

《技術報告》

シグナルプロセッサ——実時間ディジタル信号 処理用プログラマブル LSI

丸 田 力 男*

1. はじめに

当初はもっぱらシミュレーションの手段として用いられていたディジタル信号処理技術も、最近ではディジタル LSI 技術の著しい発達によって、信号処理装置の具体的な実現手段として極めて魅力あるものになってきた。特に最近シグナルプロセッサ (SP) あるいはディジタルシグナルプロセッサ (DSP) と呼ばれる高速の信号処理用マイクロプロセッサが出現してきたことによって、信号処理装置実現上の技術的課題がハードウェア的なものからソフトウェア的なものへと移行し、ソフトウェア上の工夫により新しい信号処理機能を実現し得るようになったため、益々この傾向が強まっている。

本稿ではこのようにディジタル信号処理技術の応用面に一大変革を与えつつあるシグナルプロセッサについて、アーキテクチャ上の特徴/汎用マイクロプロセッサとの差異、使用方法、並びに開発動向などについて概観することにした。

2. シグナルプロセッサの目的と所要機能

シグナルプロセッサはアナログ信号を A/D 変換して得られるディジタル数値系列に対し、遅延、論理演算、加減乗除などの代数演算、非線形写像、など所望の信号処理実現のための演算/操作をソフトウェア的に施し、結果を D/A 変換器に通してアナログ信号に戻す。シグナルプロセッサを用いた信号処理装置は、A/D 変換器や D/A 変換器を内部にもつか外部と直接ディジタル的にインターフェースするか、シグナルプロセッサを単独使用するかマルチプロセッサとして用い

るか、など幾つかの実現形態上のバリエーションはあるものの、ハードウェア的にみればその実現方法はほぼ定形化されており、その機能上の特質は内部に組込まれたソフトウェアによって定められることになる。

シグナルプロセッサはこのように従来ハードウェア的に実現されていた機能をソフトウェアによって置き換えようとするものであるから、取扱う信号の帯域によって定まる一定の標本化周期内に所望の機能を実現するために必要とされる全ての演算を実行し得るだけの高速性を備えていなければならない。例えば電話音声帯域 (300~3400Hz) 内の 8 周波を図 1 に示すようなバイカッド (双 2 次) フィルタにより検出することを考えてみても、電話音声信号は 8 kHz で標本化されるから $1/8\text{kHz}=125\mu\text{s}$ の標本周期内に図 1 に示される演算を 8 回繰返し実行しなければならないことになり、それぞれ 32 回の乗算と加算、それに控えめにみても 16 回のメモリアクセスを必要とする。通常のマイクロプロセッサでは 1 回の加算に数 μs 、乗算には数 $10\mu\text{s}$ 以上の時間を要するため、とても上の処理をこなさなければならない。また信号処理装置に要求される機能は一般に上の例より一層複雑であるから、シグナルプロセッサは通常のマイクロプロセッサに比し桁違いの高速信号処理能力を必要とすることがわかる。

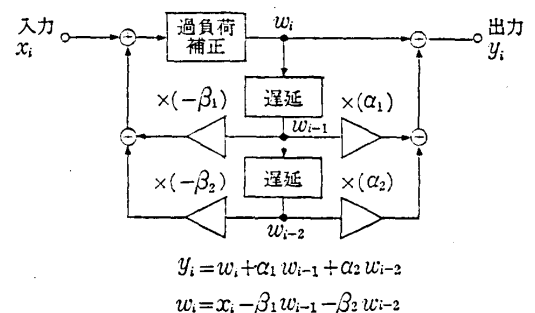


図 1 デジタル信号処理回路の例—バイカッド (双 2 次) デジタルフィルタ

Signal Processors——Programmable LSI Devices for Real Time Digital Signal Processing. By Rikio Maruta (C & C Systems Research Laboratories, NEC Corporation).

*日本電気(株) C & C システム研究所

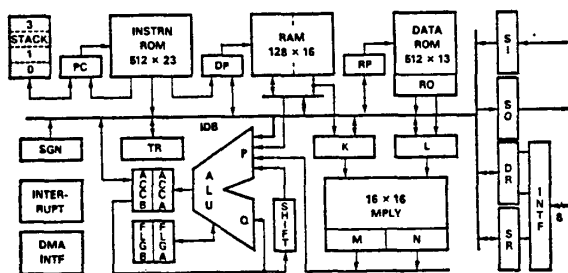


図2 μPD7720 シグナルプロセッサのアーキテクチャ

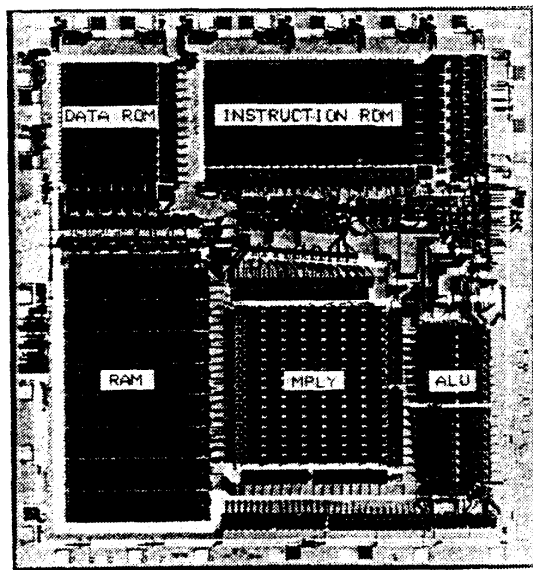


図3 μPD7720 チップ写真

3. シグナルプロセッサのアーキテクチャ

このようにシグナルプロセッサには高速性が要求されるものの、LSIとして実現する上では通常のマイクロプロセッサと同一の技術基盤に依存せざるを得ないから、その高速化はデバイス技術的というよりむしろアーキテクチャ的な工夫によってはじめて可能になるものである。以下、具体例に沿ってどのようなアーキテクチャ上の工夫によりシグナルプロセッサの高速信号処理能力が実現されているかを探ってみよう。

図2は現在市販され最も広く用いられているシグナルプロセッサであるμPD7720 (NEC)¹¹⁻¹³⁾のアーキテクチャを示すブロック図である。μPD7720はこのブロック図に示された全ての機能を図3に示す5.47mm×5.20mmのチップ上に40,000以上の素子を集積化して実現した16ビットのNMOS 1チップマイクロコンピュータであり、250nsの命令サイクルで動作する。

従来の汎用マイクロプロセッサとの対比に於て気づく第1の特徴はチップ面積の相当部分(10%)を割いて実現している16×16ビットの並列乗算器の内蔵で

ある。この乗算器がALUと独立に用意され250nsという高速乗算能力を有することによって、乗算が多用されるデジタル信号処理の実時間動作能力を飛躍的に高めている。

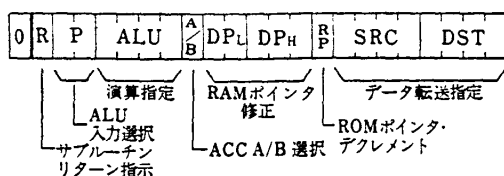
アーキテクチャ上の工夫における第2の特徴は命令用のメモリ及びバスをデータ用のメモリ及びバスから分離したいわゆるハーバード・アーキテクチャの採用である。このようなメモリの分離は命令とデータに対するメモリ容量配分を固定化することにはなるが、データに対する演算/操作の実行を命令のフェッチ/デコードと時間的に併行して行なえるためスループット向上の観点から極めて有効な手法であるといえよう。

第3の特徴は内部のデータ転送を効率化するための工夫である。高速並列乗算器の組込みによって乗算操作そのものが高速化されても、乗数/被乗数の設定や乗算結果の処理を効率よく行なえなければ、全体的なスループットの改善は中途半端なものとなってしまう。そこでμPD7720ではデータ転送上の制約によって演算機能が遊休化することのないように、データメモリのROMとRAMへの2分割、2変数を同時にRAMから読出す特殊構成、複数データの同時転送を可能にするマルチデータバス構成、などの工夫を施している。これによって、例えば乗算を行なう場合に①ROMとRAM、②ROMと他の内部レジスタ、③RAMと他の内部レジスタ、あるいは④RAMの*i*番地と(64+*i*)番地、からの乗数/被乗数同時設定が可能な他、乗算器で乗算を行ないながらALUで乗算結果を累積する操作を連続的に実行することもできる。

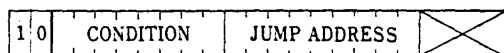
第4の特徴は水平形マイクロ命令の採用によるチップ内の多くのハードウェア機能の同時並行制御である。命令語長を長くすれば同時に多くのハードウェア機能を制御できるがその分メモリサイズが大きくなるので、所要メモリ容量と実現機能のバランスを考えた上で命令形式を定める必要がある。μPD7720では図4に示すように1命令を23ビットとし、OP(演算)命令、JMP(分岐)命令およびLDI(データ設定)命令の3種の命令形式が用意されている。この内OP命令では、①ALU演算、②データメモリ(RAM/ROM)のアドレス修正、③内部レジスタ間のデータ転送、④サブルーチンからの復帰、の4種を直接指定できる他、⑤乗算の実行、も自動的に行なわれるので、250nsの命令サイクルの間に全部で5機能を同時に実行し得るなど極めて効率化されている。

特徴の第5はパイプライン構造の採用である。コンボリューション計算などデジタル信号処理で多用さ

(a) OP 命令



(b) JMP 命令



(c) LDI 命令

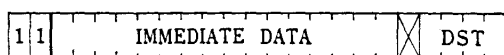
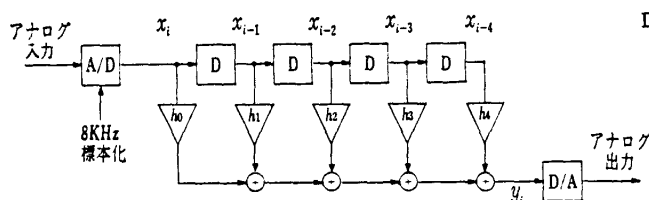
図4 μ PD7720 の命令フォーマット

図5 5タップ FIR (非巡回形) デジタルフィルタ

れる多項積和計算における「乗数／被乗数設定」→「乗算実行」→「乗算結果の累積」という一連の操作をパイプライン処理で実行することによって、250nsの1命令サイクルであたかもこれら全ての操作を実行したと同様な効果が得られている。

以上述べたようなアーキテクチャ上の工夫によってシグナルプロセッサがほぼ同程度のクロック速度で動作する汎用マイクロプロセッサに比し格段の高速信号処理能力を実現し得ている訳である。

4. シグナルプロセッサの応用手順

それでは実際にシグナルプロセッサを応用するにはどうすればよいか、図5の例について具体的に考えてみよう。図5は8kHzで標本化されA/D変換された信号を5タップFIR(非巡回形)フィルタに通した後D/A変換して再びアナログ領域に戻すものである。

まず検討すべきことは要求される演算処理がシグナルプロセッサの能力内にあるか否かである。能力内であれば1チップで実現できるが、能力を上廻る場合にはマルチチップで機能分担させたり、汎用マイクロプロセッサとの組合せを考えたりすることが必要になる。 μ PD7720は命令サイクルが250nsであるから $1/8\text{kHz}=125\mu\text{s}$ の標本周期内に500命令の実行が可能であるのに対し、図5のFIRフィルタで必要とされる乗算回数は5回、加算回数は4回にすぎないから、

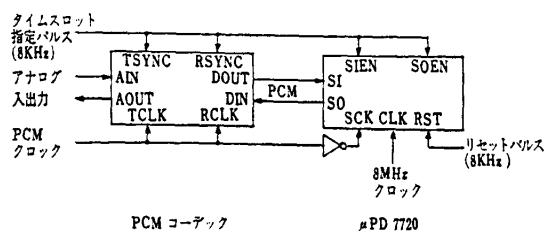


図6 PCMコーデックとの組合せによるハードウェア構成

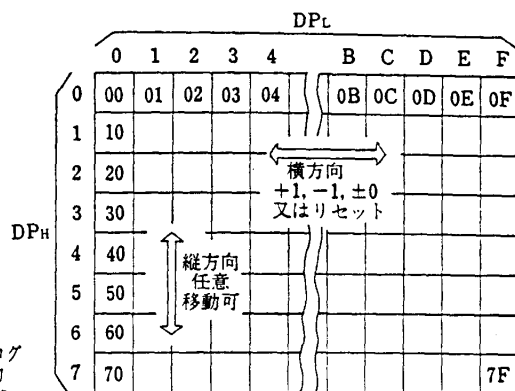


図7 データ RAM のアドレス配置と OP 命令内アドレス修正フィールドでの指定

その他メモリの書込／読出操作などに要する命令ステップを考慮しても十分命令サイクル数には余裕がある。また遅延を実現する RAM、係数を格納する ROM、などの容量も十分である。したがって図5の機能は1チップで実現できるであろうことは容易に判断できる。しかし実現すべき機能が複雑になってくると、この判断のためにはある程度プログラムの検討を進めてみるが必要になってくる。

1チップで実現できるとなればハードウェアの構成は簡単である。A/D 及び D/A に1チップ PCM コーデックを用いることにすると、必要なハードウェア構成は図6に示すようなものとなる。

次のステップはプログラムの検討である。まず演算処理の手順を考えながらデータ RAM とデータ ROM の割付けを考えることが必要になる。 μ PD7720 のデータ RAM は128語であり、7ビットのアドレスポインタ (DP) によりアドレス指定されている。図7はこの RAM のアドレス配置を DP の上位3ビット (DPH) と下位4ビット (DPL) の2次元構成で示したものであるが、OP (演算) 命令中のアドレス修正フィールドで DPH に対しては任意の変更、DPL については +1, -1, ± 0 またはリセットの4種の変更を直接指定できるようになっている。この機能が十分活用できるように RAM への変数割付けを行なうのが能率的なプログラム作成のコツである。一方データ ROM は512語で

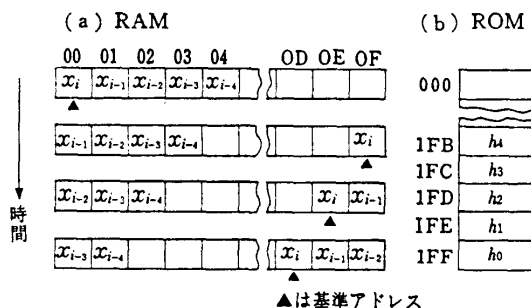


図8 5タップFIRフィルタ実現のためのRAM/ROM割付け例

表1 図5のFIRフィルタの実現プログラム

命令 ステップ	命令	命令
0	LDI	@RP, 1FFH ;
1	LDI	@A, 000FH ;
2	OP AND MOV	ACCA, IDB @NON, TR; ;
3	OP MOV	@DP, A ;
4	OP MOV DPDEC	@A, SIM ; ;
5	OP MOV	@TR, DP ;
6	OP XOR MOV	ACCA, IDB @MEM, A ;
7	OP MOV DPINC RPDEC	@KLR, MEM ; ;
8	OP ADD MOV DPINC RPDEC	ACCA, M @KLR, MEM ; ;
9	OP ADD MOV DPINC RPDEC	ACCA, M @KLR, MEM ; ;
10	OP ADD MOV DPINC RPDEC	ACCA, M @KLR, MEM ; ;
11	OP ADD MOV	ACCA, M @KLR, MEM; ;
12	OP ADD	ACCA, M ;
13	OP MOV	@SOM, A ;
14	WAIT: JMP	WAIT ;

あり9ビットのアドレスポインタ(RP)によりアドレス指定されるが、OP(演算)命令中ではRPに対しデCREMENT(-1)の指定のみ行なえる。

図8は図5のFIRフィルタに対するRAM/ROMの割付け例を示したものである。RAMは16進表示で0X_H(X=0~F)のアドレス部分に図5のデータ $x_i, x_{i-1}, \dots, x_{i-4}$ を割付けている。図5によればある標本時点における入力 x_i は次の標本時点では遅延Dを通ることによって x_{i-1} となる訳であるが、この遅延動作を実際にRAM内のアドレス位置間のデータ移動によらずとも、アドレスの基準位置を標本時点毎に順次ずらしていくことによっても実現できる訳で、図8(a)はその方法を図解している。一方ROMのアドレス割付けはRPのデCREMENT機能を考慮してアドレス1FF_Hから1FB_Hにフィルタ係数 $h_0 \sim h_4$ を格納するようにしている。

このようなメモリの割付けがすめば、あとは図5の実現に必要な演算を図2のブロック図上でどのようなデータ移動により実現するか考えながら適切な命令の流れを決めて行けばよい。表1はこのようにして得られた図5のFIRフィルタ実現のためのソースプログラムリストである。

ステップ0ではLDI命令によりROMポインタRPの内容を1FF_Hにセットし、ROMよりフィルタ係数 h_0 が得られるようにしている。ステップ1~3はTRレジスタ内にある前標本時点の基準アドレス番号をDPレジスタにセットする命令で、DP_Hを0にするためまずステップ1でLDI命令によりACCAに000F_Hをセットし、ステップ2でOP命令によりACCAでTRレジスタからの出力と000F_HとのANDをとり、ステップ3でその結果をDPに移している。次にステップ4ではSIレジスタにあるA/D変換器からの入力 x_i をACCAに取込むと同時に、現標本時点の基準アドレスを得るためDP_Lを-1する。ステップ5ではその基準アドレスを次標本時点迄保存するためTRレジスタに移す。ステップ6ではACCA内の入力データ x_i をDPで指定されたRAMアドレスに格納しACCAの内容を0にする。ステップ7ではRAMから x_i をKレジスタにROMから h_0 をLレジスタに移した後DP_Lを+1、RPを-1する。ステップ8ではMレジスタに得られている乗算結果 $h_0 \cdot x_i$ をACCAに移すと同時にRAMから x_{i-1} をKに、ROMから h_1 をLに移し、その後DP_Lを+1、RPを-1する。ステップ9ではM内に得られた $h_1 \cdot x_{i-1}$ をACCA内の $h_0 \cdot x_i$ に加算し

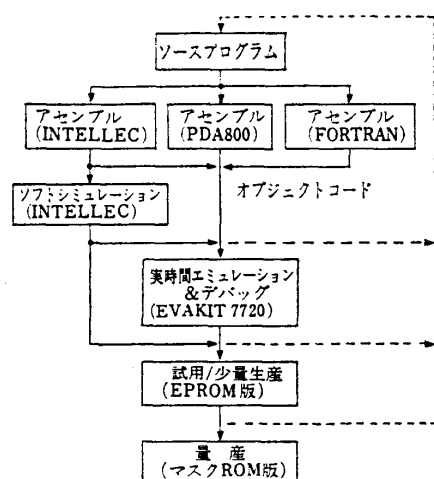


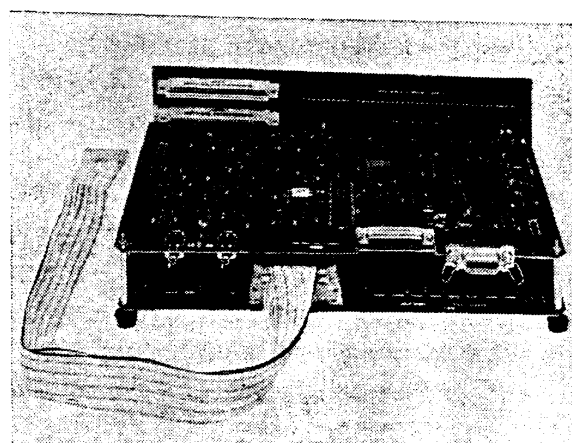
図9 ソフトウェア開発ツールとその使用手順

ACCA の内容を $(h_0 \cdot x_i + h_1 \cdot x_{i-1})$ に更新すると同時に RAM から x_{i-2} を K に ROM から h_2 を L に移し、その後 DP_L を +1, RP を -1 する。

このような操作を続けるとステップ12で ACCA に最終出力 $y_i = h_0 \cdot x_i + h_1 \cdot x_{i-1} + \dots + h_4 \cdot x_{i-4}$ が得られるから、ステップ13ではこの結果を SO レジスタに出力する。以上で1標本分の処理を終了するので、ステップ14の JMP 命令で次の標本時点迄待機を続け、次の標本時点で外部からリセットパルスが加わると再びステップ0から命令の実行を再開する。

結局図5の機能の実現に要す命令サイクル数は15であるから最初の見通しどおり1チップで十分実現できることが確認される。また FIR フィルタのタップ数が +1 されても命令数が1増えるだけであることも表1の例から明らかであり、より大きな FIR フィルタの実現も同様な方法で可能である。なお上の例では A/D, D/A 変換器とも線形と仮定したが、PCM コーデックのように非線形符号を用いたものであれば、プログラムの最初に非線形/線形変換、最後に線形/非線形変換のための命令ステップを追加することが必要になる。

このようにしてソースプログラムが作成された後は、アセンブラによりオブジェクトコードへの変換が行われ、その結果がシグナルプロセッサ内の命令 ROM に書込まれて実際に使用されることになる。しかしながらソフトウェアの開発過程では一般に誤りはつきものであるし、特にシグナルプロセッサのソフトウェア開発ではプログラム自体の正しさ以外に実現アルゴリズムが所望の特性を満たし得るものか否かの検証も必要とされるので、これを支援するに十分な開発ツールを用意しなければならない。図9は $\mu PD 7720$

図10 エバキット 7720 ($\mu PD 7720$ 用実時間エミュレータ)

のソフトウェア開発用ツールとその使用手順を示したものである。ソフトウェア開発を進める上で実時間ハードウェアエミュレータの存在は極めて効果的である。 $\mu PD 7720$ にはエバキットと呼ばれる実時間エミュレータ(図10)が用意されており、これをホストコンピュータ(パソコンなど)とターゲットシステムの間にごく置くことによってソフトウェア開発作業を能率的に進めることができる。すなわち、ホストコンピュータでソースプログラムのキーインとアセンブルが行なわれると、オブジェクトコードがエバキットに転送されターゲットシステム上で実時間動作に入る。実時間動作で所望の特性が得られない場合には、エバキットでのブレークポイント設定、実行過程トレース、メモリダンプなどをホストから指示しながらデバッグ作業を進めることができる。デバッグ終了後のオブジェクトコードは、エバキット内で EPROM 版の $\mu PD 7720$ に書込むことができる。 $\mu PD 7720$ には EPROM 版があるため、研究開発や少量生産機種への適用が手軽にできる他、大量生産機種のための試験使用に於てもマスク ROM 版と全く同一の使用環境の下で評価できるので非常に便利である。

5. 各種のシグナルプロセッサと発展動向

シグナルプロセッサとしての最初の製品は1979年2月発表⁴⁾のインテル 2920 であり、次が同年8月発表⁵⁾の AMI S2811 である。インテル 2920 は最初の LSI シグナルプロセッサであるだけでなく、A/D および D/A 変換器をオンチップでもちアナログ信号のディジタル処理を単一 LSI で可能にするという画期的な特徴をもつ点では高く評価されるものの、A/D と D/A の精度が直線9ビットに過ぎないこと、高速乗算

器を内蔵していないこと、内蔵メモリ容量が小さいことなど全体的な演算処理能力の点で実用域に達しているとは言いがたい。また AMI の S2811 も乗算器精度の不足、デバイス量産技術上の問題などから、実用に至らなかった。

このような事情から、最初の本格的なシグナルプロセッサとして広範な応用が図られ現在でも主力となっているのは1980年の ISSCC で同時に発表された NEC μ PD7720 と Bell DSP^{6),7)}である。これら1979/80年発表のシグナルプロセッサを第1世代第1群とすると、それら先行モデルを基に1982年以後次々と発表^{8)~12)}されてきつつあるシグナルプロセッサは第1世代第2群と分類し得よう。それらの演算機能、性能、適用可能領域をみたときに何がしかの改良は当然見受けられるものの、第2世代として明確に分類できる程画期的な飛躍は未だあらわれていない。このへんは同じ LSI でも2年一昔の速さで世代交代するメモリと大きく異なるところである。

とはいえ表2の仕様比較からも明らかなように、第2群には新しい技術の潮流や将来の発展動向を示唆す

るものも少なくない。CMOS 技術による低消費電力化は第2世代へ向けての明らかな潮流であるし、浮動小数点演算方式によるダイナミックレンジ拡大/マニュアルスケーリングの不要化も今後の発展方向を示すものであろう。 μ PD7720 で採用された RAM の2語同時読出機能は NEC μ PD 3384、富士通 DSP、東芝 DSP に於ける2 RAM 構成へと発展したが、他にもより使用上の柔軟性をもつ2ポート RAM 構成をもつものもあり、今後の展開に期待がもてる。 μ PD7720 が経済的なスタンドアロンのシグナルプロセッサを目標に小チップ/小パッケージを追求し外部メモリ拡張や外部プログラムメモリ機能をもたない構成をとったが、その後のシグナルプロセッサには大形パッケージで外部拡張性をもつものが多い。しかしこれはいずれも必要であって最近の東芝 DSP のように大小2種類のパッケージをもつファミリー構成が展開される可能性もある。また今後はインテル2920のように A/D、D/A 機能を内蔵したアナログシグナルプロセッサも LSI 技術の一層の発展を背景に再登場してくることも考えられよう。

表2 各種シグナルプロセッサの仕様一覧

発表時期	1980年		1982年			1983年	
品名	NEC μ PD 7720	Bell DSP	TI TMS 320	NEC μ RD 3384	日立 HSP	富士通 FDSP III	東芝 DSP 1A/1B
デバイス技術	3 μ NMOS	5 μ NMOS	3 μ NMOS	3 μ NMOS	3 μ CMOS	2.3 μ CMOS	2 μ CMOS
チップサイズ	28.40mm ²	68.50	43.81	58.78	49.26	91.20	49.4/50.9
パッケージピン数	28ピン	40	40	64	40	88	64/28
電源	+5V/0.9W	+5V/1.25W	+5V/0.95W	+5V/1W	+5V/0.2W	+5V/0.29W	+5V/0.2W
EPROM版	有	無	無	無	無	無	無
命令サイクル時間	250ns	800ns	200ns	300ns	250ns	Typ. 100ns	250ns
データ語長	16ビット (倍精度可)	20 (Acc: 40bit)	16 (Acc: 32bit)	16	12+4 浮動 (Acc: 16+4)	16 (Acc: 26bit)	16 (Acc: 18bit)
乗算器構成	16 $\times$ 16 $\rightarrow$ 31	20 $\times$ 16 $\rightarrow$ 36	16 $\times$ 16 $\rightarrow$ 32	16 $\times$ 16 $\rightarrow$ 16	12 $\times$ 12 $\rightarrow$ 16 (仮数部)	16 $\times$ 16 $\rightarrow$ 26	16 $\times$ 16 $\rightarrow$ 32
乗数/被乗数同時設定	可	不可	不可	可	可	可	可
プログラム ROM	512 $\times$ 23bit	1K $\times$ 16 (外付 1K $\times$ 16)	1.5K $\times$ 16 (+2.5K 外付可)	512 $\times$ 32	512 $\times$ 22	1K $\times$ 24 (外付 1K $\times$ 24)	外部/512 $\times$ 16
データ ROM	512 $\times$ 13bit	—	—	512 $\times$ 16	128 $\times$ 16	—	外部/512 $\times$ 16
RAM	128 $\times$ 16bit	128 $\times$ 20	144 $\times$ 16	128 $\times$ 16 2組	200 $\times$ 16	(128 $\times$ 16) 2組 (外付 1K $\times$ 16)	(64 $\times$ 16) 2組
外部インタフェース	8bit 並列 I/O+直列 I & O	16bit 並列 I/O+直列 I & O	16bit 並列 I/O	16bit 並列 I/O +直列 I & O with FIFO	16bit 並列 I/O+直列 I & O	16bit 並列 I/O (アドレス機能付)	8bit 並列 I/O+直列 I & O

6. おわりに

プログラムによる汎用性と音声帯域信号などの実時間処理に特徴をもつシグナルプロセッサについて、そのアーキテクチャ的な特質／汎用マイクロプロセッサとの差異、応用の具体的手順、並びに開発動向を述べた。シグナルプロセッサの活用は信号処理装置の製品化のみならず、信号処理の新しいアイデアの実時間シミュレーションによる検証など研究開発の推進にも大きなインパクトを与えるものである。本解説がこれからシグナルプロセッサを活用しようと考えておられる諸兄に何らかの参考になれば幸いである。

参 考 文 献

- 1) Y. Kawakami et al.: A single chip digital signal processor for voiceband applications, ISSCC '80 Dig. Tech. Papers, 40/41 (1980)
- 2) T. Nishitani et al.: LSI signal processor development for communication equipment, ICASSP '80 Conf. Rec., 386/389 (1980)
- 3) T. Nishitani et al.: A single-chip digital signal processor for telecommunication applications, IEEE J. Solid-State Circuits, SC-16, No. 4, 372/376 (1981)
- 4) M.E. Hoff and M. Townsend: An analog input/output microprocessor for signal processing, ISSCC '79 Dig. Tech. Papers, 178/179 (1979)
- 5) R.W. Blasco: V-MOS chip joins microprocessor to handle signals in real time, Electronics, 131/138 (1979)
- 6) J.R. Boddie et al.: A digital signal processor for telecommunication applications, ISSCC '80 Dig. Tech. Papers, 44/45 (1980)
- 7) J.R. Boddie et al.: Digital signal processor, Bell Syst. Tech. J., 60, 7, Part 2 (1981)
- 8) S.S. Magar et al.: A microcomputer with digital processing capability, ISSCC '82 Dig. Tech. Papers, 32/33 (1982)
- 9) M. Yano et al.: An LSI digital signal processor, ICASSP '82 Conf. Rec., 1073/1076 (1982)
- 10) Y. Hagiwara et al.: A single chip digital signal processor and its application to real-time speech analysis, IEEE Trans, ASSP, ASSP-31, No. 1, 339/346 (1983)
- 11) H. Kikuchi et al.: A 23K gate CMOS DSP with 100ns multiplication, ISSCC '83 Dig. Tech. Papers, 128/129 (1983)
- 12) 白川, 菅居: デジタル信号処理用マイクロプロセッサとその高速モデムへの応用, 電気学会研究会資料 IP 83-42