COBOL マシンの評価†

山本昌弘†† 中崎良成††
横田 実†† 梅村 護††

本論文は Cobol 言語処理に専用化して作られた Cobol マシンについて，応用プログラムを実行させることによって行った評価について論じている．評価は言語処理に専用化することによって得られる効果を定量的に分析するために，汎用処理を意図して作られた汎用計算機との比較によって行っている．ここでは，Cobol マシンのハードウェアやファームウェア量，オブジェクトメモリ量，メモリアクセス回数，実行速度およびハードウェアの構造を規定するパラメータについて詳細に分析を行う．おもな評価結果として，汎用計算機と比べた場合に，オブジェクトメモリ量は命令部について平均約 26% 減少し，メモリアクセス回数は平均約 35% 減少し，実行速度は平均約 2.1 倍向上することが得られた．また，今後，Cobol マシンの商品化を進めるときに有効な各種の詳細な評価データが収集された．

1. はじめに

著者らは，Cobol 言語[1]で書かれたプログラムの高速処理を目的とした付加プロセッサ形式の Cobol マシンを開発したが，本論文では，Cobol プログラムを，Cobol マシン上で動作させることにより測定したデータを基に行った評価について論じる．

高級言語マシンはこれまで各種の高級言語に対して多くの研究がされているが，評価結果についての報告は非常に少なく，また，Cobol マシンについては皆無に近い．ここでは，Cobol 言語処理に対する専用化による効果について，汎用計算機と比較することによって定量的に明らかにする．このため，まず，Cobol マシンを構成するファームウェアやハードウェア量，オブジェクトメモリ量，メモリアクセス回数および実行速度について評価する．さらに，Cobol マシンのハードウェアの構造を規定するパラメータについて分析する．

2. Cobol マシンの概要

Cobol マシンのアーキテクチャ[2]およびハードウェア[3]について，評価に関連する項目を中心に概要を述べる．**図1**は Cobol マシンを用いたシステム構成と Cobol マシンのハードウェア構成を示す．Cobol マシンは汎用中型計算機 Acos システム 300[4]をもとに作られたホストプロセッサに接続されて動作し，ホストプロセッサ上で動く Cobol マシン・トランスレータ[5]によって翻訳されて作られたソースステートメントに近い高度な命令列を実行する．したがって，中間言語方式の高級言語マシンである．Cobol マシンは Cobol の算術，移送，データ操作，条件，実行順序制御などのステートメントを実行し，入出力や通信などのステートメントはホストプロセッサに処理を依頼する．

Cobol マシンのアーキテクチャは Cobol 言語で使用できるデータに直接対応する多種類の内部データ，複雑な構造のデータへアクセスするための高度なアドレス機構および Cobol ステートメントに対応する高機能命令をもち，Cobol ステートメントは大部分 Cobol マシンの 1 命令へ変換される．

Cobol マシンはパイプライン制御される三つのプロセッサ複合体（命令取出しプロセッサ：IFP，オペランド取出しプロセッサ：OFP，命令実行プロセッサ：EXP）と先取り制御部，主記憶インタフェース制御部から成っている．命令取出しプロセッサは Cobol のステートメント動作を反映した命令列の先取り[6]とアドレス展開処理を行い，オペランド取出しプロセッサはデータの取込みとデータ形式変換を行い，命令実行プロセッサは演算処理と結果の格納を行う．先取り制御部は演算結果が主記憶にセットされるオペランドの先取りが可能か否かを検査するための制御を行い，主記憶インタフェース制御部は三つのプロセッサからのメモリ要求を選択し，主記憶に依頼する制御を行う．

3. 評価環境

Cobol の言語処理に専用化した Cobol マシンの評価

† COBOL Machine Evaluation by MASAHIRO YAMAMOTO, RYOSEI NAKAZAKI, MINORU YOKOTA and MAMORU UMEMURA (C & C Systems Research Laboratories, Nippon Electric Co., Ltd.).

†† 日本電気(株) C & C システム研究所

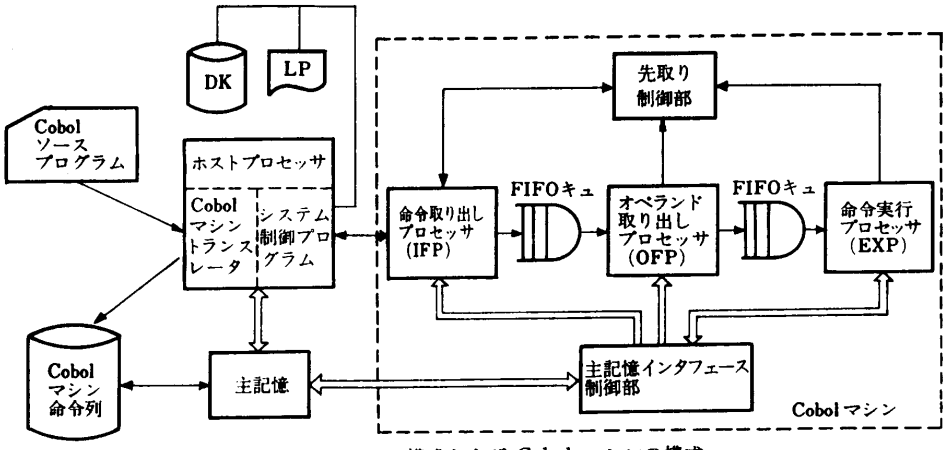

図 1 システム構成および Cobol マシンの構成

Fig. 1 System configuration and Cobol machine configuration.

について，汎用処理を意図して作られた汎用計算機との比較により行う．そして，Cobol マシンがホストプロセッサの基本となっている Acos システム 300 と同じハードウェアを用いて実現されていることから，比較対象の汎用計算機として，Acos システム 300 を選んだ．

評価はステートメントレベルとプログラムレベルの二つの観点から行う．評価に用いるステートメントとしては，ユーザのプログラムを分析して設定した 15 個のステートメント（演算処理だけで，入出力処理は含まれない）[7]を選んだ．また，使用したプログラムは実際に業務で用いられている約 90 本のプログラムから成り（静的ステップ数は約 2 万 8,000，動的ステップ数は約 2,500 万），これらは，三つの分野（第 1 は比較的演算が多い事務処理プログラム，第 2 は文字列データを対象とするテキスト処理プログラム，第 3 は典型的な事務処理プログラム）から成っている．

表 1 ベンチマークプログラムの特性

Table 1 Benchmark programs feature.

		プログラム		
		P_1	P_2	P_3
ステートメント頻度(%)	GO TO	21.0	0	15.2
	IF	37.8	30.1	38.3
	PERFORM	0.9	0.2	1.2
	MOVE	6.6	11.7	20.4
	Arithmetic	24.8	0.1	13.8
	Data Handling	0	0.4	0
	Table Handling	0	29.0	0
	I/O operations	0.1	0.1	9.1
	Others (EXIT, STOP)	8.6	28.4	2.0
ステートメント数（ステップ）	静 的	320	129	925
	動 的	867098	836666	1564969

また，詳細な評価のときには，上記三つの分野のなかから代表的な三つのプログラムを選んだ．これらのプログラムの特性を**表 1** に示す．プログラム P_1 は迷路チャート作成プログラム，P_2 は英単語のチェックプログラム，P_3 は貸借対照表作成プログラムである．

ハードウェアに関する詳細なデータの収集には，テスデータ社のハードウェアモニタ MS-88 を使用した．

4. ハードウェアおよびファームウェア量

Cobol マシンのアーキテクチャは汎用計算機と比べて高度な機能を備えている．したがって，これを高性能に実現するために，Cobol マシンは汎用計算機よりも高度なハードウェアを備えており，規模が大きくなっている．**表 2** は Cobol マシンのハードウェア量を機能別に示したものである．表 2 において，機能 1 から 5 は Cobol マシンを実現するための個有のものと考えられる．しかし，機能 6 は付加プロセッサ機構のないホストプロセッサに Cobol マシンを接続するために設けられた機能であり，したがって，マルチプロセッサ構成を考慮した接続付加機構をもつホストプロセッサの場合には除去できる．また，本 Cobol マシンは，詳細

表 2 Cobol マシンのハードウェア量

Table 2 Cobol machine hardware capacity.

	機 能	IC 数(個)
1	命令取出しプロセッサ	1700
2	オペランド取出しプロセッサ	1720
3	命令実行プロセッサ	3110
4	FIFO キュメモリ	270
5	先取り制御部	710
6	主記憶インタフェース制御部	1610
	合 計	9120

表 3 Cobol マシンのファームウェア量
Table 3 Cobol machine firmware capacity.

ステップ数(%)

命令, 等 ＼ プロセッサ	命令取出しプロセッサ	オペランド取出しプロセッサ	命令実行プロセッサ
演算 (Add, Mult. 等)	74(7.2)	553(18.7)	639(27.1)
移送 (Move)		376(12.7)	153(6.5)
条件 (GoTo, If, Perform. 等)	146(14.2)	397(13.5)	336(14.2)
データ操作 (Inspect, String. 等)	112(10.9)	793(26.9)	651(27.6)
表操作 (Set, Search. 等)	94(9.1)	195(6.6)	131(5.5)
その他 (Transform. 等)	14(1.4)	67(2.3)	59(2.5)
共通ルーチン	587(57.2)	571(19.3)	393(16.6)
合　計	1027	2952	2362

な評価データを収集する機能などを組み込んだ実験機である．したがって，このような点を考えると，商用 Cobol マシンとして実現した場合には，ホストプロセッサの基本となっている汎用中型計算機 Acos システム 300 の CPU (約 5000 IC) とほぼ同程度のハードウェアで構成できるものと予想される．

表 3 は三つのプロセッサを構成するファームウェアの機能別ステップ数を示す．全体で約 6.3k ステップであり，汎用計算機と比べやや多めである．そのおもな理由は，命令取出しプロセッサの高度なアドレス展開処理，オペランド取出しプロセッサが行うデータのタイプ変換処理（表 3 の共通ルーチンの部分）および命令実行プロセッサの複雑な命令の処理，などに多くのファームウェアが必要なためである．

5. 中間言語アーキテクチャの評価

中間言語方式の高級言語マシンでは，中間言語をどのレベルに設定するかが非常に重要である．本章では，Cobol マシンの中間言語アーキテクチャを，(1) 高級言語指向度，(2) オブジェクトメモリ量，(3) メモリアクセス頻度，の 3 点から考察する．

(1) 高級言語指向度

Cobol マシンの中間言語アーキテクチャが Cobol 言語にどの程度指向しているかを計るために，Cobol ステートメントが何個の命令に変換されるかを示す値 (IPF : instructions per function) を求めた．**表 4** は 3

表 4 高級言語指向度 (IPF)
Table 4 High-level language proximity (IPF).

	Cobol マシン	Acos-4	IBM 370*
静　的	1.16	3.07	5.74
動　動	1.15	3.61	6.76

* 推定値

表 5 オブジェクトメモリ量
Table 5 Object memory capacity.

(%)

プログラム	命令部	データ部	全　体
P_1	74.9	99.1	88.8
P_2	64.2	99.2	97.2
P_3	83.4	93.1	87.2

汎用計算機を 100% とする．
また，ディスクリプタはデータ部に含む．

章で述べた応用プログラムについての Cobol マシン，Acos-4 および IBM 370 アーキテクチャに関する静的および動的な IPF 値を示す．ただし，IBM 370 の値は 3 章で述べた 15 個のステートメントの IPF 値を基に推定したものである．この評価によって，Cobol マシンは IPF 値が 1 に近く，また，ステートメントの複雑度によるばらつきが汎用計算機と比べて少ないことがわかった．

(2) オブジェクトメモリ量

表 5 に，3 章で述べた 3 本の応用プログラムのオブジェクトメモリ量について，汎用計算機との比較を示す．オブジェクトのなかで，命令部は大幅に減少する (平均数 26%) が，データ部はほとんど減少しない．命令部が減少するおもな理由は，Cobol マシンでは，高度な命令を備え，かつ，添字づけやデータタイプ変換処理がハードウェアやファームウェアで行われていること，などがあげられる．

(3) メモリアクセス頻度

オブジェクトメモリ量が減少すると，実行時のメモリアクセス回数もそれと対応して減少する．**表 6** はメモリアクセス回数を汎用計算機と比較したもので，プログラム全体で平均約 35% 減少する．

通常，プロセッサの高速化と並列処理化によって，システムはメモリリミット状態に近づくが，このように，メモリアクセス回数が絶対的に減少することによって，メモリネックになることを防ぐことができる．

したがって，このメモリアクセス回数の減少は高級言語マシンの重要な効果の一つであり，これによっ

表 6 メモリアクセス頻度
Table 6 Memory access frequency.

(%)

プログラム	命令リード	データリード	データライト	全　体
P_1	65.5	52.3	36.1	55.7
P_2	55.2	82.7	48.4	64.9
P_3	98.5	55.9	83.5	74.9

汎用計算機を 100% とする．
また，ディスクリプタリードはデータリードに含む．

て，本Cobolマシンは本質的に実行速度が向上すると考えられる．

メモリアクセス回数が減少する理由として，オブジェクト命令数が少ないために命令読出し回数が減ること，中間結果がハードウェアレジスタに蓄積されることやデータタイプ変換，小数点桁合せ処理がハードウェアでサポートされているので主記憶中のワーク領域を用いて命令によって行う必要がないこと，などがおもな理由である．

6. 実行性能の評価

Cobol 言語の場合には，一度作成されたプログラムは頻繁に実行される．したがって，汎用計算機では，コンパイラによって生成されたオブジェクトが実行される場合が多いことを想定すると，Cobol マシンでの中間言語の実行性能が非常に重要である．ここでは，ステートメントレベルと応用プログラムレベルについて評価した．

（1） ステートメントの実行速度

表7 はユーザの応用プログラムを基に選ばれた15個のCobolステートメントに関するステートメント実行速度について汎用計算機との比較を示す．Cobol マシンは命令列の先取りを行っているために，実行速度は If や Perform などの条件分岐命令の分岐率に依存する．分岐率が50%の場合で約2.5倍高速になる．ここで現われる15個のステートメントはデータ構造が単純（添字づけ，小数点位置合せ，データのタイプ変換などがない）であり，また，命令も簡単（Add, Sub, Move, If などが中心）なものが大部分である．そこで，ユーザプログラムのなかから，これらより複雑なデータや命令（String, Inspect, Search など）を備えるステートメント10種を選んで評価を行ったところ，約7.5倍高速になった．

（2） 応用プログラムの実行速度

本Cobolマシンを使用したシステムでは，演算，移送，条件，データ操作などのCobolの中核ステートメントの処理がホストプロセッサからCobolマシン側へオフロードされることになる．したがって，ここで

表7 ステートメント実行速度比

Table 7 Statement execution speed ratio.

分岐命令の分岐率(%)	汎用計算機との比(倍)
0	2.67
50	2.52
100	2.38

表8 プログラム実行速度比

Table 8 Program execution speed ratio.

処理対象 ＼ プログラム	P_1	P_2	P_3
オフロードされた部分の実行速度比(倍)	3.7	1.8	1.8
オフロード率(%)	98.3	93.6	30.4
プログラム全体の実行速度比(倍)	3.5	1.7	1.2

（汎用計算機との比）

は，オフロードされた部分がCobolマシンで処理された場合の実行速度の向上度を求め，これを用いて，プログラム全体の実行速度向上度をオフロード率（汎用計算機上で実時間によって測定したもの）を基に算出した．

表8 は三つのプログラムについての結果を示す．オフロードされた部分の実行速度は約1.8～3.7倍向上し，プログラム全体としては，約1.2～3.5倍（平均約2.1倍）向上する．

（3） ステートメントの複雑度と実行速度向上度

一般に，実行時に処理しなければならない機能が多いステートメントほど，高級言語マシンの効果は大である．このことをCobolマシンについて明らかにするために，ステートメントの複雑度（汎用計算機のIPF値を用いる）と実行速度の向上度との関係を求めた（**図2**）．とくに，複雑な処理を必要とするデータ操作や乗除算ステートメントの向上度が大きい．GO TO ステートメントでは汎用計算機より悪い原因は，Cobol マシンの GO TO 命令が複雑なアドレッシング機能をサポートできるようになっているために単純な場合に長くなるためである．

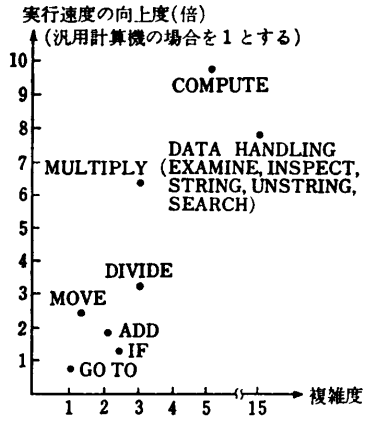

図2 複雑度と実行速度向上度との関係

Fig. 2 Statement execution time improvement.

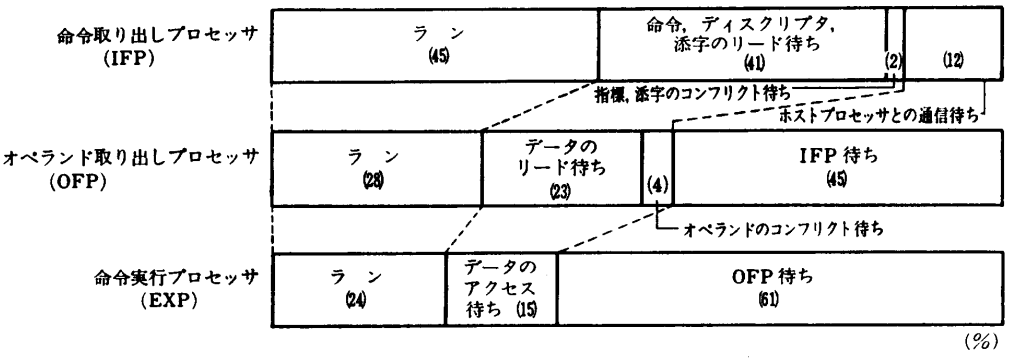

図 3 プロセッサの動作状況
Fig. 3 Processors status.

7. ハードウェアパラメータの分析

マシンアーキテクチャが高度になると，それを効率よく実現するには高級なハードウェアが必要になる．ハードウェアの高度化を進めるときには，プロセッサ構成，先取り方式，ハードウェアとファームウェアのトレードオフなどの点で各種の実現方式が可能である．これらの方式を決定するときには，ユーザプログラムの使用環境に合わせることが重要である．ここでは，Cobol マシンのハードウェア構造を規定するパラメータについて，応用プログラムを実行させてハードウェアモニタを用いて測定した結果を示す．

(1) プロセッサの動作状況

Cobol マシンでは図1に示したように，一つの命令は三つのプロセッサで分割処理される．**図3**は三つのプロセッサの動作状況を示す．

各プロセッサが有効な処理を行っているラン状態の割合が比較的小さいのは，主記憶のアクセス時間がプロセッサのマシンサイクル (200 nsec) の約6倍のためで，このために，メモリリミット状態にあるといえる．また，命令取出しプロセッサのラン状態の割合が他プロセッサより大きいのは，命令取出し処理は通常，他プロセッサとの待合せなしに常に行えること，ディスクリプタの取出しなど比較的複雑な処理が多いためと考えられる．

(2) FIFO キュメモリの使用状況

三つのプロセッサの処理時間のばらつきによって生じるプロセッサの待合せを小さくするために，プロセッサ間に命令用とオペランド用の2種類の FIFO キュメモリ（それぞれ 16 語の容量）を備えている．

表9は FIFO キュの利用状態を一定間隔でサンプリングすることによって求めたそれらの利用状況を示す．たとえば，90%を保証するには，それぞれ，4，8，2，1語でよいことがわかる．表9において，キュ長が0のときは，後段のプロセッサが待っているときと，二つのプロセッサの処理時間がバランスしてキュにたまらない状態の両方を含んでいる．

(3) 先取り命令数

Cobol マシンは汎用大型計算機と比べて高度な命令列の先取りを行っており，たとえば，Perform や

表 9 FIFO キュメモリの使用分布
Table 9 FIFO queue memories usage.

(%)

キュ長 (語)	IFP と OFP 間キュ		OFP と EXP 間キュ	
	命令キュ	オペランドキュ	命令キュ	オペランドキュ
0	38.7	50.5	75.9	89.5
1	26.0	11.5	11.7	5.3
2	18.2	18.6	7.0	0.9
3～4	8.7	7.9	3.6	1.5
5～8	5.3	3.2	1.5	1.7
9～16	3.1	8.3	0.3	1.1

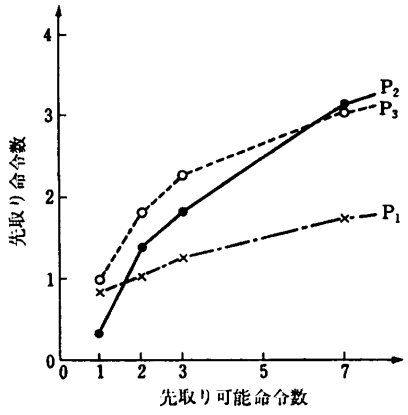

図 4 先取り制御の特性
Fig. 4 Prefetching control characteristics.

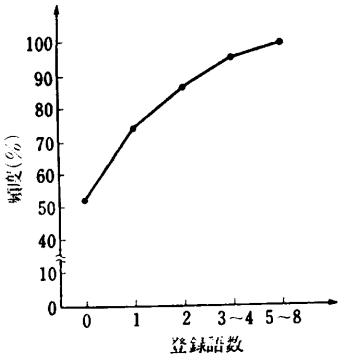

図 5 先取り制御レジスタの使用状況
Fig. 5 Usage of advance control registers.

Search に対してはソースステートメントの動作を反映した先取りを行う．**図4**は，先取り可能命令数（命令取出しプロセッサが命令実行プロセッサより何個先の命令までフェッチすることを許すかを示す値）をパラメータにして，実際に Cobol マシンにおいて何命令先を先取りしているかを実測したものである．図4から，平均2～3命令先をフェッチしていることになる．

（4） 先取り制御部の登録語数

命令やデータを先取りする場合には，演算結果が主記憶に蓄積される前に，古い値が後続命令のために取り込まれる可能性がある．このために，命令取出しプロセッサにおいて，演算結果が主記憶に蓄積されるオペランドは先取り制御部に登録されるが，このために Cobol マシンは8語のレジスタをもっている．**図5**はこのレジスタの動的な使用状況を測定し，累積分布で示したものである．登録の9割を保証するには3語ほどあればよいことがわかる．

8. むすび

Cobol マシンの性能およびアーキテクチャやハードウェアの評価結果について述べた．

Cobol マシンでは，同程度の技術レベルの汎用計算機と比べ，オブジェクトメモリ量はプログラムの命令部について平均約26%減少し，メモリアクセス回数は平均約35%減少し，実行速度は平均約2.1倍向上することが明らかになった．そのおもな理由として，

- Cobol 言語仕様に合った中間言語アーキテクチャを設定したこと
- パイプライン制御方式による三つのプロセッサから成る複合体で構成したこと
- Cobol 言語処理に最適化したハードウェアおよびファームウェアを備えたこと

などが大きいことがわかった．

本 Cobol マシンで達成された性能向上は APL マシンや Lisp マシンと比べると小さい．これは，Cobol で要求される機能が APL や Lisp と比べて単純であり，また，現在開発されている応用プログラムは簡単な Cobol ステートメントを多用しているためである．しかし，図2で示したように，本 Cobol マシンは複雑なステートメントに対して実行速度の向上が高い．このため，ストラクチャード Cobol にみられるような構造化プログラミング機能やオフィスオートメーション分野などで要求される文字列操作機能などの複雑なステートメントが多く使用される環境下では，本 Cobol マシンはさらに高い性能向上を期待できる．

さらに，詳細な評価データに基づく分析の結果，本 Cobol マシンのおもな改良項目として，以下のものが得られた．

1) 条件分岐命令の先取り制御方式の改善
たとえば，分岐履歴情報を用いた先取りを行う．

2) プロセッサ間のデータ転送方式の改善
たとえば，FIFO メモリが空のときには後続プロセッサのレジスタにデータを直接転送する．

最後に，ここで得られた Cobol マシンのハードウェア構造を規定する各種の設計パラメータについての詳細なデータは，今後，Cobol マシンの商品化を進める上で有効であると考えられる．

謝辞 本 Cobol マシンは通産省工業技術院大型プロジェクト「パターン情報処理システム」の一環として開発したもので，本プロジェクトの関係各位に深謝します．また，本 Cobol マシン開発に当たり，研究期間を通じて有益なご指導とご支援を下さった当社のソフトウェア生産技術研究所祢津所長代理，コンピュータシステム研究部三上部長，箱崎課長，および Cobol マシンの開発，評価に参加下さったプロジェクトメンバ，関連会社の担当者に深く感謝します．

参考文献

1) *American National Standard Programming Language COBOL*, X3.23 1974, American National Standard Institute, Inc., (1974).
2) 山本，箱崎，梅村，永井，熊野，中崎：COBOL マシンのアーキテクチャについて，情報処理学会第17回全国大会，pp. 307-308 (1976).
3) 中崎，山本，梅村，箱崎：COBOL マシンのハードウェア構成，情報処理学会第17回会国大会，pp. 309-310 (1976).

4) 北村，市古：ACOS シリーズ 77 NEAC システム 300/400/500 のアーキテクチャ，日本電気技報，No. 114, pp. 6-12 (1975).
5) 山本：COBOL 用高級言語マシントランスレータ，電子通信学会情報・システム部門全国大会，pp. 416 (1979).
6) 山本，横田，中崎：高級言語ステートメントの実行形態を反映した高級言語マシン，情報処理学会第 19 回全国大会，pp. 21-22 (1978).
7) 北村，金森，徳永：高水準言語指向の CPU 性能指標について，電子通信学会電子計算機研究会資料 EC 74-20, pp. 59-69 (1974),

(昭和 56 年 10 月 8 日受付)
(昭和 56 年 12 月 17 日採録)