

解説

6. 命令セットアーキテクチャの具体例



6.7 マイクロプロセッサ NS 32000 ファミリ†

松 浦 豊 竹

1. アーキテクチャ概要

ナショナルセミコンダクター社（以下 NS 社と略）の汎用 32 ビットマイクロプロセッサファミリ Series 32000 は、DEC 社の VAX 780 を基本に設計された。

図-1 は NS 32032 の内部ブロックダイアグラムである。現在ファミリとして 8 種類の CPU があるが、基本構成は変わらない。Series 32000 は、マイクロコードパイプラインマシンであり、各 CPU の性能はパイプライン段数により異なる。アーキテクチャおよび命令セットは、各 CPU で完全に互換性が保たれているため、ソフトウェアはオブジェクトレベルですべての CPU で実行可能である。

汎用、専用レジスタは、各 8 本用意されている。専用レジスタは、高級言語のサポートおよびソフトウェアのモジュール構造をサポートする機能をもつ。

2. Series 3200 命令セット

命令長は可変長であり、プログラムコードが最大限メモリ効率を良くするために、次の点が工夫されている。

- 1) 使用頻度順に命令をコード化
- 2) バイト単位の命令構成
- 3) 可変長ディスプレイメント
- 4) 高級言語対応

2.1 2 アドレスアーキテクチャ

表-1 は Series 32000 のデータ転送命令である。オペランドの欄にある gen は、ジェネラルオペランドの意味で、後で説明する 9 つのアドレッシングモードのどの組合せでも、指定可能であることを示す。

ニモニックの最後にある i は、データタイプを示し

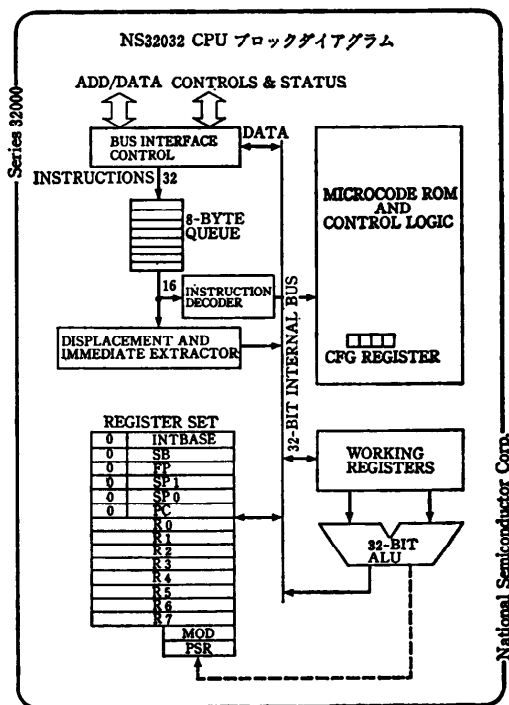


図-1 NS 32032 内部ブロックダイアグラム

ている。ユーザが指定する i および gen のフィールドは、組合せ上制限がない。

9 種類あるアドレッシングモードのうち、即値、絶対、レジスタ相対、メモリ空間、レジスタの 5 種類は、一般の CPU でも用意されている。これらに加えて高級言語向けとして、4 種類のアドレッシングが用意されている。

2.1.1 メモリ相対 (図-2)

オペランドはメモリにあり、アドレスは disp 2 とメモリにある 32 ビットのポインタとの合計で与えられ、このポインタアドレスと disp 1 と指定したレジスタの内容を加算することにより生成される。

† An Example of Instruction Set Processor Architecture Series 32000 by Yutaka MATSUURA (Hardware & System Support Engineer Microprocessor Strategic Products Group Marketing Div. National Semiconductor Japan Ltd.).

竹 ナシ ■ ナルセミコンダクター・ジャパン(株)マーケティング部

2.1.2 トップオブスタック

オペランドはメモリ空間の現在のスタック最上部に置かれている。

2.1.3 スケールドインデックス (図-3)

汎用レジスタの値に 1, 2, 4, 8 を乗算して、その結果を有効アドレスに加算し、最終オペランドアドレスを決定する。

2.1.4 外部 (図-4)

各モジュールが、外部モジュールのデータエリアに対してアクセスする際に使用する。

2.2 モジュール構造

Series 32000 のユニークな点は、ソフトウェアのモジュール構造をサポートしていることである。この構造の長所としては、開発時の柔軟性 (モジュールを個々に独立して開発、デバッグできる) や、リンク時にコード変更の必要がない (再配置が可能であり個々に ROM 化できる) などが考えられる。たとえば一つのモジュールを別のアドレスに移す場合でも、テーブル

disp 2 (disp 1 (FP)) フレームメモリ相対
disp 2 (disp 1 (SP)) スタックメモリ相対
disp 2 (disp 1 (SB)) スタティックメモリ相対

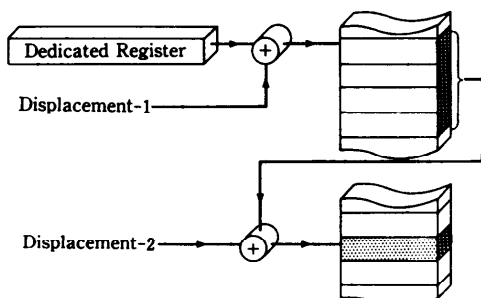


図-2 メモリ相対アドレッシングモード

disp (reg) [Rn: i] スケールドインデックス

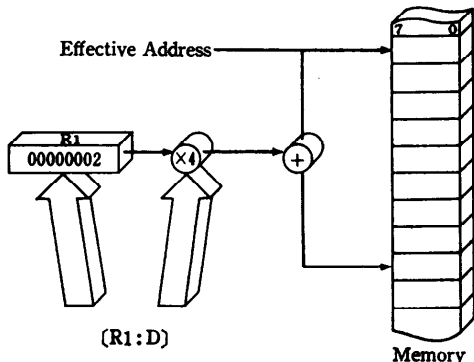


図-3 スケールドインデックスアドレッシングモード

表-1 データ転送命令

• データ転送			
フォーマット	オペコード	オペランド	概 略
4	MOV _i	gen, gen	データの転送
2	MOVQ _i	short, gen	符号付き 4 ビット定数を拡張して転送
7	MOV _{Mi}	gen, gen, disp	disp (1 から 16) で指定した数だけ転送
7	MOVZBW	gen, gen	上位にゼロを拡張して転送
7	MOVZiD	gen, gen	上位にゼロを拡張して転送
7	MOVXBW	gen, gen	符号付きで拡張して転送
7	MOVXiD	gen, gen	符号付きで拡張して転送
4	ADDR	gen, gen	実効アドレスを転送

を改新するだけでこの作業を完了する。

このモジュール構造をサポートするために、モジュールテーブル (図-5) とリンクテーブル (図-6) が設けられている。モジュールテーブルは、各モジュールのスタティック変数領域のアドレスポインタ、リンクテーブルアドレス、プログラムベースアドレスの情報が入力されている。リンクテーブルには、実行しようとするモジュールのアドレス情報が入力されている。

図-7 は CPU が、他のモジュールに処理を移す動

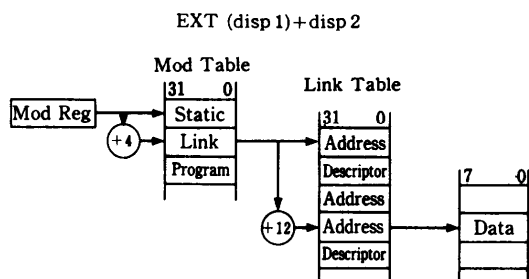


図-4 外部アドレッシングモード

アドレス	31	0
MOD:	スタティックベース	
MOD+ 4:	リンクベース	
MOD+ 8:	プログラムベース	
MOD+12:	(今後の使用のため保留)	

図-5 モジュールテーブルの形式

エントリ	タイプ	31	16	15	0
0	変 数	絶対アドレス			
1	変 数	絶対アドレス			
2	プロシージャ	オフセット		モジュール番号	

図-6 リンクテーブルの形式

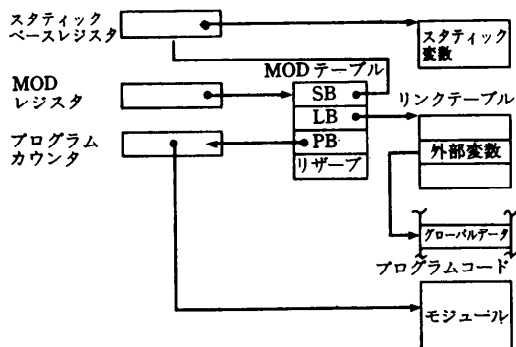


図-7 モジュールの管理

作を示している。CPU内のMOD(モジュール)レジスタにより、どのモジュールテーブルを参照するかが決められる。参照したモジュールテーブルより、スタティックベース、プログラムカウンタが入力される。割込み処理、トラップ処理も、モジュール構造になり、条件が発生するとCPUはモジュールを移行する。

先に説明した外部アドレッシングモードは、他のモジュールの変数をアクセスするために使用され、モジュールテーブル内のリンクテーブルより、ディスクリプタの参照アドレスが決定される。外部アドレッシングは、リンクテーブルの先頭アドレスを参照し、disp 1を加算したアドレスにある32ビットのアドレスポインタに disp 2を加算した値が、オペランドのポインタ位置として与えられる。

3. スレーブプロセッササポート

Series 32000では、CPUがスレーブプロセッサ命令を管理している。命令体系も他の命令と同じ構造であり、アドレッシングモードなども全てサポートされている。スレーブプロセッサ命令であっても、ステータスレジスタの改新の必要な場合には自動的にCPU内のステータスビットが改新される。図-8に示す浮動小数点比較命令も、比較結果によりCPU内のステータスビットが改新されているため、条件ブランチ命令もそのまま実行可能である。

4. 高級言語対応アーキテクチャ

UNIXおよびC言語をサポートする機能を、オンチップでもっているのもSeries 32000の特長の一つである。

たとえばフレームポインタは、ENTER命令によってプロシージャの初めに設定され、アロケートされた変数をアクセスする際に参照される。図-9に示す

浮動小数点演算命令 使用例

8087		32000	
FCOM	ST(1)	CMFP	F0, F1
FSTSW	FLGTMP	BEQ	PRG 1
FWAIT			
MOV	AH, BYTE PTR		
	FLGTMP+1		
SAHF			
JZ	PRG 1		
計	6 ステップ	2 ステップ	

図-8 浮動小数点命令比較

Series 32000

ENTER (reg list), disp

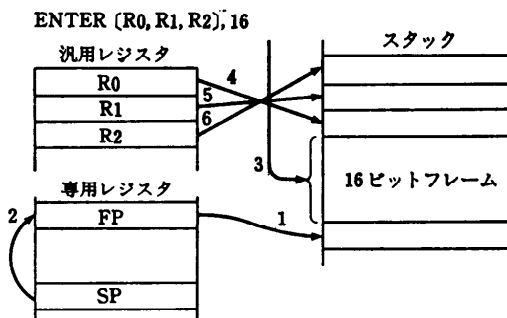


図-9 フレームポインタアロケーション

ENTER命令は、スタック領域に現在のフレームポインタをコピーし、dispで指定されるデータ領域を確保する。C言語での関数コール時のパラメータの転送は、この領域にアロケートされフレームポインタを使用しアクセスされる。

スタティックベースレジスタは、プロシージャのローカル変数領域へのデータアクセスに使用される。

5. まとめ

NS社のSeries 32000は、マイクロコードベースのCISC型のマイクロプロセッサである。内部は各ブロックで独立したパイプライン構造になっており、その管理をマイクロコードで行っている。この方式では、各データパスでコンテンションが発生しないかぎり、並列にデータが処理される。性能を向上させるため、NS 32532では4段の命令パイプラインを用意し、最大7つの命令がパイプライン上で処理される。

NS社ではRISC型のマイクロプロセッサに対して、外部ウェイト挿入された際の影響、プログラムサイズ、バス使用頻度(マルチCPU構造のサポート)などから、CISC型マイクロプロセッサを今後もベースとする。

(昭和63年11月1日受付)