

# オータムフェスタ特講 コンピュータ考古学

～コンピュータの歴史と動作原理～

---

LA-Linux 専任講師 矢越昭仁

2010/11/06

コンピュータが誕生し現在の形になるまでの歴史と、その動作原理について解説します。昨今、IT 業界でもコンピュータがなぜ動くか知らないエンジニアが増えている事への反省も含め、コンピュータの動作原理を中心にお話しします。



## 目次

1.はじめに .....	4
1.1 黎明期 .....	4
1.2 揺籃期 .....	4
1.3 飛躍期 .....	4
2.黎明期（コンピュータ誕生前） .....	5
2.1 アルゴリズム.....	5
2.2 計算機械.....	6
2.3 ブール代数 .....	6
3.揺籃期（コンピュータ誕生） .....	8
3.1 第二次世界大戦中 .....	8
3.2 戦後のコンピュータ .....	9
3.3 コンパイラ .....	10
3.4 インタプリタ.....	11
3.5 OS .....	11
4.飛躍期（コンピュータの普及） .....	12
4.1 Micro-processor と電卓戦争 .....	12
4.2 GUI.....	13

## 1.はじめに

コンピュータは第二次世界大戦での要求によって誕生した計算機械でしたが、その基本技術もしくは数学的論理はそれ以前に完成していたといってもよいでしょう。また現在利用されているテクノロジーの殆どは 1960～1970 年代に登場しています。最新技術といわれる ITC 世界でも十分に温故知新を語る事ができます。

今回の講義では、黎明期として 19 世紀の数学の話、揺籃期としてコンピュータ誕生の戦後から 1960 年代頃の数々のテクノロジーの話、飛躍期としてマイクロコンピュータ（パーソナルコンピュータ）誕生を中心とした 1970～1980 年代の話題を解説してゆきます。

### 1.1 黎明期

コンピュータでプログラムを作るためには、計算する方法を順序立てて整理する必要があります。このようなプログラムを作るための本質・考え方をアルゴリズム(Algorithm)といひ古くはユークリッド言論（紀元前 3 世紀、エジプト・アレキサンドリアの数学者）に遡る事ができると言われています。この章では論理学の基礎を築いた数学者やその論理、また計算機械（計算機関）の誕生などを解説します。

### 1.2 揺籃期

いわゆる電子計算機の発明について、その社会的背景（時代の要求）を含め解説してゆきます。コンピュータが誕生してからの 20 年で現在基本とされる情報技術の殆どが生まれている事に驚く事でしょう。応用分野が飛躍的に広がり社会生活に影響を及ぼし始めるころです。

### 1.3 飛躍期

1980～1970 年代で最もエポックメイキングだったのはパーソナルコンピュータの出現です。それまでは大企業の一部で、専任のオペレータが専用の部屋の中で操作していたコンピュータが、一般の市民の目に触れ数多くの中堅企業にも広まった頃です。

## 2.黎明期（コンピュータ誕生前）

「はじめに」で概要を紹介しましたが、この章では「解を得る手順」であるアルゴリズムやロジックの基本を学びます。

### 2.1 アルゴリズム

人類最古のアルゴリズムは「ユークリッド原論」（紀元前3世紀頃）と言われています。実際には数学者ユークリッドが全てを発明した訳でなく、その頃にあった種々のアルゴリズムをまとめた物と言われています。最も簡単なアルゴリズムとしては、ユークリッド幾何学の面積問題などがあります。例えば、

「三角形の面積は、底辺の長さ×高さ÷2」

といったものがあります。平行四辺形の面積や立体の体積なども用意されていて、円の面積では本来は微分を行わないと求まらないのですが、そこはまた別の話です。

ここでアルゴリズムが上記の「三角形の面積の公式」だとすれば、

$$S = \frac{a \cdot ha}{2}$$

但し、 $a$  は頂点  $A$  の対辺の長さ、 $ha$  は  $A$  から  $a$  までの高さ

となります。このように解釈がぶれないよう（高さは見る方向で変わるとか…）方程式で表現すること、をプログラムと呼ぶこともできます。

更に、 $a$  を 5、 $ha$  を 10 とした場合、面積は 25 となり、この 5、10、25 といった値をデータと呼べばコンピュータのプログラミングにそっくりです。

日本でも今ほどコンピュータ用語が氾濫するまでは、十分に気を使って翻訳できたもので、プログラムを楽譜に見立てて「算譜」、データを「算量」、アルゴリズムを「算法」と呼んでいました。それで名著と呼ばれる翻訳本（古典）のタイトルには「～算法」といったものが数多くあります。また日本でも「和算」とよばれる独自の数学体系がありました。

### エラトステネス<sup>1</sup>の篩の方法

単純な計算だけではなく、繰り返しや判断といった処理を含む、もう少し複雑なアルゴリズムの例を考えます。今回は指定された範囲で素数を求める「エラトステネスの篩」と呼ばれるアルゴリズムを紹介します。

素数は、ご存知のように1と自分自身以外で割り切れない数をさします（2、3、5…）。それ以外は合成数と呼ばれています（4、6、8…）。

以下がその手順になります。

- 1) 2から指定された上限までの数列を用意

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	…
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	---

- 2) 数列を先頭から捜査し、2（数列の先頭の数）の2以上倍数にチェックを入れる

2	3	<del>4</del>	5	<del>6</del>	7	<del>8</del>	9	<del>10</del>	11	<del>12</del>	13	<del>14</del>	15	<del>16</del>	17	<del>18</del>	19	<del>20</del>	21	<del>22</del>	…
---	---	--------------	---	--------------	---	--------------	---	---------------	----	---------------	----	---------------	----	---------------	----	---------------	----	---------------	----	---------------	---

<sup>1</sup> 紀元前300年ほど前の古代ギリシャの数学者。地球の大きさを始めて測定した事で有名

3) 数列全ての捜査が終わったら、再び3から(2)と同じ事を繰り返す。

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	...
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

4) (3) を数列の最後まで繰り返す。

5) 最後に、数列でチェックのついていないものを出力する。

2、3、5、7、11、13、17、19...

実際には(3)の開始位置や、どこまで繰り返せばよいか?といった工夫を行うことで全体の作業量を少なくする事ができます。

## 2.2 計算機械

1800年代にはいると、産業革命がおこり多くの科学技術とりわけ機械工業が発達します。この頃は電子的な装置は一切ありませんでしたが、その元となるアイデアは数多く発明されました。

ジャガード織で有名な織機は模様のパターンをパンチカードで換える事ができました、ある意味ゲーム機のROMのような記憶媒体の走りと考えてよいでしょう。これを応用した集計機も登場します。特に1880年代、米国の国勢調査局は極端な移民の増加に手を焼いており10年ごとの調査の集計が、10年たっても間に合わないという絶望的な状況に陥っていた事態を救ったのがホレリス・カードでした。それは今でいうマークシートのようなカードを用意し、それを機械的に読み取る(集計する)仕組みも登場しました、

また歯車を複雑に組み合わせた計算機が登場します。たとえば「人間は考える葦である」で有名なパスカルは、歯車を用いた初の機械式計算機(パスカリーヌ)を発明しましたし、バベッジはさらに大規模な階差機関を発明(多項式を用いて対数や三角関数を扱う事ができる)します。さらに複雑な解析機関(微積分)を着想しましたが、完成には至りませんでした。

解析機関以外は、基本的に歯車を回し繰り返し演算を行うといった手法をとります。たとえば電卓で $2 \times \times \text{=====}$  とするようなものです。

## 2.3 ブール代数

今までの発明は、主に数を数える、計算をするといった「算術演算」とよばれる機能でした。しかしコンピュータでは複雑な判断を伴う処理も行います。例えば人口統計データのうち、「女性で20歳以上40歳未満かつ関東(神奈川、東京、千葉、埼玉、群馬)に住むみ専業主婦でない人の合計」といった処理です。

この「〇〇かつ」「〇〇または」「〇〇でない」という操作は「論理演算」と呼ばれます。数学の方程式のような分野があり、物事の間関係を記号化して表現、さらに種々の法則までそんざいします。この「物事の間関係」を対象とした数学の分野が論理数学で、19世紀中ごろにジョージ・ブールによって考案されたと言われています。基本的な法則は以下の6つです。

1) 冪等律:  $X \text{ and } X == X \text{ or } X == X$

- 2) 交換律 :  $X \text{ and } Y == Y \text{ and } X, X \text{ or } Y == X \text{ or } Y$
- 3) 結合律 :  $(X \text{ and } Y) \text{ and } Z == X \text{ and } (Y \text{ and } Z), (X \text{ or } Y) \text{ or } Z == X \text{ or } (Y \text{ or } Z)$
- 4) 吸収律 :  $(X \text{ and } Y) \text{ or } X == X, (X \text{ or } Y) \text{ and } X == X$
- 5) 分配律 :  $(X \text{ or } Y) \text{ and } Z == (X \text{ and } Z) \text{ or } (Y \text{ and } Z), (X \text{ and } Y) \text{ or } Z == (X \text{ or } Z) \text{ and } (Y \text{ or } Z)$
- 6) 補元律 :  $X \text{ or } \text{not}X = \text{全て}, X \text{ and } \text{not}X = \text{無}$

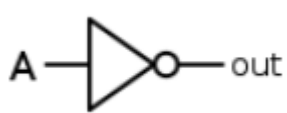
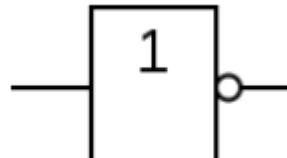

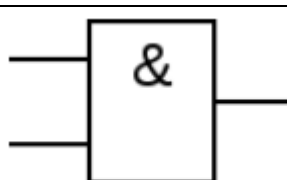

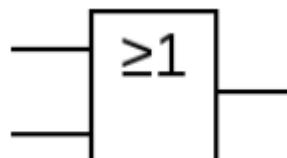
実際には集合  $\Delta$  に対し  $\text{and} = \wedge, \text{or} = \vee, \text{not} = \neg, \text{全て} = T, \text{無} = \perp$

このブール代数で扱う集合を 2 値 (0, 1) とすることで、様々な論理演算が可能となりました、今のコンピュータは全てこの論理式に基づき動いています。

余談として、ド・モルガンの法則は上記の応用辺で、数多くの電子回路の基本的な考え方となっています。

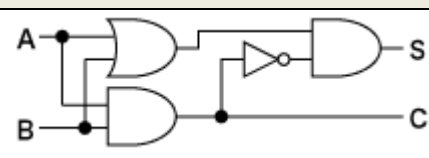
$$\text{not} (X \text{ or } Y) == \text{not } X \text{ and } \text{not } Y$$

$$\text{not}(X \text{ and } Y) == \text{not } X \text{ or } \text{not } Y$$

論理式	MIL 記号	JIS 記号
NOT		
AND		
OR		

実際の IC(LSI)では、からさらに AND、OR の出力を反転 (NOT) にした、NAND、NOR 回路が多く用いられます。

この論理回路を組み合わせることで、2 進数の足し算を行うことができ、半加算器(Half Adder)と呼ばれます。これがコンピュータの原点となります。

A	B	C	S	論理回路
0	0	0	0	
0	1	0	1	
1	0	0	1	
1	1	1	0	

この回路では、A,B2つを入力とし、S が和、C が桁あふれとした結果を返すことができます。ちなみに S に相当する論理回路は XOR (排他的論理和) と呼ばれます。

このように、アルゴリズムを解くためには単に計算をするだけではなく、論理的な仕組み（≒プログラミング）も必要だという点が発見された時代でした。

### 3. 揺籃期（コンピュータ誕生）

20 世紀に入ると、産業革命の恩恵をうけ流通業（商店）も発達してきます。米国の西部開拓史に登場する駅馬車(オリジナルは 17 世紀にパスカルが考案した 5 ソル馬車でした)や、大草原の小さな家に出てくる町の商店などが、この頃に商店から企業へ発展した頃です。

#### 3.1 第二次世界大戦中

先に登場した集計機（ホレリス・カード）や、機械式キャッシュレジスターを売っていた会社が合併し 1911 年 IBM(The International Business Machines)が誕生しました。

当時の IBM はまさにレジスタや集計機などを作っていました。第二次世界大戦が始まると多くの計算が必要となる軍部・政府との結びつきが強くなってゆきます。精密機械を作る技術を応用し自動小銃なども作るようになります<sup>2</sup>。

特に世界初の電子計算機(digital computer)と呼ばれる ABC(Atanasoff-Berry Computer)は暗号解読に使われ、ENIAC は大砲の弾道計算に使われていました。

WWII 中は暗号開発競争が激化していて、特にドイツのエニグマ暗号機は素晴らしく、現在の暗号化の基礎を作ったともいえます(後に Z1、Z2、Z3 と進化。古い UNIX の passwd(4)にもエニグマの記載があった)。そのエニグマを解読したのは英のアラン・チューリングが係った Colossus で、その結果ノルマンディ作戦が成功したとも伝えられます。

Z3 : 1941 年独、プログラム可能機電気算機、リレー 2200 個

ABC:1942 年米、世界初の電子計算機、真空管 31 本

Colossus: 1943 年英、世界初のプログラム可能電子計算機、真空管 2,500 本、

ENIAC:1944 年米、真空管 17,468 本

#### チューリング・マシン

第二次大戦中、英国で暗号解読など複雑な数学と向き合っていたアラン・チューリングは、現在のコンピュータの基本ともいえる数学モデル「チューリング・マシン」(1936 年)を提唱しています。数学のいち分野として「オートマトン」というモデルが存在しますが、その中でも極めて単純化されたモデルがチューリング・マシンと言えます。その特徴は

読み書きできる無限に長いテープ

テープの情報を読み書きするヘッド

内部状態を保持するメモリ

となりますが、いわば IPO-Model (Input/Process/Output)の概念を始めて定義した事となります。コンピュータで解ける問題を数学的に証明することを、「チューリング・マシンで解ける事を証明する」といったりします。また人工知能の分野で用いられる、電話先の相手が人間か機械かを見極めるテスト「チューリング・テスト」も考案しています。

---

<sup>2</sup> 「IBM とホロコースト」 ISBN4-7601-2158-7



コンピュータ界のノーベル賞といわれる ACM<sup>3</sup>チューリング賞は、これら業績をたたえるものです。

### ノイマン型コンピュータ

ジョン・フォン・ノイマン（1903-1957 ハンガリー）20世紀を代表する科学者の一人、アインシュタインとともにマンハッタン計画（原爆開発）に携わり、フェルミとともに水爆を開発。ENIACの開発にあたっては、自分の次に計算の速い奴が出来たと喜んだ…など、驚異的な計算能力・思考力をもった人でした。

そのノイマンが提唱したコンピュータの原理は今でも全てのコンピュータに応用されており「ノイマン型コンピュータ」と呼ばれています。

ノイマン型コンピュータ（コンピュータの5大要素）：

1. 計算装置が独立している（ALU: Arithmetic Logic Unit）
2. 制御装置をもつ（チューリング・マシンのヘッド、テープ送り機能に相当）
3. メモリに状態を記録する
4. 入力装置
5. 出力装置

また、メモリにはデータとプログラム両方を搭載することでプログラム自信を書き換え自己増殖的な機能をもたせる「ストアード・プログラミング方式」を提唱しています。

またプログラムを上から1つずつ順番に実行（正確には制御装置が示す場所から）するといった、一見当たり前に見える考え方もノイマン型コンピュータの特徴です。

### 3.2 戦後のコンピュータ

戦争がおわると、暗号や弾道計算・各種実験専用だったコンピュータがビジネス分野でも再び使われるようになります。

ENIACの開発に携わったジョン・エッカートとジョン・モークリーが立ち上げたECC(Electronic/Eckert Computer Company)が資金不足に陥りレミントンランド社傘下になったUNIVAC-Iが世界初の商用コンピュータとなりました。1951年に発表され50台近くを販売、当時米国でコンピュータといえばUNIVAC(Universal Automatic Computer)と言われるほどで、1952年の大統領選挙では過去の開票結果に基づき大方の予想に反し、アイゼンハワー当選予想を的中させたと言われます。

IBMは1953年にIBM701を世界第2の商用コンピュータとして発売します。後発ですがUNIVAC-Iと比較すると性能は見劣りするものでした。（UNIVAC-I:加算100,000/秒、磁気テープ、IBM701:21,000/秒、紙カード）しかし、後続機IBM704で初めてコンパイラが登場します。

また1953年には日本初のコンピュータFujicが誕生します。ちなみにFujicは製品ではなく、富士フィルムが非球面レンズの計算をするために、自社内で使うために作ったコンピュータでした。

---

<sup>3</sup> ACM: Association for Computing Machinery. 米国の情報工学学会

### 3.3 コンパイラ

1957年にIBM704で稼動したFORTRANコンパイラが初の高級言語とされます。IBMのガッカスが設計し、いまでもコンピュータ言語やプロトコルを定義する方法としてバックス・ナウア記方(Backus-Naur Form)が用いられています。

FORTRAN(Formula Translator)は名前の通り、コンピュータに  $A=\text{SIN}(0.5)*2$  といった数式を直接入力し処理できる言語として開発されました。それまでのコンピュータは専門のプログラマが直接マシン語に変換する必要があり、とても素人が扱える代物ではありませんでしたが、FORTRANでは人の言葉(数式ですが)を直接コンピュータに与えることで仕事をさせることが可能となりました。普通の人を理解できる数式をコンピュータの言語(マシン語)に変換するプログラムをコンパイラ(翻訳家)といいます。

また当時はOSという概念がなく、FORTRAN3では、言語機能に加えモニタとよばれるBIOSに似たハードウェア制御機能が組み込まれていました。

このFORTRANは1960年代のアポロ計画でも利用され(しかしコンピュータそのものはUNIVAC Vanguardでしたが)、その時にNASAが作成した膨大な科学技術演算ライブラリのため現在も利用されています。

この頃になると、UNIVAC,IBMの成功により、様々なコンピュータ・メーカーが乱立することになります。メーカーが異なるとプログラミング言語も異なり、またメーカーが自社の特徴を生かすべく機能追加を行うため利用者からは使いにくいものとなっていました。そこで1959年に米国のCODASYL(Conference on Data Systems Languages)が結成され、商業向け言語(つまり科学者ではなく事務員が使える言語)の仕様を開発・公表します。そこで登場したのが、より普通の言語(英語)にちかいCOBOLです。

COBOL(COMmon Business Oriented Language)は、より英語表記に近い文法と、コンピュータの環境定義(たとえば入出力装置を割り当てるといった)を含む機能のおかげで個ハードウェアに依存しないプログラムを実現しました。また主なコンピュータ言語のなかで唯一10進数を正しく扱う事ができる言語です。

米国の独走にコンピュータ危機感をもった欧州勢はFORTRAN,COBOLと対抗すべく全く別の言語を開発しました。FORTRANが科学技術演算、COBOLが商業用途であったため、この言語はアルゴリズムの研究のために開発されALGOL(ALGORithmic Language)と名付けられました。この開発にはBNFのNであるナウアも参加しています。

名前の通りプログラミングに必要とされる主要な機能はこのALGOLで実現されており、構造化プログラミング(モジュール化)、再帰呼び出し、局所変数、カプセル化が実装された最初の言語でもあります。登場からACMで、アルゴリズムを記述する標準言語としても用いられていました。

しかしながらハードウェア構造を無視した理想を追求した言語は実装が難しくパフォーマンスにも難があったため一般に普及する事が少なく、JIS規格でも1983年には廃止されています。

この様に用途によって言語が異なる事態を打開すべく、1964年にIBMがFORTRAN, COBOL両方の性格を併せ持つPL/I (Programming Language One)を開発、発表しました。評価は分かれるところですが、PL/Iは二つの言語をそのまま取り込んだため、プログラマによってはFORTRANに似た書き方や、COBOLに似た書き方ができるといった2面性をもつ言語となりました。またコンパイラ自信が非常に巨大なものとなり、大型コンピュータでしか動作しないものとなり、あまり普及しなかった言語です。

日本では銀行の第三次オンラインプロジェクト(ATMを普及させた)で多くの都銀が採用していました。またMULTICSプロジェクトではシステム記述言語としても採用された事例があります。

### 3.4 インタプリタ

1960年代には、科学技術向けのFORTRAN、会計処理向けのCOBOL、コンピュータのアルゴリズム記述用ALGOLだけでなく、文字列処理に特化した言語が登場します。文字列処理を進めると人間の言葉が理解できることから、人工知能向け言語としても有名なのがLISP(List Processing language)です。最初のLISPはFORTRAN上で実装されたインタプリタで、FORTRANにつぐ史上2代目の言語となります。FSF創始者のリチャード・ストールマンが作ったGNU EmacsはLISPを使って自己拡張できる設計になっています。

インタプリタで最も有名な言語は1965年にケメニーとクルツが開発したBASICでしょう、1980年代の第一次パソコンブームでは各社がBASICを採用していましたし、その影響でMicrosoftが台頭してきたのでした。BASICはBeginner's All-purpose Symbolic Instruction Codeの略で、その名の通り素人(文系の人)が扱えるよう親しみやすい文法を目指して設計されました。当時はGUIもなくFORTRANに似た言語でした。

さらに1967年にはタートルとよばれる簡単なロボットを操作できるLogoが登場します。子供にコンピュータを学ばせる事を目的とした言語でした。BASICがFORTRANの影響を受けているとすると、LogoはLISPの流れをくみます。

### 3.5 OS

この頃のコンピュータでは、OSという概念がなく言語そのものがハードウェアの制御をおこなっていました。一言でいえば「コンピュータを使う=プログラミングを行う」という図式になっていました。

初のOSは1964年にIBMが開発したOS/360だといわれています。このOSはバッチ処理(予約実行・一括処理)を基本としていて、予め計画された仕事を効率よくこなすことを目的としていました。そのため現在でも金融など信頼性を求められる業界ではよく用いられています。実行するアプリケーションプログラムとコンピュータ資源管理を分離し、その橋渡しをするJCL(Job Control Language)も特徴の一つです。

しかし、複数のユーザがそれぞれの仕事を「使いたい時に使う」オン・デマンド処理への

要望が高まり、初の TSS(Time Sharing System)が MULTICS として GE、MIT、AT&T ベル研究所の共同プロジェクトとして 1965 年にスタートします。OS 自信を高級言語 PL/I で記述するといった画期的なプロジェクトでありましたが、あまりにも巨大で複雑すぎプロジェクトは 10 年近くかかってしまいます。その間に GE はコンピュータ事業から撤退し製品は Honeywell から販売されることとなります。

また、1969 年にはベル研究所が当プロジェクトから離脱、その結果 UNIX が登場するといった皮肉な事も起こっています。

1960 年代は他にも ARPA-Net プロジェクトが開始されたり、テッド・ネルソンが Hypertext やノート PC の概念を発表するなど、現在のコンピュータ技術のほとんどが定義された年代といってもよいでしょう。

#### 4. 飛躍期（コンピュータの普及）

1970 年代に入ると、パソコンが誕生し個人でもコンピュータを利用する時代となります。この時代は米軍およびその周辺にあったコンピュータが、民間で利用できる時代になったとも言えます。また「コンピュータ＝軍事技術（安全保障）」であったなかで、米国以外で唯一日本だけがスーパーコンピュータを作る事が出来たという点も特筆すべきことです。

##### 4.1 Micro-processor と電卓戦争

1970 年代日本は高度成長期で、いろんな企業が躍進を始めた時期でした。それとともに計算すべき事が増え、計算機の需要も伸びました。シャープ、カシオといった新興ハイテクメーカーは計算機の低価格化、高速化、小型化にしのぎを削り「電子卓上計算機＝電卓」を生み出します。当時の競争は「電卓戦争」とも呼ばれ、TV 番組にもなりました（電子立国日本の自叙伝）。

その中で消え去ったメーカーも存在しますが、その会社が画期的な発明を行います。メーカーの名前はデジコン社で、競争の激化する電卓戦争に勝つためには「汎用的な計算用 IC」が必要と考えていました。当時は計算機ごとに CPU 設計するような仕組みとなっており、すべてハードウェアの設計にかかっていた。これをソフトウェアによってこなそうという画期的な発想をデジコン社もっていました。しかし当時の国内はおろか、米国にも CPU を単一チップに実装しようとした会社はなく、デジコンのエンジニアである嶋氏は米国の半導体メーカーを放浪します。唯一チャレンジをしてくれたのは、ベンチャーとして誕生したばかりの Intel でした。

1971 年史上初のマイクロプロセッサ 4004(4bit)がインテルから発表されます、ほどなくしデジコン社は解散してしまい、嶋氏はインテルへ転職、その後も CPU の設計屋として Zilog を始め多くの半導体メーカーを渡り歩く事になります。

このマイクロプロセッサの誕生が、今まで機械式制御であった種々の装置を、多機能な製品へと転化させていくこととなります。計算機としてだけではなく、全自動洗濯機や自動車エンジンの電子制御などに応用されてゆきます。

初のパーソナルコンピュータ MITS Altair8800 (1974 年 MITS)もインテル社の 8008 を使っていました。

## 4.2 GUI

1973年には Xerox のパロ・アルト研究所(PARC)がワークステーションの概念を発表します。オフィス環境をコンピュータ上に仮想的に実現する GUI (マウス、アイコンの概念) を搭載した Alto を発表します。PARC はさらにオブジェクト指向言語 (環境) である Smalltalk も開発します。1972年に初版が誕生しましたが最初は BASIC コンパイラ上に実装されたためパフォーマンスは良くなかったと思われます。

オブジェクト指向プログラミングは正に GUI と相性がよく、後のプログラミング言語やユーザインタフェースの基礎となりました。

ちなみに PARC は Ethernet も開発しており、現在の PC を初めて具現化したといってもいいでしょう。

また ALGOL 設計に携わったチューリッヒ工科大学のニクラス・ヴィルトは、この Alto に触発され Lilith を開発。また ALGOL を進化させた Pascal 言語も作成する事になります。Pascal は UCSD p-System にも採用される事になります。UCSD p-System は OS ですが、そこで実行される UCSD Pascal はハードウェアに依存することなく動作する仮想コンピュータを実現しました。これは現在の Java の基礎概念と同じものです (1978年)。

同様に Apple 社も遅れる事 1983年に、Pascal を基本言語とした GUI 対応の Lisa を発表します。商業的には失敗しましたが、いわゆるパソコンで初めてマウスが付いた製品でした。X の登場も同時期でしたが、Ver 11 の後半で大幅な改定を行いオブジェクト指向に移行しました。