

VHDL で記述されたシステム仕様の プロセスレベルでの HW/SW 分割の一手法

城代佳範 北嶋 暁 武内良典 今井正治

大阪大学 大学院基礎工学研究科 情報数理系専攻

560-8531 大阪府豊中市待兼山 1-3

E-mail: {jiyoudai, kitajima, takeuchi, imai} @ics.es.osaka-u.ac.jp

あらまし 本稿では並行に動作するプロセスでモデル化された VLSI システムに対しハードウェア・ソフトウェア分割を行う手法を提案する。本手法では、入力としてシステム仕様およびシステムへの入力系列、個々のプロセスを HW で実装したときの面積・処理時間、システム全体の面積制約を与えると、プロセス単位での分割に対し各分割候補ごとにシステム全体の処理時間の見積もりを行い、面積制約を満たす処理時間の短い分割候補を選択する。処理時間の見積もりでは、プロセスの依存関係をもとにスケジューリングを行うことで、入力系列に対し最小に近い処理時間を求めている。ATM の一部機能を例題とした実験では与えた入力系列に対し準最適な分割が得られた。

キーワード 設計自動化, システムレベル設計, HW/SW 分割, 最適化, 面積制約, プロセス依存関係

A Hardware/Software Partitioning Method for Process-level System Specification in VHDL

Yoshinori Jodai, Akira Kitajima, Yoshinori Takeuchi, and Masaharu Imai

Department of Informatics and Mathematical Science
Graduate School of Engineering Science Osaka University

1-3 Machikane-yama, Toyonaka, Osaka, Japan 560-8531

E-mail: {jiyoudai, kitajima, takeuchi, imai} @ics.es.osaka-u.ac.jp

Abstract This paper proposes a Hardware/Software partitioning algorithm for system models written in VHDL. System models are assumed to be specified by concurrent processes. Given the system specification, input sequences, each process hardware area and execution time, and design constraint like system chip area, proposed algorithm determines the HW/SW partitioning where system has the shortest execution time. Proposed algorithm is composed of two steps. First, estimate system execution time for each HW/SW partitioning candidate and renew the partitioning candidates under area constraint. Secondly, find semi-optimal execution times using list scheduling of process dependencies and decide the HW/SW partitioning. From experimental results of ATM model, semi-optimal system partitioning can be acquired.

key words design automation, system level design, HW/SW partitioning, optimization, area constraint, process dependency

1 はじめに

半導体製造技術の発達により1チップに搭載されるシステムが大規模化し続けている一方、システムの機能が複雑になり、従来の設計手法のままでは、チップに搭載可能ではあってもシステムの設計自体が困難になるということが予想されている [1]。これは設計対象となるシステムのアーキテクチャ探索とそれにもとづく最適化が困難になるためである。

システム仕様の設計段階で設計を効率化する方法として、ハードウェア/ソフトウェア (以下 HW/SW) 分割を一定の評価基準のもとで行う手法が提案されている。Gupta らは、HardwareC を用いてタイミングと同期にもとづく分割手法 [2] を、Barros らはデータ依存性・実装しやすさ・リソースの共有から分割を行う手法 [3] を、また Peng らはシミュレート結果と演算の結合強度にもとづいて分割する手法 [4] を、Jerraya らは階層化されたプロセス間通信モデルから設計制約を元に分割を行う手法 [5] をそれぞれ提案している。

本稿では、VHDL で記述されたシステム仕様の設計段階において仕様記述からプロセス間の依存関係にもとづいて HW/SW 分割を自動的に行う手法を提案する。本手法で対象とする VLSI システムのモデルでは、システムの入力系列に対し、複数のプロセスが並行に動作してデータの処理を行い、処理結果の出力を行う。本稿では、このモデルで表されるシステム仕様に対し、各プロセスを HW または SW のいずれかに割り当てるという、HW/SW 分割問題を定式化し、システムのある入力系列に対する、面積制約のもとで最小の処理時間となる HW/SW 分割を行うという問題を扱う。本手法の特徴として、データ処理過程における各プロセスの動作の依存関係に着目し、各プロセスの実行順序 (スケジューリング結果) の違いによるシステム全体の処理時間の違いも考慮していることが挙げられる。

提案する HW/SW 分割アルゴリズムの概要は以下のとおりである。(i) 各分割候補について、システムの見積もり面積の見積もりを行う。(ii) 面積制約に合う分割について、システムの実行時間の見積もりを行う。(iii) システムの実行時間が最短である分割を選択する。面積および実行時間の見積もりでは広大な

解空間を持つため、本稿では効率的に解候補の削減を行い見積もり値を得る。その結果、実行時間の見積もり値に関しては準最適解を得ている。

本手法を用いることにより、VHDL で記述されたシステム仕様記述から、HW/SW 分割とその分割を行った時のシステムの見積もり面積と実行時間の見積もりが実行時間内で得ることが可能となる。

本手法の有効性を確かめるために、ATM の一部機能を例題として評価実験を行った。その結果、本手法により、与えたシステムへの入力系列に対し最適な分割を得ることができた。本実験により、本手法がシステム設計において有用であるという見通しを得た。

本稿は以下のとおり構成されている。第2章では、HW/SW 分割問題の定式化を行う。第3章では提案手法のアルゴリズムを説明する。第4章では ATM の一部機能をモデルとした実験を行い、提案手法を用いた場合の設計早期におけるアーキテクチャ決定のための指針が得られることを示す。第5章で本稿をまとめる。

2 HW/SW 分割問題

2.1 対象とするシステムのモデル

本稿で扱うシステムモデルは図1のように表される。システム S は組 (P, I, O, E) で表す。ここで、 P はプロセス P_i の集合、 I は入力ポートの集合、 O は出力ポートの集合、 E は実行依存関係の集合をそれぞれ表す。

プロセスとは、システムの部分機能の動作定義である。各プロセスは並行に動作する。入力ポートは、データを外部から受け取るポートである。出力ポートは、データを外部へ渡すポートである。実行依存関係は、 $(P \cup I) \times (P \cup O)$ の部分集合であり、 $(P_1, P_2) \in E$ は P_1 の実行に対して P_2 の実行が依存していることを表している。具体的には、 P_1 からの出力データに対し、 P_2 がその出力に対する動作を開始する。

システムへの入力、入力ポート $I \in I$ と入力データ d の組 (I, d) の系列として表す。

システムへの入力系列が与えられたとき、各プロ

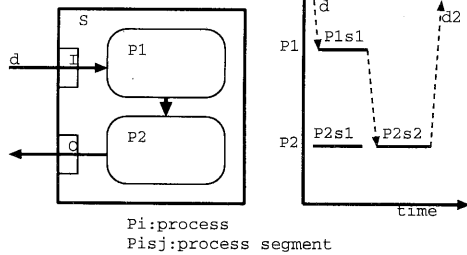


図 1: システムモデル

セス P_i は、データを入力してから次にデータのを行うまでの一連の動作を行う。この一連の動作のそれぞれをプロセスセグメント P_{isj} と呼ぶ。

2.2 VHDL 記述と提案モデルとの対応

前節で述べたシステムモデルは、VHDL 記述に対して図2のように対応させることができる。すなわち、以下の対応となる。

- システム → VHDL 記述全体
- プロセス → プロセス文
- プロセスセグメント → wait 文から次の wait 文の直前までの動作
- データ入力の変化 → シグナルのイベント

VHDL 構文規則より、コンカレント文は等価なプロセス文に書き直すことができる。全ての VHDL 記述はプロセス文のみにより記述することができる。センシティブティ・リスト付きのプロセス文は、wait 文で表現されたプロセス文に等価に書き直すことができる。wait 文を含まないプロセス文は今回は扱わない。

本稿では提案するモデルでのプロセスは、VHDL 構文でのプロセス文と一致する。プロセスセグメントについても対応する VHDL 記述が存在し、提案モデルを VHDL 記述によって記述することが可能である。

2.3 プロセスの HW/SW への実装

2.3.1 実装方法の仮定

本稿では、プロセスごとに、そのプロセスを HW または SW として実装する。すなわち、システムは、各プロセスの動作を実現する HW および CPU とかなる回路へ実装する。

HW として実装される場合、プロセスの機能は実現する回路として実装される。その実現方法は、一方法のみを対象とする。

SW として実装される場合、実現システム上の CPU でプロセスの機能を実現するプログラムとして実装される。一つのプロセスは、同一 CPU 上でのみ実行される。CPU では、プロセスセグメント単位でスケジューリングが行われる。

2.3.2 実装に対する面積・処理時間

プロセスを HW で実現した場合の面積はそのプロセス専用の HW の面積を表す。SW で実現されたプロセスの面積はプログラムが占める ROM/RAM の面積に相当する。分割によらず常に CPU のための面積は必要であるとし、常に必要な面積と考える。プロセスの機能を HW で実現した場合は、SW で実現した場合に比べ HW 実装での面積が大きいとす。プロセス間のデータ伝達のためのインターフェース部分の回路は、他の HW の面積と比べて十分小さいと仮定する。

2.4 HW/SW 分割問題の定式化

以下の入力を与えられたとき、システムの実行時間 T_S が最小である HW/SW 分割を求める。ここでシステムの実行時間とは、システムへの入力系列が与えられてから全ての出力を得るまでに要する時間を指す。またシステムへの入出力データはバッファを経由するものとし、処理時間はイベント待ちの時間を含まないとする。

- CPU の個数
- システム S および S への入力系列 I_S に対する各プロセスごとのプロセスセグメントの系列

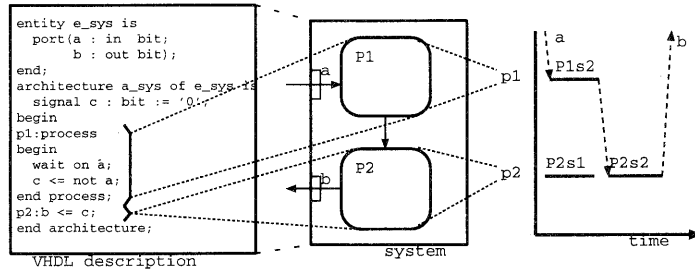


図 2: VHDL 記述とシステムモデルへのマッピング

- プロセスの処理を実現する上で必要な回路面積 A_{P_i} およびプロセスセグメントの実行時間 $T_{P_i s_j}$
- 面積制約 A_{MAX}

3 提案アルゴリズム

提案手法は、大きく 2 段階に分けて HW/SW 分割を決定する。初めに分割候補に対し面積・実行時間の見積もりを行い、次に見積もり値から HW/SW 分割解を求める。

3.1 各分割候補に対する面積・実行時間の見積もり

3.1.1 システムの面積見積もり

システムの面積 A_S は各プロセスの面積の和で求める。

$$A_S = \sum_{i=0}^n A_{P_i}$$

ここで A_{P_i} はプロセス P_i を HW で実装した時の面積である。

面積は各プロセスの実装が決まると求めることができる。すべての分割候補に対し、面積制約条件と比較して、それより大きい面積の分割候補を除くことにより、分割候補を絞り込むことができる。

3.1.2 システムの実行時間見積もり

HW/SW 分割候補が決まり、各プロセスの実現方法が一意に決まったとしても、システムの実行時間の見積もり値 T_S は面積見積もりのような簡単な式で表すことはできない。実行時間は線形問題ではないため、実行時間最短のスケジュールを求めするためには、あり得る全てのスケジュールを尽くす必要がある。

スケジューリングを行った結果に対しては、各プロセスセグメントの処理時間と依存関係にもとづき、システム全体の処理時間を求めることができる。

3.2 HW/SW 分割解の決定

各分割候補に対する面積・実行時間の見積もりにより、分割候補の中から入力系列に対し最適な HW/SW 分割を決定する。

まず、分割候補の中で実行時間最短である分割解を選択する。次に、選択された候補の中から面積最小の候補を HW/SW 分割解として決定する。

3.3 見積もり時間の高速化

分割候補数は、システムを構成するプロセス数を n 、プロセス P_i の実装方法を I_i 通りとしたとき、

$$\prod_{i=1}^n I_i$$

通り存在する。分割候補数が n に対して指数的に増加するため計算時間の高速化が重要になる。提

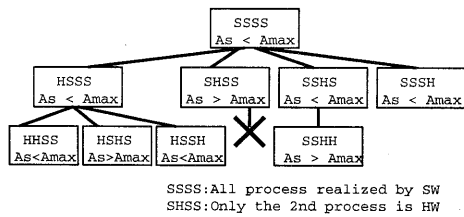


図 3: 分枝限定法による解候補の削減

案手法では、プロセスの面積は HW 実現された時は SW 実現された時よりも大きいという仮定から、分枝限定法を用いて解候補の削減を行っている。分枝限定法を利用した解候補削減のアルゴリズムは以下のとおりである。

- (1) 全てのプロセスを SW で実現した時の面積 A_S を求める
- (2) $A_S \geq A_{MAX}$ ならば、解なし。そうでない時 (4-1) へ
- (3) 先ほど (4-1) で HW 実現としたプロセスを SW 実現に戻し、(4-1) へ。HW で実現していないプロセスがない時はこのアルゴリズムを終了する
- (4-1) SW 実現されているプロセスの中で、まだ HW で実現していないプロセスを HW 実現するとした時の面積 A_S を求める。HW で実現していないプロセスがない時は (3) へ
- (4-2) $A_S \geq A_{MAX}$ ならば先ほど HW 実現としたプロセスを SW 実現に戻し (4-1) へ、そうでない時 (4-3) へ
- (4-3) 現在のプロセスの実現方法の組み合わせを 1 つの解候補とする。(4-1) へ

解探索における解候補の削減の直観的な様子を図 3 に表す。

また実行時間の見積りに対して、各 HW/SW 分割解候補について、与えられた入力系列に対して n 個のプロセスがそれぞれ PS_i 個のプロセスセグメントからなるとき、全てのスケジューリングを行うと

$$\frac{(\sum_{i=1}^n PS_i)!}{\prod_{i=0}^n (PS_i!)}$$

通りの候補が存在し NP 困難な問題である。したがってこの見積りについても分割解候補の枝刈りは重要である。提案手法では、スケジューリングアルゴリズムの一つであるリストスケジューリング [6][7] を用いた。このスケジューリングでは準最適解を得ることができ、計算量は $O(\sum_{i=1}^n PS_i)$ である。

4 評価実験

4.1 実験結果

分割解に対するシステムの実行時間の見積り値の分布及び見積りに要する時間について調べた。また、全数探索で行った場合とリストスケジューリングを使用した場合とで、得られる解に対する実行時間の見積り値および、本手法により解を得るのに要した CPU 時間を比較した。ここで、与えられる、プロセスを HW として実装した場合の面積は全てのプロセスで一定とし、各プロセスセグメントの実行時間についても全てのプロセスセグメントで同じ値であるとする。

実験に用いたモデルの規模を表 1 に、与えた入力を表 2 に示す。モデルは ATM の OAM 機能の一部 (セルの中継ならびに AIS セル生成) をモデル化したもので、[8] を参考にした。入力系列に対するプロセスの依存関係を図 4 に示す。用いた VHDL 記述は ATM の一部の機能を実装しており、入力系列は ATM セルである。表 1 では、プロセスの特徴として、各プロセスに含まれるプロセスセグメント数を表している。入力系列として与えられているものは (a) AIS セル生成 (b) USER セル入力 (c) EMPTY セル入力 (d) OAM セル入力であり、表 2 では、入力系列の違いによるシステムの振る舞いがどのように異なるのかを示している。

入力系列に対する見積り値の分布を図 5 で表し、入力系列ごとに選択された分割解を表 3 に示す。図 5 では、全ての分割解候補についての見積り値の分布を示し、分割解の空間を表している。提案アルゴリズムでは、横軸の値が 5000 以上の候補については、面積制約を満たさないため、実行時間の見積りは行わない。

4.2 考察

表3より、入力系列が異なると得られる分割解も異なることが分かる。表3に示している内容は次のとおりである。提案手法により、分割解候補の中から入力系列に対して最適な分割が選択されている。各入力系列に対して最適な分割は各入力系列の表の一番上の行である。また、比較のために他の入力系列を与えた時に得られた分割解は各入力系列の表の2行目と3行目に示した。プロセス番号0から順に16までの実現方法を表しており、HならHWをSならSWでの実現を示している。例えば入力系列emptyでは、分割解ではプロセス番号0はSWでの実現を表している。また分割解の最後に(u)とあるのはuserと呼ばれる入力系列に対する分割解である。入力系列userで最適と思われる分割は、入力系列emptyではシステムの実行時間は1.25倍かかるという見積もりが得られた。この比較からある入力系列に対しての最適分割は必ずしも他の入力系列に対して最適ではないということが分かる。従い与える入力系列は実際の運用に近いものが必要である。

表4から、分割解候補の数および各分割における実行時間の見積もりにかかる計算量が膨大であることが分かる。提案手法による解候補の削減の結果、例題では全分割解候補8192個のうち面積制約条件から5785個の解候補の枝刈りを行った。その結果、全解候補を見積もる場合に比べ計算時間が約半分になっている。また高速なスケジューリングアルゴリズムを用いないと実行時間の見積もりにおいては、計算時間が実用的ではないことも確かめられた。

5 おわりに

本稿では、HW/SW分割のためのシステムモデルを提案し、このモデルへの入力系列およびシステムの面積制約条件が与えられたとき、準最小の処理時間を持つハードウェア・ソフトウェア分割を求める問題を定義した。この問題に対しプロセス間の依存関係から解を求める方法を提案すると共に、本手法を例題に適用した結果を示した。また、プロセスセグメントのスケジューリングに注目し、探索空間を大幅に縮小し準最適解を求める方法を示した。そし

表 1: モデルの規模

コンポーネント名	プロセス番号	プロセスセグメント数
Timer	0	4
	1	3
Cell Interface	2	2
	3	2
	4	2
	5	2
共通処理部	6	7
UPC 処理部	7	2
OAM 処理部	8	7
セル挿入部	9	2
	10	2
セル挿入部	11	3
	12	4
Cell Header 変換部	12	4
Testbench	13	5
	14	6
	15	9
	16	2
計		64

表 2: 与えた入力

面積制約	5000
CPU の個数	1 個
プロセスの面積	プロセスを構成するプロセスセグメント数に比例
プロセスセグメントの実行時間	一樣(HW実装時はSW実装時の半分)
システムへの入力系列 (総プロセスセグメント数)	(a)AISセル入力(29個) (b)USERセル入力(39個) (c)EMPTYセル入力(36個) (d)OAMセル入力(39個)
プロセスの実現方法の固定	TestbenchはHW実現固定

て実験によりその有効性を実証し本手法が有用であるとの見通しを得た。

本手法は、実験の結果からVHDLで記述されたシステム仕様記述の設計段階で設計妥当性を評価するのに有効であると思われる。また適用モデルが大規模な場合でも、問題が分割できればトップダウンで各部分問題に本手法を適用できると考えられる。

今後の課題としては、プロセス数や依存関係の異なった応用例に対しても実験を行い、提案手法の有用性を確認することである。プロセス間のデータ伝達のインタフェースのコストの考慮、複数CPU搭載システムのCPUの面積など、より精度の高い見積

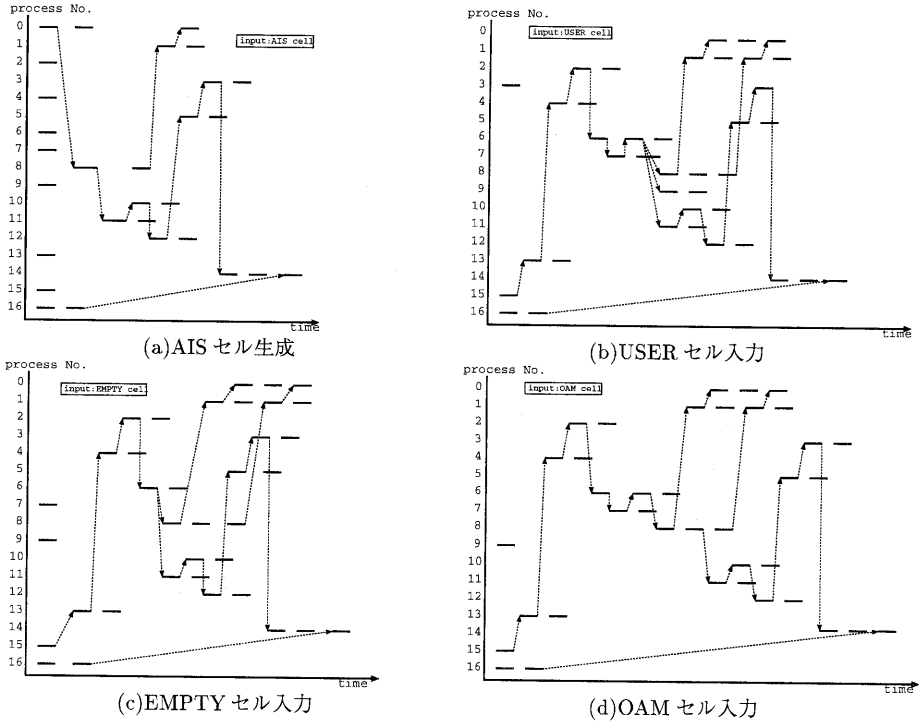


図 4: 入力系列に対するプロセスセグメントの実行依存関係

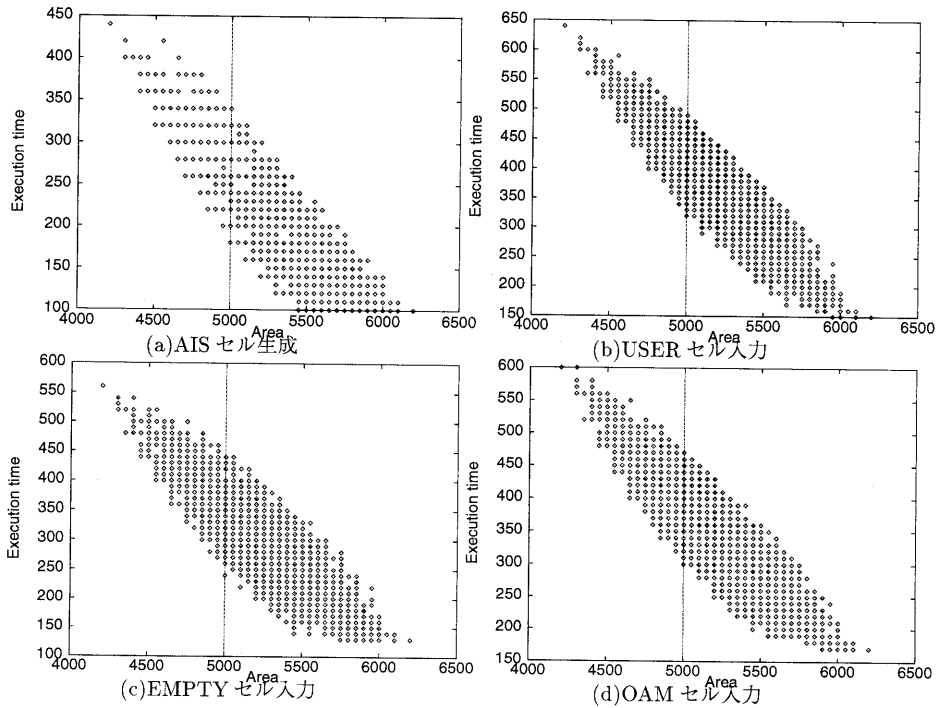


図 5: 入力系列に対する見積み値の分布

表 3: 分割結果

入力系列	最適分割	面積	実行時間
ais	HHS HSHSSSSHHSHHHH	5000	180
	SHSHSHSHSHHHSHHHH(u)	5000	200
	SHHHHHSSSSHHSHHHH(e)	5000	200
計算時間			0.95 秒
user	SHSHSHSHSHHHSHHHH	5000	320
	SHHHHHSSSSHHSHHHH(e)	5000	340
	HHS HSHSSSSHHSHHHH(o,a)	5000	340
計算時間			1.12 秒
empty	SHHHHHSSSSHHSHHHH	5000	240
	SHSHSHSHSHHHSHHHH(u)	5000	300
	HHS HSHSSSSHHSHHHH(o,a)	5000	260
計算時間			1.16 秒
oaml	HHS HSHSSSSHHSHHHH	5000	300
	SHSHSHSHSHHHSHHHH(u)	5000	350
	SHHHHHSSSSHHSHHHH(e)	5000	330
計算時間			1.10 秒

表 4: 高速化の結果 (時間は AIS セル入力系列)

	探索解数	計算時間
提案手法	2407	0.95 秒
全分割候補を計算	8192	1.62 秒
全スケジュールを計算	$8192 \sum_{n=0}^{29} n!$	3 日以上

(Sun UltraSparc 296MHz x1)

もりを提案手法に取り入れていきたいと考えている。

参考文献

- [1] SEMATECH, "EDA Industry Standards Roadmap — 1996," 1996.
- [2] R. K. Gupta, "Co-synthesis of Hardware and Software For Digital Embedded systems," Ph.D Thesis, Stanford, Dec. 1993.
- [3] Edna Da Silva Barros, "Hardware/Software Partitioning using UNITY," Ph.D Thesis, Stanford, Dec. 1993.
- [4] Petru Eles, Zebo Peng, Alexai Doboli, "VHDL System-Level Specification and Partitioning in a Hardware/Software Co-Synthesis Environment," Proceedings of the Third International Workshop on Hardware/Software Co-Design, pp.49-55, 1994.
- [5] Ismail, T. B., Abid, M., O'brien, K., Jerraya A., "An Approach For Hardware-Software Codesign," Proceedings of The Fifth International Workshop on Rapid System Prototyping, pp.73-80, Jun. 1994.
- [6] R. D. Adam, J. Kjelstrup and H. C. Torng, "An Instruction Issuing Approach to Enhancing

Performance in Module Functional Unit Processors," IEEE Transactions on Computers, Vol. C-35, pp.815-828, Sep. 1986.

- [7] Fisher, J. A., "The Optimization of Horizontal Microcode Within and Beyond Basic Blocks; An Application of Processor Scheduling Resources," Ph.D. Dissertation, Technical Report C00-3077-161, Courant Mathematics and Computing Labo, New York Univ., New York, Oct 1979.
- [8] 富永英義, "わかりやすいB-ISDN技術," 新日本 ITU 協会, Oct. 1993.