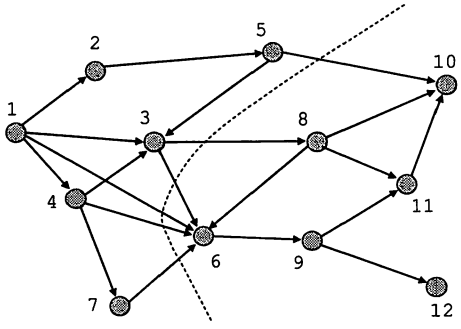

図6　多重化段数を考慮した分割の例

この結果から，我々は分割時に大きな段数のパスを分割しなければ，多重化段数の削減に繋がると考えた．しかし，単純に大きな段数のパスの分割を避けただけでも不十分である．例えば，図7の場合を考える．

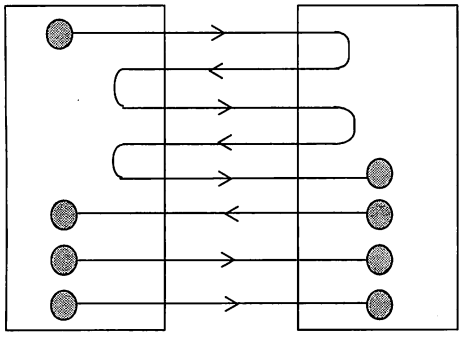

図7　多段パスを分割する例

これは，多段のパスを少数回分割するだけならば，多重化段数を大きくしなくてもピン数制約を満たせるからである．

以上の観察より，我々は，1) カット数をできるだけ小さく，2) FF間段数の大きいパスはあまりカットしない，ような分割を求めることとした．そこで，図8に示すようなアルゴリズムを提案する．ここで，kは段数に対するコスト係数である．

step 1: 各信号を通過する最大パス段数の計算
- **step 1.1:** 入力側からのレベル付け
  - **step 1.1.1:** FFをレベル0に初期化
  - **step 1.1.2:** FFに到達するまでトポロジカルオーダ順にレベル付け
- **step 1.2:** 出力側からのレベル付け
  - **step 1.2.1:** FFをレベル0に初期化
  - **step 1.2.2:** FFに到達するまでトポロジカルオーダの逆順にレベル付け
- **step 1.3:** 各信号に対し，入力側のレベル数 (step1.1 の結果) と出力側のレベル数 (step1.2) の最大値の和に信号自身の1を加えて，その信号の最大段数を計算

step 2: step1 の結果に基づいて各信号線の重みの決定
- **step 2.1:** 最大段数の平均μと標準偏差σの計算
- **step 2.2:** 信号線の最大段数と段数の平均μとの比較
  - **step 2.2.1:** もし 最大段数 $\leq \mu$ ならば 各信号線の重み = 1.
  - **step 2.2.2:** そうでないならば 各信号線の重み $= 1 + k \times \frac{\text{最大段数} - \mu}{\sigma}$.

step 3: カット重み最小分割を求める

図8　提案分割アルゴリズム

図9に図5や図6の回路に対する図8のstep1及びstep2の結果を示す．各点のラベルは(入力側のレベル数, 出力側のレベル数)，信号線のラベルは$k = 0.3$としたときのその信号線の重みを表わす．このアルゴリズムを適用すると，図6が得られる．

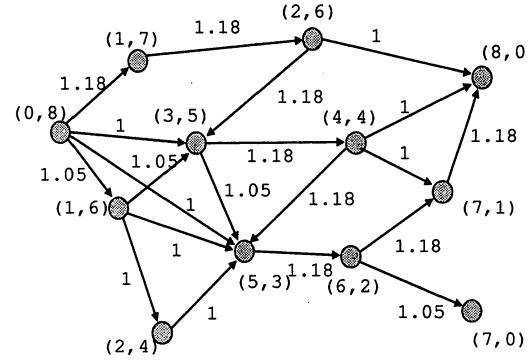

図9　アルゴリズムの適用結果

5. 実験結果

提案分割手法の有効性を確認するため，計算機上に提案アルゴリズムを実装し，実験を行なった．実験環境は，CPUがAMD Opteron 265 (1.8GHz)，メモリが3GB，OSがCentOS 4.3である．また，分割ツールとしては，hMetis [11] を利用した．実験回路として，ISCAS89の回路と，企業からのデータを利用

表1 回路データ

回路名	セル数	信号線数
s1196	575	561
s1423	753	748
s5378	3042	2993
s15850	10470	10383
s38584	20995	20717
industry	253267	253217

表2 実験結果

回路名	k	カット数	$N=1$		$N=2$		$N=3$	
				割合 (%)		割合 (%)		割合 (%)
s1196	0	39.275	20.74	52.81	28.835	73.42	33.795	86.05
	0.2	39.015	23.73	60.82	30.845	79.06	35.805	91.77
s1423	0	36.925	20.56	55.68	29.635	80.26	34.66	93.87
	0.2	36.935	20.405	55.25	29.865	80.86	34.72	94.00
s5378	0	109.32	80.415	73.56	103.315	94.50	107.61	98.44
	0.2	111.46	81.655	73.26	105.09	94.28	109.765	98.48
s15850	0	110.345	86.325	78.23	102.75	93.12	107.675	97.58
	0.2	115.745	89.43	77.26	107.545	92.91	113.02	97.65
s38584	0	109.225	91.825	84.07	108.29	99.14	109.07	99.86
	0.2	110.695	93.07	84.08	109.885	99.27	110.625	99.94
industry	0	903.335	796.34	88.16	890.715	98.60	902.17	99.87
	0.2	879.365	780.01	88.70	869.12	98.83	878.83	99.94

した．表1に，回路データを示す．

実験結果を表2に示す．段数コスト k は，予備実験の結果，最も結果良好であった0.2を選択し，$k=0$(=カット数最小分割)を比較対象とした．表の各列は，カット数が分割時のカットされた信号線数，$N=1$，$N=2$，$N=3$ が分割結果に対し，[9]の手法により多重化される信号線数及びカット数に占める割合を示す．各項目は200回の試行の平均である．

実験結果を見ると，カット数はカット数最小分割とほぼ同等であるが，多重化できる信号線数が増加している．このことより，式(5)から，多重化度が減少し，その結果，動作速度が向上する．

ここで，industryデータに着目する．FPGAのピン数 (I =)500であると仮定し，多重化段数1段のときの遅延を評価する．カット数最小分割のときは，多重化度は $M \geqq 2.03$ より3となる．一方，$k=0.2$ のときでは，多重化度は $M \geqq 1.95$ より2となる．よって，システムの動作速度は，カット数最小では3，提案手法では2となる．以上より，提案手法が実際に動作速度を向上させることが確認できた．

6. まとめ

本稿では，各信号線を通るパスの最大段数を計算し，その段数に従って信号線に重みを付けて，カット重み最小化により，回路分割を実現する手法を提案した．実験により，カット数最小分割と比較して，カット数はほぼ同等の状況で，多重化される信号線数を増やすことができることが確認できた．よって，システムの動作速度の向上が可能となり，提案手法の有効性を確認した．

今後の課題は，最適な段数コストの決定法や，多分割への適用，階層構造を持つ回路への適用などである．

文献

[1] N. S. Woo and J. Kim, "An efficient method of partitioning circuits for multiple-FPGA implementation," *in Proc. of DAC*, pp.202–207, 1993.

[2] Y. Katsura, T. Koide, S. Wakabayashi and N. Yoshida, "A new system partitioning method under performance and physical constraints for multi-chip modules," *in Proc. of ASP-DAC*, pp.119–126, 1995.

[3] J. Gateley, M. Blatt, D. Chen, S. Cooke, P. Desai, M. Doreswamy, M. Elgood, G. Feierbach, T. Goldsbury, and D. Greenley, "UltraSparc-I emulation," *in Proc. of DAC*, pp.13–18, 1995.

[4] H. H. Yang and D. F. Wong, "Circuit clustering for delay minimization under are and pin constraints," *IEEE Trans. on CAD*, pp.976–986, 1997.

[5] K. R. Azegami, M. Inagi, A. Takahashi, and Y. Kajitani, "An improvement of network-flow based multi-way circuit partitioning algorithm," *IEICE Trans. on Fundamentals*, pp.655–663, 2002.

[6] A. B. Kahng and X. Xu, "Local unidirectional bias for cutsize-delay tradeoff in performance-driven bipartitioning," *IEEE Trans. on CAD*, pp.464–461, 2004.

[7] J. Babb, R. Tessier, and A. Agaral, "Virtual wires: Overcoming pin limitations in FPGA-based logic emulators," *in Proc. of IEEE Workshop on FPGA-based Custom Computing Machines*, pp.142–151, 1993.

[8] R. Tessier and S. Jana, "Incremental compilation for parallel logic verification systems," *IEEE Trans. on VLSI*, pp.623–636. 2002.

[9] M. Inagi, Y. Takashima, Y. Nakamura, and A. Takahashi, "Optimal Time-multiplexing in inter-FPGA Connections for Accelerating Multi-FPGA Prototyping Systems," *IEICE Trans. on Fundamentals*, pp.3539-3547, 2008.

[10] M. Inagi, Y. Takashima, Y. Nakamura, and Y. Kajitani, "A Performance-Driven Circuit Bipartitioning Method Considering Time-Multiplexed I/Os," *IEICE Trans. on Fundamentals*, pp.924–931, 2007.

[11] hMetis, http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download .