

組み込み用途マイコンに関する高速演算器接続方式

森河 徹 檜垣 信生 宮地 信哉^{*1} 西道 佳人^{*2} 春名 修介
松下電器産業(株)マルチメディア開発センター
松下電子工業(株)半導体事業本部LSI開発センター^{*1}
松下電器産業(株)半導体研究センター^{*2}

近年、マイコンの高速化が進み、またマイコンコアに演算強化のために専用演算器を内蔵することにより、高性能・低価格なマルチメディアマイコンが実現しつつある。しかし、専用演算器を内蔵してマイコンを作成することは、マイコンの品種展開毎に回路を設計し直さなければならず、コスト高の原因となっている。本稿では専用演算器を内蔵する場合と同じ性能を維持しつつ、品種展開に応じてこの専用演算器の接続・切り離しを行うことができる高速演算器接続方式を考案し、応用アプリケーションを用いて評価を行った。

A METHOD OF CONNECTING HIGH-PERFORMANCE AND HIGH-FUNCTION EXECUTION UNIT FOR ENBEDDED MICROCONTROLLERS

Toru Morikawa, Nobuo Higaki, Shinya Miyaji ^{*1}, Yoshihito Nishimichi, Shuusuke Haruna
Matsushita Electric Industrial CO.,LTD.
Matsushita Electronics Corporation ^{*1}

Recently, it is possible that high-performance and low-cost microcontrollers become to execute many multimedia applications which microcontrollers can't quickly execute until now, because recent microcontrollers have extremely high performance, and also have high-performance and high-function embedded execution units for multimedia applications such as a multiplier. However, such embedded execution units cause a increase of cost, because all consumers doesn't need such units for their applications, and these units are useless for them.

This paper shows a new method of interface between microcontroller and high-performance and high-function execution unit, which can realize to easily connect and separate these units corresponding to every applications. And the evaluation by using some multimedia applications also are indicated.

1. はじめに

従来マイコン応用は、家電機器を中心とした制御処理を行うものが多かった。画像、音声データ等を扱うマルチメディア応用のように高速に処理することが必要とされる応用は、マイコンの性能では処理できなかった。しかし半導体プロセス技術の向上によるVLSIの微細化・高集積化、計 算 機 ア ー キ テ ク チ ャ の 技 術 向 上 が 進 む こ と に よ り、マイコンの高速動作が実現できるようになった。

また、例えば乗算のようにマイコンで処理する場合多サイクル処理となり、マイコンの処理能力を越える性能が必要とする場合や、例えばビット処理のようにマイコンで処理するには大幅な計算量の増大を引き起こす場合が存在する。このような場合、乗算器やビット処理演算機等の専用演算

器をマイコンコアに内蔵し、専用演算器への命令をマイコンコアの命令セットに用意し、これにより制御することにより、専用演算器の1サイクルを実現し、従来マイコンが苦手とされてきた処理を高速に実行することが可能となった。

しかし、上記のようにマイコンコアに専用演算器を内蔵してマイコンを作成することは、マイコンの品種展開毎に全ての回路を設計し直さなければならず、コスト高の原因にもなる。

本稿では専用演算器を内蔵する場合と同じ性能を維持しつつ、品種展開に応じてこの専用演算器の接続・切り離しを行うことができる高速演算器接続方式を考案した。実際の応用アプリケーションを用いて詳細に評価を行う。

まず2章で演算拡張について述べ、3章で今回

新たに考案した専用演算器接続構成について述べ、4章では実際のマルチメディア応用アプリケーションを用いた評価について述べる。

2、演算拡張

マルチメディアマイコンの実現のためには、画像、音声等を高速に処理する必要がある。この処理には乗算、積和、除算、ビット操作等の処理が含まれ、これらの処理は従来のマイコンでは高速に処理することはできない。そこで演算拡張として、これらの演算を高速に行う専用演算器が用いられる。ここでは、この専用演算器とマイコンとの従来の接続方法について説明する。

2.1 従来の専用演算器接続方式

乗算、積和、除算、ビット操作等の処理を高速に実行することは、従来マイコンの不得意なものもある。例えば乗算のようにマイコンで処理する場合多サイクル処理となり、マイコンの処理能力を越える性能が必要とする場合や、例えばビット処理のようにマイコンで処理するには大幅な計算量の増大を引き起こす場合が存在する。そこで、マイコンコア内の内部バスに専用演算器を結合する専用演算器内蔵マイコンが作られるようになってきた。これは、マイコンが持つ記憶手段と専用演算器とを内部バスによって接続し、専用演算器への命令をマイコンコアの命令セットに用意しこれにより制御することにより、専用演算器へのアクセスが高速に行うことができる。この専用演算器の実行サイクルは1サイクルから多サイクルのものまで様々である。

しかし、マイコンの使用状況に応じて自由に接続・切り離しを行うことが不可能であるため、品種展開時には、各品種毎にマイコンを作り替える必要があるかもしれない回路がマイコン内に残ってしまう欠点がある。

3、専用演算器接続構成

3.1 拡張性とその問題点

2章で述べたように、マイコンが専用演算器を持つことは有用であるが、その拡張性のため、データバス部で接続・切り離しを可能とする構成がある。図.1にその構成図を示す。このモデルは、インターフェイスがコアと拡張演算器のデータ転送用バスにバッファ、ラッチを設け、それに

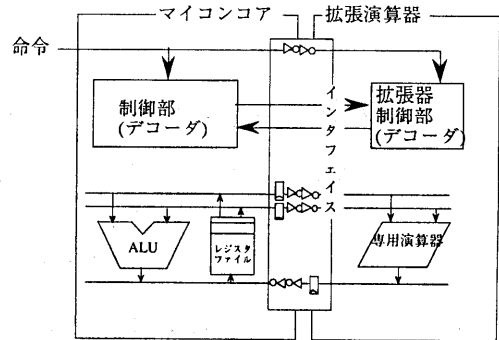


図.1 拡張演算器を切り離し可能にしたマイコンの構成

加えてマイコンコアのパイプライン制御を拡張演算器に伝える信号、拡張演算器の演算実行状態を伝える信号を持つ。このように接続部分にインターフェイスを設けることにより、マイコンコアと専用演算器を持つ拡張演算器を接続・切り離しを可能にすることができる。しかしこの接続インターフェイスは、データ転送時にラッチとバッファの回路を通ることにより、そのための遅延が発生する。図.2に拡張演算器とインターフェイスを用いたパイプライン図を示し、以下ではこれを用いて動作を説明する。ここで例として、ここでは拡張演算器において1サイクルで実行される演算を挙げる。拡張演算器はマイコンコアのパイプライン制御信号によってマイコンコアと同期して動作する。マイコンコア及び拡張演算器へ拡張演算器の実行命令が入力されると、時刻1において、マイコンコア及び拡張演算器のデコーダで同時に解釈されその結果、マイコンコアでは拡張演算器命令であることを解釈し、拡張演算器では拡張演算の内容を解釈する。時刻2において、マイ

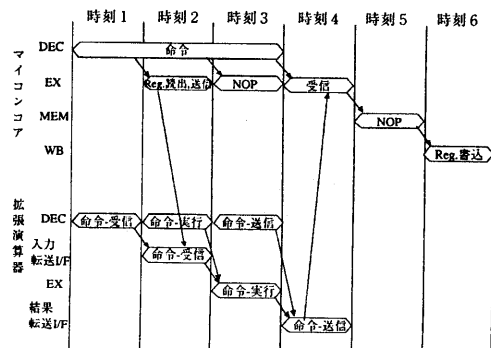


図.2 拡張演算器接続時のパイプライン

コンコアは、レジスタから読み出した演算入力データを拡張演算器に送信する。一方、拡張演算器は、演算入力データを受信する。時刻3において、マイコンコアは拡張演算器より命令終了信号が来ないので次のサイクルまで何も実行しない。一方、拡張演算器は、拡張演算器を用いて演算を実行する。このとき拡張演算器のデコーダは次のサイクルで演算結果をマイコンコアへ送信することを解読するので、命令終了信号を出力する。時刻4においても、マイコンコアは命令終了信号を受け付けることにより、拡張演算器の演算結果を受信する。一方、拡張演算器は、演算結果データを送信するし命令を終了する。時刻5において、マイコンコアは演算結果データをレジスタに書き込む。しかしインターフェイスによるデータ転送の遅延のため拡張演算器命令実行において2サイクルのペナルティが発生する。また、2サイクル以上の実行サイクルを要する拡張演算器も多いので、このパイプラインでは高速に実行できない。

3.2 拡張演算器接続方式

図.2のパイプライン図において、拡張演算器の入力、結果の転送は別の時刻に行われている。しかし、図.1の構成に示すように、拡張演算器の入力、結果の転送は同時に行うことができる。また、拡張演算器を用いる演算は繰り返し演算が多く、拡張演算の結果は次拡張演算実行にあればよい。図.3に繰り返し演算のプログラム例を示す。このプログラムでは、ループ中に置かれたデータの送受信を同時に行う拡張演算器命令を実行する。その後8命令がマイコンコアで実行されると同時に、拡張演算器にて拡張演算器を用いて演算が行われ、その結果を拡張演算器にて保持される。ループが継続され、再び拡張演算器命令を実行し、マイコンコアから拡張演算器へ演算入力データが転送され、同時に拡張演算器よりマイコンコアへ前回の演算結果が転送される。マイコンコアはこの演算結果を保持するように動作する。このループ処理を繰り返し、拡張演算器命令を実行する。この時マイコンコアの拡張演算器命令はデータの送受信サイクルの1サイクルで終了する命令として扱うことができるので、その結果本拡張演算器接続方式を導入することによる遅延や拡張演算器の実行サイクルが1サイクル実行を行っているように見える。

```

CLR  D0
CLR  D1
MUL  D1,D0
loop;
MOV  (A0),D1
MOV  (0x2,A0),D0
★ MUL  D1,D0
INC  D3
INC  A1
MOV  D0,(A2)
INC  A2
CMP  #0x60,D3
BLT  loop

MUL  D1,D0
MOV  D0,(A2)

```

★印は拡張演算器命令

図.3 繰り返し演算のプログラム例

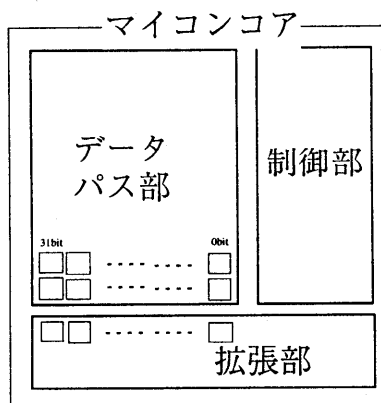


図.4 拡張演算器接続方式の式チップレイアウト

また、本拡張演算器接続方式は、図.4のようにVLSIのレイアウト設計において、マイコンコアと拡張演算器のデータパス部のビットスライスが等しい。よって、レイアウトが容易に設計できるように配線が短く設計できるので配線遅延を少なくできる。

図.5にパイプライン図を示し、以下ではこれを用いて動作を説明する。ここで例として、斜線部分のステージが拡張演算器命令を示し、これは拡張演算器において3サイクルで実行される演算を挙げる。拡張演算器はマイコンコアのパイプライン制御信号によってマイコンコアと同期して動作する。マイコンコア及び拡張演算器へ拡張演算器命令が入力されると、時刻1において、マイコンコア及び拡張演算器のデコーダで同時に解読されその結果、マイコンコアでは拡張演算器命令であることを解読し、拡張演算器では拡張演算の内容

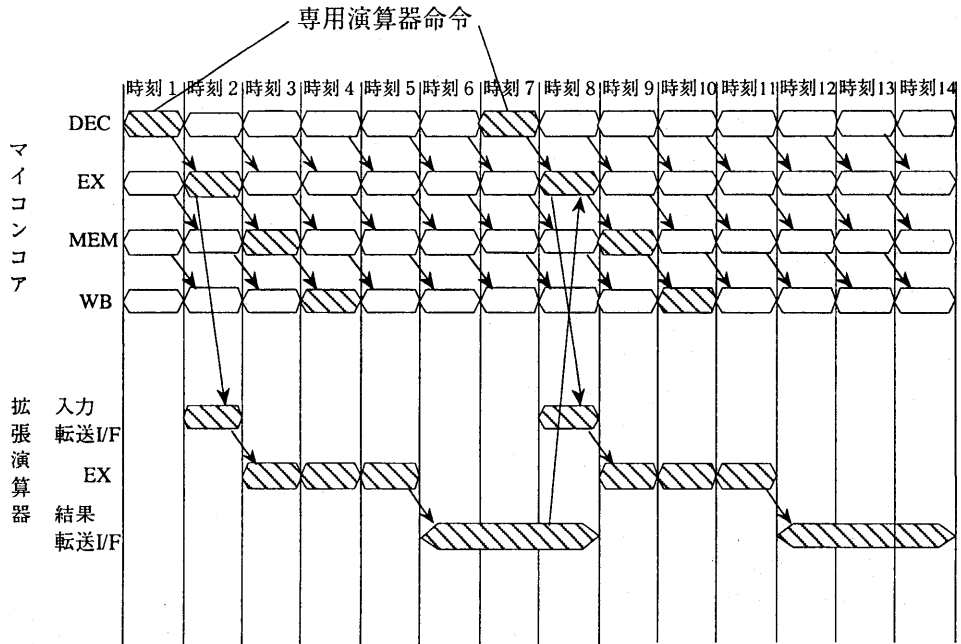


図.5 本方式のパイプライン

を解説する。時刻2において、マイコンコアは、演算入力データを送信すると同時に、演算結果データを受信する動作を行い、命令を終了する。一方、拡張演算器は、演算入力データを受信すると同時に、前回の演算結果データを送信する。時刻3～5において、マイコンコアは拡張演算器命令のメモリアクセスステージ(MEM)、レジスタ書込ステージ(WB)を含むマイコンコアで実行する命令を実行し、拡張演算器は、拡張演算器を用いて演算を実行する。時刻6において、拡張演算器は、演算を終了させて結果を出力可能な状態となると同時に演算終了を信号線を通じてマイコンコアに伝達する。マイコンコアは、拡張演算器の演算終了信号を受け取るまで次の拡張演算器命令を実行できないので、もし拡張演算器の演算終了前に拡張演算器命令が解説された場合、マイコンコアはパイプライン処理を停止させて拡張演算器の演算終了を待つ。図.5の場合、拡張演算器側のデコーダは時刻4まで動作するため、次の拡張演算器命令は時刻5まで受け付けることができない。つまり、この拡張演算器命令を動作する場合、次

の拡張演算器命令まで4サイクル以上の間隔をあげる必要がある。そして、時刻7において再びマイコンコア及び拡張演算器のデコーダで同時に解説されその結果、マイコンコアでは拡張演算器命令であることを解説し、拡張演算器では拡張演算の内容を解説する。時刻8において、マイコンコアは、演算入力データを送信すると同時に、演算結果データを受信する動作を行い、命令を終了する。一方、拡張演算器は、演算入力データを受信すると同時に、前回の演算結果データを送信し、時刻9においてマイコンコアのレジスタに演算結果が書き込まれる。

このように拡張演算器を用いる演算は繰り返し演算が多いことを利用し、演算結果を転送を次の拡張命令の入力データ転送時に同時に行うことで、マイコンコアのパイプライン処理のthroughputを1サイクルにできる。

4、評価

4.1 評価モデル

図.6に今回評価も用いたモデルを示す。モデル

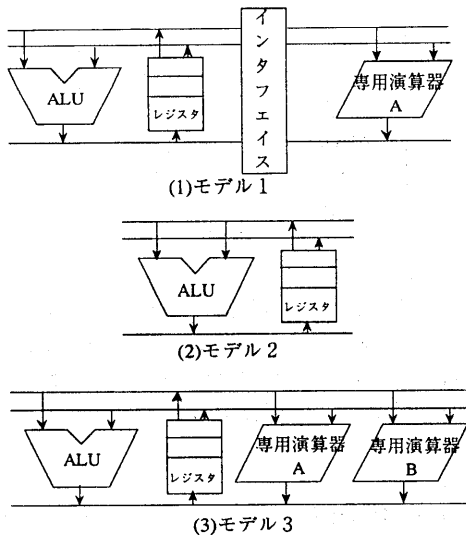


図.6 評価モデル

1はマイコンコアと専用演算器1個とその間のインターフェイスから構成され、今回考案した手法を用いるモデルである。モデル2はマイコンコアのみから構成されるモデルである。モデル3はマイコンコアと専用演算器複数から構成され、従来の専用演算器を内蔵したモデルである。回路規模を評価するために、モデル2であるマイコンコアの回路は命令供給器、レジスタ、ALU、バレルシフト、フラグレジスタ等のデータバス部とその制御部からなるマイコンとする。モデル3はマイコンコアに拡張演算器として絶対値減算器、32×32ビット積和演算器、ビットサーチ演算器回路を付加した。モデル1はマイコンコアにモデル2の拡張演算器をアプリケーションの必要に応じてそれぞれ1つの拡張演算器とインターフェイス回路を付加した。

このマイコンコアは命令フェッチ・命令解読、命令実行、メモリアクセス、レジスタライトバックの5段のパイプライン構成を持つロード・ストアアーキテクチャマイコンである。

4.2 評価プログラム

現在、マルチメディア応用としては、MPEG等を用いた動画・静止画・音声の圧縮伸張処理、音声・文字認識処理、グラフィック処理等が上げられる。ここでは評価プログラムとして、マルチメディア応用の具体的な応用例としてMPEG復号処

○ビットサーチ演算ループ

```

前処理
MOV (ビット入力),D3
MOV (A0),D2
INC A0
★ BSCH D2,D3
loop;
MOV (A0),D2
★ BSCH D2,D3
BNE found
INC A0
★ BSCH D2,D3
found;
ベクトル作成処理
MOV (ビット入力),D3
MOV (A0),D2
INC A0
★ BSCH D2,D3
CMP #0x0f,D3
BLT loop

```

★印は専用演算器命令

・レジスタの意味

A0:結果格納アドレス
A1:入力格納アドレス
A2:入力格納アドレス
D2:累和数据
D3:ループ回数

図.7 評価プログラム

理について行う。MPEG復号処理において、拡張演算器命令を繰り返し演算を用いる処理として、ビットストリームを処理するときにも多用されるビットサーチ処理に着目した。

図.7にこれらの演算の繰り返し部分であるループ部のアセンブラプログラムを示す。ここでBSCH Dm,Dnはビットサーチ命令である。また、評価時には拡張演算器命令以外は全て1サイクル実行命令とした。

4.3 性能とハード規模評価結果と考察

表1に4.2で述べた評価プログラムを用いて、ループ部分の処理1回当たりの動的サイクル数、実際のアプリケーション上でこのループ部が処理する全演算の動的サイクル数を示す。このように、本方式の繰り返し演算を利用したマイコンコアのパイプライン処理によって、拡張演算器内蔵マイコンと同じ性能が得られる。

他に文字認識処理等のアプリケーションの評価によっても、回路規模の増加量を抑えながら性能が向上することが確認できている。このように、

演算方法		モデル 1	モデル 2	モデル 3
ビットサーチ演算	処理 1 回の 実行サイクル数	1200	480	495
	ループ部の 全実行サイクル数	3326400 (1.00)	950400 (0.29)	980100 (0.29)

表.1 各モデルに対する実行サイクル数の比較

本手法によって拡張演算器を接続・切り離しを行うことにより、低コストでかつ特定のアプリケーションに対して高性能なマルチメディア処理に柔軟に対応可能なマイコンを実現することができる。

5、おわりに

以上、新たに考案したマイコンの専用演算器接続方式について、内蔵する場合と比較して性能を

劣化することなく、専用演算器を接続・切り離しを可能にすることについて述べた。今後、本稿の評価プログラム以外のアプリケーションに対してさらに評価していく予定である。

謝辞

日頃研究活動を指導していただいている松下電子工業 出口副参事、松下電器産業 間野主担当、坂尾主担当に感謝いたします。