

アニメーション・システムについて

名大・工 榎本 美久・山中 謙一

(1988年9月26日受理)

近年、パターン形成の動力学の問題が盛んに研究されています。言うまでもなく、パターン形成の動力学は、非平衡・非線形物理の主要テーマの一つであり、物性物理は言うに及ばず、宇宙・天体、化学反応、流体、生体等に現われる現象をも含んでいます。

この分野においては、界面等の位相的欠陥と呼ばれている概念を使ったモデル方程式の解析的研究のほかに、モデル方程式のシュミレーションも重要な研究手段です。例えば、シュミレーションの結果と電子顕微鏡による”その場観察”の結果とを比較・検討することは、モデルの正当化・解析の確認のために役立つと思われます。

筆者らは、さらに進んで、シュミレーションの結果をあたかも実際の現象を直接観察しているかのように動かしてみたい(アニメーション化)と考えました。この方向としては、広島大学理学部数学科の三村・小林両先生のグループが大がかりなビデオ撮影等を行っています。

我々は、もっと手軽にアニメーション化できないかと考え、そのためのマイコン用のソフトを作成しました。このソフトは、数枚のセル(1枚のセルは 128×128 の正方格子で出来ている。各格子点は0~6の整数で色づけされている。)の絵(シュミレーションの結果)を、マシン語を使ってマイコンの画面上に連続的にしかも瞬時に写し出すことによってアニメーション化するものです。なおマイコンはNECの98シリーズを使用しました。VMで45セル、それ以上の機種では75セルの絵を処理できます。筆者らはいくつかのモデル方程式についてアニメーションを作ってみました。45セルでも充分鑑賞に耐えるものができます。しかし、単純なソフトなので、まだまだ改良すべき点があると思います。御意見、御批判下されば幸いです。尚、このソフトの原型は、九州大学理学部物理の安田敬君によって作成されたものです。ここに感謝の意を表します。

システム(VM用)は2つの部分からなっています。以下、各部分のプログラムを簡単に説明します。最後にプログラムのリストを載せました。

128×128 ドットの1枚のセルは、ディスプレイ1画面に15枚収めることが出来ます。セル1枚につきメモリが1800Hバイト必要ですので、VMで使えるメモリを考えると3画面、45枚が最大限です。(増設RAMを用いれば、1680Kバイトに付き

1 画面増加できます。)

メインプログラムは以下の2本です。(図1)

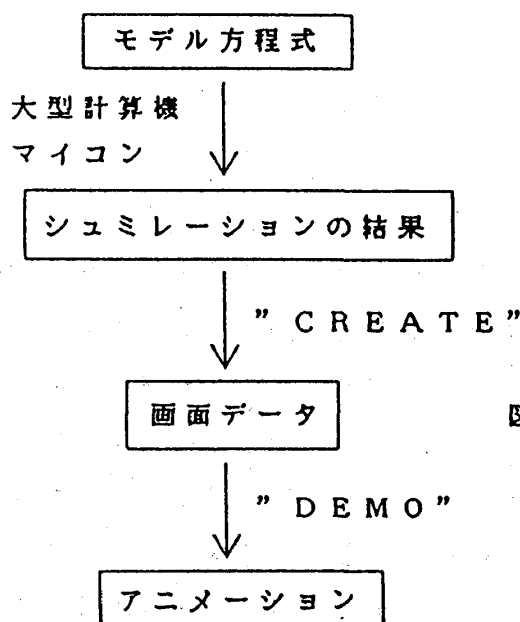


図1. アニメーション作成のための概略図

<1>" CREATE"

シュミレーションで得られたデータファイルより<2>で用いる画面データを作成するプログラム。

<2>" DEMO"

<1>で作成した画面データを用いて各セルをディスプレイ上の同一場所に順番に表示することによりアニメーションにみせるプログラム。

<1>について:

シュミレーションで得られたデータは、0から6までの整数1つで1ドットを表したシーケンシャルファイルです。このデータから、画面に15セルを描画し、それを1単位としてファイルを作成します。シーケンシャルファイルより1ドットずつPSETコマンドで描画するため、1単位の画面データを作成するのに30分強の時間が必要です。作成した画面データを読み込むときには1単位10秒ほどで済みます。

< 2 >について：

	縦	横	
1枚のセル	128	128	ドット
	= 128	× 16	バイト = 800H
3原色分必要なので1枚のセルに必要な総バイト数	= 1800H		

メモリアドレス	セルNo.	色	ファイル名
1C000H	1 - 15	(青)	OST1. B
23800H		(赤)	OST1. R
2B000H		(緑)	OST1. G
32800H	16 - 30	(青)	OST2. B
3A000H		(赤)	OST2. R
41800H		(緑)	OST2. G
49000H	31 - 45	(青)	OST3. B
50800H		(赤)	OST3. R
58000H		(緑)	OST3. G

ある番地Xを指定した時に、「横：XからX+16，縦：XからX+126」のメモリをグラフィックRAMの指定した番地へ転送すれば画面上に1つのセルが現われます。これを繰り返せば、あたかも絵が動いているように見えます。

(サブプログラムの説明) (図2)

MAIN. m : 上記の転送を行う。

MACRO. m : 1ドットを2×2ドットに拡大して上記の転送を行う。

(結果として256×256ドットの画面が得られる。)

TRANS. m : メモリに読み込んだ1画面(15枚分)を一度に表示する。

(主な変数の内容)

X(1-15) : セル15枚一組になったデータにおいて、その先頭からみた各々のセルの、画面上での相対座標

Y(1-15) : アニメ画面の大きさを変えるフラグ。SWON=1で大、SWON=0で小画面

A(1-3) : メモリ上にある画面データのセグメント

SG : マシン語サブルーチンWOコールする際にデータのセグメントを上記のA(1-3)を代入することによって指定する。

- A D 1 : 指定した1枚のセルのメモリでの先頭アドレス
A D 2 : A D 2 によって指定されたグラフィック R A M のアドレスにセルの画面データを転送する。

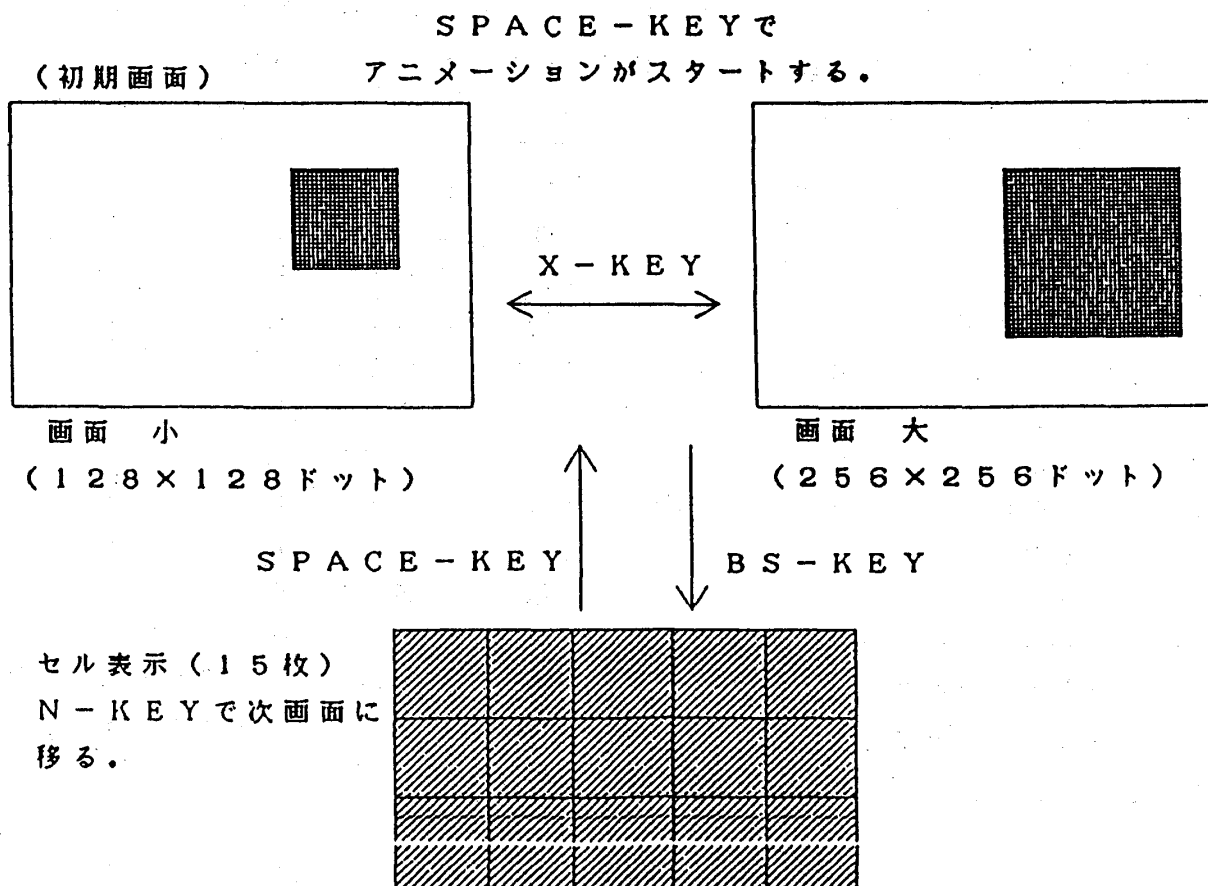


図2. プログラム"DEMO"における主なキー操作

プログラム "CREATE"

```

990 ' "CREATE"
1000 '-----フレームの位置
1010 DIM X( 15 ), Y( 15 )
1020 FOR I = 1 TO 15
1030 ID = I - 1
1040 X( I ) = ( ID ¥ 3 ) * 128
1050 Y( I ) = ( ID MOD 3 ) * 128
1060 NEXT I
2000 '-----初期設定
2010 CONSOLE , , 0 , 1
2020 WIDTH 80 , 25
2030 SCREEN 3 , 0 , 0 , 1
3000 '-----パラメーターの入力
3010 *INIT
3020 CLS 3
3030 INPUT "データドライブ" , IDR$
3040 INPUT "データファイル名" , IFILE$
3050 INPUT "スタートファイルナンバー" , IFNUM$
3060 INPUT "出力ファイルドライブ" , ODR$
3070 INPUT "出力ファイル名" , OFILE$
3080 INPUT "出力ファイルナンバーオプション" , OFNUM$
3090 INPUT "何画面作成しますか" , FRAMES$
3100 INPUT "1画面作成後、ポーズをとりますか" , PAUSE$
3110 IF IDR$ = "2" OR IDR$ = "B" OR IDR$ = "b" OR IDR$ = "B:" OR IDR$ = "b:" THE
N IDR$ = "2:" ELSE IDR$ = ""
3120 IF ODR$ = "2" OR ODR$ = "B" OR ODR$ = "b" OR ODR$ = "B:" OR ODR$ = "b:" THE
N ODR$ = "2:" ELSE ODR$ = ""
3130 IF IFNUM$ > ":" OR IFNUM$ < "0" THEN IFNUM$ = "1"
3140 IF OFNUM$ > ":" OR OFNUM$ < "0" THEN OFNUM$ = "1"
3150 IF FRAMES$ > "3" OR FRAMES$ < "1" THEN FRAMES$ = "3"
3160 IF PAUSE$ = "N" OR PAUSE$ = "n" THEN MES$ = "1画面作成度にポー
ズをとります。" : ELSE MES$ = "1画面作成度にポー
ズをとります。"
3170 FD$ = IDR$ + IFILE$ + IFNUM$ + ".DAT"
3180 B$ = ODR$ + OFILE$ + OFNUM$ + ".B"
3190 R$ = ODR$ + OFILE$ + OFNUM$ + ".R"
3200 G$ = ODR$ + OFILE$ + OFNUM$ + ".G"
3210 CLS
3220 PRINT " データを " ; FD$ ; " より読み込みます。"
3230 PRINT " 画面データを " ; B$
3240 PRINT " " ; R$
3250 PRINT " " ; G$ ; " より出力します。"
3260 PRINT FRAMES$ ; " 画面"
3270 PRINT MES$
3280 PRINT : INPUT "よろしいですか" , A$
3290 IF A$ = "N" OR A$ = "n" THEN *INIT
3300 COUNT = VAL( FRAMES )
4000 '-----メイン・ルーチン
4010 *LOOP
4020 COUNT = COUNT - 1
4030 CLS 3
4040 FOR L = 1 TO 3
4050 OPEN FD$ FOR INPUT AS #1
4060 INPUT #1,C:IF C=0 THEN 4090 : REM-----DATA NI ERROR GA ARUTOKI NI 1 KAI
4070 CLOSE : REM-----KARAYOMI WO SURU.TADASHI 2KOIJYO
4080 OPEN FD$ FOR INPUT AS #1 : REM-----NO ERROR HA KANTI SINAI
4090 FOR K=1 TO 5
4100 OFFSET=(L-1)*5
4110 GOSUB *GRAF
4120 NEXT K
4130 CLOSE
4140 IFNUM$ = CHR$( ASC (IFNUM$ ) + 1 )
4150 FD$ = IDR$ + IFILE$ + IFNUM$ + ".DAT"
4160 NEXT L
5000 '-----画面データのセーブ

```

```

5010 DEF SEG = 8HA800 : BSAVE B$ , 0 , 8H7800
5020 DEF SEG = 8HB000 : BSAVE R$ , 0 , 8H7800
5030 DEF SEG = 8HB800 : BSAVE G$ , 0 , 8H7800
5040 OFNUM$ = CHR$( ASC( OFNUM$ ) + 1 )
5050 B$ = ODR$ + OFILES$ + OFNUM$ + ".B"
5060 R$ = ODR$ + OFILES$ + OFNUM$ + ".R"
5070 G$ = ODR$ + OFILES$ + OFNUM$ + ".G"
5080 IF COUNT = 0 THEN *COMP
5090 IF MES$ = "" THEN *LOOP
5100 CLS 3
5110 PRINT "1画面作成終了しました。ポーズ中です。"
5120 PRINT "なにかキーを押して下さい。"
5130 A$ = INKEY$ : IF A$ <> "" THEN *LOOP ELSE 5130
6000 '-----終了
6010 *COMP
6020 CLS 3
6030 PRINT " 全作業終了しました。"
6040 PRINT " 最終画面データは " ; B$
6050 PRINT " " ; R$
6060 PRINT " " ; G$ ; " です。"
6070 END
7000 '-----描画
7010 *GRAF
7020 FOR I = 1 TO 128 : X = X( K + OFFSET ) + I
7030 FOR J = 1 TO 128 : Y = Y( K + OFFSET ) + J
7040 INPUT #1 , C
7050 PSET ( X , Y ) , C
7060 NEXT J , I
7070 LINE ( X( K + OFFSET ) , Y( K + OFFSET ) ) - STEP( 127 , 127 ) , 6 , B
7080 RETURN

```

プログラム " DEMO "

```

990 ' "DEMO"
1000 CLEAR ,8H1B00
1010 DEF SEG = 8H1B80 : BLOAD "MAIN.m"
1020 DEF SEG = 8H1BA0 : BLOAD "TRNS.m"
1030 DEF SEG = 8H5F80 : BLOAD "MACRO.m"
1040 DEF SEG = 8H1C00 : BLOAD "1:OST1.B"
1050 DEF SEG = 8H2380 : BLOAD "1:OST1.R"
1060 DEF SEG = 8H2B00 : BLOAD "1:OST1.G"
1070 DEF SEG = 8H3280 : BLOAD "1:OST2.B"
1080 DEF SEG = 8H3A00 : BLOAD "1:OST2.R"
1090 DEF SEG = 8H4180 : BLOAD "1:OST2.G"
1100 DEF SEG = 8H4900 : BLOAD "1:OST3.B"
1110 DEF SEG = 8H5080 : BLOAD "1:OST3.R"
1120 DEF SEG = 8H5800 : BLOAD "1:OST3.G"
1200 '-----
1210 CONSOLE , , 0 , 1
1220 DEFINT A-Z
1230 WIDTH 80 , 25
1240 CLS 3
1250 DIM X( 15 ) , Y( 15 )
1300 '-----
1310 FOR I = 1 TO 15
1320 ID = I - 1
1330 X( I ) = ( ID * 3 ) * 128
1340 Y( I ) = ( ID MOD 3 ) * 128
1350 NEXT I
1400 '-----
1410 L1$ = "DISPLAYING FLAME No.1 - No.15"
1420 L2$ = "DISPLAYING FLAME No.16- No.30"
1430 L3$ = "DISPLAYING FLAME No.31- No.45"
1440 SWON = 0 : SWOFF = 1
2000 '-----
2010 *DO
2020 X2 = 340 : Y2 = 50
2030 SCREEN 3 , 0 , 0 , 1 : CLS 3
2040 A(1)= 8H1C00 : A(2)= 8H3280 : A(3)= 8H4900
2500 '-----
2510 *MAIN
2515 LOCATE 0 , 25 : PRINT "[X] change cell size [BS] show 15 cells [SPACE]
start moving";
2520 SG=A(1)
2530 X1 = X( 1 ) : Y1 = Y( 1 )
2540 GOSUB *GT
2550 A$=INKEY$
2560 IF A$=CHR$(8) THEN *HOLD
2570 IF A$="X" OR A$="x" THEN SWAP SWON,SWOFF :GOTO 2030
2580 IF A$="" THEN 2550
2590 FOR M=1 TO 3
2600 SG = A(M)
2610 FOR I = 1 TO 15
2620 X1 = X( I ) : Y1 = Y( I )
2630 GOSUB *GT
2640 NEXT I
2650 NEXT M
2660 FOR W = 1 TO 10000 : NEXT W
2670 GOTO *MAIN
2680 END
3000 '-----

```

```

3010 *HOLD
3020 Z = 0
3030 SCREEN 3 , 0 , 1 , 17
3040 DEF SEG = &H1BA0 : CLS 3
3050 TRNS = 0 : CALL TRNS( A( 1 ) )
3060 LOCATE 21 , 10 : COLOR 6 : PRINT L1$ : COLOR 5
3065 LOCATE 0,25 : PRINT "[PJ pick 5 cells [W] zoom 1 cell [N] next 15 cells
[SPACE] animation";
3070 A$ = INKEY$
3080 IF A$ = " " THEN *DO
3090 IF A$ = "X" OR A$ = "x" THEN SWAP SWON , SWOFF : GOTO *DO
3100 IF A$ = "P" OR A$ = "p" THEN *PICK
3110 IF A$ = "W" OR A$ = "w" THEN *ZOOM
3120 IF A$ = "N" OR A$ = "n" THEN SWAP L1$ , L2$ : SWAP L2$ , L3$ : SWAP A( 1 ) ,
A( 2 ) : SWAP A( 2 ) , A( 3 ) : GOTO 3040
3130 GOTO 3070
4000 '-----
4010 *GT
4020 AD1 = X1 ¥ 8 + Y1 * 80
4030 AD2 = X2 ¥ 8 + Y2 * 80
4040 MAIN = 0
4050 DEF SEG = &H1B80 + &H4400 * SWON : CALL MAIN( AD1 , AD2 , SG )
4060 RETURN
5000 '-----
5010 *PICK
5020 SCREEN 3 , 0 , 0 , 1 : CLS 3
5030 IF SWON = 1 THEN SWAP SWON , SWOFF
5040 INPUT "FRAME No." , F( 1 ) , F( 2 ) , F( 3 ) , F( 4 ) , F( 5 )
5050 FOR I = 1 TO 5
5060 IF F( I ) < 1 OR F( I ) > 45 THEN 5100
5070 GOSUB *ADRS
5080 X2 = 128 * ( I - 1 ) : Y2 = 128
5090 GOSUB *GT
5100 NEXT I
5110 A$ = INKEY$ : IF A$ = "" GOTO 5110
5120 GOTO *HOLD
6000 '-----
6010 *ADRS
6020 N = ( F( 1 ) - 1 ) ¥ 15
6030 SG = &H1C00
6040 IF N = 1 THEN SG = &H3280
6050 IF N = 2 THEN SG = &H4900
6060 M = 1 + ( F( 1 ) - 1 ) MOD 15
6070 X1 = X( M ) : Y1 = Y( M )
6080 RETURN
7000 '-----
7010 *ZOOM
7020 SCREEN 3 , 0 , 0 , 1 : CLS 3
7030 INPUT "FRAME No." , F( 1 )
7040 I = 1 : GOSUB *ADRS
7050 X2 = 340 : Y2 = 50
7060 AD1 = X1 ¥ 8 + Y1 * 80
7070 AD2 = X2 ¥ 8 + Y2 * 80
7080 DEF SEG = &H5F80
7090 MAC = 0 : CALL MAC ( AD1 , AD2 , SG )
7100 A$ = INKEY$ : IF A$ = "" GOTO 7100
7110 GOTO *HOLD
9999 END

```

プログラム "MAIN. m"

```

0000 8B37      MOV     SI,[BX]
0002 8B4702    MOV     AX,02[BX]
0005 8EC0      MOV     ES,AX
0007 26        ES:
0008 8B04      MOV     AX,[SI]
000A 50        PUSH    AX
000B 90        NOP
000C 90        NOP
000D 90        NOP
000E 8B7704    MOV     SI,04[BX]
0011 8B4706    MOV     AX,06[BX]
0014 8EC0      MOV     ES,AX
0016 26        ES:
0017 8B3C      MOV     DI,[SI]
0019 8B7708    MOV     SI,08[BX]
001C 8B470A    MOV     AX,0A[BX]
001F 8EC0      MOV     ES,AX
0021 26        ES:
0022 8B34      MOV     SI,[SI]
0024 B800A8    MOV     AX,A800
0027 8ED8      MOV     DS,AX
0029 58        POP     AX
002A 90        NOP
002B 90        NOP
002C 90        NOP
002D 8EC0      MOV     ES,AX
002F 90        NOP
0030 B90300    MOV     CX,0003
0033 51        PUSH    CX
0034 56        PUSH    SI
0035 57        PUSH    DI
0036 B98000    MOV     CX,0080
0039 51        PUSH    CX
003A B90800    MOV     CX,0008
003D BB0000    MOV     BX,0000
0040 26        ES:
0041 8B00      MOV     AX,[BX+SI]
0043 53        PUSH    BX
0044 01FB      ADD     BX,DI
0046 8907      MOV     [BX],AX
0048 5B        POP     BX
0049 83C302    ADD     BX,0002
004C E2F2      LOOP    0040
004E 59        POP     CX
004F 83C650    ADD     SI,0050
0052 83C750    ADD     DI,0050
0055 E2E2      LOOP    0039
0057 5F        POP     DI
0058 5E        POP     SI
0059 59        POP     CX
005A 8CD8      MOV     AX,DS
005C 050008    ADD     AX,0800
005F 8ED8      MOV     DS,AX
0061 8CC0      MOV     AX,ES
0063 058007    ADD     AX,0780
0066 8EC0      MOV     ES,AX
0068 E2C9      LOOP    0033
006A CF       IRET

```

プログラム "MACRO. m"

0000 8B37	MOV	SI,[BX]
0002 8B4702	MOV	AX,02[BX]
0005 8EC0	MOV	ES,AX
0007 26	ES:	
0008 8B04	MOV	AX,[SI]
000A 50	PUSH	AX
000B 8B7704	MOV	SI,04[BX]
000E 8B4706	MOV	AX,06[BX]
0011 8EC0	MOV	ES,AX
0013 26	ES:	
0014 8B3C	MOV	DI,[SI]
0016 8B7708	MOV	SI,08[BX]
0019 8B470A	MOV	AX,0A[BX]
001C 8EC0	MOV	ES,AX
001E 26	ES:	
001F 8B34	MOV	SI,[SI]
0021 B800A8	MOV	AX,A800
0024 8ED8	MOV	DS,AX
0026 58	POP	AX
0027 8EC0	MOV	ES,AX
0029 B90300	MOV	CX,0003
002C 51	PUSH	CX
002D 56	PUSH	SI
002E 57	PUSH	DI
002F B98000	MOV	CX,0080
0032 51	PUSH	CX
0033 B90800	MOV	CX,0008
0036 BB0000	MOV	BX,0000
0039 26	ES:	
003A 8B00	MOV	AX,[BX+SI]
003C 51	PUSH	CX
003D 53	PUSH	BX
003E D1E3	SAL	BX,1
0040 01FB	ADD	BX,DI
0042 E82D00	CALL	0072
0045 88E0	MOV	AL,AH
0047 83C302	ADD	BX,0002
004A E82500	CALL	0072
004D 5B	POP	BX
004E 59	POP	CX
004F 83C302	ADD	BX,0002
0052 E2E5	LOOP	0039
0054 59	POP	CX
0055 83C650	ADD	SI,0050
0058 81C7A000	ADD	DI,00A0
005C E2D4	LOOP	0032
005E 5F	POP	DI
005F 5E	POP	SI
0060 59	POP	CX
0061 8CD8	MOV	AX,DS
0063 050008	ADD	AX,0800
0066 8ED8	MOV	DS,AX
0068 8CC0	MOV	AX,ES
006A 058007	ADD	AX,0780
006D 8EC0	MOV	ES,AX
006F E2BB	LOOP	002C
0071 CF	IRET	
0072 BA0000	MOV	DX,0000
0075 D0E0	SAL	AL,1
0077 7304	JAE	007D
0079 81C2C000	ADD	DX,00C0
007D D0E0	SAL	AL,1
007F 7303	JAE	0084
0081 83C230	ADD	DX,0030
0084 D0E0	SAL	AL,1

0086 7303	JAE	008B
0088 83C20C	ADD	DX,000C
008B D0E0	SAL	AL,1
008D 7303	JAE	0092
008F 83C203	ADD	DX,0003
0092 D0E0	SAL	AL,1
0094 7304	JAE	009A
0096 81C200C0	ADD	DX,C000
009A D0E0	SAL	AL,1
009C 7304	JAE	00A2
009E 81C20030	ADD	DX,3000
00A2 D0E0	SAL	AL,1
00A4 7304	JAE	00AA
00A6 81C2000C	ADD	DX,0C00
00AA D0E0	SAL	AL,1
00AC 7304	JAE	00B2
00AE 81C20003	ADD	DX,0300
00B2 8917	MOV	[BX],DX
00B4 895750	MOV	50[BX],DX
00B7 C3	RET	

プログラム " T R N S . m "

0000 8B37	MOV	SI,[BX]
0002 8B4702	MOV	AX,02[BX]
0005 8EC0	MOV	ES,AX
0007 26	ES:	
0008 8B04	MOV	AX,[SI]
000A 8EC0	MOV	ES,AX
000C B800A8	MOV	AX,A800
000F 8ED8	MOV	DS,AX
0011 B90300	MOV	CX,0003
0014 51	PUSH	CX
0015 B9003C	MOV	CX,3C00
0018 BB0000	MOV	BX,0000
001B 26	ES:	
001C 8B07	MOV	AX,[BX]
001E 3E	DS:	
001F 8907	MOV	[BX],AX
0021 83C302	ADD	BX,0002
0024 E2F5	LOOP	001B
0026 59	POP	CX
0027 8CD8	MOV	AX,DS
0029 050008	ADD	AX,0800
002C 8ED8	MOV	DS,AX
002E 8CC0	MOV	AX,ES
0030 058007	ADD	AX,0780
0033 8EC0	MOV	ES,AX
0035 E2DD	LOOP	0014
0037 CF	IRET	