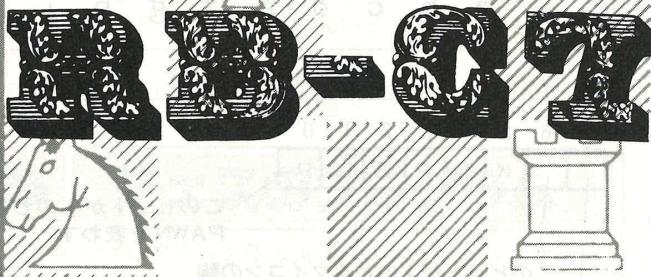


MZ-80B

チエス・プログラム

CHESS PROGRAM



このプログラムは、次の3つの目的をもって作りました。
 ①CHESS
 ②ENDINGの練習
 ③チエス問題を解く
 CCCC番地でスタートさせると、上の3つのゲームを選択するブロックにやってきます(写真1)。

TOMUC ● 馬場隆信 / COMPAC

1

CHESS

C キーを押すと、コンピュータとチェスの対戦することになりますが、その前に先手か後手かを選びます(写真2)。[W] (白) が先手で [B] (黒) が後手です。

次にLEVEL を選びます(写真3)。僕としては [3] を中心にプログラムを作ったつもりですが、最初のうちはレベル [1] で序盤定跡を覚えてから、[3] に進むのがいいと思います。

チェスの駒の動かし方は、国際表記法を使っています。チェスの表記法には大きく分けて2つあり、他の1つは英米書式です。私達がチェスの本を読むときは、たいてい英語ですから、英米書式も覚えておく必要があるのですが、最近ではアメリカでも国際表記法を使い始めているので、このプログラムでも国際表記法を使いました。

図2を見てください。g2のピショップを、f3へ動かすときは、Bf3と書きます。キー入力では、[B] [f] [3]となります(図3)。

写真1

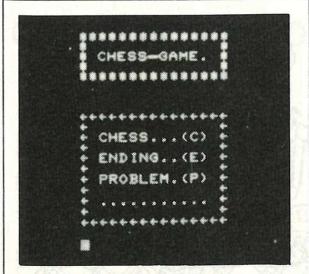


写真2

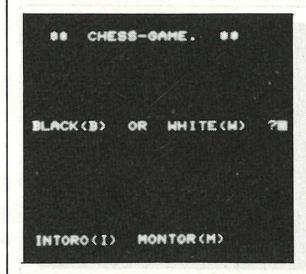


写真3

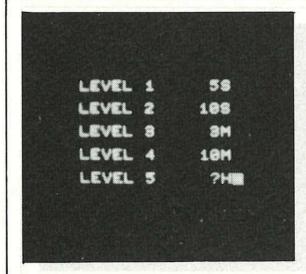
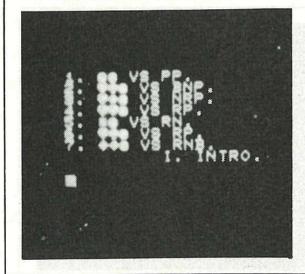


写真4



e1のナイトを、f3へ動かすときは、Nf3と書きます。キー入力は [N] [f] [3] です。では、e1のナイトをc2へ動かすときはどうでしょうか? このときはNec2 ([N] [e] [c] [2]) となります。なぜかと言うと、a1のナイトもc2へ進めるために、どちらのナイトかを区別する必要があったからです。

では、d1にあるルークをd3へ進める場合はどうでしょうか? a3にあるルークもd3へ進むことができるため、どちらのルークかを区別する必要があります。この場合は、R1d3あるいは、Rdd3と書きます。キー入力は [R] [1] [d] [3] です。

さて、ポーンの動かし方は少し変っています。たとえば、g3のポーンをg4に動かすときには、Pg4とせず、単にg4とします。また、a4のポーンでb5の敵駒を取る場合は、ab5と書きます。これは、aファイルのポーンで、b5の駒を取るということを表わしています。

キャスリングは、K側をO-O、Q側をO-O-Oと書くのですが、キー入力では、[k] をキング・サイド・キャスリング、[q] をクイン・サイド・キャスリングとしました。

チェス・プログラム

なお、キー入力の後には、必ず **CR** キーを押してください。

プリンタへの出力

ゲーム進行中に、次の命令が使えます。

T M CR : 画面上1行にメッセージを書き、**CR** でプリントへ出力。

T T CR : 手筋を国際表記法で印字する。

2

ENDING 練習

写真1で **E** のキーを押すと写真4になり、どんなENDINGを希望するかを **1** ~ **9** のキーで指定します。すると、乱数で適当に駒を並べて問題を作ってくれます。後は1のCHESSと同じです。

相手のキングを詰ますには、最低限、図4のような駒が必要です。これを、メイティング・マテリアルと言います。

このブロックは、乱数の作り方の失敗で、同じパターンの問題が出てしまします。

X CR キーで駒設定ブロック呼んで問題を作り、**X CR** で戻してENDING練習してください。

3

チェス問題を解く

P キーを押すと、このブロックにきます(写真5)。まず、チェス・ボードに駒を並べます。これはメイン・キー ボードを使ってください。**P f 3** と **CR** でボーンの駒が f 3 のマス目に置かれます。

次に、深度を設定します。**S** キーを押すと、DEPTH? と聞いてくるので、2手詰のときは **4** を、3手詰のときは **6** を押してください(写真6)。

そして、**V** キーを押すと、答を画面右側に表示して、駒も1回分を動かしてくれます。

次に、**&** キーを押すと白黒を入れ換えてくれます。そして、再び **V** キーを押します。これを繰り返すと、解答の全手筋が解ることになります。これは、詰めチェスのみではなく、NEXTムーブでも使えます。

I で写真1へ戻ります。

4

序盤データ・エディタについて

1のCHESSから、**O CR** でこのブロックに来ます。これにより、序盤データを自分で作ることができ、練習したい手筋を作れるわけです(写真7)。

このブロックは、チェス盤の換わりもしてくれるので、手筋の研究にとても便利です。

図2

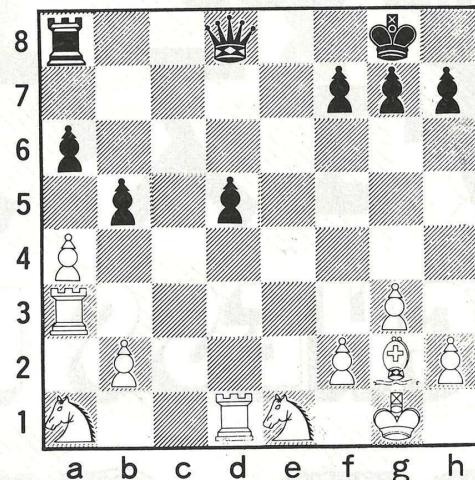
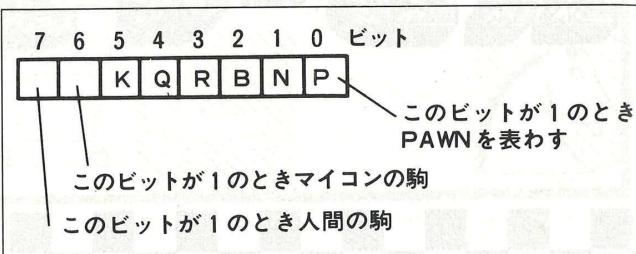


図3 駒コード



-	0	1	2	3	4
	5	6	7	8	DEL
CR	A	B	C	K (0-0)	Q (0-0-0)
	D	E	F	G	H

写真5

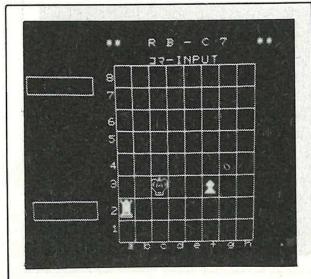


写真6

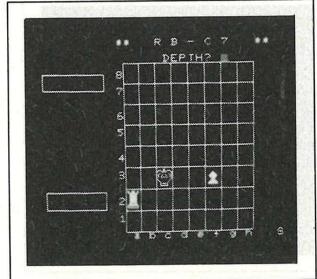


写真7

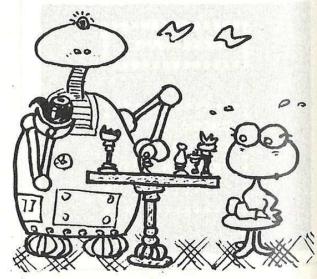
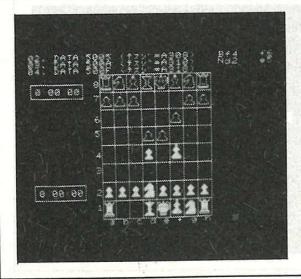


図4 メイティング・マテリアル

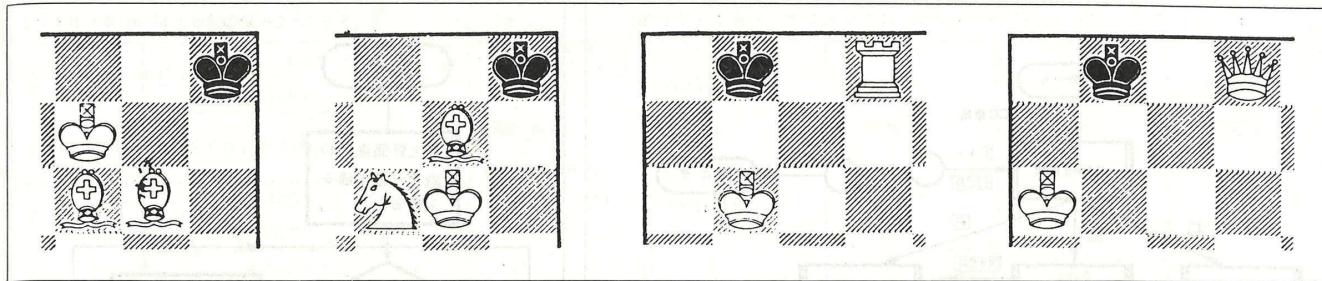
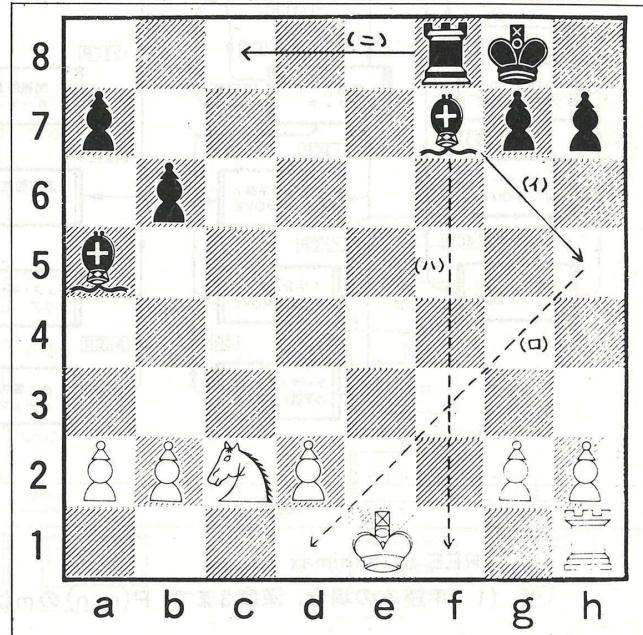


図9



— キーを押すと、1手分、駒の動きを戻してくれます。
/ キーは、1手戻すと同時に、序盤データのポインタも1手分戻し、訂正に使います。チェスの1手は将棋の2手にあたるので、これらのキーを使うときは、白黒の1手を動かした後に押す必要があります。

[π] キーは、ポインタの変更に使います。データを継ぎ足すのに使います。

なお、コマンドの実行には、[CR] を押すことが必要です。いろいろと図1を見ながら試してみてください。

5 チェスのプログラム

チェスのプログラムを作る場合、まず、チェス・ボードを用意します。これは適当なRAM上に決めるのですが、今回はA0mn番地(m, n = 0 ~ 7)の64バイトを使っています。次に、駒コードも適当に決めて、チェス盤のRAM上に並べます。そして、実際に駒をあれこれ動かしてみて、次の1手を決めてくるわけです。

たとえば、1.5手読みの場合、まず、自分の駒を動かし、次に相手の駒を動かし、そして、もう1回、自分の駒を動かして初めて、その1.5手読みの評価点が決まります。そして、この評価点の最大のものを、次の1手として駒を動かしてくるのです。

6 評価点の種類

では、どのようにして評価点を付けていくのかを考えみたいと思います。これらの判断基準を多く使えば、それだけ良いプログラムができるわけですが、逆に処理速度は遅くなってしまいます。また、先を何手も深く読めば読むほど、プログラムは強くなり、それにより不必要になってくる判断基準も出てきます。

GAIN：駒の獲得

相手の駒を取ったときに与える評価点で、最優先に近い点を与えます。一般に、駒の価値は、P : N : B : R : Q = 1 : 3 : 3 : 5 : 9というのがよく知られていますが、このプログラムでは1 : 3 : 3 : 7 : 15を使っています。これは、実際に走らせてみて調整した比率です。

CONCENTRATION：駒の集結

たとえば、図9の場合、f7のビショップを矢印(i)のように動かしたとき、ビショップのきき筋が点線(iv)のように通り、キングは動きにくくなります。また、ルークのきき筋も点線(iv)のように通り、一層キングは動きにくくなります。このような手に与える評価点がこれです。

この判定はプログラムの中で次のようにして見つけます。たとえば、ビショップの場合、まず(i)のように動かしま

す。次に白のキングを動かしてみます。すると、点線の通っているマス目でチェックを受けることになります。この場合4回です。この回数に比例して、この評価点は高くなるわけです。

ATTACK：攻撃

図9において、ルークを矢印(ii)のように動かすことは、C2のナイトを攻撃したことになります。当然、逃げられて駒を取ることはできないのですが、それなりの有効点を与えるわけです。

STRUCTURE：構成

a7のポーンがb6のポーンを守っています(図9)。また、b6のポーンとa5のビショップとは互いに守りあっています。こういう手が積み重なって、堅固な構成ができるわけです。

POSITION：駒位置

(i)絶対駒位置

たとえば、将棋のプログラムなどで、穴熊を使うとか、矢倉を使うとかのときに、その動きに評価点を付けます。

(ii)相対駒位置

これは非常に面白い評価点なのですが、処理時間が多く取りそうです。たとえば、ルークを二重に使うダブル・ルーク(将棋でいう二枚飛車)、囲碁などで、大桂馬に何点、小桂馬に何点というように付けたり、簡単な詰め碁の型やコンビネーションに評価点を与えます。

チェス・プログラム

図 1

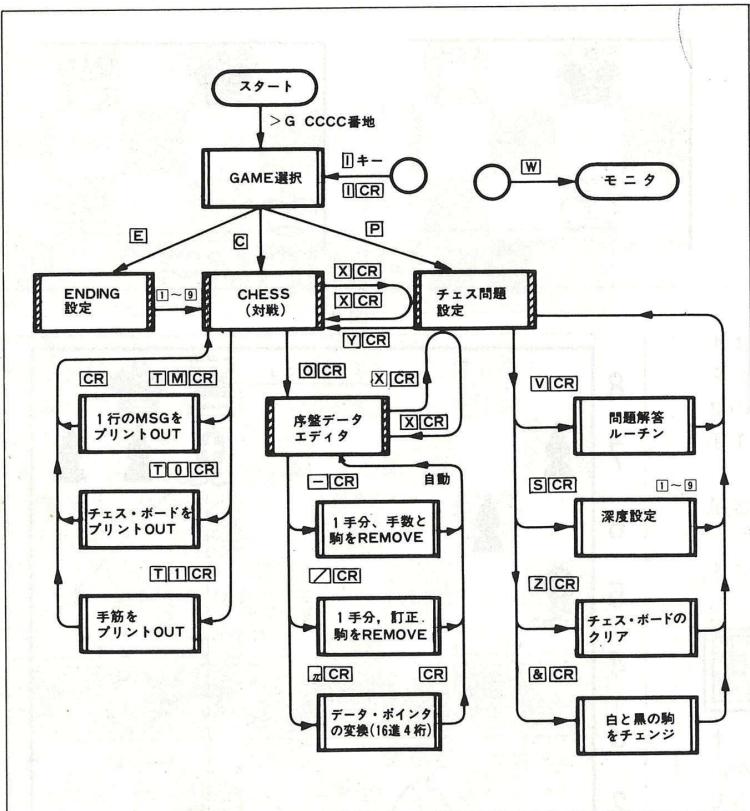
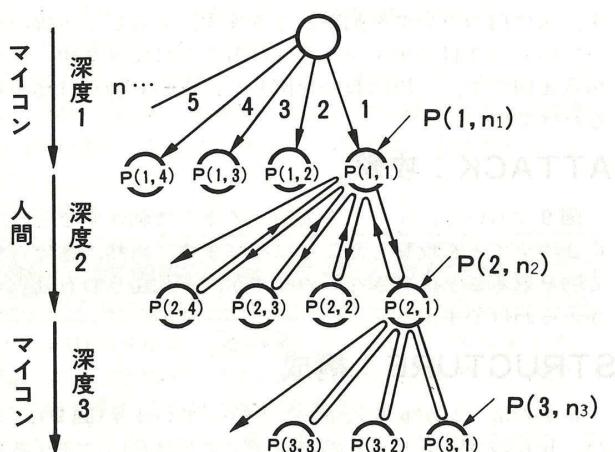


図 10

TREE of minimax
(1.5手読みの場合、深度3まで。 $P(m, n)$ のmは深度)

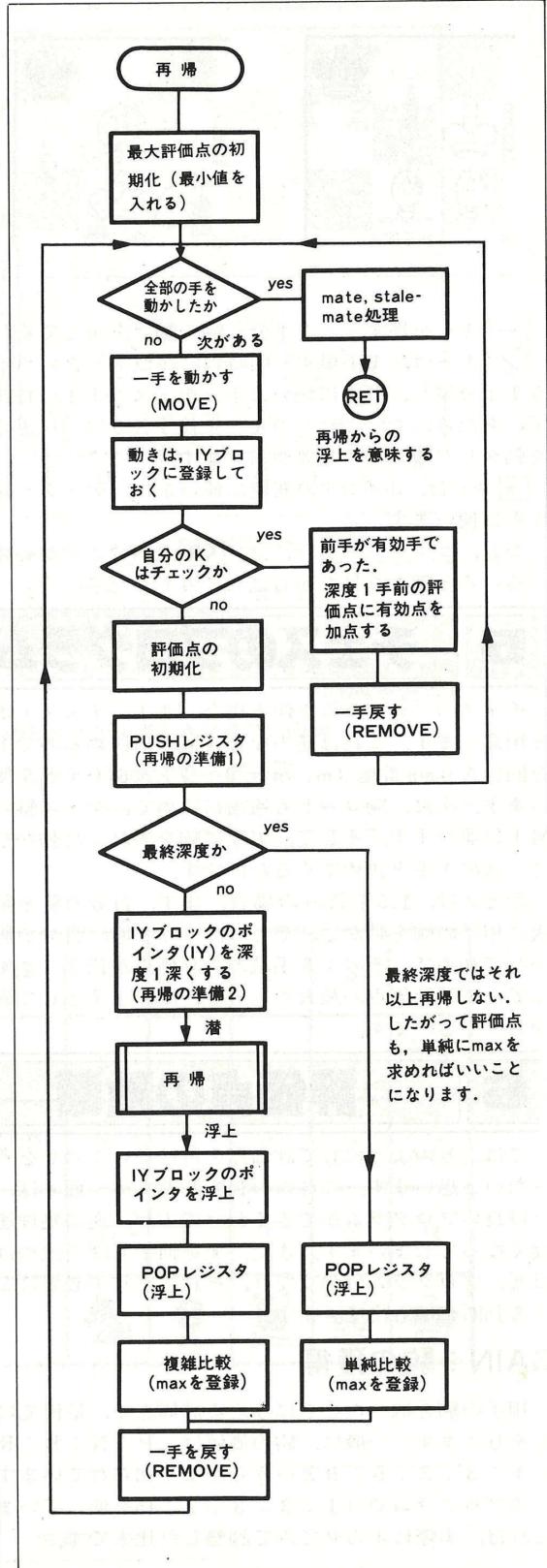


$P(1, \text{max})$ は、真の $P(1, n_1)$ の中で最大の値を取る手を意味する。
 真の $P(1, n_1) = P(1, n_1) - \{P(2, \text{max}) - P(3, \text{max})\}$
 $= P(1, n_1) - P(2, \text{max}) + P(3, \text{max})$

その他

キャスリングなどの特殊な手に与えます。他にもいろいろ考えられると思いますが、いま述べた評価点の付け方が主流になると思います。他には、駒の動きやすさを考慮したMOVINGやDEFENCE関係がありそうです。

図 8



7 Tree of mini-max

mini-maxに関する正式な知識はないのですが、以下に述べるのがそれだと思います(図 8, 図 10, 図 11)。

まず、マイコンで n 手読みのチェス・プログラムを作ろうとする場合、メモリ容量の制限や、同じようなルーチンをいくつも作る労を省くために、再帰型のmini-maxルーチ

図11

1 Y ブロック(16バイト単位のワーク・エリア)	
1 Y + 0 0	L(0) 動かす駒の下位番地
0 1	BOX(0) その駒
1 Y + 0 2	L(N) 駒の目的地の番地
0 3	BOX(N) そのマス目にある駒
1 Y + 0 4	対応する深度
0 5	フラグ
0 6	アンパサン番地
	その他
1 Y + 0 A	評価点
0 B	"
1 Y + 0 C	その深度における
0 D	最大評価点
1 Y + 0 E	L(0) max
0 F	L(N) max

ンを作る必要があります(図8)。

1.5手読みを前提とした場合、図10のように(マイコン)(人間)(マイコン)と、3回、駒を動かしてみることになります(チェスの1手は将棋の2手)。

まず、1回、駒を動かすわけですが、この動きには次の3つの要素が含まれています。

- ①どの駒を動かすか?
- ②どの方向に動かすか?
- ③その方向に何ステップ動かすか?

さて、最初の動きP(1,1)によって得られる評価点は真的評価点ではありません。なぜなら、深度3まで読むのが1.5手読みだからです。そのため、とりあえずこの評価点をワーク・エリアに登録しておく必要があります。これが、図11の1Yブロックです。この1Yブロックは、深度1用、深度2用…とn手分必要です(図12)。そして、再帰するたびに、その深度に合わせたブロックを使います。

結局、真的評価点P(1,n)は図10の下の式で求まることがあります。

8

$\alpha-\beta$ Algorithm(枝切り)

枝切りに関しては、私の力量不足から、このプログラムの中では具体化できなかったというのが事実です。ただ1つ使った枝切りは、1手動かしたときに自分のキングがチェックになったときです。抜き王手や、相手のきき筋に自

図12

1 Y ブロック用エリア(A100~16バイト単位)

↓ 1 Y レジスタでポイントする。深度(再帰回数)と呼応して動く。

A 100	A 110	A 120	A 130	A 140
深度0用 (人間手用の ワークエリア)	深度1 (マイコン手 minimax 用エリア)	深度2	深度3	深度4

図6

図5

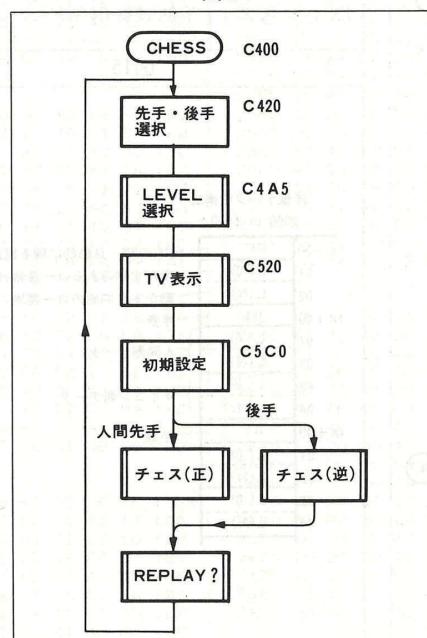
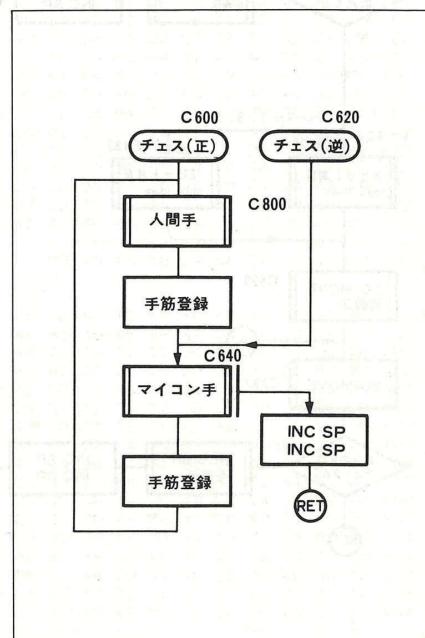


図7



分のキングを動かしたときがこれです。図9で、ポーンをd3へ動かしたときです。もし、このように動かしたら、当然相手はキングを取って勝つてしまうので、この手は禁じ手となり、また、この手以後の深度の手は考慮に入れる必要がなくなるわけで、これが技切りの一つとなります。

注) 入間側の入力に関して、禁止手のチェックを行なつていません(キングが一度動いた後でもキャスティングできる)。打ち手の側で判断してください。



Computer Chess

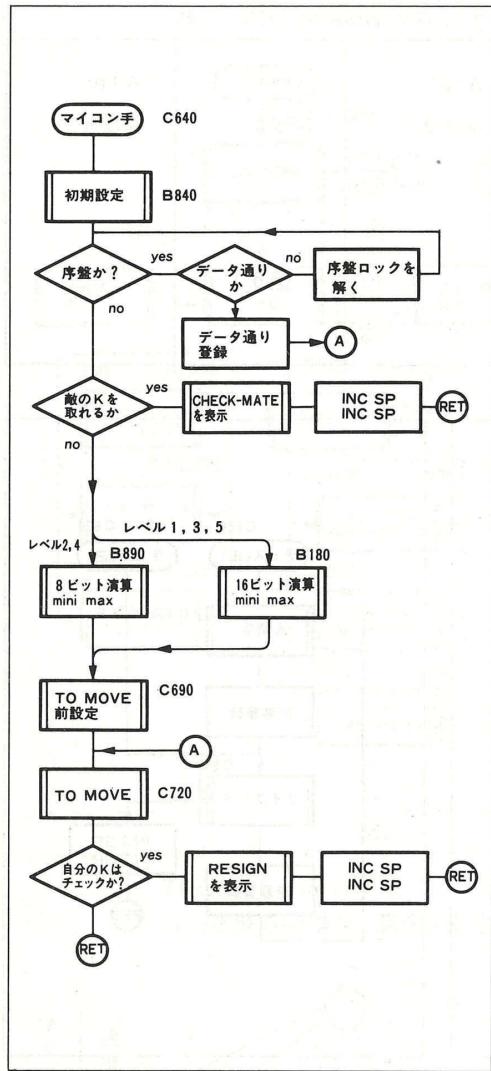
欧米ではコンピュータのチェス大会が盛んなようです。実態は知りませんが、BYTE誌などに載っているんじゃないでしょうか。見たことは、ないので解りませんが。

以前、Sargon 2.5というチェス・プログラムについて少し話したことがあります。Sargon 3ということができているようです。これはおそらく3手先を読むのだと思うのですが、問題なのは、3手先をどの程度の時間で読むのかということと、どんなCPUで何MHzで走っているのかということです。

いま、僕が残念に思っているのは、チェスのプログラムを作ったのはいいのですが、どの程度の実力を持ったプログラムなのか解らないということなのです。このプログラムがどの程度の実力かを知る方法はただ1つです。つまり、他のチェス・プログラムと戦わせてみるということです。

チェス・プログラム

図7



TIやAPPLEにはチェスのソフトがあるらしい、ぜひ戦わせてみたいものです。

さて、もう一度話を戻します。

Bellという人がMachine Plays Chessという本を書いています。この人の作ったチェス・プログラムは、1980年の世界コンピュータ・チェス選手権で優勝したそうです。

Bellというと、あなたも何かを連想するでしょう。そうです、Bell研究所を連想してしまいます。しかし、名前が同じだからと言つて現実検討抜きで結論付けるのは、早合点というものですから、僕は結論を延期していました。しかし、どうやらそのようですね（確信はありません）。Bellというプログラム・ネームで、マシンはかなり高速で走る大型コンピュータのようです。

10

あとがき

チェスのプログラムを作つて、一番感じるのは処理速度で、8ビットのCPUでは限界を感じてしまいます。早く、8MHzで走る68000でプログラムを組んでみたいものです。このとき、普通の人はまったく勝てなくなると思います。

人間同志のチェス大会はどうでしょうか？ FIDE（世界チェス連盟）が2年に一度、チェス・オリンピックを開い

図13

レジスタ名	使 用 目 的
B レジスタ	深度
DE レジスタ	進行方向値のポインタ
HL レジスタ	チェス盤のマス目のポインタ
B'C' レジスタ	IY レジスタ加算用
D'E' レジスタ	" 減算用
IY レジスタ	IY ブロックのポインタ
IX レジスタ	手筋登録用など

図14

B 09 E	キング番地(マイコン側)
B 09 F	" (人間側)
B 0A 0	フラグ：人間先手……00
B 0A 1	LEVEL
B 0A 2	DEPTH (深度)
B 0A 3, 4	使用時間(マイコン側)
B 0A 6, 7	" (人間側)
B 0A 8	フラグ：序盤時……00
B 0A 9, A	ポイント番地(序盤データ)
B 0AB	現在の手数
B 0AC, D	ポイント番地(手筋ストア・エリア)
B 0AE	手数(プリント用)

図15

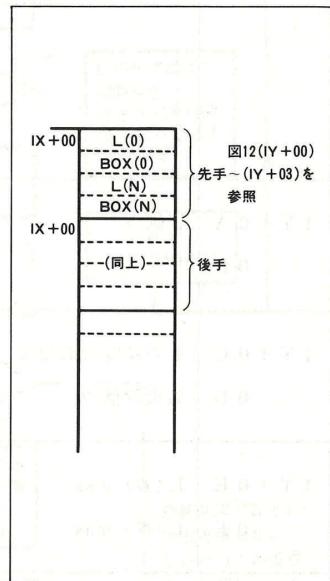
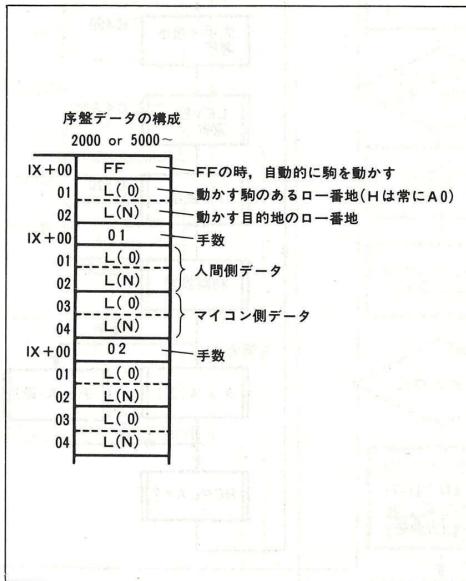


図18
評価点の例 (16ビット演算を使う場合)

駒	評価点	GAIN	DEFENCE	STRUCTURE	ATTACK	CONCENTRATION
PAWN	00A0	0000	PAWNのみ P × 0001	0000	0000	
NIGHT	0200	0000		0000	0000	
BISHOP	0202	0001		0000	0000	
ROOK	0409	0003		000C	0018	マス目1につき
QUEEN	0900	0005			001E	001E
KING	1400 + α	000A			001E + β	(キングのみ)

注： α や β は深度によって値が変わること、たとえば、KINGは先に取った方が勝ちだから、深度が浅い程評価点は高くなる。

ています。日本には将棋や囲碁があるのでほとんど知りませんが、かなり大きな国際大会です。1980年は地中海のマルタで開かれ、82箇国が参加しました。1982年はイスラエルで開かれる予定で、100箇国が参加するのではないかと言われています。

日本でチェスを楽しんでいるのは500人ほどしかいません。そのうち、研究熱心な人は100人程度です。つまり、その気になれば、あなたも日本代表として世界大会への参加が可能なのです。特に、19才以下の世界ジュニア選手権大会ではそうです。

日本代表は毎回10名程度で、日本チェス協会の選抜大会で決められます。脳細胞に自信のある人は、試してみればどうでしょうか。