

## 再利用モジュールのオンライン評価を取り入れた ハードウェア・ソフトウェア協調設計方式とその適用評価

井上 聡 遠藤 祐 荒川 雄一 小泉 寿男

ハードウェア/ソフトウェア協調設計におけるハードウェアとソフトウェア間のトレードオフは重要な課題である。莫大なソフトウェアを開発する組込みシステムでは、再利用の実現性が開発成否の鍵となる。本稿では、ハードウェア/ソフトウェア協調設計において、トレードオフの条件として、開発工数、処理時間、メモリ、ゲート数を取り入れる。さらに、再利用の手法として、再利用部品のリアルタイム評価を取り入れた協調設計方式を提案する。本方式を実現する方法として、システムレベル言語 SpecC を用いる。本方式を ITS(Intelligent Transport Systems)安全運転支援システムに適用し評価を行った。

### A hardware/software co-design method using real time evaluation of re-use components and its evaluation

Satoshi Inoue Yu Endo Yuichi Arakawa Hisao Koizumi

Abstract: The trade-off between hardware and software is an important matter in hardware/software co-design. In embedded systems which develop immense software, the implement ability of software re-use serves as a key of development success. In this paper, we propose a co-design method using trade-off conditions of development man-hours, processing time, memory, and the number of gates. Also in this method we propose a method which incorporates real-time evaluation of re-use parts. The system level language SpecC is used for realizing the proposed method. We applied the method to ITS (Intelligent Transport Systems) safe driving support system and evaluated the effectiveness of the method.

#### 1. はじめに

マイクロプロセッサと LSI は組込みシステム (Embedded System) の主構成要素であり、マイクロプロセッサ上で動作するソフトウェア部分と LSI 化されるハードウェア部分との機能分担を行うトレードオフ作業が重要視されてきている<sup>[1][2]</sup>。従来、ハードウェアとソフトウェアのトレードオフは設計者の経験に依存する要素が多く、最適解が得られているとは言えない。一方、マイクロプロセッサの性能が大幅に向上し、ソフトウェア部分によってカバーされる範囲が増加している。膨大な量のソフトウェアを開発する組込みシステムでは、ソフトウェア部品の再利用が開発成否の重要な要素のひとつである。しかし現状では、所望の再利用部品を迅速に見つけ、動作確認を行うことは難しい。

筆者らは、制御対象モデルと設計対象モデルを結合させたシミュレーションを行い、これをもとにして設計仕様をハードウェアとソフトウェアに機能分担させ、それぞれを段階的に詳細設計していく協調設計方式をすでに提案した<sup>[3][4][5]</sup>。この方式を ITS(Intelligent Transport System) の画像処理系・制御系開発への適用評価を行い、その実現可能性を見出した。

しかしながら、これらの方式においては、目標性能を満足する設計条件下でハードウェアのゲート数を最小にすることにトレードオフの条件を限定している。さらに、これらの方式においては、ハードウェア、ソフトウェアのコンポーネントを再利用する手法を協調設計方式に取り入れていない。

本稿では、トレードオフの条件として、工数、性能、ソフトウェアモジュールの使用メモリ、ハードウェアモジュールのゲート数の複合要因を加味し、この条件下でソフトウェア再利用を実現する方式を提案し、その評価を行う。再利用においては、まず再利用可能なモジュールをネットワーク上のリポジトリから選択し、ネットワーク上でモジュールの動作を評価・確認する。次に、再利用可能な複数のモジュールから複合要因を加味したトレードオフによって最適なモジュールを選択する方法を提案し、その評価を行う。本方式においては、システム機能仕様をハードウェアとソフトウェアの区別なしに同じ文法によって統一的に SpecC で記述する方法を活用する。

本方式の適用評価対象を ITS 安全運転支援システムとした。

#### 2. ハードウェア/ソフトウェア協調設計

提案する協調設計方式のフローを図 1 に示す。本方式では、まず、システム構築における設計目標を

決める。次に目標システムが導入される側の状態を表す制御対象モデルと、システムそのものである設計対象モデルの両者をシステムレベル設計言語 SpecC で作成する。その際、再利用可能な部品の検索を行う。そのモデリング結果を設計目標や制約条件からハードウェア・ソフトウェアに機能分担する。そして、ハードウェア/ソフトウェア設計フェーズで、ソフトウェア部分は SpecC を C 言語に変換し、PC 上で動作させ、ハードウェア部分は SpecC から VHDL (Very high-speed integrated circuit Hardware Description Language) に変換し、FPGA (Field programmable Gate Array) によってハードウェア化する。最後に、比較検証として、FPGA 評価ボードとソフトウェア動作 PC を接続して動作させる。その結果をシミュレーション結果と比較・検証しシステムを作り上げていく。

### 2.1. 設計目標

設計目標では、工数、処理時間、使用メモリ、ゲート数の上限をそれぞれ  $K_{MAX}$ ,  $T_{MAX}$ ,  $M_{MAX}$ ,  $G_{MAX}$  として式 (1) に示す制約条件式をたてる。このときハードウェア/ソフトウェアの機能作成工数をそれぞれ  $k_{HW}$ ,  $k_{SW}$ , 処理時間を  $t_{HW}$ ,  $t_{SW}$ , 新規作成 SW 使用メモリ、再利用部品で構築された SW 使用メモリを  $m_{SW}$ ,  $m_{SWIP}$ , 新規作成 HW ゲート数、再利用部品で構築された HW のゲート数を  $g_{HW}$ ,  $g_{HWIP}$  とする。

$$\left. \begin{array}{l} \text{工数} : K_{MAX} \geq k_{HW} + k_{SW} = k \\ \text{処理時間} : T_{MAX} \geq t_{HW} + t_{SW} = t \\ \text{使用メモリ} : M_{MAX} \geq m_{SW} + m_{SWIP} = m \\ \text{ゲート数} : G_{MAX} \geq g_{HW} + g_{HWIP} = g \end{array} \right\} (1)$$

### 2.2. モデリング

目標システムが導入される側の状態を表す制御対象モデルと、システムそのものである設計対象モデルの両者を結合し、その動作をシミュレーションによって計算し、相方のモデルのチューニングにより設計対象モデルを確定する。この全体モデルを協

調設計方式を進めていく上での機能確認モデルとする。

また、本研究において SpecC によってシステム仕様の記述を行う。SpecC は ANSI C をベースとし、ハードウェア記述のための並列性、同期・通信、タイミングなどの構文を追加した言語である。SpecC ではシステムの構成要素の動作 (計算) をビヘイビア、構成要素間の通信をチャンネルと呼ばれるオブジェクトで表す<sup>[6]</sup>。

### 2.3 再利用モジュールのオンライン評価

再利用においては、まず再利用可能なモジュールをリポジトリから検索し、ネットワーク上でモジュール動作のオンライン評価を行い、選択する方法をとる。

#### (1) 検索

再利用部品を使用する場合、クライアントは構築対象の機能が組織内の設計資産やネットワーク上の IP (Intellectual Property: 再利用部品) ベンダに存在するのかわ、Web サーバに処理名やキーワードで検索し、その検索結果を受け取る。検索結果のページには IP の性能、作成者、実行環境などが記述されている。

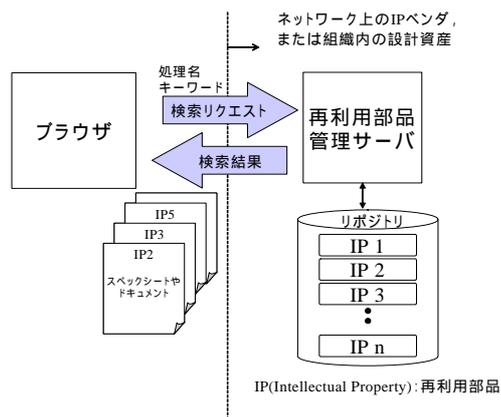


図2 機能検索

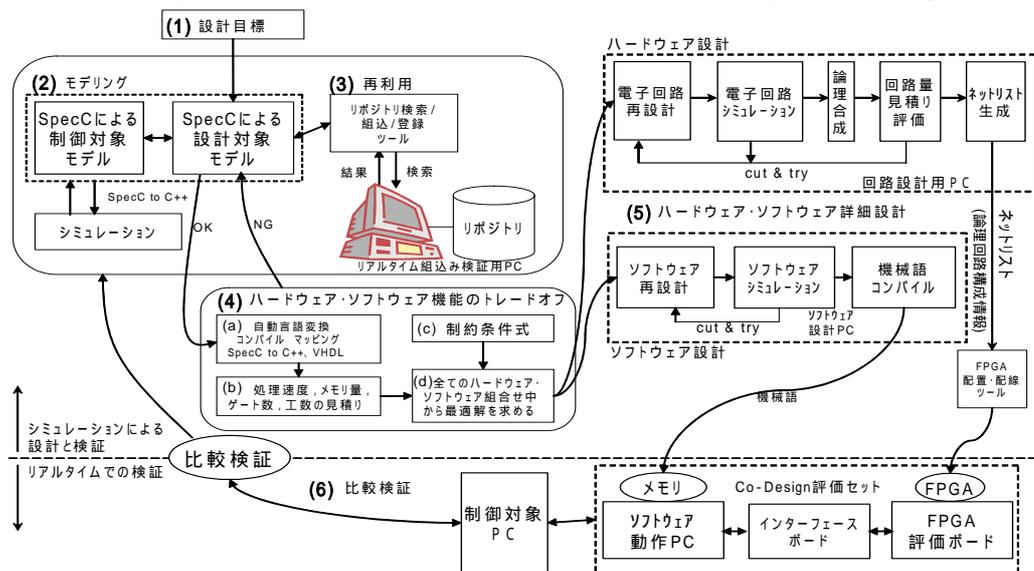


図1 ハードウェア/ソフトウェア協調設計フロー

## (2) オンライン評価

検索の結果、該当した全ての IP を設計 PC 上で動作している再利用しない部分のプログラムと、RPC(Remote Procedure Call)を利用しネットワークで結合しシミュレーションを行う。この際、設計者のプログラムに IP データ構造適合ルーチンや RPC 用のインターフェースコードを付加し、シミュレーションを行う。

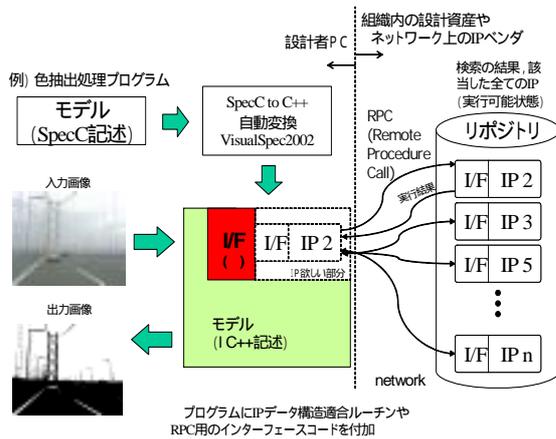


図3 オンライン評価

## (3) 選択

シミュレーションの結果、IP の処理品質が満足できるものでなかった場合は、再利用候補から外す。IP が見つからなかった場合には、新規に作成、もしくは検索キーワードを変更、及び参考となる IP を探し、それを元に作成する。

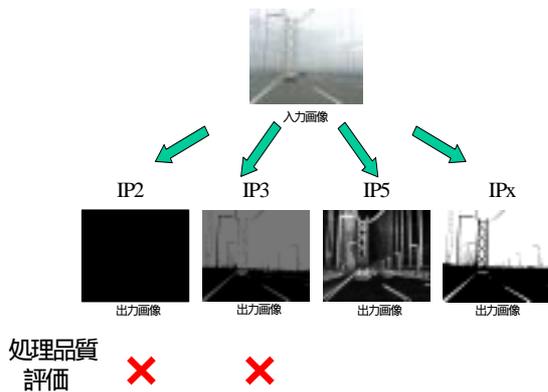


図4 選択

以上に示したオンライン IP 評価方式を用いることにより、ネットワークに接続された環境であれば、短時間で多くの IP の組込み、評価、動作確認を行うことができる。

## 2.4 ハードウェア/ソフトウェア機能のトレードオフ

モデリング結果をハードウェア・ソフトウェアに機能分担する。トレードオフ方式の詳細手順を下記

に示す。

### (1) トレードオフ準備

機能分担に使用する次の項目を VisualSpec2002 にて解析・見積りする。システム全体の処理時間、機能の処理時間、機能の実行回数、機能同士の通信頻度、機能間 I/F の処理時間、機能間 I/F の実行回数、メモリ(ソフトウェアの場合)、ゲート数(ハードウェアの場合)。項目 ~ は、ハードウェアの場合とソフトウェアの場合の2通り見積もる。IP ベンダなどから IP を購入する場合、購入前にはソースを取得できないため、IP の見積もりが行えない。そこで、本研究では、メモリ、ゲート数、処理速度などを問い合わせ可能なメソッドが各 IP に装備されているものとする。

### (2) 大分類

(1) の見積もりの結果から、ある1つの処理で処理時間やメモリサイズが、システム全体での処理時間やメモリサイズをオーバーしている場合はハードウェアに機能分担する。また、1つの処理ゲート数がシステム全体でのゲート数を超過していた場合、ソフトウェアに機能分担する。

### (3) グルーピング

次に、後に行う詳細分類でのハードウェア/ソフトウェア全通りの組合せ見積り時の負荷を軽減するために、(1) の見積もり結果である処理の実行頻度、通信頻度などから、機能間の繋がりの強さ、依存度などを求めて同系列処理のグルーピングを行う。複数候補ある IP は、それぞれの実行頻度、通信頻度が同じとは限らないのでグルーピングしない。

### (4) 詳細分類

#### (i) 制約条件の各要素の単位統一作業

詳細分類では、まず、制約条件にある要素の異なる単位を point という単位に統一する。これは、詳細分類で最適解を point の大小で分かりやすく評価できるようにするためである。まず、工数 1 人日を 1 point としたとき ( $K_{USER} = 1$ )、1 point 相当の処理速度  $T_{USER}$  [ms] と、メモリ  $M_{USER}$  [Kbyte]、ゲート数  $G_{USER}$  [Kgate] を決める。この係数を基に、単位が point で統一された関数を導く。

$$\left. \begin{aligned} P_K &= (K_{MAX} - k) / K_{USER}, & P_T &= (T_{MAX} - t) / T_{USER} \\ P_M &= (M_{max} - m) / M_{USER}, & P_G &= (G_{MAX} - g) / G_{USER} \end{aligned} \right\} (2)$$

#### (ii) 制約条件の各要素の重み付け

単位系が統一された各制約条件要素の重み付けを行う。あるハードウェア、ソフトウェア組合せの機能分担結果を評価する評価関数を  $F$ 、工数、処理時間、メモリ、ゲート数の重み付け係数を  $U_K$ 、 $U_T$ 、 $U_M$ 、 $U_G$  とし、以下の式 (3) を導く。

$$F = U_K P_K + U_T P_T + U_M P_M + U_G P_G (3)$$

なお、(i)、(ii) で二度重み付けをしているが、(i) の単位統一は、例えば会社規模での重み付け係数と想定しており変動が少ないものとする。(ii) の重み付けはプロジェクト単位などの設計管理者レベルでの重み付けを想定している。

### (iii) ハードウェア/ソフトウェア全通りの評価

グルーピングされた処理に対して、ハードウェア/ソフトウェア全通りの組合せで制約条件式と評価関数 F を計算する。制約条件式を満たしており、かつ評価関数 F が最大のものを最適なハードウェア/ソフトウェアの機能分担結果とする。制約条件を満たさなかった場合、アルゴリズムの改善や、設計目標、制約条件の見直しを行う。

## 2.5 ハードウェア/ソフトウェア設計

### (1) ハードウェア設計

ハードウェアは SpecC から VHDL に変換し、シミュレーションや回路量見積もりを繰り返しながら詳細設計を行う。IP を利用した部分は SpecC からの自動変換した VHDL ではなく、IP ベンダなどから支給されるゲート数や消費電力の面での効率を重視してチューニングしてある VHDL を組込む。次に、設計結果を論理合成しネットリストを生成し、ネットリストの結線情報を用い、FPGA によってハードウェア化する。

### (2) ソフトウェア設計

ソフトウェア部分は SpecC からの C 言語変換結果をコンパイラで機械語に変換し PC 上で実際に動作させ、その結果がモデル実行時の動作結果と同じか確認する。さらに処理速度が設計目標・設計環境条件を満たしているかどうかを繰り返し評価、検証しながらソフトウェアのプログラムを作成していく。

## 2.6 比較検証

FPGA 評価ボードとソフトウェア動作 PC を、インターフェースボード経由で連動させる。その動作結果を、PC 画面上に入力し、その結果をシミュレーション結果と目視やデータログで比較検証を行い、目標システムを作り上げる。

## 3. 協調設計環境の構築

本方式における再利用環境、ハードウェア/ソフトウェア混合システムシステムを動作させるためのハードウェア/ソフトウェア間通信の環境を構築した。全体構築図を図 5 に示す。

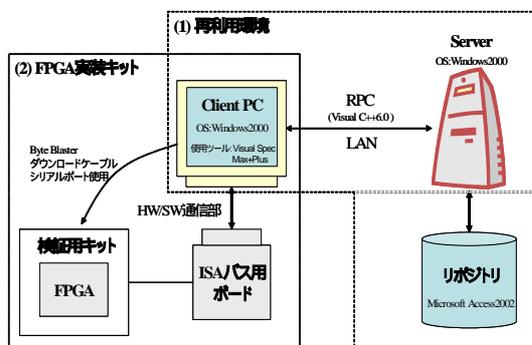


図 5 構築図

### (1) 再利用部品オンライン評価の構築

再利用部品のオンライン評価の構築環境として、クライアントは Web ブラウザを用いて再利用部品

検索、及び結果を受け取る。リポトリには Microsoft Access2002 用い、クライアントとサーバの RPC 連携には Microsoft Visual C++6.0, それぞれの OS は windows2000 を用いた。

クライアントは再利用部品詳細ページに記載してある RPC 接続コードをクライアントのプログラムに記載し、再利用部品と連動を行う。

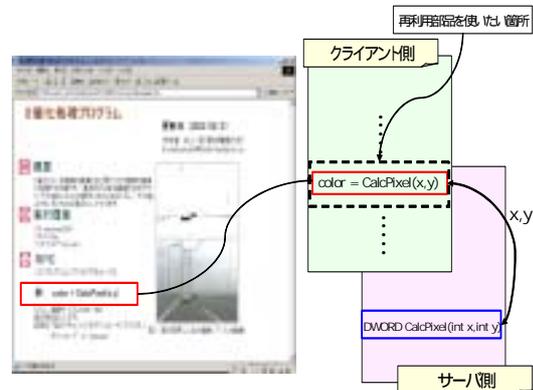


図 6 RPC 接続コード

リポトリにはコンパイル済みの実行形式になっている再利用部品があり、設計者は自身のローカル PC で再利用部品を含まない部分をコンパイルして実行できるようにしておく。そして、あらかじめクライアント・サーバ間を通信する I/F コードをクライアント側のプログラムに組み込んでおき、サーバ上で動作している再利用部品と設計者のローカル PC 上で動作している設計対象をネットワークで接続し、実行させデータをやりとりする。

サーバ側は継続的に実行されており、再利用部品を実行することが必要な RPC クライアントからは常にこの再利用部品を使用できる状態が維持されている<sup>[8]</sup>。

### (2) FPGA 実装キット

本方式による設計結果をリアルタイムの動作で実証するために FPGA コアとして ALTERA 社の FLEX10K50GC403(50K ゲート) を搭載した PowerMedusa MEB200-A50 を用いた FPGA 評価ボードとソフトウェア動作 PC(CPU:Pentium 233MHz, RAM:192MB)の両者をハードウェア/ソフトウェア・インターフェースボードにより連動させる。また、そのハードウェア・ソフトウェア間の通信速度は 4MBPS である。



図 7 FPGA 実装キット

#### 4. ITS 画像処理システムへの適用評価

本方式の適用例として、ITS 開発分野を取り上げ、システム構築し評価を行い、その有効性を確認する。

##### 4.1 ITS 画像処理システムの仕様

ITS は9つの開発分野で構成されているが<sup>[9]</sup>、本研究では安全運転の支援分野を対象とする。画像処理による前方の障害物や停止車両等を検出しドライバーに危険警告を行い、同時に車両に回避操作補助を行う ITS 安全運転システム(前方障害物衝突防止支援システム)に適用をする。ITS 画像処理システムの概要を図8に示す。

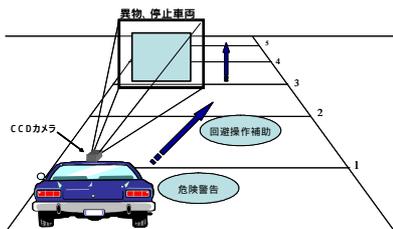


図8 ITS 画像処理システムの概要

入力機器として、自動車上部に設置したカラーCCDカメラ(車上カメラ)から前方道路動画像を危険検出画像処理の入力処理が受け取り、その動画像を用いて色抽出処理がアスファルト、車線、遠・中・近距離異物、道路線形を抽出する。その後、必要十分な画像クリッピング処理や画像平滑化処理を経て、検出処理が前方の道路構造や車両、障害物、レーン、分合流部などを検出する。その結果をもとにドライバーに前方異物による危険を警告するものである。本システムの動作概要図を図9に示す。また、設計目標を表1、設計環境条件を表2に示す。

表1 設計目標

共通	自動車専用的高速道路を対象とする
	専用レーン内を自律走行するシステムとする
	一般車両との混合交通とする
	システムの具体化にあたっては、車載自律型システムを基本とし、インフラの支援は必要最小限とする
	160m先の異物を検出できなくてはならない
制御処理 (運転補助、 自動運転)	入力画像は256*256画素24bitカラーとする
	画像処理は単眼式ビジョンシステムとする
	高速道路で時速100km走行時160m先の異物を回避できなくてはならない
	レーン移動にて回避するのか停車して回避するのか判断できなくてはならない
	異物が静止体なのか移動体なのか判断できなければならない
	移動体であればその速度と方向がわからなければならない
	異物に衝突しない停車制御目標を立てなければならない
	障害物手前10mで停車する
	異物に衝突しないレーン移動制御目標を立てなければならない
	停車制御目標にそった停車制御が出来なくてはならない
レーン移動制御目標にそったレーン移動制御が出来なくてはならない	
常に道路状態を把握し、制御目標を逐次更新できなくてはならない。	
通常走行時は車速・車間・レーンを維持して走行できなくてはならない	

表2 設計環境条件

シミュレーション による検証	Specにてモデリングし、シミュレートする。
FPGA実装キット による検証	前方画像はハードディスクに記録されている動画像情報をもちいる。
	前方画像は、ビデオデッキより再生したものを動画像入力インターフェースボード(A/D変換)に入力する。
共通項目	実証は、動作結果を制御対象PC上に出し、自動車の制御状態や警告状態の確認を行う。
	ハードウェア動作には50KゲートのFPGAを使う。
	ボイスアラームには、PC用スピーカーを使用する。
	ITS DisplayにはPC用モニターを使用する。
	自動車ステータスはPC用シリアルポートを介して入力する。
	ソフトウェア設計・動作には汎用PC( CPU:1GHz,memory:512MByte )を使う。
	外部制御入出力はシリアルポートを介し9500BPSである。
FPGAとソフトウェア動作PCとの通信はISAバス(4MBPS)を使う。	
実証は、入力画像に対する制御信号、警告信号の出力結果のログを取り合う。	

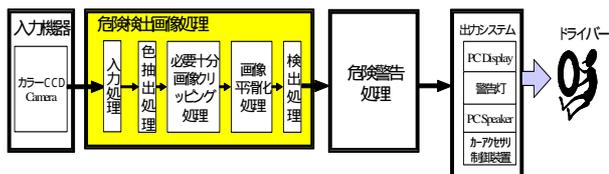


図9 ITS 画像処理システムの動作概要図

#### 5. 結果と考察

図9より本システムに必要なモデリングに使用可能な再利用部品の検索・オンライン評価の一連の動作確認テストを行った。画像処理プログラム、色抽出処理(2値化処理)の再利用部品の検索から評価までを行った。"2値化"というキーワード検索を行い、その結果画面、及び詳細ページを図10、11に示す。



図10 結果画面 図11 詳細画面

詳細ページには、更新日、作成者、概要、実行環境の他、ドキュメント形式でまとめてあるソースコード、仕様、設定可能なパラメータ、実行時の状態や性能等が記述されているものをダウンロードできるようにになっている。

図12をクライアントの入力画像とし、実行結果の出力画像を図13示すとともに、再利用部品のオンライン評価システムを満たす動作を確認した。このオンライン評価システムを用いることにより、今まで再利用部品を自分の設計環境に持ってこないこと、その再利用部品の機能・動作確認ができないことからの選択ミス、選択に慎重になりすぎて時間がかかってしまう問題など<sup>[10]</sup>に対応できると考えられる。また、ネットワークに接続された環境であれば、いつでも評価が可能、また、設計と並列に利用できる環境であり、容易に再利用部品を選べる環境である。

しかし、今のシステムでは再利用部品の容量が大きい場合や大量のデータを流したときにオーバーヘッドが起きる可能性がある。現在、キーワード・処理名などによる検索が行われているが、再利用部品を多くのカテゴリ別に区分けを行い、検索向上を行う必要がある。

再利用部品を利用したモデリングの段階のシミュレーション結果の一部を図14に示す。



図 1 2 入力画像



図 1 3 出力画像



図 1 4 シミュレーション実行画面

## 5. まとめ

筆者らは、協調設計方式を ITS 画像処理系・制御系開発に適用し、構築・評価を行ってきた<sup>[3][4][5]</sup>。本稿では、システム全体を SpecC でシミュレーションを行い、トレードオフの条件として、開発工数、処理時間、メモリ、ゲート数を取り入れ、リアルタイムに再利用部品を評価・組み込む環境を加えたハードウェア/ソフトウェア協調設計方式を提案し、設計環境の構築とその評価を行った。

再利用モジュールのオンライン評価に関しては、ネットワーク上のリポジトリからの再利用候補モジュールの検索、モジュール機能のネットワーク上での評価、その結果に基づく最適モジュールの選択という一連の方式の検証を行い、その有効性を確認した。

今後は、本方式の評価を充実するとともに、協調設計方式の改善を行っていく予定である。

## 参考文献

- [1] Thomas D.E., Adams J.K, and Schmit H., "A Method and Methodology for Hardware-Software Codesign", IEEE Design and Test of Computers, Vol. 10, No. 3, pp6-15(1993)
- [2] 内藤貴志, 山田啓一, 山本新: 撮像位置にロバストなナンバープレート認識方式, 電子情報通信学会論文誌 A, Vol. J81-A, No. 4, pp. 536-545(1998).
- [3] 遠藤, 小泉, 清尾, "ハードウェア・ソフトウェア協調設計方式と ITS 画像処理開発への適用検証", 電気学会論文誌 D, T. IEE Japan, Vol. 120-D, No. 10, pp. 1118-1126(2000).

[4] 遠藤 祐, 吉田 健, 井上 聡, 飯田庸介, 小泉寿男, "ITS 画像処理系・制御系開発におけるハードウェア・ソフトウェア協調設計方式", 情報処理学会論文誌, Vol. 43, No. 12, pp3745-3755(2002).

[5] 井上, 遠藤, 吉田, 飯田, 小泉, 清尾, "SpecC を用いたハードウェア/ソフトウェア協調設計方式と ITS 安全運転の支援システムによる評価", 情報処理学会 ITS 研究会, (2003)

[6] DANIEL D.GAJSKI 他著, "SpecC 仕様記述言語と方法論", CQ 出版株式会社(2000)

[7] 本田 晋也, 高田 広章, 中島 浩, "SpecC によるソフトウェア記述の実装記述への変換", 情報処理学会論文誌 コンピューティングシステム, Vol. 44, NO. SIG 11, pp236-245(2003)

[8] M・ブレイン+ R・リーブス 著, "Win32 システムサービスプログラミング 改訂版", ピアソン・エデュケーション(2002)

[9] 国土交通省道路局 ITS のホームページ: <http://www.mlit.go.jp/road/ITS/j-html/>

[10] 望月 洋介, "ブラック・ボックス IP 目指した設計環境(前編)", NIKKEI MICRODEVICE 1月号, pp. 141-147(2002)