

## 解説



## FORTRAN†

高橋 延 匡††

## 1. ま え が き

本学会の活動として、日本の電子計算機の草創期におけるソフトウェアの研究開発活動について、現代的な視点から明確にしたいという考え方は大変意義のあることと思う。

たまたま筆者は、昭和32年4月より昭和52年3月まで日立製作所中央研究所において、初期の国産機のソフトウェアの研究開発に携わってきた。この間、草創期であったという幸運に恵まれ、HIPAC 103のFORTRANコンパイラHARP 103や、HITAC 5020モニタや5020 TSSなどの研究開発に没頭することができた。

また、現在研究・教育機関に属する立場から、情報処理の分野に携わる方々に、一つの研究開発の様子を事例として提示することにより、当時の問題点を明らかにすると同時に、何等かの参考になれば幸いであるとの立場からまとめることにした。したがって、このテーマの副題は“Early Activities on Software Projects at Hitachi Central Research Lab.”というのが妥当かも知れない。

また、草創期においては主役はハードウェアでソフトウェアは脇役的存在であったため、発表も少なかった。FORTRANの場合には、さらに企業間の競争も激しかったことから、社外発表がおさえられていた面もあり学会発表は極端に少ない。したがって、とてもそれ等を公平に述べることは不可能であるので、以下、筆者の独断と偏見でまとめたことを、あらかじめお断りしておく。

最後に、FORTRANに関しては、全国大学共同利用の東京大学大型計算機センター納入のHARP 5020について重点を置く方が技術的側面にも興味深い点が多

多あるが、草創期ということで、むしろ、HARP 5020以前に焦点をあてた。したがって、その後のFORTRANのJIS化の問題等、重要な問題にもかかわらず触れていない部分もあることに関してはご寛容いただきたい。

## 2. HIPAC-Mark I から HIPAC 103 の誕生まで

## 2.1 HIPAC-Mark I の誕生

日立製作所におけるデジタル計算機の開発は、昭和31年頃から中央研究所で始まった。その研究開発の動機は、日立電線(株)から送電線の弛張度に関する膨大な量の計算を処理するための装置の作成を考えてほしいという研究依頼がなされたことによる。その計算内容を検討した結果、ストアードプログラム方式のデジタル計算機の開発に着手することになった。

HIPAC-Mark I<sup>†</sup>のハードウェアは高田昇平(現、リンク(株))、嶋田正三(現、法政大)、萱島興三、岩上秀夫等の諸氏が中心となって行われた。論理素子としてパラメトロン、主記憶装置として磁気ドラム(1024語)が採用された。昭和32年12月、はじめてストアードプログラム方式で動き出した。その後直ちに電線の弛張度計算にとりかかった。この問題は、発電所と消費地を結ぶ送電線に加わる荷重(風圧と氷雪による荷重)が最悪条件下でも一定の限界内にあり、しかもどの鉄塔に加わる力もその両側で釣合っている必要があるが、このような条件を満たすためには、個々の鉄塔間の電線がどのような形で、また、どのような力のかかり方で張られればよいかを求めることである。従来、30人位の計算手が図・表などを頼りに2~3カ月カンズメ状態となって計算したとのことである。計算の内容は5次方程式の根および数多くの多項式や有理式の計算の羅列であり、式としてはそれ程困難ではなかったが、HIPAC-Mark Iが固定小数点方式であったため、そのノーマライゼーションにかなりの時間と努力が払われた。計算の速度は1径間(鉄塔と鉄塔の

† Early Activities on FORTRAN in Japan by Nobumasa TAKAHASHI (Faculty of Technology, Tokyo University of Agriculture & Technology).

†† 東京農工大学工学部

間) 当り 1 分 50 秒で、これは熟練した計算手の約 1 日余りの計算分量とのものであった。

当時のプログラミング・グループは嶋田正三主任の下に電気試験所で ETL-Mark II の経験を豊富に持った藤中恵研究員と筆者で構成されていた。上記のプログラミングは藤中氏の指導の下でコーディングを筆者が担当した。この時の最大の問題点は、ハードウェアの熱設計の経験不足から来る信頼性の低さにあった。これは筆者に、ハードウェアは誤動作をするのが当たり前であり、プログラミングは要所要所のチェックが大切であることを身に浸り込ませた。この経験は、その後、ハードウェアの虫取りや、その後のオペレーティングシステムの設計時に、「常に、ここでハードウェアが故障や誤動作を起こしたら、プログラムはどのような動作をするか」という見方を定着させた。

HIPAC-Mark I は、その後 HIPAC 101 A と改良され、昭和 34 年 6 月パリの万博に出品され、その高い信頼性から好評を博した。これは、実装設計の改善に依る所が大きい。その後、商用機としては、昭和 35 年から、主記憶装置を 2 倍に拡張した HIPAC-101 B が開発された。

計算機が動き出すと同時に、中央研究所の内外から数値計算の要求が殺到した。昭和 33 年 4 月からは、氏家一彬氏(現、日立電子(株))、高德嘉子(加藤)氏がプログラムとして参加した。また、原子力コードの開発と FORTRAN の普及に貢献した新井公雄氏もユーザとして計算機室に加わった。同時に高德一恵氏が計算機の保守・管理専任者として加わった。その意味で、中央研究所におけるソフトウェアの研究は、昭和 33 年 4 月から本格的に増強の一途をたどることになった。一方、日立社内の技術計算は、当時の有隣電機や IBM の計算センタを利用していましたが、それ等のユーザも一部この HIPAC Mark I を利用しはじめた。

この数値計算の経験は、筆者にプログラミングのおもしろさを満喫させると同時に、二つの問題点を投げかけた。第一の問題は固定小数点方式のため、式をノーマライズする手間とチェックの手間が大変であり、これを軽減したいということであり、第二の問題はコーディングの手間を軽減したいということであった。

第一の問題の解は浮動小数点方式の演算装置の導入であるが、これはハードウェアのコスト・アップになることから、HIPAC-103 まで実現できなかった。一方、浮動小数点のサブルーチン(インタプリタ方式)は、プログラムは作成されたが、性能上の理由からは

ほとんど使用されなかった。当時は計算速度が比較されるため、プログラムの関心は苦勞して少しでも速い処理プログラムを作成することにあつたためである。また、主記憶装置の容量も少ないことから、ステップ数の短いプログラムを作成することが要求されたためでもある。

第二の問題の解は、ライブラリ・サブルーチンの充実と自動プログラミングの研究の必要性の認識と研究の開始であった。

上記のような状況から、森口繁一先生にご指導をいただくことになった。すなわち、昭和 33 年度に、森口繁一先生、高田勝先生等と MH 委員会という研究会を持ち、以後、昭和 37 年頃までプログラミング全般に亘ってご指導をいただいた。森口研究室からは、三浦大亮氏(現、東レ)、松谷泰行氏(現、新日鉄)、清水留三郎氏(現、東大)、五十嵐滋氏(現、筑波大)等の若手の研究者も加わり、話題は新しいプログラミングの話題から、数値計算における累積誤差の問題、常微分方程式や偏微分方程式の解法の評価の問題、自動プログラミングの問題等、広範囲にわたった。また、高田勝先生には、HIPAC 101 のマニュアル<sup>2)</sup>の書き方についてまでご指導をいただいた。また、清水留三郎氏にはコンパイラの技法に関し、種々の方式についてご指導をいただいた。これ等は日立のプログラミング・グループのレベルアップに大いに寄与した。

## 2.2 自動プログラミングの研究の開始

昭和 33 年、34 年はユーザの数値計算が中心であったが、自動プログラミングの研究への願望が強まって行った。森口先生を中心として、1パス・アセンブラの記号入力ルーチン SIP (Symbolic Input Program) が提案され、電子協に納入された HITAC 301, NEAC 2203 を対象に作成されたことも刺激となった。

ここで、HIPAC Mark I, 101 のアーキテクチャ<sup>1)</sup>について触れておく。当時の日本の計算機システムは大かれ少なかれ、EDSAC の影響を受けていたと言える。EDSAC ではプログラムの入力にイニシャル・インプット・プログラムと称する 40 語のアセンブラの原型のような入力ルーチンがあった。これを用いることにより機械語の 2 進コードや 10 進コードを直接使用せず、Add なら、“A”, Subtract なら“S”と書け、番地部も 10 進整数で相対的に書けるようになっていた。ところが、HIPAC の開発時の設計思想として、“このような入力プログラムは、元来ハードウェアとして実装してしまうべきである”を採用してい

たため、HIPAC 101 A までは入力命令を持たなかった。この点に関しては、ストアードプログラム方式の理解が不足しているのではないかと批判もあったが、なるべく、ソフトウェアをハードウェア化したいというメーカーの基本的な考え方も理解できる。

上記の理由から、HISIP 101 の開発は HIPAC 101 B が商品化され、入力命令が実装されたことにより開始された。昭和 35 年の 2 月に開始され 7 月に完成した。これを契機として、自動プログラミングの研究に力を注ぐことになっていった。当時、日立の創業時代から研究開発にタッチしてこられた初代中央研究所所長で、当時名誉所長の馬場顧問からは、HISIP に関して、以下の含蓄に富むコメントをいただいた。「之は非常にむずかしい仕事也。昔弘法大師の仮名を造れる時と似た仕事。これは何としても使う人の便・不便を十分に聞いてからやれ。これは一遍決めたらそれで良いという事にはなかなかならないらしい。細目ネジの歴史を調べられたし。参考になろう」。このコメントから、計算機言語の選定にあたっては、所内外の意見を聞く習慣が定着した。また、ネジの歴史は、ユーザにとって互換性の問題が如何に重要であることを示唆した。この辺が明治生まれの教育者の真髄ではあるまいか。恐らく、我々の多くは「互換性の問題はどうか考えるか」というような直接的な質問になりそうな気がする。

昭和 35 年 4 月には、自動プログラミングの研究が重視され、筆者の他に、吉村一馬氏（現、日立マイクロコンピュータ・エンジニアリング(株)）、中田育男氏（現、筑波大学）等が入社し加わった。当時の話題は ALGOL と FORTRAN であり、興味の焦点は自ら、そちらに向いて行った。その後、HIPAC 103 が計画され、自動プログラミングの研究グループは、103 を対象にコンパイラの研究開発に向かった。

### 2.3 HIPAC 103

HIPAC 103 は<sup>3)</sup>、当初関西電力(株)において、電力経済負荷配分装置 (ELD) を計画するにあたり、アナログ計算機と組み合わせた、ハイブリッド計算機としての構成を意識して設計が開始された。また、同時に単独に使用して、高度の科学計算にも使用できることという要求から、信頼性を重視して、演算素子としてはパラメトロンを採用し、主記憶装置には磁心記憶装置、補助記憶装置には磁気ド

表-1 HIPAC 103 の主要な仕様

|            |                                                                                              |
|------------|----------------------------------------------------------------------------------------------|
| 形 式:       | プログラム記憶式電子計算機                                                                                |
| チエック方式:    | 入出力部におけるパリティチェック                                                                             |
| 回路方式:      | パラメトロン、トランジスタ、ダイオードによる並列同期方式                                                                 |
| 主要演算素子:    | パラメトロン 約 8,900 個<br>トランジスタ 1,500 本<br>ダイオード 2,500 本<br>真空管 40 本                              |
| 数値表現:      | 2進法、絶対値表示、固定小数点および浮動小数点                                                                      |
| 演算方式:      | 1 $\frac{1}{2}$ アドレス、1語2命令                                                                   |
| 命令の種類:     | 約 110 種類                                                                                     |
| 主記憶装置:     | コアメモリ 1,024 語または 4,096 語、磁気ドラム 8,192 語                                                       |
| 演算速度:      | 加減算 固定 650 $\mu$ s 浮動平均 1.3ms<br>乗算 固定 2.3ms 浮動平均 2.4ms<br>除算 固定 8.1ms 浮動平均 6.9ms            |
| 紙テープ入出力装置: | MTR 500字/分<br>PTR 200字/秒<br>さん孔タイプライタ 500字/分<br>接続可能台数 各 1 台                                 |
| ラインプリンタ:   | タイプホイール式<br>数字、英字 (大、小文字)、記号計 94 種                                                           |
| 1 分間の行数:   | 300 行 (最大)                                                                                   |
| 接続可能台数:    | 1 台                                                                                          |
| 磁気テープ装置:   | テープの種類 幅 $\frac{1}{2}$ インチ、長さ 3,600 フィート<br>記憶容量 700,000 語/リール<br>クロック周波数 12kc<br>接続可能台数 8 台 |
| その他接続可能装置: | AD 変換器 1 チャンネル<br>DA 変換器 1 チャンネル<br>補助磁気ドラム 51,200 語                                         |

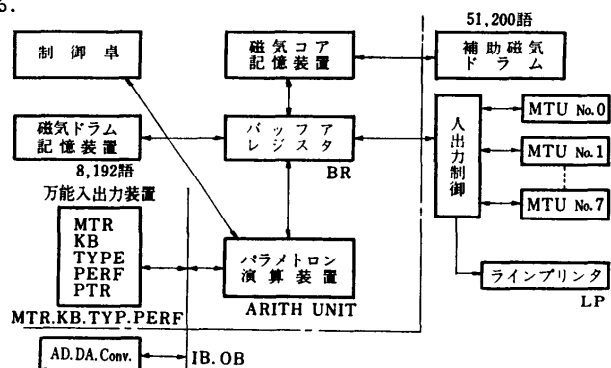


図-1 HIPAC 103 の構成ブロック図

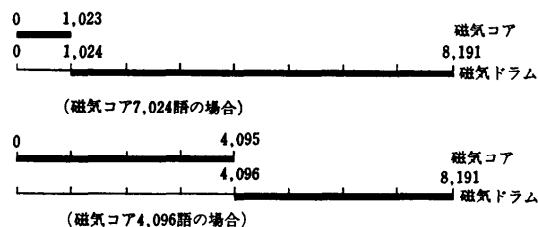


図-2 磁気コアと磁気ドラムのアドレス

ラムを採用した。また、設計目的から、AD、DA 変換装置が接続されている。

また、日立製作所として最初の、国産商用機としても最も早い時機に FORTRAN コンパイラの開発に成功したことから、かなり成功した計算機（販売台数 31 セット<sup>5)</sup>）となった。

ここで、HIPAC 103 について簡単に触れておく。

HIPAC 103 の主要性能は表 1 に示す。その構成上の特徴を以下に述べておく。主記憶装置は、アドレス部が 13 ビット (8192 語) のうち、磁心記憶装置として、1024 語と 4096 語のどちらかを選択できる。また補助記憶装置として、磁気ドラム装置 8192 語とを、図-2 のように組み合わせて使用する。図において、太線で示した部分は、1 語単位で直接アクセスできるが、細線で示した部分は、最大 256 語単位のブロックとして、入出力命令を用いて磁心記憶装置に転送することによって使用可能となる。また、1 命令は 24 ビットで構成されており、1 語に左命令、右命令のように 2 命令が入る。

| 左命令  |      |       | 右命令  |      |       |
|------|------|-------|------|------|-------|
| 命令部  | 修飾部  | 番地部   | 命令部  | 修飾部  | 番地部   |
| 7ビット | 4ビット | 13ビット | 7ビット | 4ビット | 13ビット |

また、数値語は、48 ビットで構成されており、特に浮動小数点は、2 進の絶対値表示であった。

|           | 指数部  | 仮数部   |
|-----------|------|-------|
| 符号<br>ビット | 8ビット | 39ビット |

このアーキテクチャのソフトウェア作成者からみた特徴としては、修飾部が 4 ビットあり、それぞれ、SCC, index 1, index 2, index 3 に対応していて、アドレス部に対し、独立に、多重に修飾ができる点は当初魅力的なものと考えた。一方、1 語 2 命令ということは、アセンブラ (HISIP 103) では、制御構造の負担増につながった。特にプログラムの修正には気をつける必要が生ずる。また、主記憶装置のアクセス・ギャップは、逆にプログラマに工夫の余地を残し、やがて、1 K 語の FORTRAN コンパイラの開発を成功させた動機となった。

また、当時、日立中研のソフトウェアの関連研究の一切が、HIPAC 101 から、HIPAC 103 に切り替えられた。その意味で、HARP 103 を頂点とする HIPAC 103 研究開発プロジェクトと言えよう。人員も増加の一途をたどった。

### 3. HARP 103 の開発と関連研究

昭和 35 年度から昭和 38 年度は、日立中研において、プログラミングの研究の重要性が認識され、多くの研究者が参加した画期的な時期であった。その中心が、HIPAC 103 であり、ソフトウェアが HARP 103 と、HISIP 103, ライブラリ・サブルーチンである。

当時、ソフトウェア研究の研究室長であった嶋田主任研究員は「計算機のプログラミングの研究は、計算機を応用するグループと研究するグループが近い所において、相互のコミュニケーションが無ければいけない」という信念を持っておられ、当時の日立の幹部に「東大医学部と東大病院」という関係になぞられて PR しておられた。これは非常に慧眼であった。

そのような視点から、大略、以下のグループを構成していた。

#### (1) 自動プログラミングの研究

主として、HARP 103, HISIP 103 の開発

#### (2) ライブラリの開発

HIPAC 103 用ライブラリの開発

#### (3) 数値解析の研究および設計計算のコンサルト

数値計算用ライブラリ開発、日立製作所内の設計計算の数学的なコンサルト、および開発

#### (4) 計算機センタの運営・管理

計算機の効率的な運用、保守

また、上記の HIPAC 103 で経験を積んだ嶋田スクールが、その後の HITAC 5020 プロジェクトの母体となっていった。また、HIPAC 103 のソフトウェア開発は、HARP 103 を除き、殆んど、研究者個人の能力に依存していた。その意味で名人芸的なソフトウェア製作の最後に近いフェーズと言えよう。

### 3.1 HAP (Hitachi Automatic Programming)

#### 言語の検討

HIPAC 103 の開発と並行して、昭和 35 年度より、本格的に ALGOL と FORTRAN<sup>6)</sup>に関して、自動プログラミング・グループによって検討が加えられた。その結果、日立独自の言語 “HAP” の仕様を昭和 35 年 12 月に完成させた。この言語の特徴は、ALGOL の美しさを FORTRAN に導入したようなものであった。この言語について、日立の社内の工場や代表的なユーザおよび、コンピュータ事業部に検討を依頼した。また、MH 委員会をはじめ、島内剛一先生 (立教大) や大学の諸先生方にもご意見をお伺いした。その結果、大多数のユーザは好意的であったが、社内の最

大手ユーザである原子力コード（原子力関係の技術計算のソフトウェア）の開発グループからは猛烈な反対意見が出された。すなわち、HAP や ALGOL などまったくナンセンスであり、米国の産業界は FORTRAN が中心である以上 FORTRAN にすべしというものであった。また、東大の高橋英敏先生からは、「日立が ALGOL でもない FORTRAN でもない独立した言語をつくった事は結構なことであるが、一旦、顧客に提供する以上、将来もこの言語の普及と改善に努める積りでないと、顧客は困る」と言われた。又、「HAP を採用しなかったとしても、このような言語仕様を研究することは大変意味があることである」とも言われた（昭和36年2月13日）。

上記の数カ月間にわたる討論や、工場・事業部等の意見を踏まえて、嶋田主任研究員の決断により、日立の自動プログラミング言語は FORTRAN に決定した。またその名称も HARP とした。国産メーカーがいずれもこのような名称を付けた理由は FORTRAN を IBM の商標と考えていたからである。

上記を現代的な視点で再考してみると、

(1) FORTRAN にせよ ALGOL にせよ、非常に明確な意図がある言語であり、それぞれ主張も明快であった。FORTRAN や ALGOL の方言的な言語では、強いユーザを引き込むことはできない。新しい言語は、新しい概念や、明確な応用を意図して生まれる

(2) 言語は、言語設計者やメーカーが育てるものではなく、ユーザが育てるものである。熱烈的なユーザ無しには、どんな優れた言語も普及しない

(3) 当時の研究開発の状況では、追いつけ、追いつけのフェーズであり、技術導入が盛んな時代であり、その意味で、FORTRAN との「互換性」の重視は、当を得た判断であった

と言えよう。ソフトウェアの開発では、特に技術者の希望や好みよりは、管理者の長期的視野に立脚した方針が重要であることを物語っている。

一方、今日の一部のメインフレームに見られる「主流の言語」を選択しそれをやってさえいれば良いという風潮も、このような点に立脚している可能性が高い。

### 3.2 HARP 103 (4K 用) の開発

HIPAC 103 用の FORTRAN コンパイラのインプリメンテーションは、吉村、中田、筆者の3人で、昭和36年3月からその作業に入った。当初の分担は、日本的な年功序列の伝統にもとづき、数式処理を吉村

が、入出力関係を中田が、制御関連を筆者が担当した。HARP 103 の仕様および、IBM 社の同種の FORTRAN との比較は表-2<sup>9</sup>の通りである。仕様の作成では、IBM 650 FORTRAN と 704 FORTRAN の中間を行く程度のものでした。

HARP 103 は、システム・プログラムとしては初めて分担開発（3人）によって作られた。作業量の見通しを立てて分担するというものではなく機能による分割しか分割の方法はなかった。約10日に1回、システム打合せ会を開いて、スケジュールや分担の調整を行いながら進めた。8月末、ほぼ全体の見通しがついた時点で、再度、分担の見直しが行われた。当時、HIPAC 103 は標準構成（≒最小構成）として、磁心記憶装置は1K語であった。しかし、上記の HARP 103 は磁心記憶装置4K語のものを対象として設計されていた。営業からの強い要請で HARP 103 (1K 語用) の開発の必要性が生じた。吉村は主として、1K 語用の HARP の開発に重点を移行した。筆者は、新たに、磁気テープ関係、ラインプリンタ関係、FORMAT 文、SUBROUTINE 文関係を、中田が、全体のデバッグ、新たに加わった高須昭輔氏がマニュアルを担当することになった。

昭和37年3月には、HIPAC 103 が中研に搬入され、早速全体のデバッグ作業が開始された。また、5月21日から所内で1週間の講習会を行い5月28日から実習というスケジュールが4月20日に決定された。それからの日程は、テストプログラムの開発とシステムのデバッグに費やされた。5月にはデバッグも完了し、実習を待つばかりとなった。実習は、昭和38年度大学卒新入社員33人と一般の研究所所員43名、計76人について、前者は3日間の講習、後者は2日半の講習を経て、数表作成、大小順の並べ替え、求根問題等を実施した。その結果、HARP のユーザは平均1プログラム（1問題）当たり1カ弱のプログラム・エラーを起した。エラーの起した比率は新人所員と一般の研究員との間には、まったく差はなかった。またエラーの要因の大部分は

- (i) 入出力ステートメントと FORMAT ステートメントのまちがい
- (ii) 添字付変数に関する配列宣言がない
- (iii) 文法に無い制御ステートメントを書いた
- (iv) 型の混合表現がある
- (v) DO ループの制御のまちがいであった。

表-2 HARP 103 と FORTRAN のステートメントごとの制限事項 (文献4)より転載)

|                                                 | HARP 103                                                                                                                                  | IBM 650                                                                                    | IBM 704                                                                                                                                                                                                                          | IBM 7070                                                                                      | IBM 709                                                                                                     |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 定数                                              | 数<br>区間 $(-a, a)$ 内の整数<br>ここで $a = 2^{17} \div 10^{16}$                                                                                   | $a = 10^{10}$                                                                              | $a = 2^{17}$                                                                                                                                                                                                                     | $a = 10^{10}$                                                                                 | $a = 2^{17}$                                                                                                |
|                                                 | 実数形 /<br>0 および区間 $[b^{-1}, b]$ ,<br>$[-b, -b^{-1}]$ 内の数が有効<br>数字 11 けたで表わされ<br>る。ここで $b = 10^{38}$                                        | 有効数字 8 けた<br>$b = 10^{30}$                                                                 | 有効数字 8 けた<br>$b = 10^{38}$                                                                                                                                                                                                       | 0 および $[10^{-90}, 10^{+8}]$<br>$[-10^{+9}, -10^{-90}]$<br>有効数字 8 けた                           | 有効数字 8 けた<br>$b = 10^{38}$                                                                                  |
|                                                 | 517 個以上の定数を使用して<br>はいけない。                                                                                                                 | 一つのステートメントに 9 個<br>以上の異なる定数を書いて<br>はいけない。                                                  | 整数形では異なる定数は 10 個<br>以下のこと。実数形では異なる<br>定数は 450 個以下のこと。<br>(ただし符号のみ異なるものは<br>同じものと計算する)                                                                                                                                            | 一つのステートメントで使用<br>できる異なる定数の個数は 40<br>以下のこと。<br>(ただし符号のみ異なるものは<br>同じものとみる)                      | 整数形では異なる定数は 100<br>個以下のこと。実数形では異なる<br>定数は一つのステートメントで<br>50 個以下で全体で 450<br>個以下のこと。(ただし符号の<br>み異なるものは同じものとみる) |
| 変数                                              | 変数の個数は 5 文字以下のこ<br>と。<br>添字付変数でないときは関数<br>名に使用したのを使ってよい                                                                                   | 103 と同じ                                                                                    | 6 文字以下のこと。<br>使用する関数名の最後の F を<br>除いたものと一致するものを<br>使用してはいけない。                                                                                                                                                                     | 103 と同じ                                                                                       | 6 文字以下のこと                                                                                                   |
|                                                 | ステートメントで関数を定義<br>するときその関数を数式で<br>使用する前に書かねばならな<br>い。                                                                                      | 103 と同じ                                                                                    |                                                                                                                                                                                                                                  | 704 と同じ                                                                                       | 704 と同じ                                                                                                     |
|                                                 | DO の範囲内で関数を定義し<br>てはいけない。                                                                                                                 |                                                                                            |                                                                                                                                                                                                                                  |                                                                                               |                                                                                                             |
|                                                 |                                                                                                                                           |                                                                                            |                                                                                                                                                                                                                                  | ユーザーの関数サブルーチン<br>で使用できるのは 20 個以内                                                              |                                                                                                             |
|                                                 |                                                                                                                                           |                                                                                            | 添字付変数で 4 つ以上の文字<br>よりなるものは最後の F とな<br>ってはいけない。                                                                                                                                                                                   | 添字付変数は最後が F であ<br>ってはいけない (103 と同じ)                                                           | 4 文字以下よりなる変数に添<br>字を付けるときは最後の文字<br>が F であってはいけない。                                                           |
| 添字                                              | 二次元まで<br>添字の許される形は V を数<br>数型変数、C、C' を整数型<br>変数とすると<br>V<br>C<br>V+C'<br>V-C'<br>C*V<br>C*V+C'<br>C*V-C'                                  | 二次元まで<br>HARP 103 と同じ                                                                      | 三次元まで<br>HARP 103 と同じ<br>ただし C' は 99 以下のこ<br>と                                                                                                                                                                                   | 二次元まで<br>HARP 103 と同じ                                                                         | 三次元まで<br>HARP 103 と同じ                                                                                       |
|                                                 | 添字は 8192 を法とする。                                                                                                                           | 不明<br>添字の個は 3 けた以内のこと                                                                      | 2 <sup>10</sup> を法とする。                                                                                                                                                                                                           | 10 <sup>4</sup> を法とする。添字の個は<br>正または 0 のこと                                                     | 2 <sup>10</sup> を法とする。                                                                                      |
| 変数の個数についての制限                                    | X: 添字のつかない変数の個<br>数<br>Y: 添字のついた変数の個数<br>$X + 2Y \leq 512$                                                                                | X に制限はない<br>$Y \leq 20$                                                                    | 添字付変数を 1,500 個以上使<br>用してはいけない。<br>添字のない変数型変数の総数<br>は右辺全体で 750 個以下左辺<br>全体で 500 個以下のこと                                                                                                                                            | 添字の数は一つのステート<br>メントで 64 個以下であること。<br>$X + 2Y \leq 153$                                        | 704 と同じ                                                                                                     |
| 関数<br>(built-in あるいは一つの<br>ステートメントで定義<br>されたもの) | 関数名の文字について<br>変数型のは 7 文字以下<br>実数型のは 6 文字以下                                                                                                | 4 または 5 文字                                                                                 | 4 ~ 7 文字                                                                                                                                                                                                                         | 2 ~ 5 文字                                                                                      | 4 ~ 7 文字                                                                                                    |
|                                                 | 組み込みの関数<br>1) ABSF<br>2) XABSF<br>3) FLOATF<br>4) XFIXF<br>5) LOGF<br>6) LOGEF<br>7) EXPF<br>8) EXPEF<br>9) SINF<br>10) COSF<br>11) SQRTF | 1) ABSF<br>2) XABSF<br>3) FLOATF<br>4) XFIXF<br>5) LOGF<br>6) EXPF<br>7) LOGEF<br>8) EXPEF | 1) ABSF<br>2) XABSF<br>3) INTF<br>4) XINTF<br>5) MODF<br>6) XMODF<br>7) MAXOF<br>8) MAX1F<br>9) XMAXOF<br>10) XMAX1F<br>11) MINOF<br>12) MIN1F<br>13) XMINOF<br>14) XMIN1F<br>15) FLOATF<br>16) XFIXF<br>17) SIGNF<br>18) XSIGNF | 1) LOGF<br>2) LOGEF<br>3) EXPEF<br>4) EXPF<br>ほかに<br>ABSF<br>XABSF が Open サブ<br>ルーチンとしてはっている。 | 704 で組み込まれているもの<br>のほかに<br>19) DIMF<br>20) XDIMF                                                            |
|                                                 | 組み込みの関数はすべて<br>Closed サブルーチンである。                                                                                                          |                                                                                            | 組み込みの関数はすべて<br>Open サブルーチンである。                                                                                                                                                                                                   | Closed サブルーチンである。                                                                             | 組み込みの関数はすべて<br>Open サブルーチンである。                                                                              |

表-2 (つづき)

|                                                                | HARP 103                                                                                                                                                                                                                                           | IBM 650                                                                                                               | IBM 704                                                                                                                                     | IBM 7070                                                                                      | IBM 709                                                                                                                          |
|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
|                                                                | 一つのステートメントで定義するとき、変数の個数は7まで、定義できる個数は20以内                                                                                                                                                                                                           | 一つのステートメントで関数を定義することはできない。                                                                                            | 50 以内                                                                                                                                       |                                                                                               | 特になし。<br>35 以内                                                                                                                   |
| ステートメント・ナンバー                                                   | 大 き さ<br>8191以下の自然数<br>個 数<br>256 個以下                                                                                                                                                                                                              | 9999以下の自然数<br>Symbol<br>Table の個数 300 以下                                                                              | 32768 以下の自然数<br>1,500 個以下                                                                                                                   | 9999以下の自然数                                                                                    | 32768 以下の自然数<br>750 個以下                                                                                                          |
| GO TO                                                          | 制限なし                                                                                                                                                                                                                                               | computed<br>GO TO の総数は25以内のこと。                                                                                        | (1) computed<br>GO TO と assigned<br>GO TO に使用するステートメント・ナンバーは 250 個を越えないこと。<br>(2) ASSIGN 型<br>IF 型, GO TO 型のステートメントの総数は 300 以下のこと (709 と同じ) | 不 明                                                                                           | (1) computed<br>GO TO と assigned<br>GO TO に使用するステートメント・ナンバーは 250 個を越えてはいけ<br>(2) ASSIGN 型<br>IF 型 GO TO 型のステートメントの総数が 300 以下のこと。 |
| DO                                                             | DO の個数<br>制限なし<br><br>DO の深さ 15 以内<br>DO の変数はループの途中で DO から出るときはそのときの値を持っているが、ループを全部終えて出るときは必ずしもそのときの値をもっていない。<br>DO の変数の値を DO の内部で変えること。また DO の変数の初期値、終りの値、きざみ幅を DO の内部で変えることは許されない。<br>DO の最後は飛越しのステートメントであってはならない。<br><br>DO の内部に関数を用いた数式を書いてよい。 | DO の個数<br>制限なし<br><br>4 以内<br><br>IBM 709 と同じ<br><br>103 と同じ                                                          | DO の個数<br>150 以内<br>(I/O での List における対応するものを含む)<br><br>50 以内<br>103 と同じ<br><br>103 と同じ<br><br>103 と同じ                                       | 27 以下<br>103 と同じ<br>(ただしそれは添字として使用のときであって、普通の変数として使用するときとは別個参照)<br><br>103 と同じ<br><br>103 と同じ | DO の個数<br>704 と同じ<br><br>50 以内<br>103 と同じ<br><br>103 と同じ<br>DO のはじめのステートメントに実行しないステートメントを書いてはいけない。<br>103 と同じ<br><br>103 と同じ     |
| PAUSE n                                                        | n は 17777 以下の 8 進法正整数または 0。n はオーダレジスタの右アドレス部に表わされる。                                                                                                                                                                                                | n は 1999 以下の正整数である。                                                                                                   | n は 77777 以下の 8 進法正整数。                                                                                                                      | n は 10 <sup>8</sup> 以下の符号のつかない整数。                                                             | 77777 以下の 8 進法正整数または 0。n はストレージレジスタのアドレス部に表わされる。                                                                                 |
| STOP n                                                         | n は PAUSE の n と同じ、表現される所も同じ<br>個数についての制限なし                                                                                                                                                                                                         | n は 9999 以下の正整数または 0                                                                                                  | STOP の総数は 300 個以下のこと                                                                                                                        | n は 10 <sup>8</sup> 以下の符号のつかない整数                                                              | PAUSE n と同じ<br>STOP ステートメントの総数は 300 以下であること。                                                                                     |
| END                                                            | END (I <sub>1</sub> , I <sub>2</sub> , I <sub>3</sub> )<br>I <sub>1</sub> =0 のとき、COMPILE のとき SWITCH I <sub>1</sub> を OFF にする。<br>I <sub>1</sub> =1 のとき、COMPILE のとき I <sub>1</sub> を ON にする。<br>I <sub>1</sub> =2 のときは何もしない。                        | END の形しかない。意味は 103 と同じ                                                                                                | 709 と同じ                                                                                                                                     | END (I <sub>1</sub> , I <sub>2</sub> , ..., I <sub>5</sub> )                                  | i=1, 2, ..., 5<br>外は 103 と同じ。<br>ただしそれによる処理は異なる。                                                                                 |
| 一つのステートメントの長さ                                                  | 200 字以内                                                                                                                                                                                                                                            | 125 字以内 (Sp は含まず)                                                                                                     | 660 字以内                                                                                                                                     | 660 字以内 (Sp も含む)<br>カードで 10 枚以内のこと                                                            | 750 字以内                                                                                                                          |
| HARP 103 に無い制御ステートメントまたは HARP 103 にあるが、その他の FORTRAN にないステートメント |                                                                                                                                                                                                                                                    | IF (SENSE LIGHT)<br>#1, #2<br>IF ACCUMULATOR<br>OVERFLOW<br>#1, #2<br>IF QUOTIENT<br>OVERFLOW<br>#1, #2<br>は 650 にない。 | ASSIGN i TO n<br>GO TO n, (#1, #2, ..., #m)<br>IF (SENSE SWITCH)<br>#1, #2<br>IF DIVIDE CHECK<br>#1, #2                                     | IF DIVIDE CHECK                                                                               | 704 と同じ                                                                                                                          |
| READ                                                           | READ #, list<br>紙テープ                                                                                                                                                                                                                               | 103 と同じ                                                                                                               | カ ー ド<br>あとは 103 と同じ                                                                                                                        | カ ー ド<br>あとは 103 と同じ                                                                          | 704 と同じ                                                                                                                          |

表-2 (つづき)

|                                        | HARP 103                       | IBM 650 | IBM 704                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | IBM 7070                                                                                                                                                                                                                                                                                                                   | IBM 709         |
|----------------------------------------|--------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| LIST                                   | 行ごと (フルマトリックスを<br>与えるとき)       | 列ごと     | 列ごと                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 列ごと                                                                                                                                                                                                                                                                                                                        | 列ごと             |
| TYPE                                   | TYPE <i>n</i> , list<br>頁式プリント | なし      | なし                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | HARP<br>103 と同じ                                                                                                                                                                                                                                                                                                            | なし              |
| PRINT                                  |                                | なし      | 103 と同じ                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | HARP<br>103 と同じ                                                                                                                                                                                                                                                                                                            | HARP<br>103 と同じ |
| FORMAT                                 |                                | なし      | FORMAT のカッコの中の<br>字数はそれを6で割ったとき<br>の商を切り上げたものが 714<br>をこえてはいけない。                                                                                                                                                                                                                                                                                                                                                                                                                                             | HARP<br>103 と同じ                                                                                                                                                                                                                                                                                                            | HARP<br>103 と同じ |
| HARP 103 以外のもの<br>にあるもの (制御ステートメントを除く) |                                |         | 1) READ INPUT<br>TAPE <i>i</i> , <i>n</i> , List<br>2) PUNCH <i>n</i> , List<br>3) WRITE OUTPUT<br>TAPE <i>i</i> , <i>n</i> , List<br>4) READ TAPE <i>i</i> , List<br>5) READ DRUM <i>i</i> , <i>j</i> ,<br>List<br>6) WRITE TAPE <i>i</i> ,<br>List<br>7) WRITE DRUM <i>i</i> ,<br><i>j</i> , List<br>8) END FILE <i>i</i><br>9) REWIND <i>i</i><br>10) BACK SPACE <i>i</i><br>11) EQUIVALENCE<br>12) FREQUENCY<br>13) CALL name<br>14) COMMON<br>15) SUBROUTINE<br>name<br>16) FUNCTION name<br>17) RETURN | 1) READ INPUT<br>TAPE <i>i</i> , <i>n</i> , List<br>2) PUNCH <i>n</i> , List<br>3) WRITE OUTPUT<br>TAPE <i>i</i> , <i>n</i> , List<br>4) READ TAPE <i>i</i> , List<br>5) WRITE TAPE<br><i>i</i> List<br>6) END FILE <i>i</i><br>7) REWIND <i>i</i><br>8) BACK SPACE <i>i</i><br>9) EQUIVALENCE<br>10) TYPE <i>n</i> , List | 704 と同じ         |

しかし、この最初の講習会は HARP 103 のインプリメンタにとっても、非常に有益であった。当時は、大学で FORTRAN など習った人は殆んど皆無であり、まったく予想もしないプログラムが現われ、コンパイラの弱点や虫のたたき出しができ、信頼性を向上させることができた。

次に、IBM の社内ユーザの使用結果であるが、IBM の FORTRAN を用いた計算結果と HARP の結果が異なるから、お前の HARP は虫がいるという強い信念を持った苦情があった。その殆んどが、変数の初期値設定に関するものであった。HARP 103 では、初期値をゼロに強制的に初期化していたが、IBM の FORTRAN では不定であり、実はユーザ側の誤りであった。現在では、ISO や JIS では初期値は不定となっているので、値が不定のまま変数の値を使用した場合にはエラーメッセージを出すようにコンパイラを作っておけば良い。

以上の経験から、システム・プログラムはエラーに対して強くないと、設計者は永久にメンテナンスに追われてしまうことを学んだ。この経験を踏まえて、HIPAC 101 を用いた HARP 103 の診断プログラムの開発も新井全勝氏、加藤正道氏によって直ちに行われた。また、後の HITAC 5020 のソフトウェアの開

発には、「ユーザのデバッグの手間を如何に軽減するか」、「ユーザの誤使用に対してどれだけ強いシステムを作れるか」が重要なテーマになり、前者に対しては HARP 5020 のデバッグ機能の強化や、後者に対してはモニタのプロテクション機能とスーパーバイザ・モードの導入という結果を生んだ。

### 3.3 HARP 103 の仕様と工夫したこと

前述の通り、HARP 103 の仕様は IBM 650 と IBM 704 の中間を狙った (表-2 参照)。特に、計算機の規模を考えると、仕様が大きく立派になった結果ユーザが使用できる磁心記憶装置が少なくなってしまう。そこで、IBM 650 FORTRAN との互換性には十分注意することとし、それ以外の機能ではデバッグや HISIP 103 のサブルーチンの活用のための追加程度にとどめた。この方針は、当時としては正解であったと思う。

IBM には無くて、HARP 103 の方に追加したステートメントは<sup>7), 8)</sup>次のようなものである。

#### (1) 磁心記憶装置と磁気ドラム装置との間の転送

二次記憶装置としての磁気ドラム装置を有効に活用するために DIMENSION ステートメントと本質的には同じである ARRAY ステートメントを設けた。そのちがいは、ARRAY ステートメントでは、記憶場



所が磁気ドラム装置の方にとられる点である。また、後述の HARP 1 K 用では磁心記憶装置と磁気ドラム装置の間におけるデータのブロック転送を行う転送ステートメントを設けた。

#### TRANSFER, A→B

のように使用した。

#### (2) 入出力ステートメントとデバッグング

プログラムのデバッグングを助けるために中間結果を印字させたい場合がしばしばある。そのために TYPE 又は PRINT ステートメントを各所に挿入しておいて、中間結果を印字させ、誤りを正した後は、このステートメントを無視したい。この目的で、以下の機能を用意した。

(A) ソースプログラムを読み込む際に、コンソール switch 4 が on のときは、ステートメント番号のついていない TYPE, PRINT ステートメントを読み飛ばす。

(B) オブジェクト・プログラムを実行する際に、switch 4 が on のときは、ステートメント番号のついていない TYPE, PRINT ステートメントを無視する。

### 3.4 HISIP 103 とライブラリの開発

昭和36年度には、新たに加わった高須昭輔氏が中心となり、アセンブラ HISIP 103 の開発を担当した。また、ライブラリの開発には、その標準化を含めて多数の協力を得た。ライブラリの取りまとめ、登録、標準的な仕様書などの作成は氏家一彬氏が担当したが、所外からは高橋道子氏（立教大）に開発の協力をいただいた。その他、数値解析の研究グループからも、新谷尚義氏（現、広島大）、渡部敏氏（現、山形大）、清水敬子（多田）氏（現、東京農工大）、宮南和子（矢野）氏等が各種のライブラリを開発した。また、殆んどの新入達もライブラリの開発から手掛けて、それぞれの専門の研究へと移行して行った。

### 3.5 HARP 103 (1 K 語用)

HARP 103 (4 K 用) は昭和37年6月から実用に供したが、殆んどのユーザが磁心記憶装置として、1 K 語しか実装していなかったもので、1 K 用を作成するのが急務であった。

昭和36年、昭和37年は、自動プログラミングの研究部隊が大幅に増員された年でもあった。特に、HARP 103 (1 K 用) は初版のものが思わしくなかったため、全面的に作成し直した。これは吉村一馬氏の下で、袴田史郎氏（現、立命館大）が中心となり、入出

力関連は稲富彬氏（現、明治大）、池田隆一氏（現、鹿山工業）等によって開発された。特に改訂版の1 K 用 HARP は主メモリが少ないにもかかわらず4 K 用 HARP と殆んど同じ仕様で性能的にもひけをとらない出来栄えのものであった。したがって、外部の多くのユーザは、実際には改訂版の HARP (1 K 用)<sup>9)</sup>を使用した。

このコンパイラは、一次メモリと二次メモリとを如何に巧妙に入れ替えながら実施するかに焦点が置かれていた。袴田氏によれば、「HARP 103 (1 K 用) の成功の原因は、1 K 用を作るんだという決断そのものにあったと思う。作ると決まれば工夫はつくものだ」と淡々と語っていたのが印象的であった。

### 3.6 HARP 103 モニタの開発とセンターの運営

センターの運営は、HIPAC 101 以来、高德一恵氏を中心に行われてきた。HIPAC 101 は信頼性が高かったが、磁気ドラム記憶装置を主記憶装置としていたため、速度が遅く、終夜運転になることが多かった。そこで、計算機に安全警報装置をつけてはどうかと考え無人運転装置を開発し、昭和35年8月より無人運転を行っていた。これは、当時としては、保守員付きの計算機の運転の常識からすれば画期的なことであった。

HIPAC 103 の導入と、HARP 103 の開発に伴い、昭和37年6月に所内のえらばれた研究員約100名を対象に HARP の講習会等の教育を行い、同年7月より、従来の制度、つまりプログラムの作成を計算センターに依頼する制度を改め、ユーザ自身がプログラムを作製するように変え、運営もオープンショップ制に改めた。ところが、数カ月運営してみると、以下のことが判明した。

(1) 一般ユーザは、すべて HARP 103 を利用した。HISIP 103 を利用したのはシステム・プログラムとライブラリの開発だけであった。

(2) 磁心記憶装置にある HARP のシステム・プログラムは、1つのソース・プログラムの完了の際にオブジェクト・プログラムによって書き換えられるので、次のコンパイルを始める前にロードし直す必要がある。この操作は、磁気テープを使用しても1～2分かかるし、結局、ユーザがオペレータに尋ねることが多くオペレータが介在することになる。

(3) ユーザのプログラムには、文法ミスがしばしば見出され、コンパイル完了までは、計算センター側のコンサルテーションに相当の時間をとられる。

そこで、HARP 103 の開発グループの協力を得て、高德一恵氏、吉田郁三氏が中心となり、HARP 103 用モニタ<sup>7)</sup>の開発を行った。HARP 103 モニタの機能は、

(1) コンパイルの途中の文法ミスや計算途中のミス (例えば、 $\sqrt{x}$  で  $x < 0$  のときなど) が発生した場合に、そのジョブをとばして次のジョブに移ること

(2) オブジェクト・プログラムを直ちに実行するか、あるいは、一旦磁気テープ記憶装置に書き込むかの決定および実行

(3) 磁気テープ装置から、指定したオブジェクト・プログラムを計算機の所定の番地にロードすること

(4) 主記憶装置の指定した領域を、浮動小数点表示でプリントすること

を満足させることに重点が置かれた。また、上記を実現するために、HARP 103 のシステム・プログラムの格納用と、オブジェクトプログラムの格納用に 2 台の磁気テープ装置を用いた。また、モニタ・ランが止まらないように HARP 103 の STOP ステートメントの変更や END ステートメントの変更などを行った。

上記の開発は昭和 37 年末に完了し、昭和 38 年からはモニタ・モードで運営が行われ、昭和 39 年にはさらにモニタ自身も改良が加えられた。

### 3.7 HARP 101

HARP 101 は、HIPAC 101 のために開発された科学技術計算用に開発されたコンパイラである。計算機自体が非常に小さいため、FORTRAN や ALGOL 言語をそのまま採用するのは無理ということもあり、既存の言語とは異なったものになっている。どちらかと言うと ALGOL に近いものである。この開発は、HARP 103 の開発と同時期にあったということもあり、島内剛一氏 (立教大) をお願いしたものである。言語仕様も簡潔で、特徴としては、ステートメントの略記法を持っている点であろう。一例をあげれば、飛び越し文では

```

は  go to L; (Lは飛び越し先のラベル)
    → L;

```

と略記しても良い。当時としては、紙テープベースということもあり、打鍵数を減らす点でも、また慣れてくるに従い、全体がコンパクトになり見易く書き易くなるという点で合理的な設計であった。このような言語設計の手法は TSS 用言語としても参考になる点を多く含んでいる。しかし残念ながら、FORTRAN と ALGOL の潮流の中では、後続機種での HARP 101

言語のコンパイラは見送らざるを得なかった。計算機言語の普及は、結局メーカの努力もさることながら、多くのユーザの強い要求や支持がなければなかなか発展、継続ができないことを意味している。

## 4. 国産他社の状況

昭和 30 年から 40 年にかけてが、恐らく自動プログラミングについて、国産各社がその開発に苦心した時代であった。しかしながら、学会等での発表は、各社とも商売上の利害がからんでいたため、その内容等は極く限られていたし、実用に用いられたか、研究試作に終わったのかは判断に苦しむ所である。

このような状況の下で、昭和 40 年 3 月、電子通信学会より「計算機用言語とプログラミング」という同学会東京支部編の著作<sup>10)</sup>が刊行されている。第 4 章に田村康男先生 (早稲田大) により「FORTRAN と ALGOL」と題して 42 頁にわたって詳細にまとめられている。表-3 は上記からの転載で 1964 年 5 月現在の国産の FORTRAN コンパイラについてまとめたものである。

以下、学会発表も 勘案して、各社の最も初期の FORTRAN コンパイラに関して述べる。

### 4.1 2203 NARC コンパイラについて

当時の国産電子計算機のベストセラーの一つであった日本電気の NEAC 2203<sup>11)</sup>用に開発された科学計算用のコンパイラが 2203 NARC<sup>12)</sup>である。これは、昭和 36 年度情報処理学会全国大会で古山良二氏、草鹿庸次郎氏によって発表されている。NARC の仕様は FORTRAN や ALGOL の影響を当然受けてはいるが、NEAC 2203 が、主記憶装置として、2040 語の磁気ドラム記憶装置を持ち、入出力装置としては光電式読取機とテープさん孔タイプライタという単純な構成であったため、当然多くは望めないし制限も多い。仕様としては、ARRAY, SWITCH, SUBR の宣言文と、数式、GO TO, IF, FOR, READ, PRINT, HALT, CALL, BP, 無効, STOP ステートメントから成る。コンパイラは 2 パス (ソースプログラムは 2 回読み込まれる) で、パス 1 (約 1100 語、命令約 2200) で、変数名表、定数表の作成、番地の割り当て、ソースプログラムの文法のチェックを行い、パス 2 (約 1300 語、命令数約 2600) でオブジェクトプログラムを作成し、紙テープに出力するというものであった。

NARC 自身は、あくまで NEAC 2203 という環境に依存したコンパイラであり、発展性は望むべくもな

表-3 国産機の FORTRAN コンパイラ (1964年5月現在) (文献10)より転載)

| System Name              | rogrammer                          | Machine       | System Type | Condition     | Capacity                    | Procedure & processor size                                                                                                                                                                        | completed  |
|--------------------------|------------------------------------|---------------|-------------|---------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| HARP 103                 | Hitachi Ltd.                       | HIPAC-103     | FORTRAN     | in use        | 12 K w 48 bit 2 instr.      | 1. source program c.→object program (paper tape)→execute<br>processor size 3,540 word                                                                                                             | May 1962   |
| FACOM 222 FAST (CDT 402) | Fujitsu Ltd.                       | FACOM-222     | FORTRAN     | in use        | 4 K w sign+12 dig 2 instr.  | 1. source program c.→object program execute<br>processor size 12,130 word                                                                                                                         | July 1963  |
| OKI-ART                  | Oki Electric Industry Co.          | OKITAC-5090   | FORTRAN     | in use        | 4 K w sign+12 dig 2 instr.  | 1. source program (card) c.→symbolic language (punched card)<br>2. symbolic language (punched card)a.→object program (punched card)<br>3. object program→execute<br>processor size 10,500 word    | Dec. 1963  |
| OKI-ART-M                | Oki Electric Industry Co.          | OKITAC-5090 M | FORTRAN     | in use        | 4 K w sign+12 dig 2 instr.  | 1. source program (card) c.→assembler language (in magnetic tape) a.<br>object program (in magnetic tape)→execute<br>processor size 10,500 word                                                   | April 1964 |
| FACOM 222 FAST (CDT 813) | Fujitsu Ltd.                       | FACOM 222     | FORTRAN     | in use        | 8 K 10 K w sign+12 dig      | 1. source program c.→symbolic language a.→object program→execute<br>processor size 12,260 word                                                                                                    | April 1964 |
| FACOM 231 FAST           | Fujitsu                            | FACOM 231     | FORTRAN     | flow-charting | 32 K dig                    | 1. source program (paper tape)c.→symbolic language (in magnetic tape) a.→object program→execute<br>processor size 55 K digit                                                                      | Dec. 1964  |
|                          | M. Moriya                          | HITAC-5020    | FORTRAN IV  | coding        | 8192 w 32 bit               | 1. source program c.→assembler language a.→object program→execute                                                                                                                                 | Aug. 1964  |
| HARP 5020                | Hitachi Ltd.                       | HITAC-5020    | FORTRAN IV  | coding        | 16 K-65 K w 32 bit          | 1. source program c.→loader language<br>object program→execute                                                                                                                                    | Dec. 1964  |
| NEAC-2206 NARC           | Nippon Electric Co.                | NEAC-2206     | FORTRAN     | in use        | 4 K w sign+12 dig           | 1. source program c.→assembly language (in magnetic tape) a.→object program (in magnetic tape and core memory)→execute                                                                            | April 1964 |
| NEAC-2230 NARC-11        | Nippon Electric Co.                | NEAC-2230     | FORTRAN     | in use        | 2,400 w sign+12 dig         | 1. source program c.→1 passing language (in memory)<br>2. source program c.→2 passing language (in memory)<br>3. object program→execute<br>processor size 4,500 word                              | March 1963 |
| NEAC-2230 NARC-11        | Nippon Electric Co.                | NEAC-2230     | FORTRAN     | debugging     | 2,400 w sign+12 dig         | 1. source program c.→object program (in magnetic tape and core memory)<br>processor size 8,000 word                                                                                               | June 1964  |
| NEAC-2203 NARC           | Nippon Electric Co.                | NEAC-2203     | FORTRAN     | in use        | 2040 w sign+12 dig 2 instr. | 1. source program (paper tape) c.→object program (punched paper tape)<br>2. object program→execute<br>processor size 2,400 w                                                                      | June 1961  |
| MUSE                     | Mitsubishi Electric Mfg. Co., Ltd. | MELCOM 1101 F | FORTRAN     | in use        | 4096 w 33 bit               | 1. source program (paper tape) c.→symbolic language (punched paper tape)<br>2. symbolic language a.→object program (punched paper tape)<br>3. object program→execute<br>processor size 8,000 word | Feb. 1963  |
| EASE                     | Tamura Kidera Masegi (Waseda Univ) | NEAC-2203     | FORTRAN     | in use        | 2040 w                      | 1. source program (paper tape) c.→object program (punched paper tape)<br>2. object program→execute<br>processor size 2,000 word                                                                   | March 1962 |
|                          | Shimauchi (Rikkyo Univ)            | HIPAC-101 B   |             | in use        | 4096 w 21 bit               | 1. source program (paper tape)c.→object program (punched paper tape)<br>2. object program→execute<br>processor size 2,000 word                                                                    | Oct. 1963  |
| HARP 101                 | Shimauchi (Rikkyo Univ)            | HIPAC 101     |             | in use        | 4096 w 21 bit               | 1. source program (paper tape) c.→object program (punched paper tape)<br>2. object program→execute<br>processor size 2,500 word                                                                   | Aug. 1963  |
| MI-AUTO CODEI            | S. Ikeno                           | MUSASI-NO-1   |             | in use        | 1024 w 40 bit               | 1. source program (paper tape)c.→object program (punched paper tape)<br>2. object program→execute<br>processor size 850 word                                                                      | Oct. 1961  |

Remarks: 1. interpret, c. compiling, a. assembling

かったが、言語仕様の作り方によってはコンパイラは容易に作成できることを示した点は評価されよう。

#### 4.2 MUSE-system

昭和36年1月の第2回プログラミング・シンポジウムにおいて、三菱電機研究所の菅忠義氏(現、学習院大)、関本彰次氏、魚田勝臣氏により「MUSE system について」という報告<sup>13)</sup>がなされた。

MUSEは三菱電機研究所で試作調整中のMELCOM-LD1のために開発されたFORTRANコンパイラである。菅忠義氏達の研究室では当初ALGOLを導入する予定であったが、1960年の春頃、全三菱でIBM7090の導入が決定され、1961年末にインストールされることになったため、MUSE言語としては709FORTRANを採用したと述べている。もちろん、当時のMELCOMと7090とでは、計算機が質的にも量的にも甚だしく異なっていたので、言語仕様を全く同一にはできなかったが、機能としては殆んど同様にし、容量的な面で制限を加えるという方針をとったとしている。その理由は、709FORTRANのチェックを意図したからであった。入出力ステートメントに関しては709FORTRANとは異なり単純化されている。

MUSE-systemはFORTRANのソースプログラムをアセンブリ言語MANAに変換するシステムであった。MELCOMのアーキテクチャは、主記憶装置に磁気ドラムを採用し、IBM650のような $2+1/2$ アドレス方式を採用しているため、アセンブラ——MANA(Mitsubishi Automatic Minimum Access coding system)は丁度IBM650のSOAPに相当するが、最適化の方式は異なっている。コンパイラの設計で一番工夫をした所はDOステートメントの処理であっただろうと予測されるが、事実、この報告でも主要部分はDOステートメントの処理にあった。この時点では試作段階であり、質疑応答で、コンパイラの大きさは現在3600語とのことであった。

国産の計算機メーカーで、恐らく最も早い時期に取り組んだFORTRANコンパイラであったと思われる。特に、三菱グループで導入したIBM7090は日本のFORTRAN人口を急増させる役割を演じた。

#### 4.3 FACOM 222 FAST (CDT 402 型)

FACOM 222 FASTはFACOM 222用に開発されたFORTRANコンパイラである。この詳細は、同社の辻ヶ堂信、丸山武の両氏によって情報処理に投稿<sup>14)</sup>され、昭和38年9月号に掲載されている。コンパイ

ラの動作環境としては、磁心記憶装置4K語、磁気ドラム装置はなし、磁気テープ装置2台、紙テープ読取機1台、紙テープ穿孔機1台、ラインプリンタ1台を持つシステムであった。言語仕様は7090FORTRANに大体似ているが、磁気テープ、磁気ドラムを制御する記述とEQUIVALENCEステートメントは使用できないとしている。

コンパイラの特徴としては以下の通りである。

(1) 当時のコンパイラはパスの数が多く、数十分位かかるものもあったため、コンパイル・タイムを短くするべく、symbolic assemblerのフェーズを省略するなどの工夫をしている。

(2) 分割コンパイルを可能にしている。

(3) インタプリタ方式で複素数演算を採り入れた。

(4) モニタという程ではないが、一つのプログラムのコンパイル終了後、さらにコンパイルを続けるか、実行をするかなどを約100語のスーパーバイザ・プログラムで制御する。

このコンパイラは、FORTRANコンパイラとすれば言語仕様上、機能上、当時とすれば十分過ぎる性能を持つものと思われるが、恐らく当時のユーザにとっては、オブジェクト・プログラムの領域の制限がきつかったであろうと推察される。すなわち、主記憶装置4K語のうち、システム・プログラムが約2K語占有してしまうため、全プログラムと定数を合わせて、1930語を越えられないという制限である。当時の状況として、約50%をシステムが占有するというのは、ややシステム・プログラム側が重過ぎたと思われる。しかし、これは、メーカーの場合には、販売政策との関連が深く良し悪しの問題では無いということをお断りしておく。

### 5. FORTRAN の関連する話題

現在では、ソフトウェアの開発とかプログラミングという中でコーディングの占める割合は、どんどん減少している。また、メインフレームが開発しているシステムプログラムですら、殆んど大部分はシステム記述言語で書かれている。

FORTRANコンパイラが開発された当時は、プログラミングをする人達は、それぞれコーディングの技術も一流であったため、FORTRANの目的プログラムの性能の悪さを指摘する意見が相当に多かった。

一方、日本でもFORTRANの先進的ユーザである

気象庁および三菱原子力等から、実務上の使用経験が各企業での科学技術計算に影響を与えて行った。特に重電機メーカー各社は、原子力コードがFORTRANで書かれていたため、FORTRANの使用には熱心であった。

FORTRANの普及と教育の面では、昭和40年に導入された東京大学大型計算機センターのHITAC 5020<sup>5)</sup>と、そのFORTRAN IVコンパイラであるHARP 5020<sup>16), 17)</sup>の果たした役割も見逃すことはできない。特に同上に関して、東大出版会より森口繁一先生の「JIS FORTRAN 入門——HARP 5020に即して」がFORTRANの普及に果たした役割は大きい。また、昭和44年4月から、始まったNHK-TVにおけるコンピュータ講座によって、FORTRANは科学技術計算用言語として大きな位置を築いた。また、FORTRANユーザの数が増加しメーカーのFORTRANコンパイラの技術水準（たとえば最適化など）も向上した結果、何でもFORTRANでプログラムを組みたいという要求も強くなり、FORTRANの言語仕様が大きくなって現在に至っている。

FORTRANユーザとしての日本の最初のユーザである気象庁からは、藤原滋水氏によって「IBM 704 をプログラムテストして」<sup>18)</sup>と題して、FORTRANの使用経験に関して興味深い報告が、昭和34年になされている。ここに、その一部を抜粋しよう。すなわち、

「昭和35年春に、気象庁に設置される予定の704の準備のため、約10名内外の技術者によりプログラムの作成が開始された。昭和34年7月からWashington, D. C.にある704を借りてテストを行った。気象庁から2名の研究者がこのテストに参加し、他に米国のIBMより1名のプログラマーが参加した。先方の都合により、テストの時間は10分とか20分とか区切られることが多かったが8月中旬までに合計15時間のテストを行い、5つのプログラムを完了した。テストに使用された実例の回数は15例位である。このプログラムはすべてFORTRAN形式のもので、2名の研究者の驚異的な努力があったにしろこのプログラムテストの速さは全く驚くべきものである。これこそFORTRANの有難さであり、米国IBMの幹部が『少なくとも6カ月のプログラムテスト期間がなければ、プログラム準備の保証はし難い』と言明したことからして、IBM自身ですらこのように便利なものとは思いつかなかっただけなのである」

とある。このことは、単にプログラムが組み易いとい

うことの他に、可読性の良さ、テスト効率の良さ、ひいてはソフトウェアの生産性の良さが語られている。

ソフトウェアの生産性向上の見地からシステムプログラムを高水準言語で記述する試みは、MITのMULTICSの開発に試みられたEPL (PL/Iのサブセット) が大きな影響を与えたが、昭和38年の第4回プログラミング・シンポジウムにおいて、西村真一郎氏より“Self-Compiler”という報告<sup>19)</sup>があった。その主旨は、これまでであるシステムプログラムを作るとき、それに用いられるシステムは少なくとも自分以下の発達の程度の低い言語でなされてきた。これを改め、たとえばFORTRANコンパイラを書くのにFORTRAN言語を使用することを考えてみたい、というものであった。また、自分自身を書き表わすことのできる言語を一応閉じた言語とでも名付け、この閉じた言語の研究は今後の問題としている。この問題は、当時のFORTRANコンパイラの処理能力の点から現実性の点で問題があると感じた人も多かったと思うが、最近のUNIXシステムにおける“言語C”などは、まさにこのような考え方を実証していると言えて良からう。

最近、日本語情報処理が盛んになってきたが、FORTRANの世界に、特に制御ステートメントに「仮名」を導入する試み（カナ文字FORTRAN）も、近藤頌子、原田賢一、土居範久の諸氏<sup>20)</sup>により昭和38年に試みられている。このような方式も再評価の時機かと思われる。

最後に、HARP 5020に関連した話題として、教育の問題について述べておく。HARP 5020は今流に言えば中田育男氏がチーフプログラマーとなり少数精鋭で設計したものである。そこには「ソフトウェア工学」の問題がたくさん出て来たが、今回は省略させていただく。

5020ソフトウェアの開発にあたっては、5020アーキテクチャの設計に関連しておられた島内剛一先生の教育について述べておこう。当時、我々のグループは、HIPAC 101, 103, 501, 502などのシステム・プログラムの作成経験者が集っていた。新たに5020の開発にあたり、システム・プログラムに現われる典型的な問題を何種類も出され、これを5020（勿論機械語で）でコーディングをし、処理速度、メモリ容量を評価するというものであった。これ等の問題は、5020の命令語の特徴を良く生かすものであった。新しいアーキテクチャに移る時、古い機械のコーディング・テク

ニックはなかなか抜けないものであり、それをリセットすると同時に、アーキテクチャの考え方の教育として実に有効であった。その後、これらの問題は更に発展し、「プログラム・テクニック集」として、新たにおもしろい使い方などは登録されまとめられていった。これはまた、新人の教育に重要な役割を演じた。現在、システムプログラムと言えども高水準言語（システム記述語）で書かれるが、少なくともアーキテクチャの考え方に関する機械語レベルのセンスなくしては良いプログラムを書くことはできないと思われる。その意味でも、益々教育が大切なのではあるまいか。

## 6. おわりに

筆者の狭い視野を通して、独断と偏見でFORTRANに関する草創期の活動を述べてきた。FORTRANとALGOLで芽生えた自動プログラミングは現在のソフトウェア工学においても重要な柱であることには変らない。

FORTRANをメーカ側からみれば、単に営業戦略の問題だけではなく、どのメーカにおいても担当者達はFORTRANの自力開発なしには、コンピュータ・メーカとして脱落してしまうという危機意識が支配していたし、それが心の支えとなり急速にそのグループが大きくなった所以であろう。

一方、ユーザ側からみれば、FORTRANはやはり便利な言語であり、実質的にはCPUアーキテクチャに依存する面が多いとは言え、国産計算機のコンパイラの中では最も枯れていて信頼性も高い言語である。一つの言語が長く生存するための条件は、結局ユーザがどれだけ欲しているかにかかっている。その意味で、教育の問題と標準化の問題は言語設計において重要な要因である。

最後に、本文をまとめるにあたり、コメントや資料の提供などをしていただいた山本欣子氏、西村恕彦氏に厚くお礼を申し上げたい。また、プログラミングという素晴らしい機会を与えていただき、常に暖かい目でご指導をいただいた嶋田正三博士、および、かつて著者のソフトウェアの研究グループに参加していただいた多くの方々に心から感謝の意を表する。

## 参 考 文 献

- 1) 萱島興三, 高田昇平, 嶋田正三: デジタル計算機“HIPAC Mk-1”の論理設計について, 電子工業, Vol. 8, No. 9, pp. 97-107 (1959).
- 2) 日立製作所: HIPAC 101 PRIMER'S PROGRAMMING MANUAL (1960年5月).
- 3) 萱島興三, 高橋延匡, 氏家一彬, 高須昭輔: 科学用パラメترون計算機 HIPAC 103, 日立評論, Vol. 44, No. 7, pp. 103-109 (1962).
- 4) 嶋田正三, 吉村一馬, 高橋延匡, 中田育男, 多田敬子: HARP 103 (HIPAC 103の自動プログラミングシステム), 日立評論, 日立製作所中央研究所創立二十周年記念論文集, pp. 154-161 (1962-9).
- 5) 日本電子工業振興協会: '65日本の電子計算機 (1966).
- 6) IBM: Programmer's Primer for FORTRAN Automatic Coding System for the IBM 704 (1957).
- 7) 嶋田正三, 高橋延匡, 吉村一馬, 中田育男, 高須昭輔: HARP 103, 情報処理学会第4回プログラミング・シンポジウム報告集 (1963-1).
- 8) 日立製作所: HIPAC 103 自動プログラミング・マニュアル, HARP 103, 昭和40年7月改訂第3版.
- 9) 高德一恵他: 停電時に対する計算機保護動作を有せしめた電子計算機, 特許出願, 昭 35. 9. 20, 特許番号 # 昭 39-24502.
- 10) 田村康男: FORTRANとALGOL, 計算機用言語とプログラミング, 第4章, 電気通信学会東京支部編, pp. 52-93 (1965-2).
- 11) 出川雄二郎: 電子計算機 NEAC-2203, 電子工業, Vol. 8, No. 9, pp. 108-112 (1959).
- 12) 古山良二, 草鹿庸一郎: 2203 NARC コンパイラについて, 昭和36年度情報処理学会全国大会.
- 13) 菅 忠義, 関本彰次, 魚田勝臣: MUSE-System について, 情報処理学会第2回プログラミング・シンポジウム報告集 (1961-1).
- 14) 辻ヶ堂信, 丸山 武: FACOM 222 FAST (CDT 402 型), 情報処理, Vol. 4, No. 5, pp. 241-248 (1963).
- 15) Murata, K. and Nakazawa, K.: Very high speed serial and serial-parallel computers HITAC 5020 and HITAC 5020 E, Proc. of FJCC, Vol. 26, pt. 1, pp. 187-203 (1964).
- 16) 中田育男, 高須昭輔, 稲富 彬: HITAC 5020 ソフトウェア・システム (2) HARP 5020, 情報処理学会第5回プログラミング・シンポジウム報告集, pp. S1-55 (1964).
- 17) 東京大学大型計算機センター: 東京大学大型計算機センター10年のあゆみ (1976-3).
- 18) 藤原滋水: 大型電子計算機 IBM 704 をプログラムテストして, 電子工業, Vol. 8, No. 9, pp. 132-139 (1959).
- 19) 西村真一郎: Self-Compiler, 情報処理学会第4回プログラミング・シンポジウム報告集 (1963-1).
- 20) 近藤頌子, 原田賢一, 土居範久: カナ文字 FORTRAN, 情報処理, Vol. 5, No. 4 (1964).  
(昭和58年1月4日受付)