高速演算処理用コプロセサ 8087

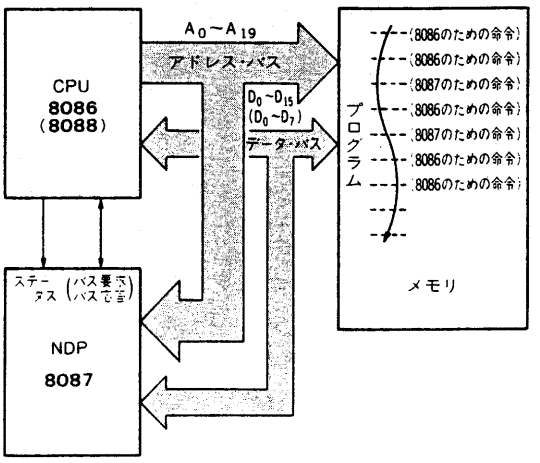
インテル ジャパン 教育センター　鎌田信夫

はじめに

16ビットのマイクロプロセサが かなり一般化し 広い分野で応用され始めている。今日市販されている16ビット マイクロプロセサは その命令セット，アドレス空間，マルチプロセサに対する備えなどの点で すでにミニコンと変らない水準にある。しかし 入出力制御や浮動小数点演算に対しては これらのマイクロプロセサは特別なハードウェアを内蔵するまでに至っていない。そこでこのLSI技術の制限の下で システム全体としての性能向上をはかるべく 採られている手法は それぞれ別パッケージの専用プロセサを用意し いくつかのプロセサ群を組合わせてシステムを構成する方法である。ここでは その一例である 16ビットCPUと密結合で使用される高速演算処理用プロセサ 8087について，その動作，性能について紹介する。

コプロセサの概念

NDP（numeric data processor）8087は独立したプロセサではない。8087は16ビットCPU 8086のコプロセサである。コプロセサとは マルチプロセサの特別な例ともいえるもので，複数のプロセサが同じプログラム セグメントの同じ命令streamを互いに同期をとりながら逐次実行していくのである。その概念を図解すれば図1のようになる。



〔図1〕 コプロセッサの概念図

両者はローカルバスを共有し，メモリ内の命令コードにはCPUとコプロセサに対する命令が混在している。両プロセサはフェッチされる命令コードを常に見守り，おのおの自分に対する命令のみを選んで確実に実行していく。このシステム構成において 一方のプロセサがある命令を実行中，他方のプロセサがつぎの命令，さらに つぎの命令と実行を先行してよい場合と そうでない場合がありうる。いずれにせよ，ローカル バスを

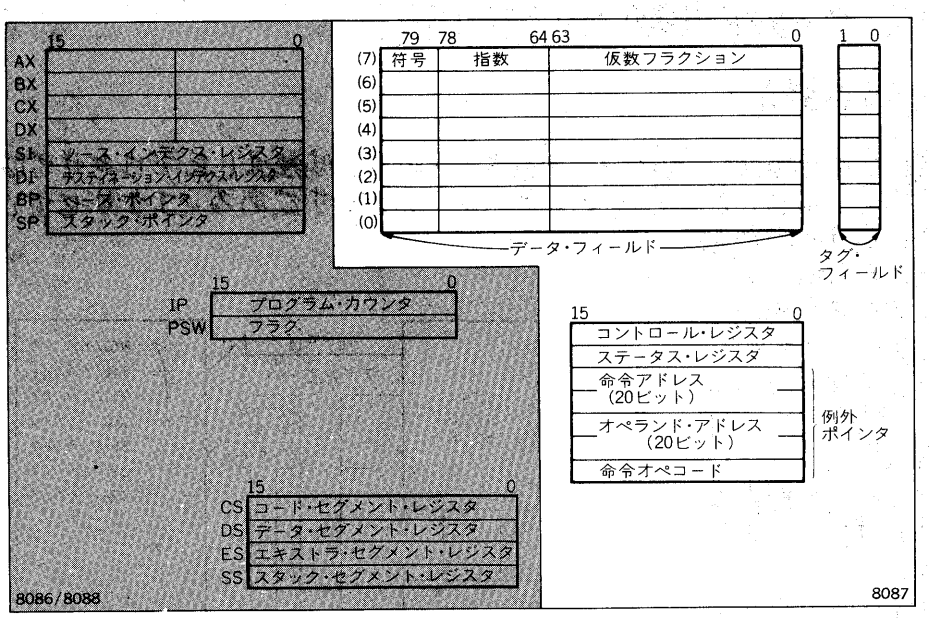
共有し 連携しながら動作するCPUとcoprocessorのシステムにおいてはつぎの三つのメカニズムが必要である。

(1) フェッチされた命令コードを見て、自分の命令であるか否かを判定する解読機構。

(2) 一方のプロセサがある命令の実行を終了するまで 他のプロセサが先行することを止めさせる機構。

(3) メモリ・アクセスのために 二つのプロセサが互いに共有バスの使用権を要求し(request), 認められる(grant)機構---- マルチ・プロセサのメカニズム。

CPUとコプロセサは上の条件を満たすべく, デコーダを内蔵し, かつ連絡をとり合う制御線で密に結ばれる必要がある。詳細は文献(2)を参照していただくとして, 二つのプロセサが具合よく命令ストリームを分担しながら実行してくれるのであれば プログラマからみて両プロセサは一体のものと考えてよい。



〔図2〕 8086/8087の一体化されたアーキテクチャ

CPUにコプロセサが追加されたことは、プログラマからみれば元のCPUに8087の持つ命令セット(浮動小数点を含む算術演算のための)およびレジスタなどのハードウェア資源が増設されたことにほかならない。コプロセサはもとからあるプロセサに対しアーキテクチャ上のextensionをもたらす。さて8087が採用した浮動小数点の表現, データ型式および例外処理などはIEEE案(文献(3))に準拠している。したがって コプロセサ8087とは「内部に算術演算のための80ビットのレジスタ・スタックを8個, ALU, 例外発生時のオペラ

ンド指定やステータス表示のためのレジスタなどを内蔵する算術演算用LSIでCPU8086(8088)と密結合で使用される」あるいは「浮動小数点に関するエミュレーションの単一チップによるhardware implementationである」と言うことができる。図2に一体となったプロセサ・リソースを示す。

CPU側の備え(1)

8087がCPUのコプロセサとして正しく動作し機能するためにCPU8086側がどのような備えを予め有していたかをまとめると次のようになる。

(1) CPUは命令の先取り方式を採用しているが，命令はフェッチされ，Queueバッファに貯えられる。このQueueの状態は2本の信号線QS_1, QS_0によって外部に示される（表1）。CPUはメモリアクセスなどを果たすバス・インターフェース（BIU）ユニットと命令の実行を行なうExecutionユニット(EU)の2つのユニットから構成される。

(2) CPUの端子$\overline{RQ}/\overline{GT}_0$, $\overline{RQ}/\overline{GT}_1$のいずれかを活性にすることによって，CPUにバスの使用権を要求できる。応答もこのラインを使う。

(3) CPUの各マシンサイクルの情報はステータス信号S_0～S_2によってCPUから外部に示される。このステータス情報はCPUの動作と歩調を合わせるコプロセッサにとって不可欠なものである。コプロセッサは特にメモリ・アクセスのマシンサイクルに注目する。

〔表1〕 命令キュー・バッファのステータス

QS_1	QS_0	キュー・ステータス
0	0	キュー状態変化なし． 先のクロック・サイクルにおいて命令キュー・バッファに何事も行なわれなかった．
0	1	最初のバイト． 先のクロック・サイクルにおいて命令コードの最初の1バイトめがキューから送り出された
1	0	キューが空． 分岐命令などのために先取りして貯えておいた命令がすべてむだになった．そのためキューが再初期化されキューは空である．
1	1	継続のバイト． 命令コードの2バイトめ以降のバイトがキューから送り出された．

0；Low　　1；High

(4) CPU8086の命令語のなかに1ワードのオペコードの命令，ESC（エスケープ）がある。そのオペコードは

11011 xxx mod xxx r/m

と表示され，2ビットのmodフィールド，3ビットのr/mフィールドはメモリ・オペランドのアドレッシング・モードを規定する。

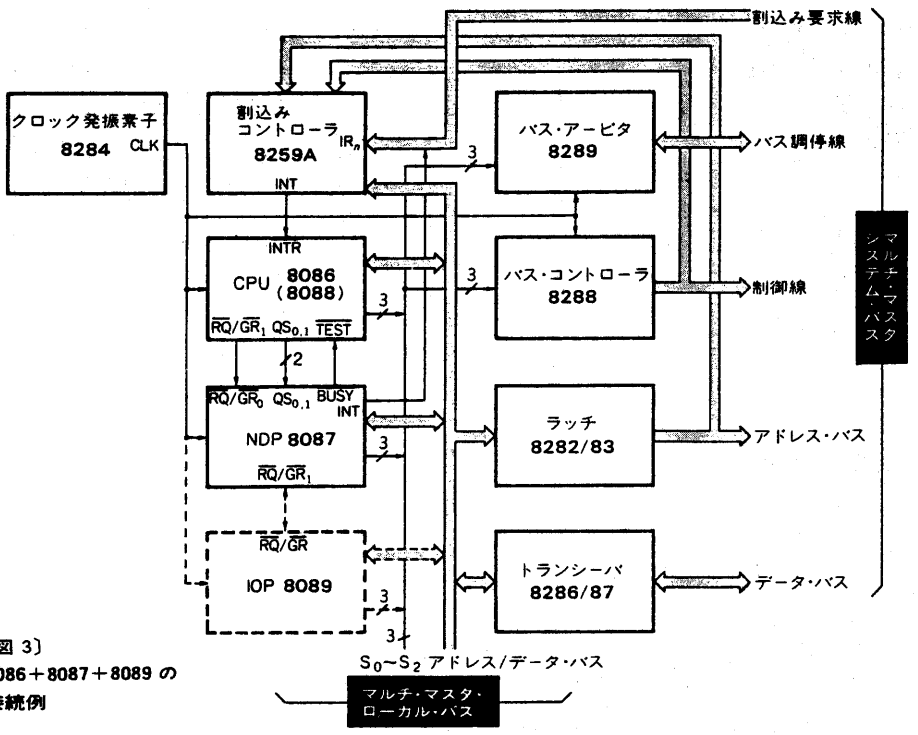
このESC命令に対するCPUの実行内容は単純である。つまり，ローカル

バスを共有しているコプロセサに対する命令であることを解読し，命令コードの2バイト目のmod，r/mのビットにもとずいて必要なアドレス計算を行ない，オペランド・アドレスをシステム バスに供給する。アドレス指定されたメモリからオペランド値がデータ バスに転送されてくるだろうが，CPUはそれをREADしない。

以上はCPU側の設計時に考慮されたコプロセサに対する備えである。

コプロセサの接続とシステム動作

図3はシステム構成を示す。[4] この図では I/O プロセサ 8089 もシステムに含めてある。各ローカル・マスタ；8086，8087，8089はこの図のように共通のクロック信号の下で動作する。 8087がある算術演算の命令の実行



〔図 3〕
8086＋8087＋8089 の
接続例

を開始するとBUSYが"H"レベルになるので これをCPUの $\overline{Test}$ に接続してソフトウエア同期を行なう。各マスタがバスの使用権を獲得し合いながらシステムバスの資源をアクセスするよう各RQ/GT線は図のように接続される。

コプロセサもCPUと同様の命令Queueを有し，CPUと同様に歩調を合わせ，フェッチした命令をQueueに貯えておく。CPUがそのQueueから ひとつの命令をとり出して実行すると その命令が8087の命令でない場合は，8087はその命令をQueueから捨て去る。

8087の命令に相当するESC命令に対して，CPUは それを解読し，メモリを参照するESC命令か否かを判定する。メモリオペランドを参照するものであれば CPUはアドレスを出力しオペランドをReadしない というダミーの読出しサイクルを実行する。そのESCがメモリオペランドを参照しない命令であればCPUは何もしないで 次の命令実行に移る。

一方，8087の方も同時にQueueから取り出したESC命令を解読し，自分に対する何らかの命令であることを知る。それは (i)オペランドをメモリからロードする命令か，(ii)オペランドをメモリにストアする命令か，(iii)メモリを一切参照しない命令か ------のいずれであるかを判定し，CPUによって与えられたアドレスを使って メモリをアクセスしたり，その他の操作を行なう。

8087のデータ・タイプ

8087で扱えるデータタイプを表2に示す。浮動小数点の形式はIEEE標準案に準拠している。

〔表2〕 8087のデータ・タイプとその表現

データ・タイプ		範囲のオーダ	精度	7 0,7 0,7 0,7 0,7 0,7 0,7 0,7 0,7 0,7 0,7 0
Word Integer	−32768〜+32767	10^4	16ビット	S I_{15} ---------- I_0 (2の補数方式)
Short Integer	-2.15×10^9〜$+2.15\times10^9$ 注1)	10^9	32ビット	S I_{31} ---------- I_0 (2の補数方式)
Long Integer	-9.22×10^{18}〜$+9.22\times10^{18}$ 注2)	10^{19} (10^{18})	64ビット	S I_{63} ---------- I_0 (2の補数方式)
Packed Binary-Coded Decimal (BCD)	−999999999999999999〜+999999999999999999	10^{18} (10^{17})	18桁 (80ビット)	S D_{17} D_{16} D_{15} D_{14} D_{13} D_{12} D_{11} D_{10} D_9 D_8 D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
Short Real	$8.43\times10^{-37} \le \lvert x\rvert \le 3.37\times10^{38}$	$10^{\pm38}$	24ビット	S E_7 -- E_0 F_1 ---------- F_{23} (F_0は暗黙に表現)
Long Real	$4.19\times10^{-307} \le \lvert x\rvert \le 1.67\times10^{308}$	$10^{\pm308}$	53ビット	S E_{10} ---- E_0 F_1 ---------- F_{52} (F_0は暗黙に表現)
Temporary Real	$3.4\times10^{-4932} \le \lvert x\rvert \le 1.2\times10^{4932}$	$10^{\pm4932}$	64ビット	S E_{14} ---------- E_0 F_0 ---------- F_{63}

注1) −2147483648〜+2147483647
注2) −9223372036854775808
〜+9223372036854775807

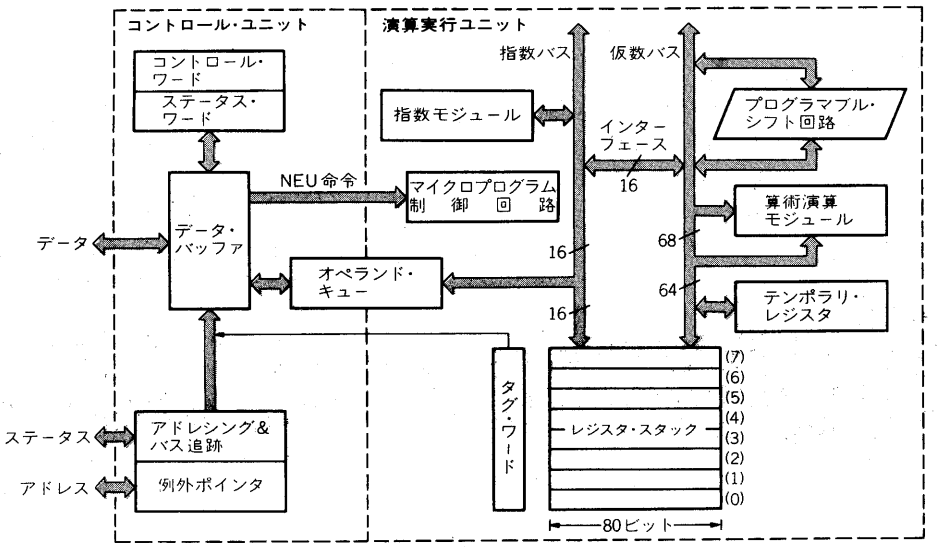
フォーマット
整数：I
Packed BCD：$(-1)^S(D_{17}\cdots\cdots D_0)$
実数：$(-1)^S(2^{E-バイアス})(F_0, F_1, F_2\cdots\cdots)$，バイアス=127(Short Real)
=1023(Long Real)
=16383(Tempoqary Real)

8087の内部構成と命令セット

8087の内部もCPUと同様，2つのブロックで構成され，それらはコントロール・ユニット（CU），演算実行ユニット（numeric execution unit; NEUと名づけられている。CUは命令をフェッチし，メモリオペランドを読み書きしたり，8087の制御命令を実行したりする。算術演算の実行はNEUの方で行なわれ，NEUに演算回路，スタック群が含まれている。

8087のブロック図を図4に示す。

〔図4〕 8087のブロック図



CUもCPUと同様の命令Queueがある。命令語はそこから5byte取り出されESCコードの演算命令がデコードされるとNEUに制御が移される。このNEUにある演算の制御プログラム（マイクロプログラム）は約3万ビットである。

8087の内部にはロードするオペランドやメモリにストアするオペランドを貯えておくオペランドQueueもあって演算とバスサイクルが効率よく行なえるようになっている。NEUの内部バスの巾は68ビットと広くとってある。

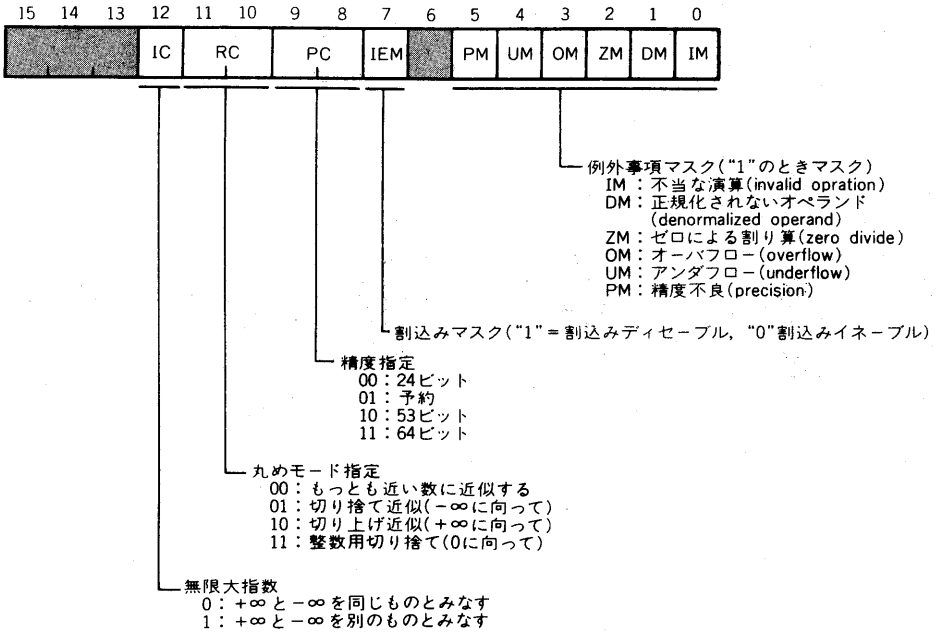
NEUにあるレジスタ・スタックは表2の最も精度の高いデータタイプに合致した長さになっており，9本の命令においてlast in - first out形のスタック的な使われ方をする。

図5に示すコントロール・レジスタは8087の動作モードのオプションを規定するもので，精度の指定，丸めモードの指定，割込みマスクの設定，例外に対する処置の指定など，プログラマが命令で初期設定する内容のものが，ビットごとに定義されている。これらのオプションのひとつがIEEE標準案に一致す

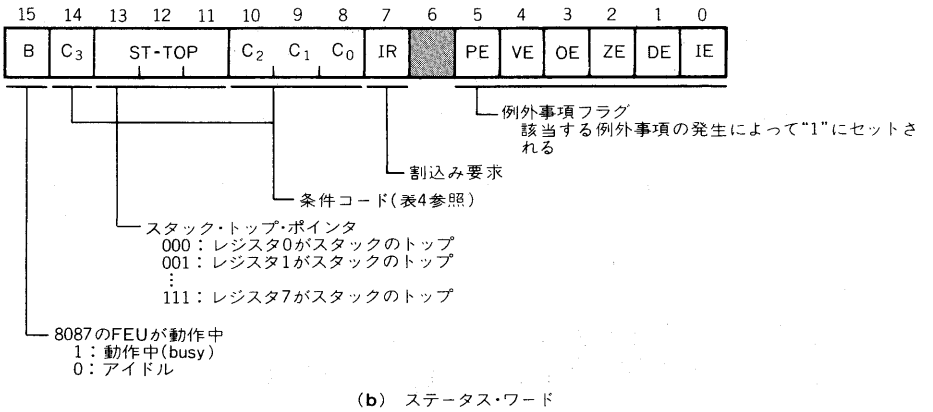
る。図5に示すもうひとつのレジスタ，ステータス・レジスタは8087の全体的な状態を記録するレジスタである。つまり例外の発生，NEUがBUSY分否か，スタック・トップ，条件コードなどの情報が記録される。この他のレジスタとして8087には「アドレシングおよびバス追跡」，「例外ポインタ」の2つがある。これらは8087が実行するESC命令のコードおよびそのアドレス，参照オペランドのアドレスを記録するもので，いずれも例外発生時のハンドラのために使われる。

8087の命令セットはニモニックで69種ある。IEEE案に示されている基本演算，つまり，加減乗除算，剰余算，平方根，浮動↔固定小数点の相互変換などの演算命令が8087の命令セットに含まれている。8087の命令セットのレパートリを表3に示す。

〔図5〕コントロール・ワードとステータス・ワード



（a） コントロール・ワード



（b） ステータス・ワード

実行時間，プログラミング（むすび）

コプロセサを付加することによって多くの算術演算はCPUのみの命令によるソフトウェアルーチンにくらべ約100倍の高速化が実現される。表4に実行時間の比較を示す。この速度はI/O素子としての演算素子（9511など）より約10倍速い。アセンブラによるプログラミングでは8087の命令語ニモニックを直接コーディングに含めてよい。高級言語では，生成されるオブジェクトに8087のマシンコードを含めるか否かの指定はリンク時に行なえるようになっている。(6)~(10)

標準案に基づく浮動小数点の扱いと演算，高速な演算，精度の高い定数値（πなど）の発生，例外に対するファームウェア化された処理などこのコプロセサによってマイクロコンピュータがさらに高度な応用に使いやすくなった。

〔表3〕 8087の命令セットのレパートリ

分類	命令
データ転送	ロードとストア（全データ・タイプに対して），交換
算術演算	加算，減算，乗算，除算，逆の減算，逆の除算，平方根の計算，スケール算，剰余算，整数への丸め，符号反転，絶対値，仮数および指数の抽出
比較	比較，検査，テスト
超越関係	正接，逆正接，2^x-1，$y\cdot\log_2(x+1)$，$y\cdot\log_2 x$
定数	0，1，π，$\log_{10}2$，$\log_e 2$，$\log_2 10$，$\log_2 e$
プロセッサ制御	コントロール・ワードのロードとストア，ステータス・ワードのストア，環境情報のロードとストア，退避と復帰，割込みイネーブルとディセーブル，例外のクリア，初期化

〔表4〕 実行時間の比較

命令	8087(5MHz)	8086 エミュレーション
Add or subtract magnitude	14/18 μs	1,600 μs
Multiply (single precision)	19	1,600
Multiply (double precision)	27	2,100
Divide	39	3,200
Compare	9	1,300
Load (double precision)	10	1,700
Store (double precision)	21	1,200
Calculate square root	36	19,600
Calculate tangent	90	13,000
Raise to the appropriate power	100	17,100

文献 （文献(5)は基礎のため，(11)-(12)は総合報告）

1. Intel Corp., The 8086 Family User's Manual, Oct. 1979

2. Intel Corp., The 8086 Family User's Manual, Numerics Supplement, July 1980

3. J.T. Coonen, "An implementation guide to a proposed standard for floating-point-arithmetic, "IEEE Computer, Jan. 1980 および J.T. Coonen, "Specification for a proposed standard for floating poing arithmetic, "UCB/ERI M78/72, Oct. 13, 1978 and Aug. 26, 1980

4. Intel Corp., Designing 8086, 8088, 8089 Multiprocessing systems with the 8289 bus arbiter, application Note, AP-51, 1979

5. たとえば T.L. Booth and Y.T. Chien , Computing, Fundamentals and Applications, Hamilton Publishin Company/John Wiley & Sons Inc., 1974

6. 8086/8087/8088 Macro Assembly Language Reference Manual for 8080/8085 based development systems, 121623-01 Rev. A

7. 8086/8087/8088 Macro Assembler Operating Instructions for 8080/8085 based development systems, 121624-01 Rev. A

8. PL/M-86 Programming Manual, 9800466-02 Rev. C (change 1)

9. 8086 Family Utilities User's Guide for 8080/8085 based development systems, 9800639-04 Rev. E

10. PL/M-86 Compiler Operating Instructions for 8080/8085 based development systems, 9800478-04 Rev. D

11. 鎌田信夫「高速演算処理用マイクロプロセサ」，『第2回マイクロコンピュータ応用国際コンファレンス予稿集』，(社)日本電子工業振興協会，July 1980

12. 鎌田信夫「コプロセッサ8087」上，中，下『インターフェース』Oct 1980, DEC 1980, MAR 1981 CQ出版