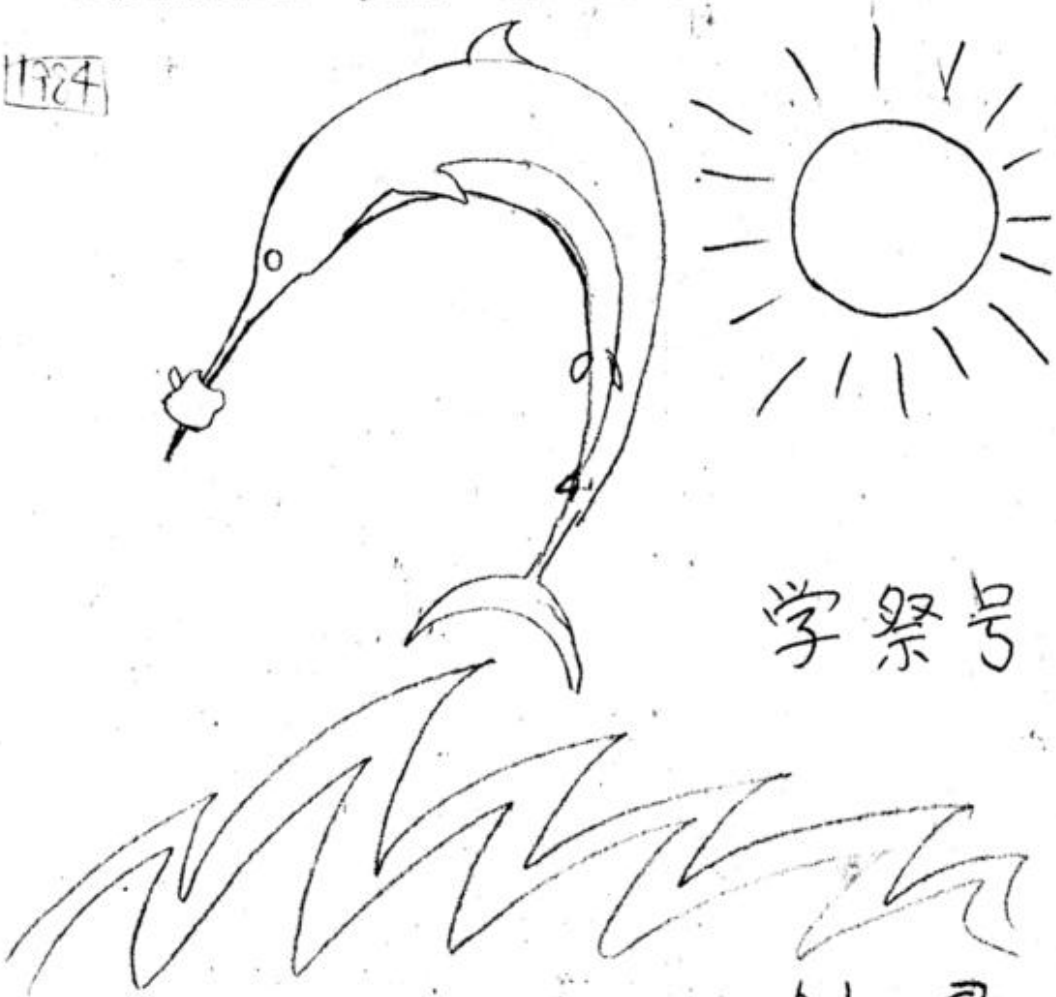
Lime

1984

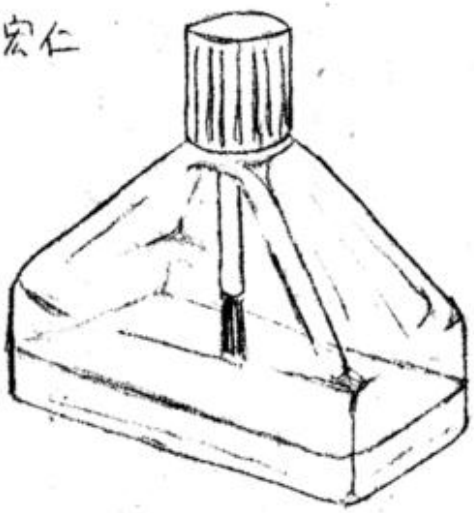


学祭号

No 8

目次

背表紙：森岡宏仁



By T.G.

6502 CPU モド記

by. mk Ⅱ

今回の学祭で私の苦労が報いられていれば、みなさんの前でうごいているであろう プログラムは、6502 CPUエミュレータ-です。ちなみに、エミュレーターというのは、異る体系のCPUの上でマシン語プログラムを走らせるためのもので、たとえて言えば、Z80のマシン語プログラムを6809 CPUの上で走らせるようなもので、ここでは、6502のマシン語プログラムをZ80 CPUの上で走らせています。ただし今回は、6502の動きがよくわかるように視覚的要素が強いものになっています。

6502 CPUというのは、あの!! apple に乗っているCPUで アキュームレーター 1ヶ に X, Y レジスタ. C.C レジスタ に. ALU が 1ヶ と なかなか シンプル (あな美しや!!) な デザインです。

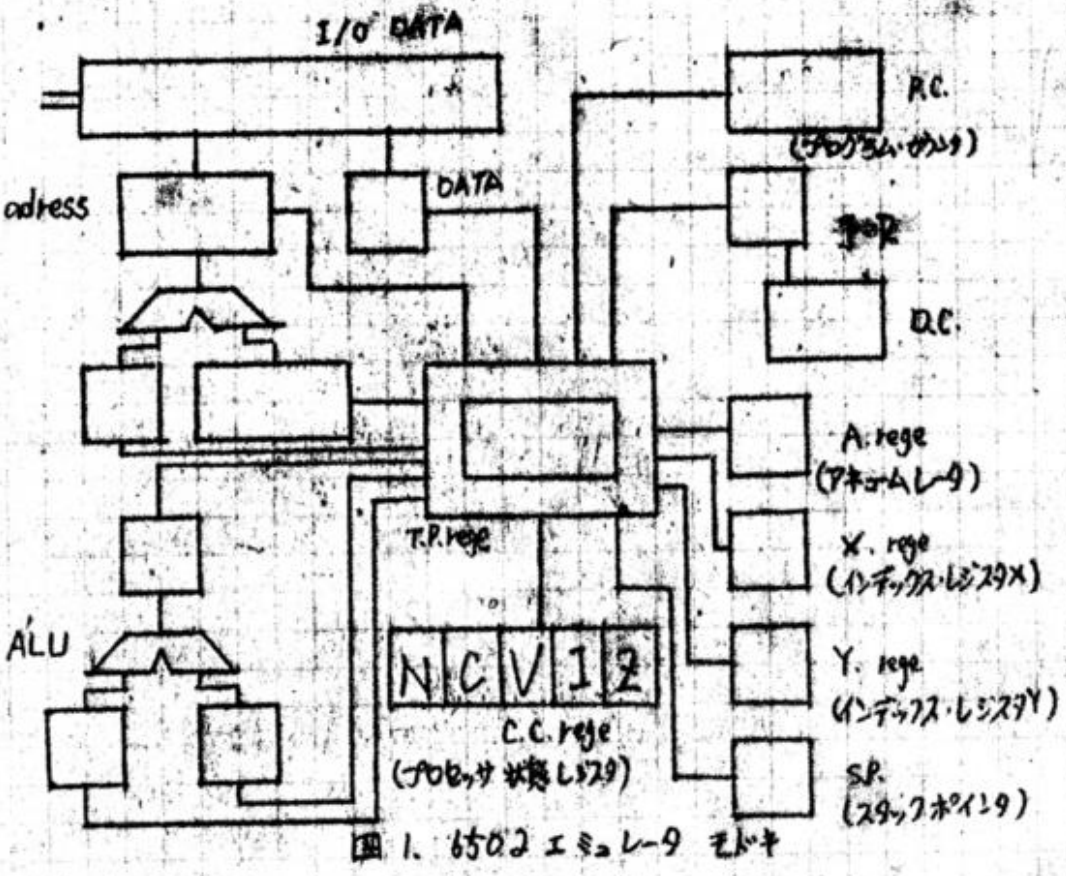


図1. 6502 エミュレータ モデル

そして表示は1図のようになります。上図で I/O DATA というのは、実は、アドレスバスとデータバスが別々に走っていなければならないのですが、わかりやすくするために、いっぱに、表示してあります。又中心の T.P. rege というのは、実は、6502は存在しないのですが、今流れている DATA を見る窓として、作ってみました。なお、6502には、ALU (加算器) は1つしか 存在 していないので、図1では、

2つ存在しているように見えるのは、6502のALUは8bitの計算しかできないのでadress(16bit)の計算では上位と下位に分けて計算しなければならず、これを表示するのはめんどうなので、仮想の16bit計算用のALUを作ってごまかしてしまいました。(ゴメンナサイ)それからデザインの関係よりALUの向きが反対になってしまいました。又、CC.regも実は、8bitでなく(N,V,0,B,D,I,ZC)と8bitですが表示の関係で二人になってしまいました。

このエミュレーターを製作するに当り、どのCPUのエミュレーターを作るかが問題でした。

始めはZ80のを作ろうかと考えました。Z80は、レジスタがやたらと多く、メモリーへのアクセスの回数が少い所が気に入りました。ところが、命令系統が統一されていず、レジスタが多く、表示の面で問題があるのでボツになりました。

6809はZ80に比べてレジスタが少なく、命令も比較的統一されているのでなかなかいいなと思いました。ところが、レジスタが少ない分、アドレッシングモード

(メモリーをアクセスするための命令)が発達していて 多くの
アドレッシングモードが有り、これを表現しないと、
6809にも5ちゃん、にの表現もなかなか困難なので、
思わず ボツ にしてしまいました。

ここで6502は 6809のようた レジスタが少なく、
命令は全て1バイト と命令系統もなかなかシンプルで、
最小限必要なコンピューターの要素だけを集めたような
CPUでエミュレータを作るのに都合のよい構造だな、
ていました。そこで私はこれに飛びついてしまいました。

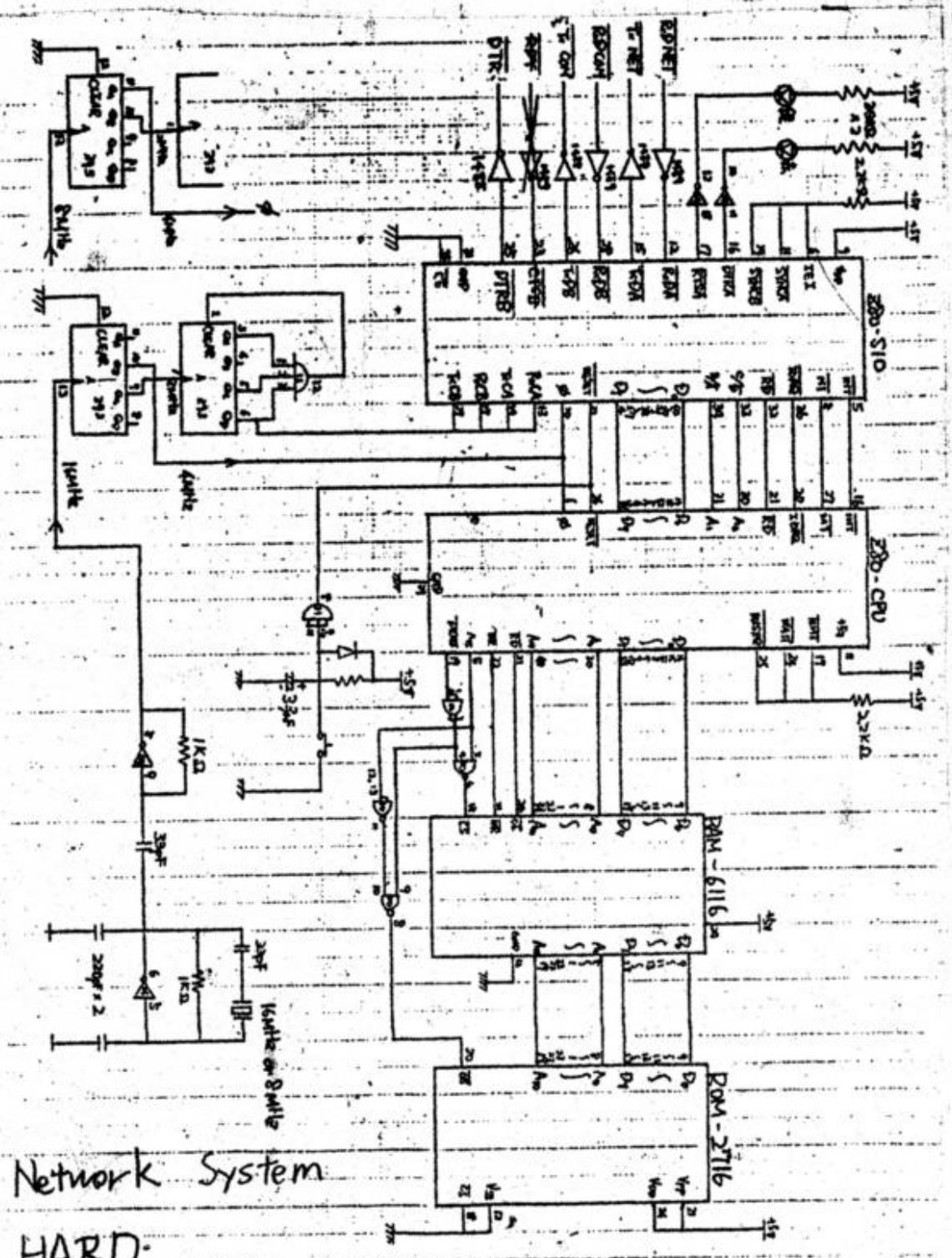
今このようなエミュレーターを作ってみて思った事は、
CPUにはおのおの性格が異っていて、製作者の思想が表
われている事です。又80ちゃん電子計算機より発達したもの
であるし、6809ちゃん大型コンピューターに近づけようとして
作られたものであるし、6502ちゃんハード的な制約をで
きるだけ少くし 汎用性や拡張性を考えて作られたもの
であるという事がわかります。

今回のエミュレーターは時間がなく そして何よりも製
作者に知識がないので かなり いいかげん なものた

なってしまいました。… T.P.reg.は何のために存在するのかわからないし、それよりなにも動かないし… それにALUから出力された DATA の位置と A reg が T.P reg を通さないと、DATA のやりとりができないのは、勿だし adress の DATA と P.C とは直接つながっているべきだし、どうみても笑ってしまいそうなものになってしまいました。

最後に 多大 なる協力をいただいた、駒嵐大先生や @ 先生, 大橋 君, 宮辺 君に大きなる、感謝をこめて このいいかげんな文を、終りたいと思います。

6



Network System

HARD

COMPUTER NETWORK SYSTEM の HARD について.

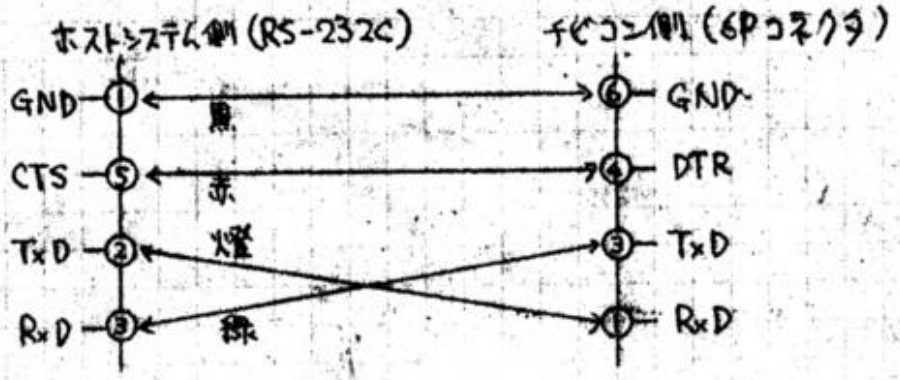
by S.O

今年は、去年のチビコンの改良ということで、すでにハードの大部分は完成していた為、配線できしむことはなかった。主な改良点は.

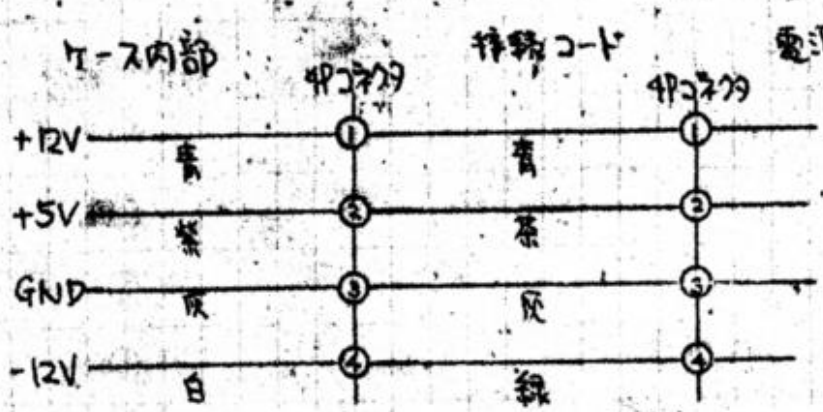
1. DTR → CTS により チビコンが受信できる状態にない時、ホストシステムからの送信を抑える。
2. 去年使用した RDY は使用しない。
3. ホストシステムとの接続には RS-232C 用コネクタを使用する。
4. 電源配線の改良
5. ケースに入れる。

などです。

各コネクタの配線は次の様になっています。



8



今回ケースに入れる為リセットスイッチを買って来たが、さすがに安物だけあって、すっすぐこわれてしまったものがあった。今後の改良点としては、電源コードを太くする。丈夫なリセットスイッチをつける。ということが考えられます。

最後に、ケース加工、及び配線を手伝っていただいた方々どうもありがとうございました。

アルゴノオト より...

「4回生に最後のシヨウモナイことを聞かせてもらおうということで、エーっと自己紹介をお願いします。」

「文学部 計量国語学科 計算機講座の@です。」

「聞きなれない学科ですね。何をする人ですか?」

「日本語を論理学的・形而学的に解析しようというものです。」

「具体的には?」

「計算機に京都の観光案内をやらせています。」

「対話形式の、いわゆる質疑応答システムですか?」

「いえ、一問一答式ですが、並みの人間には答えられないような質問にも答えます。」

「たとえば?」

「『開館時刻が閉館時刻が京都会館と同じ会館と同じ会館を教えて下さい』といった」

「最後には"。"をつけて下さいね。なるほど。でも、データベースの会館という分類には開館時刻といった項目はなかったんじゃないかなあ....

ええ、たしかにありませんね。」

「おっとそうでしたか。まあ僕がつくったものじゃありませんがSo」

「話はかわりますが，どうしてこのクラブに入ったんですか?」

「はっきり覚えていませんが，はじめはTV Gameがしたかったからでしょう。でもシステム班というのに入ってからは，少し変わったと思います。」

「そこには，何かやりたいことがあった?」

「いえ，システム班ではタダでCoffeeが飲めるということで。それにコンピューターのことを全く知りませんでしたからSo」

「システム班というからには，OSとか言語とかやる人でしょ」

「そうです。僕が入ったのは1回の後期で，既にLimeプロジェクトがはじまっていて，毎週コロラドで話し合いをやっていました。」

「コロラドというと，修学院駅前の。」

「高木町の方ではなくて。」

「Limeというと。この本の名前もそうですが。」

「詳しくは既刊のLimeをみてもらうとして，簡単にいうと，processor Aという Virtual Machineと，そのA codeを出力するCompilerを作成する，そのCompile言語がLimeです。」

「Limeも人も仕様がはっきりしないのですが。」

「そうですが。おかしいなあ。前部長の田中君か，竹岡さんにでも聞いて下さい。」

「言語Limeの作成はその後どういう影響を?」

「原稿に描いたような話の展開ですね。ええ，それからプログラミング環境，とくに言語に興味をもちましたね。」

「最近よく New Paradigm がどーのこーのといいますね。」

「そうですね。まあ昔から言ってはいるみたいですけどね。最近の特徴としては A.I. がかっている点でしょうか。」

「Object-oriented とかもですか？」

「Object-oriented も Knowledge Representation という。」

「Frame とかですか？」

「Inheritance な人が，そうでしょうし，Message Passing も Demon と同じという説が。まあ少しちがいますが。」

「恣意的な term の使用された会話になってきましたね。」

「C 的ですよ。これからは。UNIX はどうか知りませんが。」

「自然言語処理をやらせているということは A.I. なんかも。」

「あれは，A.A.I. かもしれないが，A.I. じゃないでしょう。少数分派の原始信仰，いや希望ですねまだ。」

「宗教論争にもならない？」

「そういう点では Programming Paradigm の方が宗教論争になっておもしろい，僕には。」

「でも機械翻訳な人が お金になるとか。」

「あなたと僕が 今こうしているような対話をするシステムをつくれますかという質問はどうでしょう。」

「きっとタメイキでしょうね。」

「そのタメイキは何に対するものでしょうか。」

「う～ん。」ApplyするほどA.I.がすすんでいない。でも，ApplyしてみないとA.I.もすすまない。」

「人間様のProgrammingすらもうまくいっていないのに。」

「話をもとにもどしまして，Limeの影響は具体的には？」

「その後Lispをかじりはじめまして。」

「Zeta Lispとかいろいろありますが。それにまたどうして？」

「A.I.に興味があって，竹岡さんにLispをやりなさいとすすめられたので。」

「それで。」

「その頃は今ほどいい本がなくて，安かった『Lisp入門』(近代科学社の方)を買ってきて，読んでいて，どうしても処理系がほしくなって，2回になったばかりの頃につくりまして。」

「『ほしくなって』じゃなく，『つくりたくなって』でしょ。」

「ええ。でもあの本の中にLispの処理系をつくってみなさいという問題がありました。古い本はみんなそうかどうか知りませんが，Implementのため中味について書かれていると思うんですが。それにLisp systemについて知らないと，Lisp Machine Manualなんて読めませんよ。」

「MacLisp系の方のManualも700ページにせまるとか。」

「昔 PL/I ってすごく太った言語だよ と聞きましたが，Zeta Lisp の Manual をみてがうは、「どこが?!」と思いましたね。」

「そうです。昔 よく 太りすぎ と言われて。でも Lisp の方が最近太ってますか。」

「いえ，小さなものがいっぱい集まって強力になっているという感じで。まあ Ada がいるから。」

「Ada といえば、Ada が死ぬまで 彼女がお金で苦労したので，今の Ada は DoD の資金を浪費させようという あくい がかかっているような。」

「プログラマへの のろい でなければよいが。」

「Link すると 90分 遊べてしまうという!?」

「LMI. の Lisp Compiler は microprogram レベルまで Compile するとか。たしかに 速くはなるだろうが，Lisp の neck は Garbage Collection にあるので，その over head をいかに小さくするのかが もっと大きな問題です。」

「Hardware で G.C. をしている!?」

「しかも real-time に。」

「Data-flow Machine のような 高度の並列化 ってのは?」

「Reduction Machine ですか。とにかく，この辺は問題がいろいろあっておもしろい。」

「でも，ここまで 読んできている人は、この人なら ショウモナイことを読む必要のない人でしょうね。」

「Topicsばかりで内容がないような。」

「別にそういうつもりじゃ。紙面も多くしいし,眠いし。」

「僕がクラブに入ったときには、まわりでやりとりされる会話というのは,わかりませんでした。。竹岡さんや駒嵐さんに少しずつ聞いたり本を読んだりしたように思います。」

「そこからだんだんと。」

「今やりたいことがないとかいう人たちをみうけますが,早くなんかに,それがわけのわからないものでも,ほんの少しの興味をもつものがあれば取りつかれてほしいなあと思います。その後少しずつ勉強していけば,どんどんおもしろくなっていくんじゃないかなあ。自然と話題も増えていくんじゃないかなあ。」うまく言えないが,それだけの心の落ちを持っていると思います。」

「うーーん と,茶化すのはよしまして。今回は趣味に走ったお話をどうもありがとうございました。

「自分で書くのもなんですが,どーいたしまして。」

FREE ALL

LOGOFF

(@)

内容のない コーナー

(もんくたれ)

by KAZU

現在84学園祭まっただ中であるが、毎年の慣例に従がって展示できるものは……。
私自身も大きな声では何もいえないけれども、化のクラブ(特に運動系)のテントや茶店で学祭中ずっと"準備中"の所はないはずだ!!
彼らは日頃はクラブの練習をし、また全員が一致団結してこの期間の成功を治めている。
来年からはずっと学祭中は、説明人が少数、店番をしているだけで、残りは化学園祭へ見学(遊びかてら)に行きたいものですね~。
難しいかもしれないが、やらなくてはならないのだ!

立ち直れ コンピューター部。

学祭で成功する方法

Part 1. 人間関係を深くして（上下両も）団結し仕事も分配すること

Part 2. 何を展示するかを早期に決定すること。

Part 3. 展示作品は夏休み中には完成の領域に達していること。

Part 4. クラブ員みなが、展示内容を理解していること。

Part 5. 学祭は祭（まつり）であるので おもいっきり Enjoy すること。

まあ、これができたら すっごいですね～。

Micro Processor の解説 〔後編〕 TAKE●

▷MN-1890

こいつがすごい!? さすが松下(だい)さん、さて、うなるか笑うか?

▶概要と特徴

1) 仮想的に構成した2つのProcessorを同一のチップ上に集積

2) 各々のProcessorが異なった仕事を時分割で交互に処理できる。

3) 高効率命令

8bit/4bit…BCD加減算、ニブル変換。1bit操作。

8bit×8bit乗算、15bit÷7bit除算。

命令リピート機能(ストリング操作、多方向分岐)

スタック利用による多重ループ命令

ROM領域内テーブル・ルックアップ

4) 割り込み:外部、タイマ、シリアル、プログラム・トラップ

5) 16bitタイマ/カウンタを2本持つ。

6) 8bitシリアル転送・インターフェイス

7) 豊富なI/Oポート端子(51本(内4本はD/A出力))

No.

8) 誤動作防止機能(暴走時トラップ割り込み, 低電圧検出)

9) 命令実行速度(Min 500ns、平均 1.5μs、8MHz発振時)

10) 外部ROM, RAM拡張可(Max 各64K Byte)

▶ 勝手説

4), 5), 6), 7), 8) から分かると思うが、組み込み用である。

ROM, RAMの内蔵量によって右の表のような型番がある。

型名	ROM	RAM
MN18942	4KB	256B
MN18962	6KB	256B
MN18982	8KB	592B
MN18922	12KB	656B

9) の命令実行速度のMin 500ns や、平均1.5μsとは、珍しい記述だね。8MHzでMin 500nsだと、多分、NOPが4Clockなので、大体の命令はZ-80と同じ程度のパフォーマンスかな?

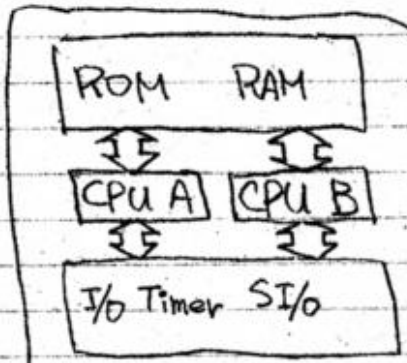
3) の命令群は、なかなか良いもんだね。乗除算もあるし、テーブル・ルックアップというのは、配列がSearchingか? 命令リピートとは8086のストリング・プリ・バイトだろう。スタック利用による多重ループは、スタックを使ったDO-Loopがネストして書けるということだね。うーむ、なかなかすごい。

しかし、本題はこれからだ。

このProcessorは、実はこんな名前がついている。

「86Pt(時分割)デュアル1チップマイクロコンピュータ」

はて?「デュアル1チップ」とは?わたくしゃ、はじめて見た時、「おっ、これはTRANSPUTERのまねして、ついに、1チップ上に、2個のプロセッサを載せたか。流石は、お手軽の松下サン、これなら発想無しで、パフォーマンスは倍だもんな。しかし、松下サンそんなに集積度上げられるかなぁ??」と思いました。実際カタログの概念図を見るとやっぱり、それ風でした。(右図⇨)



時分割デュアル1チップマイクロコンピュータの概念図

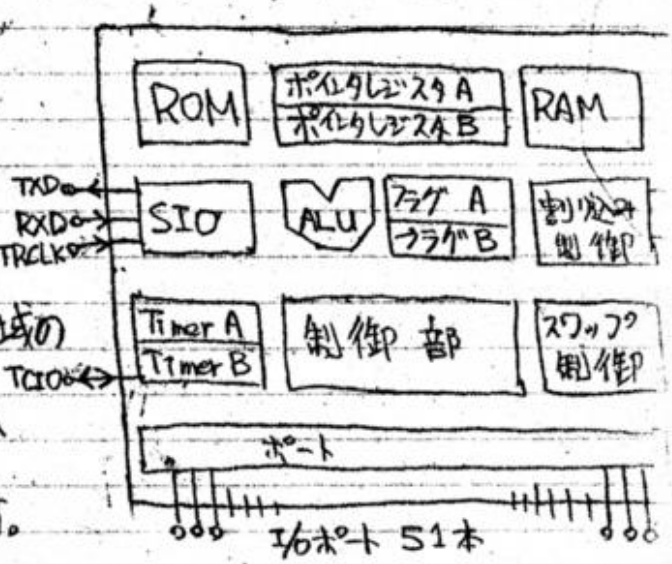
しかしながら、こんな記述もあります。

- 共有する内部ハードを時分割使用することにより仮想的に2つのマイクロコンピュータをLSI上に構成したものです。
- 各マイクロコンピュータは、プログラムステータス(命令ポインタ、スタックポインタ、フラッグ)を固有ハードウェアとして専有しますが、ROM、RAM、I/Oポート、

No.

演算制御部は共有されます。

- 各マイクロコンピュータ間共有するROM, RAMの新領域を自由に設定可能、かつ一部領域の共有化可能であり、制御システム設計上の自由度が大きくなります。
- プロセッサ間通信が共有RAMを介して簡単に行えます。



↑ブロック図だそうです。

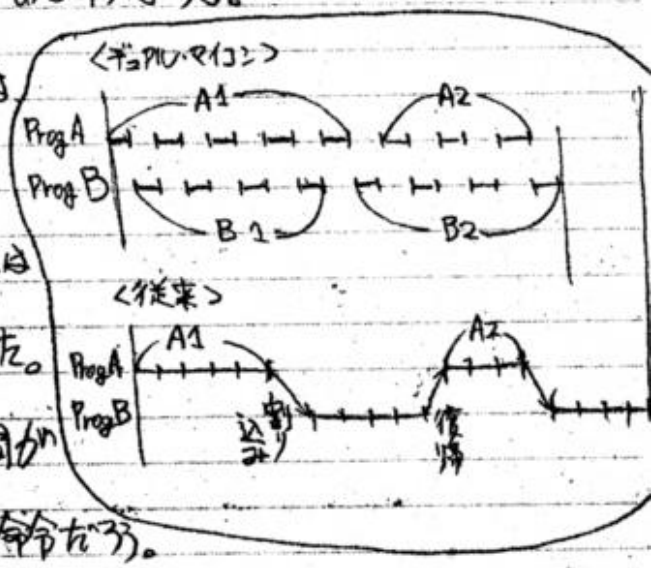
▶ これら上の記述より(ブロック図を見れば明らか)、Hardwareのうち、プロセッサ2個分は全部持ち込みではなく、時分割できるリソースは共有して使うという節約の精神。うーむ、えらい、せこい、どっちだ?!?

▶ ポインタレジスタ君は、何本あるか全くわからないが、上の記述より、プログラムカウンタとスタック・ポインタは存在して、プロセッサごとに別々に持っている。これはしごくあたりまえだ。

割り込みのSource信号線がA, Bと別々にあり、これも、プロセッサ2個分。

動きをするには必要だろう。同様に、カウンタ/タイマもA,B別々に管理でき、組み込み用マイコン1台分を1つ盛り込んだのと、同じことを可能にしている。このカウンタ/タイマは、普通の汎用マイクロ・プロセッサなら一本も内蔵していなくても差しつかえない。ROM, RAMの専有領域を自由に設定可能だとか、共有領域も作れると、書いてあるが、さて、設定するという機能は本当にあるのだろうか？これは、単にわたくしめのカンですが、多分、プロセッサ間の、メモリ・プロテクションとか、セグメント化などは、全然ないのだろう。なから、アッセンブラが多少、かしこいだけなのではないかな。それでもって、DEBUGは、大へん有のではないかな？これは全くのあてずっぽうよ。

▶さて、それでは、時分割プロセッサとはど人な？これは、もちろんプロセッサを時分割で使うんではない。カタログには右図のような、動作比較が載っていた。これでは、リソース切替えの基本時間が不明だが、大方(おおかた) 1Clock、悪くても1命令だろう。



No.

Hardとしては、1Clockごとに、切り替えても、難しくない、ひょっとすると、かえって簡単なので、多分そうだろう。うん、これなら、6800/6809の、裏Clockと、似ているね。でも、内部Busは、多分裏Clockでもつまっているから、1Clockづつの実行になってしまうのが有。

▶さて、デュアル・マイコンと従来の比較を見ると、従来型は、割り込み処理のコンテクスト・スイッチングに時間がかかるので、明らかに、デュアル・マイコンの方が、全体の処理は、勝っている。うーむ。しかし、例えばA1自身の処理は、従来型の方が早く終っている。これは、両者とも、無駄の無い動きをするしごく当り前の結果だね。だから、A1の処理が、ぎりぎりだったりしたら、このデュアル・マイコンは使えないことになる。要は、使用場所(時間)の正しい選択か。

▶しかし、このデュアルマイコンが良いのは、2つのProcessを実行するのに、マルチタスクモニタは一切いらない。割り込み的発想で2つのProcessを実行するにしても、割り込み処理ルーチンも不用なので、これはやっぱりすごい。評価できることだね。しかしながら、2つのProcessを同期させる訓練のできてない軟弱者に、使わせると、簡単に虫が出て、I/O等のAccessをむちゃくちゃにするだろう。例えば、

割り込みで、Secondaryな Processが動く時に、クリティカルな
操作をする時は、従来、割り込み禁止命令を発行するだけで良かった
のが、デュアルマイコン(従来なら)だと、Secondaryな Processがクリティカルな操作の
前に何りかの(セマフォ等)チェックをしてもらわねばならなくなり、やっかいだね。
これは、昨日や今日マイコンをいじりはじめた人には、けっこう難しいと思うが…。
(そこで、プロになろうと思ってるキミ、最近の電子レンジや、洗濯機の
プログラムは、割り込みや、スキャンするものがたくさんあって難しいんだよ。)

▶なれしろ、プロセスが複数あれば おもしろいことがいっぱい
起こり、楽しくなる。それでも、スキャンするものがいっぱいあると、
マルチタスクモニタを使って、別々なプロセスにしたくもなるし、実際
プログラムの見通しも良くなる。そんな時、だまってても、2Processあれば
うれしいし、プログラムエリアも有効にできたうれしい。また、全くちがう
2つの仕事がある時、それも、やっぱり、マルチタスクモニタ不在で、やりとげる。
しかも、組み込みのスペースは一個分と、非常に有利で、
こいつは、 

と、思いますね。(CPU)でも、「せこい」という感も
まぬがれない。

No. ( )

▶もう少し、具体的に解説をすれば、このプロセッサで、個別に
もっている、ポインタ・レジスタ、フラグレジスタやタイマは、コルーチンの個有変数、
オブジェクト・オリエンテッド言語の変数にあたり、したがって、インスタンスごとに
個別に持つのは、いたって自然。そして、このデュアル・マイコンにおいては
インスタンスは、Forkして実行される。また、このプロセッサで、共有しているもの
は、ほとんどが、コルーチンのプロセッシング部分、オブジェクト・オリエンテッド
言語のメソッドに写像できることがわかる。そして、共有部の「スワップ制御部」と
「割り込み制御部」が、スケジューラや、プロセッサ・オブジェクト等、System管理の
部分にあたる。ただし、ROM, RAM, I/Oポートの管理は、(さっきからの予想では)
ユーザー割り当てになっているので、それらの考えへ写像させ方が良いだろう。
(つまり、管理が低級なので、話題としておもしろくない)ということで、色々、考えられる、
MN-1800でした。たった一枚の紙きれで、なかなか楽しめるおもしろいChipです。
でも、とっても偉いと思いつつ、心のどこかでセコイとつぶやくのは、ボクだけでしょうか?
～♪ひとーり、じょおーずとよばれて♪～

さて、どんじりにひかえしは、

DμPD7281D

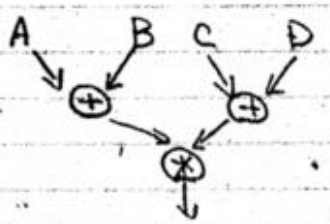
▶ Image Pipelined Processor (ImPP) 画像処理に向けて
パイプライルで処理を進めるプロセッサ。

▶これだけでは何のことか分からないが、実は、このProcessorは
Data Flow Machine (でーた・ふろー・ましん) ですっ!!!!

▶こんなLSI世界でも全く初めて、だれも実用的製品が今頃
出来るとは思わなかった。画期的だね。

▶さて、Data Flow Machineとは、話せば長くなるんで、Referenceの
⑦で、古い歴史を、⑥と④で最近の物を知れば良い。⑥、⑦は、最近特に
有名なのである。と、言っていてはきもフタも無いので、若干解説すれば、
(Limeの近号に、『のいまん v.s. のんのいまん (きるちふろうせすのお勉強、番外編)』と、
いうのを、載せる予定で、その中で、詳しく説明したいが)

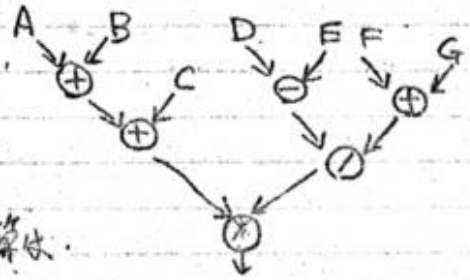
▶例えば (A+B)*(C+D) は



と、なり、(A+B)と(C+D)を全く、独立した

別々な⊕でやるから、速い。それでは

((A+B)+C)*((D-E)/(F+G)) はというと、



となる。

▶これは、何が有難いかというと、独立な演算は、

どんどん勝手に進む。独立かつ並列な所は、同時に複数処理

される。無駄な時間は、演算間の時間の差による待ち合わせだけと

なる。ということで、パイプライン処理と同じ特長の他に並列に違う演

算がいくつも起きているという特長がある。

▶それでは、結構すごいか？というと、『⊕や⊖が、有限個しか用意できない。』

いう、本質的問題のため、パラメータの待ち合わせや、スケジューリング等でやっかいな

問題が生じているのでした。また、素人(しろうと)のキミ達には、このキカイで、条件判断

Loopを、どう解決するかは、わからないだろう。(ザマミロ、『のいまん VS. のんのんさ』

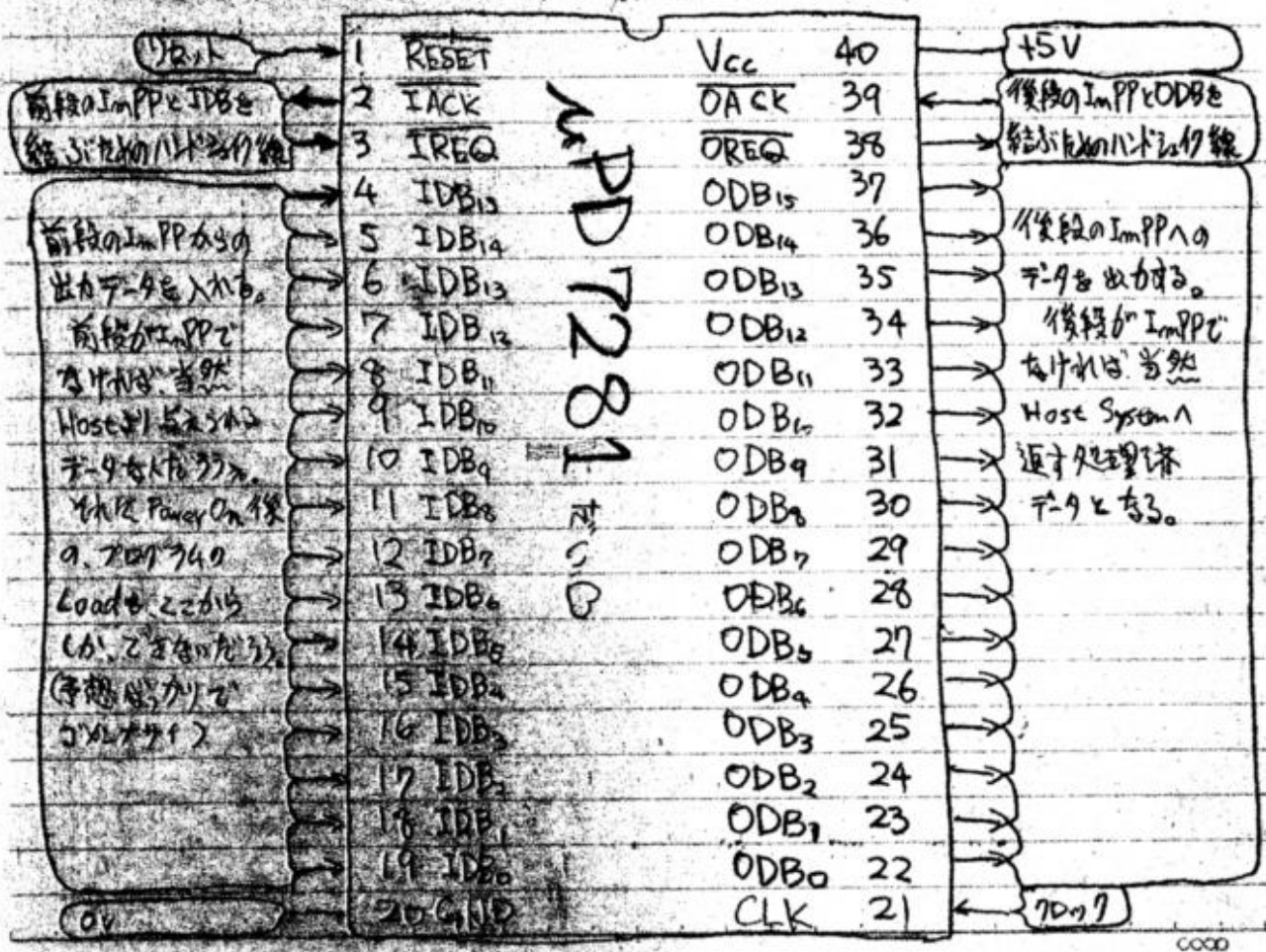
に、期待しておくれやす)しかし、若干の疑問は、μPD72001の解説でも、とけるだろう

▶それでは μPD7281 の解説。

ちゃんとした、技術資料を読んでないので、いまいち不安だが、評価はできる。

▶まずは、このLSIを使うつもりで攻めてみる。今もとのドキュメントだけでは、ちょっと資料が足りないが。

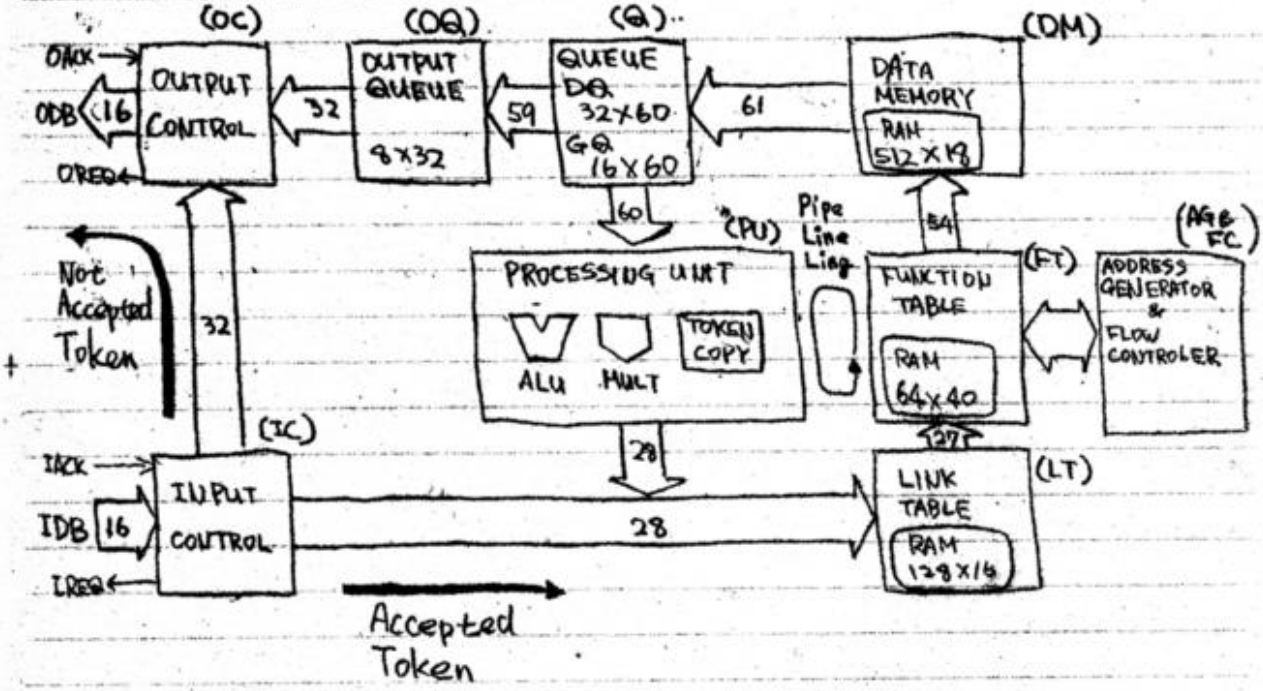
▶これが『μPD7281D』だ!!



No.

『なんや、これは!?』と、思ったのでは、ないかな。全く、こいつは、データ・バスしか

ついてない。ふん。それでは、ブロック図は、



うぬぬ。これは、これでもProcessorか。そーなんですね。やっぱり Non-Neum

なんですね! それでは、これで どんな動作をするのか。

▶うごき

・はじめに、お断わりしておきますが、これは、ほとんど bit 誌を参照しているので、退屈したらご免なさい。(Reference ⑤)

・まず、DATA だが、DATA には、DATA 自身の情報と、行先や属性をもつフィールドを持っていて、それらをまとめたものを ImPP では、トークンと呼んでいる。(フィールドの詳細は、不明)

・最初に、アプリケーションの実行に先立って、外部の Host (当りまえ) から IC からプログラムがロードされる。このプログラムのこともオブジェクトと呼ぶ。いかにも、Data Flow らしい名前だね。

・DATA が ImPP へ着くと、IC へ入力され、そのフィールドから、この ImPP の中で処理するか否かが判断され、受け付けられなかった (Not Accepted) の時は、OC から後段へ出力される。

・Accept された Token は、次に LT へ送られる。ここで、行き先フィールドを、LT 内のメモリから Pick up した新しい行き先と、置き替える。ブロック図から明らかな様に、新しい行先 Pipe Line Ling は、(は) 一周した後に、再び LT をアクセスする時に、使われる。ここで、一つ疑問 (問題) がある。

∴もし、PUを通って、演算の終っているTokenで、しかも、後段のIn PUで処理されるべきだとすると、Pipe Line Lingをむざむざ一周無駄に回らなければならない。ここらへんのMachineの手抜きが見える様な気がするが……

・さて、LTより出たTokenはFTへ入る。FTでは、PUで行なう演算のオペコードをPick upする。それと供にAGとFCを使いながら、DMで、データをリード/ライトするアドレスを生成したり、LTで得られた行き先情報を修飾する制御を実行する。ここで、なぜDMをリード/ライトしなければならないか？

・Data Flow Machineは本来メモリの無い機械なのである。メモリとプロセッサの間のバンド・ワイズの狭さが、ノイマン・ボトルネックと呼ばれる通りなので、メモリがあると、ノイマンと同じではないか？　また、僕が思うに、他入の機構自体がマルチ・プロセッシングを目指んでいるのだ。なぜなら同一の場所を使用するから実行の順序が重要となり、また、同期のメカニズム複雑になるのだ。

それでは、このDMとは何か？実は、Data Flow Machineと、いえども、何らかの形でDataをLaechするメカニズムが必要なのである。(そうでなくてどうやって、記憶を実現できるだろう!!?!)具体的に、どこに必要かと言えば、例えば、3引数関数triPara(x,y,z)を、データフロー的に表わせば

となるが、x,yが既に計算済で、triParaにzが到達しているのに、その計算が終っていない時、何らかの形で、triParaは、xとyを保持しておかねばならない。

これでDMの機能は明らかになった、DMは、Tokenの待ち合わせを行なうのである。FITはAG&FCを使いながらTokenを待っているDM中のデータを捜しだすアドレスを作るか、又は、Tokenに対応するデータがDM中になかったのでそのTokenをDM中で待たせるアドレスを作るのである。

・そして、DMへ着いたTokenは、DMへ格納されるか、これのTokenを呼び出し、Qへと向かう。尚、DMは片割れのTokenを16個保持できる。

・DMから出たTokenはQで待ち合わせを行ない、PUへ入り、オペコードに従ってPUで演算される。

No. ( )

・一通りImPP内で処理されたTokenはQからQQを通りOCから出力され、後段のImPPまたはHostへ渡される。

▶さて気になる速さだが、各ブロック内200nsのパイプ・ライン・サイクルで同期して動作し、LT, FT, DMAは1パイプ・ライン・サイクル、QとPUは2パイプ・ライン・サイクルを要するので、パイプ・ライン・リング一周で7パイプ・ライン・サイクルだそうだ。PUは、Logical Operation (AND, OR等)、Arithmatic (ADD, SUB, MULTI, Shift, Compair等の命令を全て200nsで実行できるらしい。むむ、MULTIが200nsとは、μPD7720を思い出すなあ。パイプ・ライン・リング一周で7サイクルなので7つ以上のTokenがImPP内にあれば、PUは最高の能率で動くということだ。

▶知っている人は知っているが、NECは大型の画像処理用のData Flow Machineを作ったことを、μPD7281の発表の数ヶ月前に公表している。その機械も、全体がパイプ・ライン・リングの形をしていて、しかも、その中の部品(?)にあたるModuleもパイプ・ライン・リングの形をしている。

・このImPPは、そのマシンの、各Moduleを作るための共通部分を取り出し、プリミティブとして、使用できる様に、開発したものだそうだ。(Reference⑩)

■評価

・なかなか、すっきりして気持ち良い、しかも画期的なProcessorだね。しかしながら、このマシンはPipeLineを基調としていることを忘れてはならない。

・画像処理用と名付けてある様に、単調な同じ処理を大量に、しかも直列に、できれば独立にできる仕事なら確かに速い。例えばLay Tracing (Convolinksで有名な)なら、各点は、完全に独立なので、IAPPの特性を生かせる。・しかしながら、インベーダーゲームはどうだ？各事象も、独立で、しかも等時性も要求され、また人間の入力に対する応答の即時性をも要求されている。とてもじゃないが、PipeLineなんかで、待ってられるか！(実は、高速に処理されるので、だいじょうぶだと思うが。)・どうしてみても、各所で、色んなことをしたい用途には、あまり向いていないようだ。しかしながら、やはり、企業で、DataFlow Machineを作る。しかも、需要も不明なのに、LSIにする。これは、すごいことだと思うね。もちろんNECなら内部での消費量も大したものだろうし、画像処理装置や音声認識装置は、もうかるが、それでもえらい。・ひょっとすると、LISP Machineの次のDataFlow Machineの時代はNECの物かもね。(LISP Machineの次はProlog Machineという説もあるか？)〔Modula, Ada, Occam ???〕

No.

▷ということで、たくさん書きなぐってしまいました（乗れ）乱筆、御免なさい。

しかし、高速信号処理用というのも、なかなか独特のアーキテクチャで楽しめたと思いますね。

・できればスズキの刀のことを、書きたかったが、時間のつごうで、ダメでした。スズキセンセも、一番しんの所では、セコかったりして、おもしろいんだけどね。

・けったいなアーキテクチャでは、Prolog Machineが、たいそうな名前だが、Concurrent Unifyする時のデータの持ち具合、転送具合が、みんな困ってて、ちっとも、根本的な解決になってない。もろに、Neumann-ボトルネックしてたかかって少しも、Non-Naumannらしくない。

・今ならアマチュアでも、考えるだけなので十分楽しめる。それにNECのImPPを手に入れれば、それなりに感動も人だろう。

・そんなこんなで、今回の異名の解説は、終りたいと思いますね。

・ここまで、読めた人、どうも有難うございました。

※ちなみに、前編は、次のLineです。

(ニソ/Nov./'84 Take

Reference. てきとーに、お問い合わせ下さい。当方は関知しません。

① μPD7720D …… NECの IA-112B JULY-30-83PK データシート集

② TMS-320 …… TIの SCJ 1111 なるカタログ

③ MN-1890 …… 松下電子工業の D-041/1(J) なるカタログ

④ μPD-7281D … NECのカタログ。マイコンショウ'84で配ってたが、カタログNo.は無い。MAY-23-84M

⑤ 黒川 泰文、大内 光郎：
データフロー制御方式を採用した非ノイマン型プロセッサ μPD7281，
bit，共立出版，7月，1984．

⑥ 雨宮 真人：関数型言語とデータフローマシン，新しい時代のソフトウェア，
共立出版，1984

⑦ 中都宮 公訓：データフロー計算機，bit，共立出版 Vol.12，No.3～9

⑧ 坂村 健：I/O COMPUTER，bit，共立出版 Vol.12，No.7～1

⑨ 平木 敬、西田 健次、島田 俊夫：
科学技術用データ駆動計算機 SIGMA-1 のハードウェア構成、
第28回情報処理学会全国大会 4E-2

⑩ 天満 勉：画像処理用のプロセッサ，情報処理 Vol.25，No.9

使いにくさの認識

-- 良きもんくたれになるために --

MASA

コンピュータを使っていると、「あれが欲しい」「これがこうなっていたら」などとよく思うものである。ところがその内容は千差万別で、時として当の本人の力量を知らしめることも多い。

不満などと言うものが、使っているシステムに対する認識やコンピュータ全般に関する知識のレベルがそのまま反映するからである。例えば、全くの素人がパーコンのソフトを買ってきたとき、そのソフトに対する不満を言わせてみれば、まったく馬鹿げたと思えるような内容かもしれないし、実際そういった経験をした人もいるかもしれない。

ところが人間というものは自分を中心にして考えやすいもので、他の人から見れば馬鹿げているのは自分の方であったということもまたよくある話である。

日ごろ、とんでもないシステムを使わされていてもんくのひとつも言わないのは、よっぽど我慢強いか、使いやすいシステムというものを知らないからであって、どちらにせよ好ましい状態でないことは確かである。しかし「知らぬが仏」の言葉通り、使いやすいシステム（＝非常に高価である場合が多い）の存在を知ってしまったがために毎日「○○が欲しい！」（○のなかにはお好きなシステム名をいれてください。）と叫ぶのがより好ましい状態であるかどうかの判断は読者におまかせしよう。

一般に毎日の職場で叫び声を発するのはいただけないが、世間にたむろするマイコンマニアなどという人種が良くできたシステム、使いやすいシステムの存在も知らずに、ハードのメーカーの作り出した環境の中で甘んじるのはまったくもってなさけない。

世の中には与えられた環境の中で最大の努力を払えばそれでよいと言う人もいるだろうが、自分に与えられた環境の限界を知りつつ努力するのとその環境がすべてであるのではまったく違うということを私は声を大にして言いたい。

必要に迫られた通常のエンドユーザーでなく、しかもプロでもないものをマニアということにすると彼らには少なからずとももんくたれの素質があるといえるが、使っているシステムを超えるだけの知識と認識力とがないことにはそのシステムにもんくをたれるなどということができるはずがない。

今こうしてワープロに向かっている私はそれほどワープロの経験があるわけではないが、やはりもんくがないわけがなく、「こんな単語もあらへん」とか「なんでここでカーソルが動かんのや」というようにぶつくさ言いながら書いているのである。
もんくの源はすべて不便に思う心である。「なんでこんなことができんのや」という心の叫びがもんくをたれ、ひいては環境改善にもつながるのである。（べつに私は宗教をやろうというのではない。）

現在、あたりを見まわしてみるととんでもないようなソフトやハードがまかり通っており、もんくをたれている人間は意外と少ない。一体世間の人々はこんなにも我慢強いのかと思いきやそのほとんどの人は泣き寝入りか、あるいはなにも知らずに使っているようで、そういったエンドユーザーに対してマニアたちはがんばってもんくをたれているかというとそうでもない。

もっと勉強してもんくをたれることを乞いたい。

編集後記

mKII：ははは…やっとLimeができてしまった。

O君：でも明日で学祭は終わりですよ。

mKII：そういう事は全部、Editerの責任にしておいて。今回のLimeを読んでどう思う？

O'君：う～ん まず 学祭号なのに学祭に関する記事が少ないですね。

mKII：今回部長がnetworkにとりとられていて原稿を書いてもらえなかったのが苦しかった。

O君：ほんとうに。学祭前からオシロスコープをながめては、何か一人でブツブツおっしゃってられますからね。

部長：なんでここがローのままやねん…ブッブッブッブッブッ

mkII：まぁあの呪いも学祭が終るころにはとけると思うから次の号にはしかりと書いてもらえると思うけな。

O君：それにmIIのPSの宮辺ってだれですか。

mkII：あれは部を辺と間違えただけでギャグと思ってもらえるとうれしいんだけど…ハハハ…

O君：それにあの原稿の内容はZ80.6809.6502の各方面から批難ゴウゴウですね。

mkII：あれなみんな筆者の氏名がさらけだされただけでみんな筆者の責任……（自分とわかってあもる）

O君：先輩の書かれた原稿にはクラブの方針を書いたものが多いですね。

mkII：やっぱり先輩方も今の部の状態を危険だと思ってらっしゃるんですよ。

O君：部員のみんなもあれを読んで真剣に考えてほしいもので…あ！あの土けむりは何だ

mkII：やっぱりLimeの費用、だって部費から使いこんだのはやばが…

（以下怒りに目をぬらしS也たジュームズ君の必殺のM-16の掃射音に消されて解読不能）