

[招待論文] コンフィギュラブルプロセッサによる 画像認識プロセッサの開発

宮森 高†

†株式会社東芝 セミコンダクター社 SoC 研究開発センター

〒212-8520 神奈川県川崎市幸区堀川町 580-1

E-mail: †takashi.miyamori@toshiba.co.jp

あらまし コンフィギュラブルプロセッサをベースにして画像認識用プロセッサ“Visconti (VIsion based Sensing, CONTrol, and Intelligence)”を開発した。3つの同時に3命令を実行できる VLIW 型のプロセッサがビデオ入出力や SDRAM コントローラなどの周辺回路と1チップに集積されている。VLIW プロセッサは、SIMD 型の命令を実行できる画像処理に専用化した VLIW コプロセッサ RISC 型のプロセッサコアに拡張している。Visconti は、0.13 μ m CMOS テクノロジーで作られ、動作周波数は 150MHz、消費電力は約 1W である。また、実際の画像認識アプリケーションの例として、車両の周辺監視と顔認識システムを紹介する。

キーワード コンフィギュラブルプロセッサ、インテリジェントビークル、画像認識、マルチプロセッサ、VLIW、SIMD

Configurable Processor MeP and its SoC Design Examples

Takashi Miyamori†

† SoC Research & Development Center, Semiconductor Company, TOSHIBA Corporation

580-1 Horikawa-Cho, Saiwai-Ku, Kawasaki, 212-8520, Japan

E-mail: †takashi.miyamori@toshiba.co.jp

Abstract We developed an image recognition processor, “Visconti,” based on a configurable processor. Three VLIW processors that execute three instructions in parallel are integrated into a single chip with peripheral modules such as video I/Os and an SDRAM controller. Each VLIW processor has a RISC processor core and a VLIW coprocessor dedicated to image processing. The coprocessor executes SIMD instructions such as 8-parallel byte. Visconti was fabricated using 0.13 μ m CMOS technology, operates at 150MHz, and consumes about 1W. We present actual application examples of Visconti, onboard surveillance for automobiles and face recognition.

Keyword Configurable Processor, Image Recognition, Intelligent Vehicle, Multiprocessor, VLIW, SIMD

1. はじめに

画像認識技術を用いた応用が実用化されてきている。例えば、走行レーンや周囲の走行車両を認識して運転の補助を行うインテリジェント・ビークルや、顔や指紋認識を用いたセキュリティシステムなどである。インテリジェント・ビークルでは、ミリ波レーダーが幾つかの車種で採用されている。次世代のインテリジェント・ビークルでは、カメラを使った画像処理によって、システムコストが低減されることが期待されている。前方障害物検出や追い越し車両検出などの、より複雑なアプリケーションを実現するためには、高い性能を持ちながらも低コスト、かつ、低消費電力で動作する画像処理 LSI が必要となってきている。

以前、我々は車載後側方監視システム用 [1] に、画像

認識プロセッサ [2, 3] を開発した。しかし、このプロセッサでは、ステレオカメラによる前方障害物検出などのアプリケーションに対しては性能的に不足していた。我々が PC ベースで評価した結果では、前方障害物検出を実現するためには、MMX 命令を使用した 1GHz 以上の PC 用の汎用プロセッサ程度の性能が必要であった。車載などの組み込み応用では、高い性能とプログラム性を維持しながらも、低コスト、高信頼性で、かつ、温度耐性や振動耐性などの厳しい動作環境が求められた。このような条件を満足する画像認識プロセッサ“Visconti (VIsion based Sensing, CONTrol, and Intelligence)”を開発した。

殆どの画像処理アプリケーションは、本質的に多くの並列性を持っており、それらの並列性を効率よく引

き出すことにより、低消費電力かつ低コストの画像処理 LSI を実現することが可能である。“Vision Instruction Processor” [4]や、“IMAP-CE” [5] などの画像処理プロセッサが発表されている。これらの LSI では、SIMD(Single Instruction Stream, Multiple Data Stream)アレイプロセッサを内蔵し、画像処理を高速化している。一方、今回開発した Visconti では、同時に 3 命令実行可能な VLIW(Very Long Instruction Word)プロセッサを 3 つ内蔵したマルチプロセッサ構成を採った。それぞれの VLIW プロセッサでは、最大 8 並列の SIMD 型命令を実行可能である。マルチプロセッサ構成によってタスクレベルの並列性、個々の VLIW プロセッサによって命令レベルの並列性、さらに SIMD 型命令によってデータレベルの並列性を引き出すことが可能である。

本論文では、はじめに Visconti のアーキテクチャとチップの実装について述べる。次に、画像認識アプリケーションの例を示す。車両の周辺監視と顔認識システムにおいて Visconti アーキテクチャの効果を議論する。

2. Visconti のアーキテクチャ

図 1 に Visconti のブロック図を示す。Visconti は 3 つのプロセッサ (MeP モジュール) を持ち、アフィン変換モジュール、SDRAM コントローラ、3 系統のビデオ入力、1 系統のビデオ出力、ホストインタフェース、Flash ROM コントローラを持つ。

2.1. MeP モジュール (カスタム VLIW プロセッサ)

Visconti は、東芝のコンフィギュラブルプロセッサである MeP [6, 7] をベースに設計された。コンフィギュラブルプロセッサコア (MeP コア) に画像処理用にカスタマイズされたコプロセッサを接続し、VLIW(Very Long Instruction Word)プロセッサを構成している。さらに、各 MeP モジュールは、DMA コントローラ(DMAC)を内蔵しており、グローバルバス I/F によって、オンチップ上のバスに接続される。

● MeP コア

MeP コア自体は、5 段のパイプライン、シングルスカラ構成のシンプルな RISC タイプのプロセッサコアである。また、キャッシュサイズ、内部命令メモリ・データメモリのサイズなどを設計時に選択することができるコンフィギュラブルプロセッサである。Visconti では以下のような構成を選択した。

- 命令キャッシュ: 4KB
- 内部命令RAM: 4KB
- データキャッシュ: 8KB
- 内部データRAM: 64KB
- 割り込みチャネル数: 17 チャネル
- オプション命令: 32ビット乗除算命令など

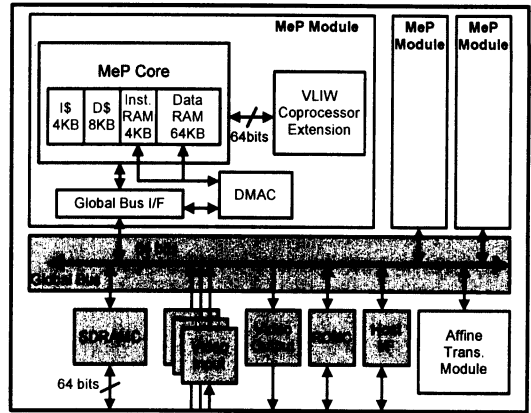


図 1. Visconti のブロック図

Visconti 中にある 3 つの MeP コアは、全て同じ構成である。この構成は、シミュレーションによる性能評価、想定するプログラムで使用する命令とデータのサイズ、ゲート数や動作周波数の見積もりをベースに構成を決定した。例えば、画像データがある程度まとめて内部データ RAM へ格納させるため、データ RAM のサイズを 64KB と大きくした。データ RAM は、4 つのバンクから構成されており、MeP コアがアクセスしている間でも、同時に DMA コントローラが他のバンクにアクセスすることが可能である。これによって、データ転送のオーバーヘッドを隠蔽することが可能となっている。

● VLIW コプロセッサ

MeP コアは、コア内部のコンフィギュレーションを選択するだけでなく、専用の命令や、ハードウェアアクセラレータを拡張することができる。これによって、プロセッサをカスタマイズして、特定の処理の性能を向上させることができる。Visconti では、画像処理に特化した VLIW 型のコプロセッサを MeP コアに追加した。

図 2 に、MeP コアと VLIW コプロセッサの構成を示す。VLIW コプロセッサを接続した場合、2 つの動作モードがある。1 つはコア動作モードであり、このモードでは MeP コアのみが、普通の RISC プロセッサとして動作する。もう一つのモードは、VLIW 動作モードであり、この場合は MeP コアと VLIW コプロセッサを合わせて、同時に最大 3 命令を実行できる VLIW プロセッサとして動作する。命令 RAM、命令キャッシュからの命令フェッチを MeP コアが行い、MeP コアと VLIW コプロセッサに 64 ビット固定長の命令を発行する。

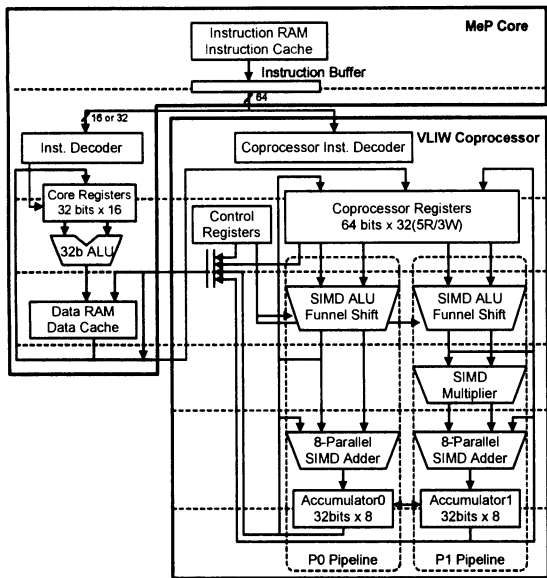


図 2. プロセッサコアと VLIW コプロセッサの構成

● 8-parallel funnel shift and multiply-accumulate instruction (CPFMADIA1.B CRq,CRp,Imm3,Imm8)

```
CPFMADIA1.B: CRq,CRp,Imm3,Imm8 //CRq <- (CRq,CRp,Imm3[2:0],Imm8)
|-----|1010_000qL_00pppL_111111|
signed 64 tmp;
tmp = ULShift(CRq,Imm3[2:0]*8) or RShift(CRp,(64-Imm3[2:0]*8));
acc1_7 = acc1_7[31:0] + (SignExtension(tmp,63:56) * SignExtension(Imm8))[17:0][31:0];
acc1_6 = acc1_6[31:0] + (SignExtension(tmp,55:48) * SignExtension(Imm8))[17:0][31:0];
acc1_5 = acc1_5[31:0] + (SignExtension(tmp,47:40) * SignExtension(Imm8))[17:0][31:0];
acc1_4 = acc1_4[31:0] + (SignExtension(tmp,39:32) * SignExtension(Imm8))[17:0][31:0];
acc1_3 = acc1_3[31:0] + (SignExtension(tmp,31:24) * SignExtension(Imm8))[17:0][31:0];
acc1_2 = acc1_2[31:0] + (SignExtension(tmp,23:16) * SignExtension(Imm8))[17:0][31:0];
acc1_1 = acc1_1[31:0] + (SignExtension(tmp,15: 8) * SignExtension(Imm8))[17:0][31:0];
acc1_0 = acc1_0[31:0] + (SignExtension(tmp, 7: 0) * SignExtension(Imm8))[17:0][31:0];
Cop:"SIMD=8,TraceOut,Group=ACC1_RW,Pipe=crq_r_crp_r_acc1_rw";
P1:
```

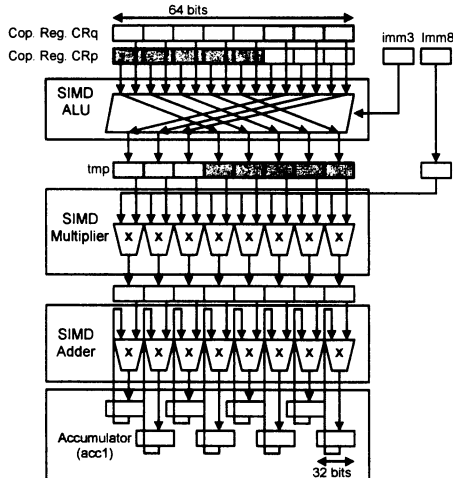


図 3. VLIW コプロセッサ命令の例

VLIW コプロセッサは、P0 と P1 の 2 つのパイプラインをもち、64 ビット幅の SIMD(Single Instruction Multiple Data)型命令を実行する。SIMD 命令は Intel 社の MMX[8]などと同様のものである。

図 2 にあるように、VLIW コプロセッサの P0 パイプラインは、SIMD 型の ALU と SIMD 型の加算器とアキュムレータをもっている。SIMD 型の ALU では、加減算に加えて、バイト単位のファネルシフト操作ができる。メモリからのロード命令で読み出されたデータは、64 ビット単位でアライメントが取られているので、ファネルシフトはバイト単位でのデータの選択に有益である。

もう一つの P1 パイプラインは、P0 パイプラインの構成に加えて、SIMD 型の乗算器を内蔵している。P1 パイプラインでサポートしている複雑な SIMD 命令の例(CPFMADIA1.B 命令)を図 3 に示す。CPFMADIA1.B 命令は、2 つの 64 ビットのコプロセッサレジスタ(CRq, CRp)から読み出した値を、定数 Imm3 で示されるバイト位置に合わせてファネルシフトする。その結果の各バイトデータに定数 Imm8 を掛け、アキュムレータ 1(acc1)に足し合わせている。図 3 に書かれている命令動作の定義は、カスタム VLIW コプロセッサ開発ツール“Arcadia”[9]の入力である。Arcadia は、このようなコプロセッサ命令の定義(コード、ニーモニック、オペランド、動作、SIMD 並列度、レイテンシなど)を基に、そのコプロセッサにカスタマイズされた、コンパイラ、シミュレータなどの開発ツールを自動的に生成する。このため、コプロセッサの命令セットの評価を、設計の初期段階に行うことができる。実際に、Visconti の VLIW コプロセッサの命令セットの設計では、5 つの命令セットの候補を評価した。ターゲットのアプリケーションで最も処理が重いと考えた正規化相関関数の性能と、ゲート数の見積りの中から、最適なものを選択した。この命令セットの設計は、1 人の技術者が 2 週間程度で行うことができた[9, 10]。

3. 実装

Visconti の開発では、コンフィギュラブルプロセッサ MeP の設計プラットフォームを利用した。前章で説明したコアの構成やコプロセッサの命令セットの設計だけでなく、グローバルバス、SDRAM コントローラなどの周辺 IP を再利用した。Visconti で新規に開発したのは、VLIW コプロセッサとアフィン変換モジュールという画像処理のキーになる部分だけであった。これらがチップ全体に占める割合は約 3 割にすぎなかった。

Visconti では、プログラムをまずソフトウェアシミュレータ上で動作確認した上で、チップの設計時の RTL シミュレータや、チップが完成した後の評価ボー

ド上で動かしている。デバッグ機能が豊富なソフトウェアシミュレータによって、開発効率が上がるだけでなく、実際のチップができる前にプログラムを先行して開発することができる。また、チップ設計時の検証においては、実際の画像処理のソフトウェアを使って、動作確認を行うことができ、検証の品質を上げることができた。この結果、最初のチップを入手してからわずか2週間で、実機ボード上で最初の画像認識デモを動かすことができた。

このように、コンフィギュラブルプロセッサの設計プラットフォームを利用することで、VLIW コプロセッサの命令セット策定を含む仕様策定からテープアウトまでを9ヶ月で行うことができた。以前に開発したチップ[2, 3]より回路規模は4倍であるにも関わらず、開発期間を40%ほど短縮することができた。

図4に Visconti のチップ写真、表1に諸元を示す。

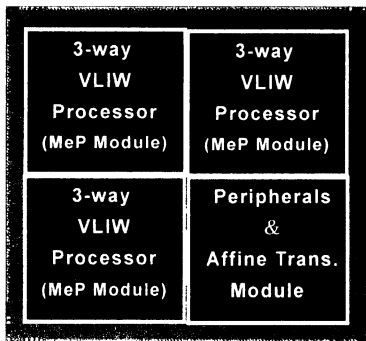


図4. Visconti のチップ写真

表1. Visconti の諸元

項目	仕様
プロセス	0.13 μ m CMOS 6層メタル
ピーク性能	18 GOPS (6 GOPS x 3 プロセッサ)
動作周波数	150MHz
トランジスタ数	21M (うち、17M はメモリ)
オンチップ RAM サイズ	260 K バイト
消費電力	1W (1.5V 時)
チップサイズ	6.98mm x 6.98mm (48.7mm ²)
パッケージ	456ピン PBGA

4. 応用例

Visconti は、車載の前方ステレオ監視[11, 12]や白線検出、後側方監視などのさまざまなアプリケーションに使用できる。ここでは、車両の周辺監視システムと顔認識について紹介する。

4.1. 車両周辺監視システム[13]

自動車を運転するときには、人間の五感のうち特に視覚から多くの情報を得ている。車両周辺監視システムでは、画像認識技術を用いて、カメラによって撮影

された画像から周辺車両を検出する。これによりドライバーを補助し、高度な安全性を提供することができる。

先行車両検出の方法について説明する。先行車両を後方から観測した場合は、その中にバンパーやナンバープレートの縁などの水平線分が多数存在する。そこで、前方の画像の中から、水平線分を検出し、検出できた水平線分を画像中で追跡し、その線分の動きの情報を取得する。動きの情報が得られた全ての水平線分から、画像中の横方向の位置がほぼ等しい縦方向に並んだ3本の水平線分を選択する。図5を使ってその方法を説明する。3本の水平線分が先行車のような立体物に属し、図5(a)のように先行車が手前に近づいた場合、その間隔の比率は一定である。一方、3本の水平線分が道路面に属する場合、図5(b)のように、その比率が変わってくる。この違いを用いて、3本の水平線分の動きが、先行車のように道路上置かれた立体物を仮定した場合に観測されるものか、あるいは、道路面を仮定した場合に観測されるものかを調べる。後側方の追い越し車両についても、同様の原理で検出できる。



(a) 先行車両の場合 (b) 道路上の標識の場合

図5. 先行車両の検出方法

本手法では、テンプレートマッチングによって、前後のフレームでの水平線分の動きを検出している。まず、追跡する水平線分の上部に矩形領域(テンプレート)を設定する。次に、次のフレームで検出されている近傍の水平成分を選択して、テンプレート間の比較を行い、類似している領域が見つかった場合に同一の水平線分と考える。

Visconti を使った場合の、先行車両検出の性能についてまとめる。ソフトウェアの開発は、PC 上で開発した C 言語のアルゴリズムをベースに、Visconti で動作を確認し、次に、Visconti の VLIW コプロセッサがサポートする SIMD 命令、DMA 転送を用いて高速化を図

った。表 2 に 320x240 画素の画像 1 フレームを、1 つの MeP モジュールで実行した場合の処理時間の例を示す。1 フレーム全体の処理を 15ms で実行でき、16 倍の高速化が達成できた。

Visconti には 3 つのビデオ入力があり、前方に 1 台、左右後側方にそれぞれ 1 台ずつ、計 3 台のカメラによって撮影された周囲画像を入力できる。さらに、内蔵されている 3 つの MeP モジュールを使って、1 チップで同時に前方と左右の後側方の監視を行うことができる。

表 2. 周辺監視での実行サイクル数

処理	高速化前	高速化後
エッジ方向の計算	10,162	3,884
水平線分の検出	74,849	2,697
水辺線分の追跡	101,369	2,793
面方向の判定	7,694	1,232
1 フレーム全体の処理合計	232,058	14,956

4.2. 顔認識システム[14, 15]

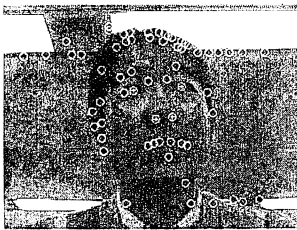


図 6. 特徴点候補の検出結果

顔画像による個人認証は、非接触で認証できること、認証と同時に顔画像を記録できることから利便性に優れている。しかし、人間の顔パターンは、顔の向きや表情、照明条件などにより多様に変化するため、ロバストな顔認識は難しく、高い処理能力のあるハードウェアが必要であった。

Visconti を用いた顔認証の手法について説明する。顔認証には、顔検出と顔認識の 2 つのステップがある。まず、顔検出では円形分離度フィルタによって 4 つの特徴点の候補を検出する。図 6 に顔特徴点候補の検出結果を示す。次に、あらかじめ作成しておいた目と鼻のパターンの辞書と候補点の近傍のパターンを比較する。これによって、図 6 中のプラス(+)で示された瞳と鼻腔の 4 点が特徴点として抽出される。2 番目のステップは、顔認識である。ロバスト性を持たせるために、ビデオからの画像系列を用いている[14, 15]。検出された特徴点から顔パターンを抽出して複数の顔パターンを主成分分析することで作成した入力部分空間と、事前に登録した辞書部分空間との類似度を計算する。

図 7 を用いて、3 つの MeP モジュールを用いたタスクレベルの並列処理について説明する。図は、左から

処理するフレームの番号、3 つの MeP モジュールに割り振られているタスクを表している。例えば、n-1 番目のフレームにおいて、顔検出が失敗した場合は、n 番目のフレームでは、新たに画像全てから顔の特徴点を検索する必要がある。このときは、画像を 3 分割してそれぞれの MeP モジュールで分担して特徴点を検出し、最後にそれをマージすることで、1 枚の画像の処理時間を短縮することができる。

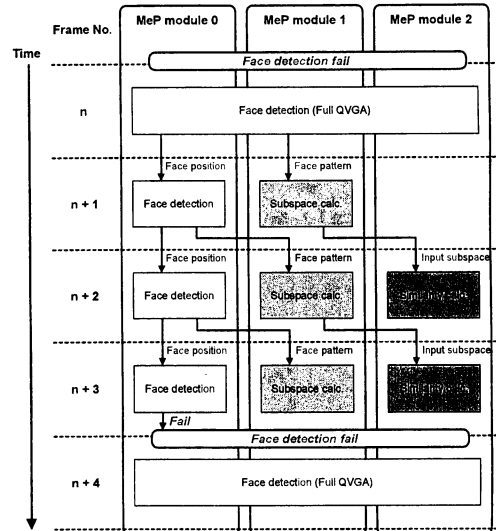


図 7. 顔認識の処理フロー

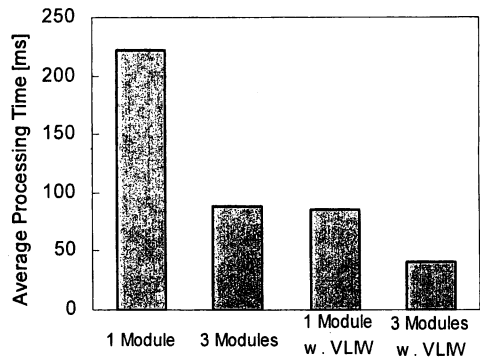


図 8. 顔認識の平均実行時間

顔検出が成功した場合は、次の n+1 番目のフレームでは、検出した顔の特徴点付近を探索する、特徴点のトラッキング処理を行う。トラッキング処理は、全探索に比べれば処理量は少ないため、MeP モジュール 1 つで行い、他の MeP モジュールは別のタスクを行う。具体的には、1 つ目の MeP モジュールは、検出された特徴点からの顔パターンの切り出しと入力部分空間の

作成を行い、もう1つのMePモジュールでは、入力部分空間と事前に登録した辞書の部分空間との類似度を計算し、認識結果を出力する。このように、顔検出の状況に合わせて、3つのMePモジュールで行うタスクを動的に変えることで、処理の負荷の平均化を図っている。

また、部分空間の演算はほとんどが積和演算である。ViscontiのVLIWコプロセッサのSIMD型の積和命令を用いることで、16ビット整数の積和演算を4並列で処理することができる。SIMD型の命令は、フィルタリング処理にも適しており、特徴点検出処理での分離度フィルタ処理も高速化することができる。

Viscontiでの顔認識処理の実行時間について評価した。Visconti評価ボードに、ビデオを入力してその処理時間を測定した。Viscontiのアーキテクチャの特徴である、3つのMePモジュールによる並列処理の効果と、SIMD型命令をサポートするVLIWコプロセッサの効果について調べた。図8に、1フレーム分を処理するのにかかった時間の平均を示す。3つのMePモジュールを使った場合、あるいは、VLIWコプロセッサを使った場合は、2倍以上高速化された。さらに両方を用いた場合は、5倍程度の高速化が図られ、約20fps(50ms/frame)の実時間処理を達成できた。

5. まとめ

我々は本論文において、新たに開発した画像認識プロセッサ Visconti について述べた。今回開発した Visconti では、3つの同時に3命令実行可能なVLIWプロセッサを持つマルチプロセッサ構成を採った。

実際の画像認識アプリケーションの例として、車両の周辺監視と顔認識システムを説明したが、Viscontiのアーキテクチャによって、プロセッサコア1つのみで実行した場合と比べて、車両の周辺監視では1つのプロセッサで約16倍、顔認識システムでは3つのプロセッサ約5倍の高速化が図られ、これらの応用を実時間で処理することができた。

文 献

- [1] Y. Taniguchi and K. Okamoto, "Automatic Rear and Side Surveillance System using Image Processing." Proc. of 6th World Congress on Intelligent Transport Systems, 1999.
- [2] Y. Kondo, et al., "A 4GOPS 3Way-VLIW Image Recognition Processor Based on a Configurable Media-processor." IEEE International Solid-State Circuits Conference (ISSCC) digest of technical papers, pp.148-149, 2001.
- [3] H. Takano, et al., "A 4GOPS 3Way-VLIW Image

- Recognition Processor Based on a Configurable Media-processor." IEICE Trans. Electron., Vol. E85-C, No.2, pp.347-351, Feb. 2002.
- [4] W. Raab, et al., "A 100-GOPS Programmable Processor for Vehicle Vision Systems", IEEE Design & Test of Computers, pp.8-15, Jan-Feb., 2003
- [5] S. Kyo, et al., "A 51.2 GOPS Scalable Video Recognition Processor for Intelligent Cruise Control Based on a Linear Array of 128 4-Way VLIW Processing Elements." IEEE International Solid-State Circuits Conference (ISSCC) digest of technical papers, pp.48-49, 2003.
- [6] T. Miyamori, "A Configurable and Extensible Media Processor," Embedded Processor Forum, 2002.
- [7] <http://www.MePcore.com/>
- [8] A. Peleg and U. Weiser, "MMX technology Extension to the Intel Architecture." IEEE Micro, pp.42-50, Aug. 1996.
- [9] A. Mizuno, "Design Methodology and System for a Configurable Media Embedded Processor Extensible to VLIW Architecture," IEEE Int. Conf. on Computer Design, pp.2-7, Sep. 2002
- [10] J. Tanabe, et al., "Visconti: Multi-VLIW Image Recognition Processor based on Configurable Processor." Proc. of IEEE Custom Integrated Circuits Conference (CICC), pp.185-188, 2003.
- [11] H. Hattori, "Stereo for 2D Visual Navigation." Proc. IEEE Intelligent Vehicles Symposium, pp.31-38, 2000.
- [12] H. Nakai, et al., "A Practical Stereo Scheme for Obstacle Detection in Automotive Use," Proc. of Int. Conf. on Pattern Recognition (ICPR), 2004.
- [13] K. Furukawa, et al., "Onboard Surveillance System for Automobiles Using Image Processing LSI," IEEE Intelligent Vehicles Symposium, pp.555-559, 2004.
- [14] O. Yamaguchi, K. Fukui, and K. Maeda, "Face Recognition Using Temporal Image Sequence." Proc. of 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition, pp.318-323, 1998.
- [15] T. Kozakaya, et al., "Development of a Face Recognition System on an Image Processing LSI chip." Proc. of CVPR Workshop on Face Processing in Video (FPIV'04), 2004.