

汎用エンジンを対象とするメモリ割付け

2G-9

蛇喰和寛 菅沼直昭 富田昌宏 平野浩太郎
神戸大学

1. はじめに

FPGAとメモリを組み合わせることで複数の処理を高速化する汎用エンジン^[1]では、複数のメモリバンクに対するデータの割付けがその処理性能を左右する。例えば、二つのデータa, bを1ポートのメモリから読み出す場合、a, bが割り付けられているメモリバンクが異なれば同時実行できるが、同一であれば2サイクル必要となる。

本稿では、論理関数の探索操作が容易な共有二分決定グラフ^[2](以下BDDと略す)を利用して、動作記述中の配列変数(メモリ変数と呼ぶ)とメモリバンクとの対応関係を決定する手法を提案する。また、大量のデータの扱いを前提とするため、静的な割付けを行う。アクセス操作の存在確率に基づいて割付けを行うことによって、全体の処理をより少ないサイクルに分割することができる。

2. 提案法

2.1 コストの算出

m個のメモリバンクにn個のメモリ変数を割り付けるとする。メモリ変数 $k_i, k_j (i, j=1, \dots, n)$ を割り付けるメモリバンクが同一の場合、バンク共有変数 $x_{ij}=1$ 、異なる場合 $x_{ij}=0$ とする。したがって、 $x_{ii}=1, x_{ji}=x_{ij}$ となる。

このバンク共有変数 x_{ij} に対応するバンク共有コスト c_{ij} を算出する。まず、動作記述中の各操作をデータフロー・グラフで表す。次に、あるステップにおける各操作の存在確率をForce Directedスケジューリング^[3]によって求める。ここで、ある二つのメモリ変数に対するアクセス操作が同一ステップ上にある場合、それらを同じメモリバンクに割り付

けることによるコストを、操作の存在確率に基づいて増加させる。少なくともその二つのメモリアクセス操作の同時実行は不可能であり、実行サイクル数が増加するためである。存在確率からバンク共有コストを算出する例を図1に示す。

2.2 制約式と目的関数

各メモリ変数 k_i に対して、バンク共有ベクトル $X_i=(x_{i1}, \dots, x_{in})$ を考える。ある割付けが無矛盾であるための必要十分条件は、各ベクトルが互いに同一または独立となることである。この条件は(以下、論理AND, OR, XORをそれぞれ \cap, \cup, \oplus で表す)、

$$\begin{aligned} (x_{i1} \oplus x_{j1}) \cup \dots \cup (x_{in} \oplus x_{jn}) &= 0 \\ (x_{i1} \cap x_{j1}) \cap \dots \cap (x_{in} \cap x_{jn}) &= 1 \\ \text{for all } i, j (i, j=1, \dots, n) \end{aligned} \quad (1)$$

で表される。式(1)を満足する場合、必要な割付け先(メモリバンク)の数がm以下であるための必要十分条件は、

$$(\text{独立なベクトルの種類数}) \leq m \quad (2)$$

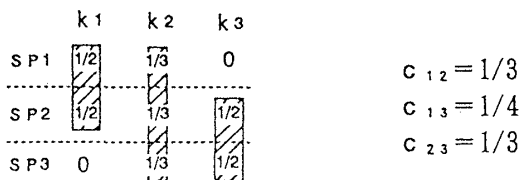
である。ここで

$$y_i = x_{i1} \cup \dots \cup x_{i(i-1)} (i=2, \dots, n) \quad (3)$$

とすると、式(2)は

$$(y_i = 0 \text{ となる } y_i \text{ の数}) + 1 \leq m \quad (4)$$

と同値である。図2に示すBDDの'1'葉に至る経路が式(4)を満足する。ただし、各ノードは式(3)で表される。また式(1), (4)をとともに満足する式が制約式となる。



(a) 存在確率 (b) バンク共有コスト

図1 バンク共有コストの算出

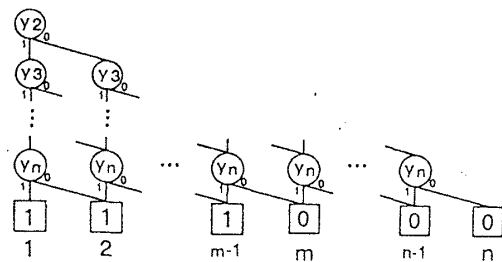


図2 式(4)を展開したBDD

表1 提案法・従来法・人手設計の比較

	X ₂₁	X ₃₁	X ₃₂	X ₄₁	X ₄₂	X ₄₃	X ₅₁	X ₅₂	X ₅₃	X ₅₄	X ₆₁	X ₆₂	X ₆₃	X ₆₄	X ₆₅	X ₇₁	X ₇₂	X ₇₃	X ₇₄	X ₇₅	X ₇₆
提案法	0	*	0	0	0	*	*	*	*	*	*	*	*	*	*	0	0	*	*	*	*
従来法	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0
	0	0	0	0	0	1	0	0	1	1	0	0	1	1	1	0	0	1	1	1	1
人手設計	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
	(0	*	0	*	0	*	*	*	*	*	*	*	*	*	*	0	0	*	*	*	*

この制約式を満足し、 k_i, k_j を同じメモリバンクに割り付けることによるバンク共有コスト c_{ij} とバンク共有変数 x_{ij} との積和で表現される総コスト

$$C = \sum c_{ij} x_{ij} \quad (5)$$

を目的関数とし、これを最小にする x_{ij} の組合せを求めて割付けを決定する。そこで、制約式を展開したBDDについて、バンク共有変数 x_{ij} に対応するノードの'1'枝の長さを c_{ij} 、'0'枝の長さを0とし、ルートノードから'1'葉に至る最短経路を探索する。

ただし、2.1節で述べた方法によって算出したバンク共有コスト c_{ij} を用いる場合、総コスト C を最小にする割付けが、実行サイクル数を最小にするという意味において必ずしも最適とは限らない。

3. 実験と評価

3.1 実験

汎用エンジン・プロトタイプRM-I上に構築した6値シミュレーション・アルゴリズムの動作記述を入力とし、RM-Iと同じくメモリバンク数を4とした。ただし、複数データのワード方向の連結を人手で行った結果を利用し、動作記述上では7個のメモリ変数を用いた。提案法、動作記述の先頭から逐次的に割付けを行う従来法^[4]、人手設計による割付けを表1に示す。括弧内は人手設計と同様のサイクル数となる割付けの一般形である。また、表中の*は制約式を満足するいずれの組合せにおいても総コストが変化しないことを表す。

PC-9801RA21+Turbo-486DX(約12MIPS)上にC言語を用いてアルゴリズムを実装し、Gnu C Compiler ver. 1.39でコンパイルした。処理時間は5秒である。

3.2 評価

表1より提案法は従来法と人手設計の割付けを包含している。したがって、より多くの割付け可能性を示していることが分かる。また、括弧内の割付けと比較しても、0の項が多いところから動作記述上の並列度をより高く抽出しているといえる。したが

って、タイミング制約が問題とならない場合はサイクル数を減少できる可能性があることが分かる。また、実験結果はメモリバンク数が3であっても同じ状態数に分割できることを示している。

4. まとめ

本稿ではBDDを用いて制約関数を表し、メモリ変数の割付けを決定する手法を提案した。今後の課題としては、同一アドレスでアクセスされるメモリ変数のワード連結や同一メモリ変数の複数メモリバンクへの割付けに対する考慮が挙げられる。また、現在の実験環境ではメモリ不足によって、制約関数のBDDによる生成が満足に行なえない場合がある。したがって、表現法の改良など省メモリ化へのアプローチが必要である。

謝辞

BDD演算パッケージに関して御協力を頂きました(株)富士通研究所ソフトウェア研究部の川戸信明部長、同プロセッサ研究部の藤田昌宏氏、松永裕介氏に深謝します。

参考文献

- [1] 菅沼直昭他, "汎用エンジンの開発と論理診断への応用", DAシンポジウム'92, pp. 89-92 (1992).
- [2] 湊真一他, "論理関数の共有二分決定グラフによる表現とその効率的処理手法", 情報処理学会論文誌, vol. 31, no. 4, pp. 532-541 (1990).
- [3] Paulin P. G. and Knight J. P., "Force-Directed Scheduling for the Behavioral Synthesis of ASIC's," IEEE Trans. Comput.-Aided Des. Integrated Circuits & Synt., vol. 8, no. 6, pp. 661-679 (1989).
- [4] 蛇蝮和寛他, "ハイレベル合成におけるメモリ変数割り付け法", 電気関係学会関西支部連合大会, G12-9 (1992).