

4. データ駆動型メディアプロセッサ

Data-Driven Media Processor by Toshiya OKAMOTO (A1206 Project Team, SHARP Corp.) .

岡本 俊弥¹

¹ シャープ (株) A1206プロジェクトチーム

1. はじめに

1997年4月、オランダでAsync97^{*}が開催された。これは非同期処理の回路設計/アーキテクチャに関する学会で、回路や応用分野に関わる各種研究が発表されている。日本からは寺田浩詔教授 (高知工科大学) が、データ駆動型プロセッサについて講演を行った¹⁾。ここでの話題は、非同期処理による低消費電力化の可能性である。また、これとは別に、東京大学・先端科学技術研究センターの南谷教授のグループでは、TITAC-2 (MIPS社R2000ベース) と呼ばれるクロックを持たない32bit非同期式マイクロプロセッサを試作している²⁾。

非同期処理が注目を集めている。この静かなブームは、非同期信号処理が将来的に同期式クロック方式よりも、低消費電力で高速な回路/プロセッサを提供しようと考えられているためである。今でこそ同期クロックを全回路に均等に分配することができているが、将来の大規模回路で同じことが保証されるわけではない。非同期転送でプロセッサができるのなら、どれだけ大きなLSIであっても、クロックスキューや同時スイッチング・ノイズといった同期設計特有の問題から解放される。

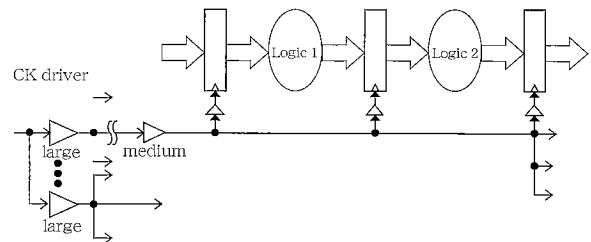
同期クロック方式と、非同期処理によるパイプライン構成の違いを図-1に示す。

図-1 (a) に従来のクロックツリー方式で設計されたパイプラインを示す。この回路では、全パイプラインに常にクロックが供給されているため、たとえデータが流れていない回路でも動作は止まらない。これらの対策として、回路ブロックごとのクロック制御機能導入や、もう少しマイクロなレベルでは、Gated Clock方式³⁾によるクロック停止回路導入などが図られてきた。しかし、いずれにしても有効なデータの有無を検知すること自体が難しいため、本質的な解決とはなっていない。

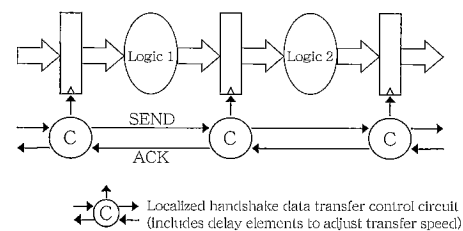
図-1 (b) にクロックレス方式によるパイプラインを示す。PLL (Phase Locked Loop) などをパイプライン制御にいったい使わない代わりに、各パイプライン段間にC素子と呼ばれる²⁾ 一種のハンドシェイク回路を設け、クロック信号を自己発生させている (自己同期回路とも呼ぶ)。このクロックは、データがパイプラインに入ったときのみ有効な信号である。言い換えるとデータがこないときには、まったく動作しない。これはデータ検知を要しない、きわめて自然な省電力回路ともなっている。

非同期処理 (=我々はクロックレス方式と称している) とデータ駆動方式、この2つは必ずしも不可分の関係にはない。データ駆動方式でもクロックを持つものがあるし、通常のプロセッサをクロックレスで作ることもできる。しかし、データ駆動の考え方と非同期処理の設計技術とは、きわめて親和性が高いといえる。

その第1の理由は、データ駆動型の方式自体が、非同期割込み入力 (処理) を前提としているからである。入力が非同期であるのなら、パイプラインも非同期であることが省電力の面から望ましい。この省電力機能が



(a) 従来型クロックツリー方式のパイプライン



(b) 自己同期 (クロックレス) 方式のパイプライン

図-1 従来型クロックツリーと自己同期方式のパイプラインの違い

^{*} THIRD INTERNATIONAL SYMPOSIUM ON ADVANCED RESEARCH IN ASYNCHRONOUS CIRCUITS AND SYSTEMS, Eindhoven University of Technology, Eindhoven, The Netherlands (1997).

²⁾ <http://www.hal.rcast.u-tokyo.ac.jp/titac2/index-j.html>

³⁾ 回路ブロックのクロック入力にスイッチを設け、クロックの停止/動作を外部から制御する方式。

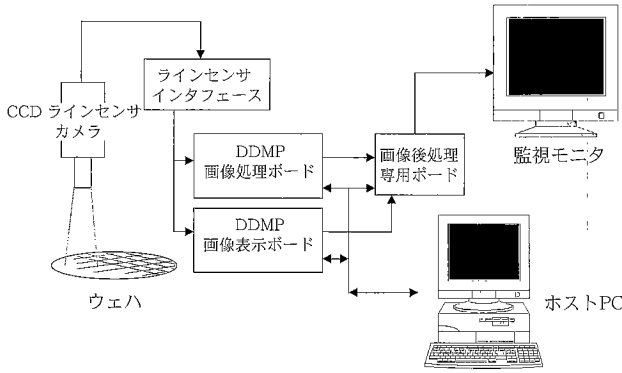


図-2 DDMPを用いたウェハ外観検査装置

は、今日の情報家電向け応用のように、動作時/待機時ともに省電力となる高速プロセッサを求めている分野では特に重要な要素となっている。

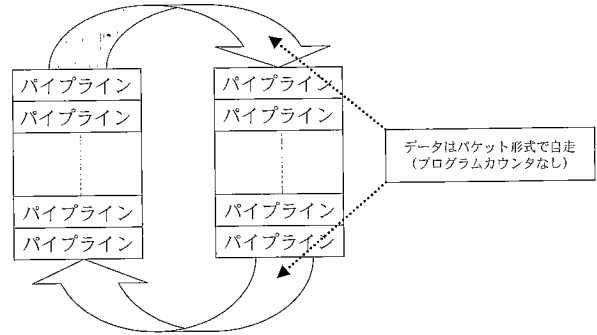
2. ハードウェアからソフトウェアへ

データ駆動型プロセッサの歴史は意外に古い。基礎を成す、データフローの概念は1972年にデニスらが提唱したことに始まった³⁾。実用化面でも、1984年に初の16bitデータ駆動型プロセッサチップが開発され⁴⁾、1991年には32bit浮動小数点演算器を内蔵したプロセッサチップが出される⁵⁾、⁶⁾など、意欲的な成果が出された。当社でも、データ駆動型プロセッサ(以下DDMP=Data - Driven Media Processorと呼ぶ)を開発し、過去画像処理向けに、32bitタイプ、映像信号処理向けに12bit⁷⁾タイプを商品化した。半導体検査装置向けの応用例を図-2に示す。

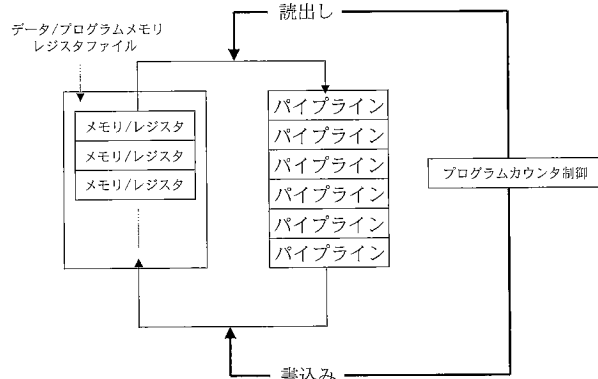
この例では、ウェハからの接写画像を4096画素のラインセンサから10~15MHzで取り込み、フィルタ処理、圧縮・伸長処理(ノイズ除去のために実施)などを行っている。その間、テンポラリなワーク用内蔵ラインメモリを除けば、いっさい外部メモリを用いず、リアルタイムに処理を実施している。9つのプロセッサを使用し、実効処理性能はおよそ2~3GOPS(ピーク性能ではない)、プログラムは約70ステップである。図中DDMP画像処理ボードで上記検査処理が行われ、DDMP画像表示ボードではモニタの解像度にあわせた画像の縮小を実施している。

しかし、産業向け画像処理分野は現状限定された市場であり、数量ベースの広がりには期待できない。コスト面でのブレイクスルーを行うためには、一般のユーザが誰でも使えるような機器への対応、すなわち民生家電機器への搭載が必要である。

長い間、高度並列処理は民生家電/情報/通信分野とは無縁の領域だった。しかし、急激なデジタル化が進んだ結果(たとえば、MPEGなどのデジタル圧縮された映像信号の処理)、まず専用のASICが現れ、次にプログラマブルなプロセッサが求められるように



(a) データ駆動型プロセッサのパイプライン構造



(b) ノイマン型プロセッサのパイプライン構造

図-3 データ駆動型とノイマン型プロセッサの構造の違い

なった。

プロセッサが目される要因として考えられるのが、ソフトウェア処理による商品開発期間の短縮効果である。いわゆるメディア処理を実現するためには、従来のASIC開発手法では1年以上の期間が必要である。しかし、現在の商品寿命はそれほど長くはない。開発に半年をかけたASICが、わずか3カ月の寿命だった例もある。ソフトウェアはハードウェアに比較して修正が簡単にできるため、短期間の再利用が可能である。少なくとも、デバイス設計完了から試作/量産完了までの生産期間が、ソフトウェアではまったく存在しない。商品出荷後のバージョンアップも自在にできる。もともと、こういったソフトウェアの利点は過去から知られていた。しかし、メディア処理のような高速演算性能を要求される分野では、ハードウェア以外の手段がなかったのである。

3. データ駆動型プロセッサの動作原理

データ駆動型プロセッサの最大の特徴は、図-3(a)に示すリング状にしたパイプライン構造にある。通常のノイマン型プロセッサでは、パイプラインは図-3(b)に示す直線状である。この違いは、従来型プロセッサがデータをメモリまたはレジスタに貯えて処理するのに対して、データ駆動型ではデータ自身がパイプライン上を流れるために生じる。ノイマン型では、データは最初にメモリから読み出され、パイプライン

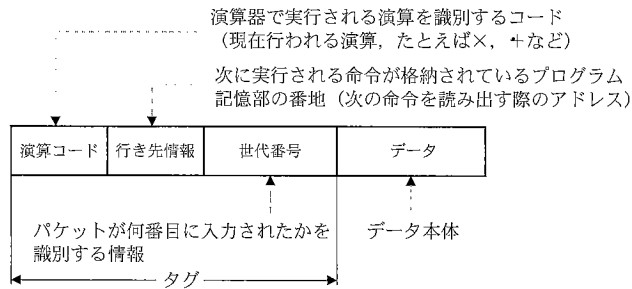


図-4 データ駆動型プロセッサの packets 形式

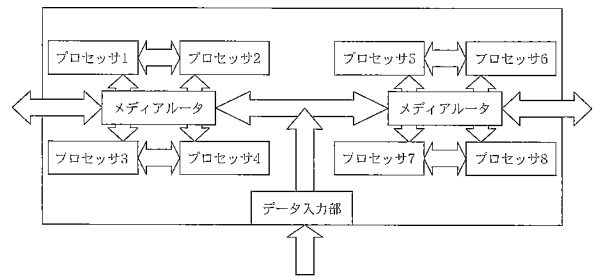


図-6 DDMP-2400Mのブロック図

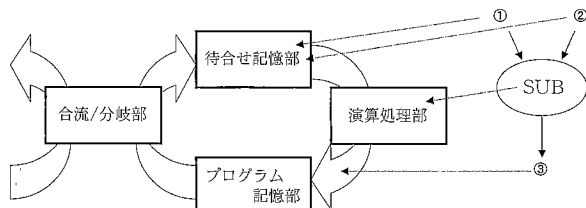


図-5 データ駆動型プロセッサのブロック図

に投入され (一連の計算を経てから), 再びメモリに書き込まれる. この読み書きの動作を制御するのはプログラムカウンタである. メモリへの読み書きが必ず前後に挟まれるために, パイプラインは一直線で済む.

一方, データ駆動型では, データは packets の形式で直接パイプラインに入ってくる. そのデータは, 原理上メモリまたはレジスタへの読み書きなしで処理される. この場合, 一連の動作を制御するプログラムカウンタはなく, その代わりに, 各データにはそれぞれのデータを管理するタグがつけられる. 図-4にデータ

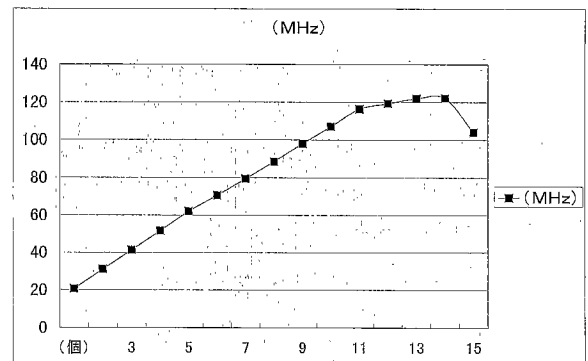


図-7 DDMP-2400Mの動作特性

駆動型プロセッサの packets 形式を示す. データ駆動型プロセッサでは, 内部で扱われるデータはすべて packets の形式をとる. これは, 電話交換機などでデータを流す際に使われる, いわゆる「packet」と等価なものと考えてよい. すなわち, データ+識別タグ (行き先情報など) で作られる1つの単位である. ここでの識別タグが, 各データ単位のプログラムカウンタと等価な働きをする. 世代番号とある部分は, プロ

表-1 データ駆動型プロセッサDDMPの性能比較

| | 40M | 600M | 2400M | 7600M*4 | | |
|--------------|------------|--------------|--------------|--------------|--------------|--------------|
| 内部転送速度 | 50MHz | 90MHz | 150MHz | 120MHz | 150MHz | 200MHz |
| 内部演算速度 | 33MHz | 60MHz | 120MHz | 100MHz | 120MHz | 170MHz |
| 1コアの演算器数 | 1 | 2 | 2 | 6 | 6 | 6 |
| 1コアの性能*1 | 33MOPS | 120MOPS | 240MOPS | 600MOPS | 720MOPS | 1020MOPS |
| コア (プロセッサ) 数 | 1 | 4 | 8 | 8 | 8 | 8 |
| ピーク性能 | 33MOPS | 480MOPS | 1920MOPS | 4800MOPS | 5760MOPS | 8160MOPS |
| 相対性能*2 | 1 | 15 | 60 | 160 | 190 | 270 |
| 消費電力*3 | — | 950mW | 330mW | 170mW | 240mW | 290mW |
| 待機時電力 | ≒0W | ≒0W | ≒0W | ≒0W | ≒0W | ≒0W |
| 電源電圧 | 5V | 5V | 3.3V | 2.5V | 2.5V | 2.5V |
| ゲート数/メモリ容量 | 70k/60kbit | 170k/110kbit | 350k/170kbit | 600k/500kbit | 600k/500kbit | 600k/500kbit |
| チップサイズ | 13mm角 | 15mm角 | 9mm角 | 8.5mm角 | 8.5mm角 | 8.5mm角 |
| パッケージ | 208QFP | 208QFP | 240QFP | 240QFP | 240QFP | 240QFP |
| 設計ルール*5 | 0.8μm2ML | 0.6μm2ML | 0.35μm3ML | 0.25μm4ML | 0.25μm4ML | 0.25μm4ML |

*1) データ長はいずれも12bit, 性能値も1データ12bitあたりの命令実行数. 単位MOPS=Million Operations Per Second.

*2) 40Mを1としたときの相対性能比. 命令の複合化などによる効果を含めている.

*3) 1コアあたりの消費電力. コア内部でpacketを無限周回させた状態での消費電力 (2400Mの場合演算速度120MHzに相当).

*4) TV用DDMP. 値はすべて設計値. 動作速度によっていくつかのモデルを想定.

*5) ML=Metal Layer (何層のメタル配線層を用いているかを表す).

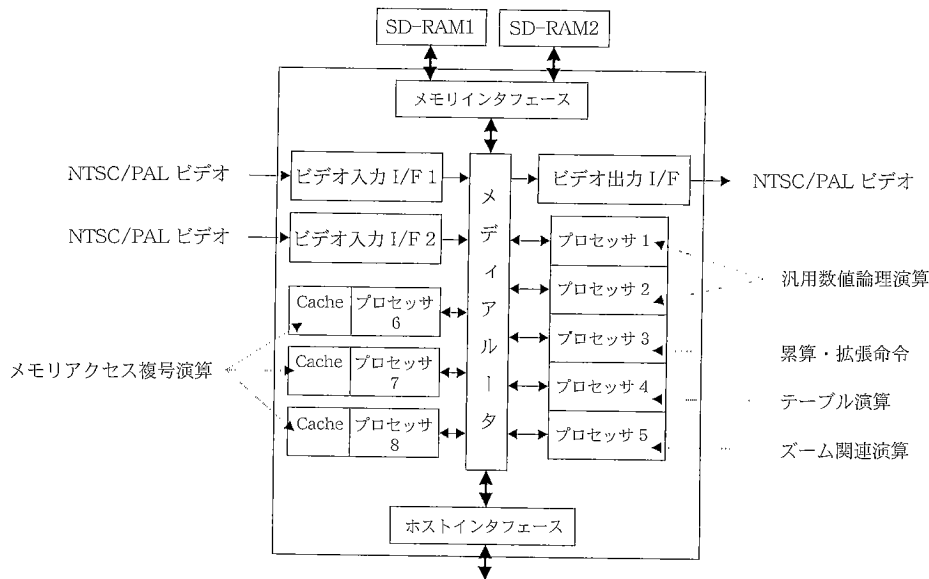


図-8 TV信号処理用DDMP

グラムにストリーム状のデータが流れ込んできた場合に、データそれぞれを識別するためにつけられる。この情報は、ライン間の相関など入力データ間の演算をする際などに使用する。

データ駆動型プロセッサでは、上記のパイプライン上に、待合せ記憶部、プログラム記憶部、合流/分岐部 (=I/O)、演算処理部を持つ。図-5にブロック図を示す。このうち、待合せ記憶部はデータ駆動型特有の機能ブロックである。この部分は、複数の非同期入力の待合せを行い、データの同期をとる。たとえば、引き算 (SUB) への2入力のデータ①と②があったとする。この2つの入力は互いに非同期に入力される。しかし、演算SUBを実行するためには、2つのデータが揃ったことをプロセッサが検出してやる必要がある^{*4}。待合せ記憶部は、2つのデータが揃った時点で、2つのデータを対にして演算処理部に送る。その後、演算 (SUB) が演算処理部で実行される。

4. コアプロセッサの性能

表-1に、これまでに我々が開発してきたDDMPの性能比較を示す。過去のDDMP-40M (1993年)、600M (1994年) に続いて、1997年10月に2400Mと呼ぶLSIを開発した。ブロック図を図-6に示す。このDDMP-2400Mは3層メタル0.35 μ mで作られ、主に画像処理、パケット通信などへの応用を目指している。

表-1で示したピーク性能の算出法は、独自に作成したテストプログラムに準じている⁷⁾。表-1で内部転送

速度とは、パイプラインにパケットを1個だけ投入した際の巡回速度、内部演算速度とあるのは演算部での処理速度である。非同期転送の場合、斉一のクロックで動いていないため、転送速度と演算速度に差が出る。DDMP-2400Mの場合は、クロック120MHz相当がピーク性能であり、パケット最大転送速度のおおよそ80%程度である。図-7に2400Mの1コアプロセッサの動作特性を表すグラフを示す。縦軸は内部演算性能 (単位: MHz)、横軸はパイプライン中に占めるパケット数 (単位: 個) を意味する。120MHzに達した後、パイプラインが溢れるまで一定の性能を保つ⁷⁾。

ここで、DDMPで特徴的なのは、データ入力に休止期間がある場合の電力消費である。TV信号のような処理の場合でも、ブランキングと呼ばれる無信号期間が存在し、その間はデータがこない。クロック同期のシステムではそのような場合でも動き続けるが、DDMPのようなシステムでは動作が停止する。表-1はあくまでもピーク時の消費電力を示している。したがって、応用対象によって大きく消費量を下げることが可能である。たとえば、待機時の電力はTVなどの年間消費電力の5~15%に達するため、これらが削減できる効果は大きい。

5. TV信号処理用DDMP

図-8に当社が開発中のTV信号処理向けDDMP (以下TV用DDMPと略記) を示す。これは、表-1に示すDDMP-7600Mに相当するLSIである。これまでのTV信号処理は、すべてASICで行われてきた。また、アナログ処理も随所で使われており、完全なデジタル化ができていないわけではない。将来的なPCでのTV処理や、デジタルTVへの統合を考えると、全体をデジタル化し、プログラム記述する必要性が出てくる。

^{*4} ノイマン型ではそのような保証はしない。その代わりに、計算の実行前に必要なデータがレジスタ/メモリに書き込まれるように、(ユーザが) プログラムを記述する必要がある。一方、データ駆動型の場合は、並列実行が可能な処理は、記述順序に関係なく演算される。したがって、待合せ記憶のようなハードウェア的な同期機構がある。

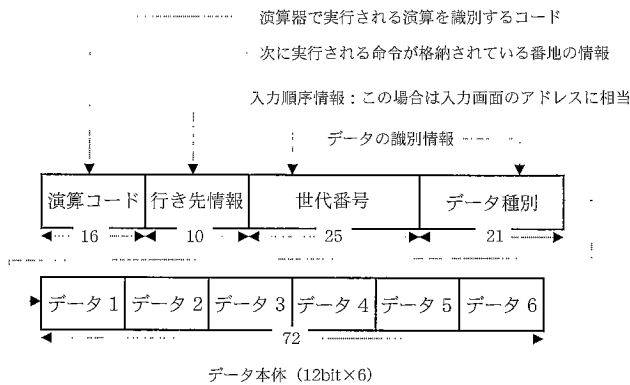


図-9 TV用DDMPのパケット構成図

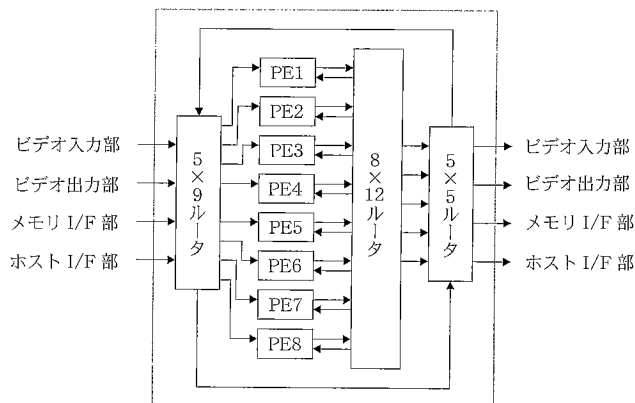


図-10 TV用DDMPのメディアルータ

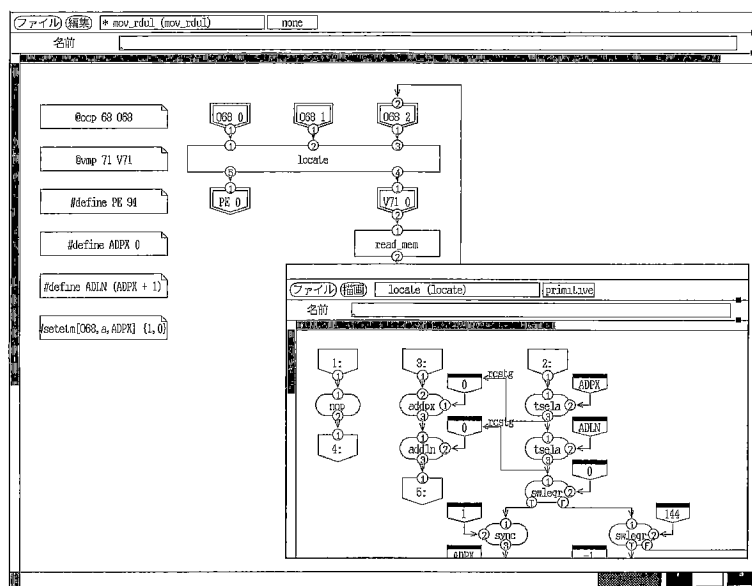


図-11 フローグラフ入力ツールFEDの画面

このDDMPでは、コアプロセッサで機能分散することにより処理の並列実行をする。NTSC/PALのビデオ信号は約14.3/17.7MHzの速度で入力されてくる。入力されたNTSC/PALデータなどは、ビデオ入力インタフェース部で、パケットの形式に再構成される。ここで、RGB各8bitのデータは、それぞれ12bitのデータ領域に収められ、さらに2つのRGB、または6つのY、または3組のCbCrデータが1パケットに畳み込まれる。

図-9にパケットの構成図を示す。ここでは、1つのパケットのタグに対して6つのデータを持たせている。この場合、タグに収められるのは、演算コード、プログラムの行き先情報、アドレス（入力順序）情報である。ここでアドレス情報とは、TV画面上のピクセル（水平方向の位置）、ライン（垂直方向の位置）、フィールド（時間方向の位置）と等価になるように与えられる。この情報を用いて、フィールド間、ライン間、ピクセル間の相関をとる。DDMP内部での処理は、6個のデータが1単位として扱われる。ここでの

6個とは、表-1に示す1コアの演算器数と等価である。これは、ある種のSIMD（Single Instruction stream Multiple Data stream）処理である。TV信号のような同一処理の繰り返しの場合、有効な方法といえる。

図-10にTV用DDMPの内部にあるメディアルータのブロック図を示す。メディアルータは高速のデータ転送を行う一種のクロスバスイッチである。そのため内部の転送速度は実効内部転送速度の2倍となっている。5×9ルータはビデオ入力インタフェース部やメモリインタフェースなど、外部からのデータを、パケット化するために使用する。ビデオ出力インタフェース部からの入力は、プロセッサコアでの処理結果を読み出すために使用する。ここで入力されたデータは、あらかじめプログラムされた行き先情報に従って、各プロセッサコア（PE1～8と表記している）宛に送られる。各プロセッサコア間のパケットの通信は8×12ルータを経由して行われる。処理の終わったパケット、または外部メモリアクセスが必要なパケットは、5×5ルータに送られる。このルータから出力される段階

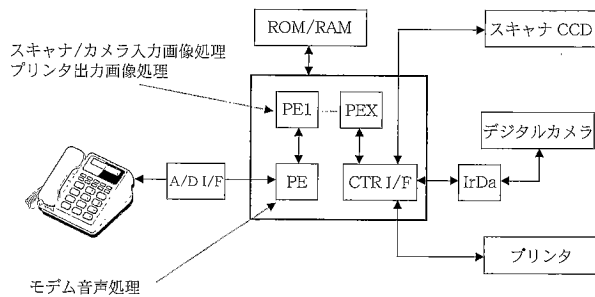


図-12 DDMPのプリンタ/FAX複合機への応用

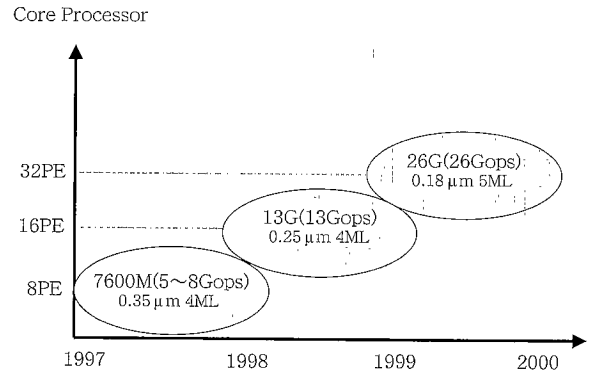


図-14 DDMPのロードマップ

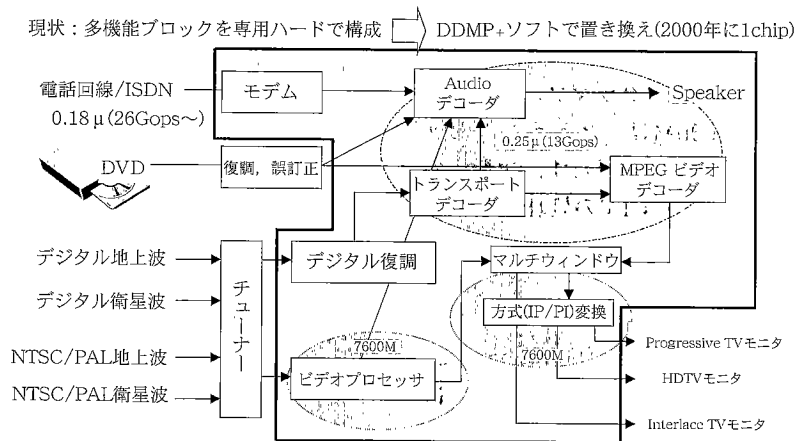


図-13 DDMPのデジタルTVへの応用

で、パケットは元のデータに分解される。ビデオ信号への変換はビデオ出力インタフェース部で行われる。

6. DDMPのプログラム

メディアプロセッサのプログラムは、C言語ベースやCで呼び出せるライブラリで提供しているメーカーもあるが、最適化を行う過程で、アセンブラソースの記述が必要な場面も出てくる。その場合、プログラムの書きやすさが問題となる。当社でも、基本的にライブラリとアセンブラの両者を提供する。DDMPのプログラムは、データフローグラフと呼ばれる、一種の図的な言語で記述することができる。データ駆動型プロセッサの場合、プログラムは記述した順序で動作するのではなく、データの到着する順序で動作する。したがって、データの流が分かりやすい記述法が求められる。データフローグラフはフローチャートや回路図に似た記述をするため、並列な流れが見やすい。並列処理を記述するためのツールとして、FED (Flow-graph EEditor)を開発中である。図-11にFEDの画面を示す。ブロック単位で階層的な記述も可能である。CADと同様にプログラム入力を行うことができる。アセンブラの記述はこのエディタを用いてもよいし、テキストベースで記述してもよい。テキストベースのアセンブラも別途用意している。

評価ボードは、PCのPCIバスに挿入して使用する。データ駆動型のプロセッサとバッファメモリを実装しており、処理結果のオフラインダンプが可能である。PCのWindows上から評価ボードに対して、プログラムのダウンロード、データのダウンロード/アップロードが可能である。ボード自体はTV用ならば、外部TV信号をダイレクトに inputsする端子を持たせる。プログラムの開発では、まずアセンブラで文法チェックをし、次にソフトウェア・シミュレータで論理的な間違いがないかを検証し、最後に評価ボードでリアルタイム検証するというステップを踏む。

7. DDMPの応用例

具体的な応用例として、図-12にプリンタ/FAX複合機への応用を示す。

現在家庭からオフィスまで、スキャナは幅広く用いられている。しかし、これらに要求される処理はかなりの部分を共通化できる。家庭用からオフィスまで、各種のデジタルプリンタ・スキャナ・カメラに対応可能である。対象によって処理の量が異なるだけなので、内蔵するプロセッサ数によって調整する。プログラムは共通化して使える。具体的には、色補正、色空間変換、解像度変換、誤差拡散処理などの画像処理である。現在生まれつつあるカラープリンタ/FAXにも対応

する。

もう1つの応用例として、デジタルTVを図-13に示す。

次世代TVではさまざまな入力考えられる。従来のNTSC/PAL（地上波、衛星）に加え、デジタル（地上波、衛星）放送がある。完全にデジタル化されるまでは、デジタル放送とNTSC/PALは共存する。また、複数の画像サイズが送られてくる。モデム、ISDNといった入力もある。出力はインタレースの従来TVだけでなく、ノンインタレースのプログレッシブCRT（現在はパソコンモニタに使われている）にも出力される。これらすべてが常にオールインワンで必要なわけではないが、内部の部品としてはすべてに対応できなければいけない。商品を開発する側からは、メディアプロセッサのようなプログラマブルな部品が望ましい。TV用DDMP-7600Mは、ビデオプロセッサとマルチウィンドウ処理の機能を果たす。次世代のDDMP-26G（図-13）でデジタルTVの処理の大半を1チップ化する。

8. おわりに

図-13に示したデジタルTV処理に対応するために、DDMPの性能向上を図っていく。1998年末には12GOPS（Giga Operations Per Second）、1999年末には24GOPSの処理が要求されるようになるだろう。コアプロセッサ数を増やすと同時に、処理内容の並列化、効率化を行う必要がある。当社で開発するメディアプロセッサDDMPは汎用のプロセッサではなく、各応用分野ごとに強化した命令セットを持つ。図-14に示したロードマップは、各時期で内蔵できる最大のコアプロセッサ数とピーク性能を指標としている。今後メディア処理で要求される機能はますます高度化する。ASICに代わって、メディアプロセッサが、

情報家電のキーデバイスとなることは間違いない。その場合、このメディアプロセッサは、従来のプロセッサの概念とはまったく違ったリアルタイム処理向けの部品となるだろう。一方、DDMPで生まれた自己同期、非同期処理の概念は、将来的に非同期制御やパケット通信処理といった新しい分野へと展開していく可能性が高い。筆者としても、単なるプロセッサに止まらない幅広い応用を望んでいる。

謝辞： 本プロセッサの開発にあたっては、高知工科大学寺田教授、岩田助教授、筑波大学西川助教授ならびに社内外関係各位から多大なご協力をいただきました。ここに感謝いたします。

参考文献

- 1) 岩田他：自己タイミング・スーパーパイプライン型データ駆動プロセッサ, D-IJ81-D-I, No.2, pp.1-8 (1998).
- 2) Mead, C.他: 超LSIシステム入門, 培風館 (1981).
- 3) Dennis, J. B.: Dataflow Schemes, Project MAC, pp.187-216, M.I.T. (1972).
- 4) Nukiyama, T. et al.: A VLSI Image Pipelined Processor, Proc. ISSCC '84, pp.208-209 (1984).
- 5) Komori, S. et al.: A 50 MFLOPS Superpipelined Data-Driven Microprocessor, Proc.ISSCC, pp.92-93 (1991).
- 6) 大橋他: データ駆動計算機EDDENの中規模並列システムの開発, 情報処理学会第43回全国大会論文集 (6), pp.131-132 (1991).
- 7) 芳田他: 映像信号処理向きデータ駆動型プロセッサ, DSP-95-109, ICD95-158 (1995).

(平成9年12月24日受付)



岡本 俊弥

1977年神戸大学工学部卒業。1979年同工学研究科システム工学専攻修了。同年シャープ（株）入社、1チップマイコン、16bitCPUなどのソフトウェア開発に従事。その後、データ駆動型プロセッサのアーキテクチャ、ソフトウェア開発に携わる。

E-mail: toshi@icg.sharp.co.jp