FAST GRAPHICS POMPACK PRIMER (MICHEAL CRANFORD)

65 HIGH LEVEL GRAPHICS COMMANDS.
MORE THAN 10 TIMES FASTER THAN BASIC.
CROSS-HAIR (4010 STYLE) JOYSTICK CURSOR.
8 TIMES THE PACKING DENSITY (>10,000 VECTORS).
SEVERAL COMMANDS ADD CAPABILITY NOT POSSIBLE IN BASIC.

BENCHMARKS FROM 13.7X FASTER TO 127X FASTER THAN A 4051.
(BENCHMARKS FROM 4.5X FASTER TO 61X FASTER THAN A 4052.)

OVERVIEW :

THE MAIN FUNCTION OF FAST GRAPHICS IS TO SUPPLIMENT THE INTERACTIVE GRAPHICS CAPABILITY OF THE 4051. THIS INCLUDES SEVERAL COMMANDS FOR IMAGE MANIPULATION, WRITE-THRU (NON-STORE), AND PROGRAMMABLE SOUND GENERATION. SEVERAL 4051 BASIC DEFICIENCIES HAVE BEEN CORRECTED AS WELL.

IN ORDER TO IMPROVE THE SPEED AND STORAGE DENSITY OF GRAPHICS, FAST GRAPHICS PACKS IMAGES INTO VALID ASCII CHARACTER STRINGS. THIS RESULTS IN ABOUT EIGHT TIMES THE STORAGE CAPACITY FOR ANY COMPLEX IMAGES. THE NEW FORMAT TAKES 3 BYTES/VECTOR (1 BIT MOVE FLAG 10 BITS X & 10 BITS Y) INSTEAD OF 24 BYTES/VECTOR (X,Y,Z). IMAGES CAN BE CONVERTED FROM FLOATING POINT TO STRING FORMAT OR FROM STRING TO FLOATING POINT FORMAT WITH EASE. IMAGES CAN BE MANIPULATED & SAVED OR READ FROM MAG TAPE FILES IN STRING FORM. NOTE THAT A FREE ADVANTAGE OCCURS WHEN IMAGES ARE IN THE STRING FORMAT - ALL OF THE 4051 BASIC STRING FUNCTIONS CAN BE USED.

POWERFULL IMAGE MANIPULATION COMMANDS SUPPORT THE STRING FORMAT AND GIVE THE 4051 INCREDIBLE SPEED WHEN COMPARED TO STANDARD 4051 (AND 4052) BASIC GRAPHICS COMMANDS.

ALL IMAGE DISPLAY COMMANDS RESTORE THE GRAPHIC CURSOR TO THE POSITION IT HAD BEFORE THE CALL WAS EXECUTED. THUS THE GRAPHIC CURSOR AND THE JOYSTICK CURSOR ARE INDEPENDANTLY MAINTAINED.

NOTE THAT EXTENSIVE ERROR CHECKING EXISTS WITHIN FAST GRAPHICS.

COMMANDS :

THERE ARE FOUR DIFFERENT CATAGORIES OF FAST GRAPHICS COMMANDS. THESE CATAGORIES ARE :

1. LOCATION : USED TO LOCATE THE GRAPHICS CURSOR OR THE JOYSTICK CURSOR OR ANY POINT OR POINTS IN ANY IMAGE.

2. DISPLAY : USED TO DISPLAY CURSORS OR IMAGES OR TEXT.

3. CREATE/MODIFY : USED TO CREATE IMAGES OR MODIFY IMAGES.

4. CHARACTER : USED TO CREATE OR PRINT OR PLAY (SOUND) SPECIAL CHARACTER STRINGS.

NOTE THAT ALL OF THE STANDARD 4051 STRING FUNCTIONS CAN BE USED TO LOCATE A VECTOR OR TO REPLACE PART OF AN IMAGE OR TO EXTRACT PART OF AN IMAGE OR TO CONCATINATE IMAGES ETC.

COMMAND FORMS :

MOST OF THE COMMANDS HAVE FOUR DIFFERENT FORMS. THE DIFFERENT

FORMS HAVE DIFFERENT PREFIXES WITH IDENTICAL SUFFIXES.
THE DIFFERENT PREFIXES ARE :

A : DISPLAY SCREEN "ABSOLUTE"
R : DISPLAY SCREEN "RELATIVE"
G : "GRAPHICS" CURSOR RELATIVE
J : "JOYSTICK" CURSOR ABSOLUTE

NOTE THAT THE COMMAND CALL SYNTAX IS CONSISTANT AMONG THE DIFFERENT FORMS. I.E. THE JOYSTICK ABSOLUTE FORM SYNTAX IS IDENTICAL TO THE ABSOLUTE FORM SYNTAX WITH AN ADDITIONAL REFRESH COUNT PARAMETER AND KEYBOARD KEY PARAMETER.

AN ADDITIONAL FORM EXISTS FOR COMMANDS THAT DO NOT FOLLOW THE ABOVE RULES. THIS IS THE "SPECIAL" FORM.

COMMAND LIST :

LOCATION :

SPECIAL : BOUNDS LOCATE

ABSOLUTE : AGIN APOINT

RELATIVE : RGIN RPOINT

GRAPHICS : GGIN GPOINT

JOYSTICK : JGIN JPOINT

DISPLAY :

SPECIAL : DASHED DOTTED KABOOM RUBBER VERTEX

ABSOLUTE : ACROSS ADRAW ADOTS APRINT AINPUT

RELATIVE : RCROSS RDRAW RDOTS RPRINT RINPUT

GRAPHICS : GCROSS GDRAW GDOTS GPRINT GINPUT

JOYSTICK : JCROSS JDRAW JDOTS JPRINT JINPUT

CREATE/MODIFY :

SPECIAL : IMAGES CHANGE DEFINE POINTS TOGGLE

ABSOLUTE : AMOVE ASCALE ASHEAR ATAPER AROTATE

RELATIVE : RMOVE RSCALE RSHEAR RTAPER RROTATE

GRAPHICS : GMOVE GSCALE GSHEAR GTAPER GROTATE

JOYSTICK : JMOVE JSCALE JSHEAR JTAPER JROTATE

CHARACTER :

SPECIAL : PRINTS INPUTS STRING SOUNDS MUZAKT

COMMAND CONDITIONS :

COMMAND LIMITATIONS :

ALL X,Y POSITIONS ARE IN GDU'S.
COORDINATES ARE LIMITED TO DEFAULT SCREEN.

COMMANDS ASSUME DEFAULT WINDOW & VIEWPORT CONDITIONS.
ALL GRAPHIC DISPLAY COMMANDS RESTORE THE GRAPHIC CURSOR.
RELATIVE IMAGE COMMANDS ARE RELATIVE TO FIRST IMAGE VECTOR.
ANY VECTORS THAT ARE BEYOND THE DEFAULT SCREEN BOUNDRY WILL
BE LIMITED TO THE DEFAULT SCREEN BOUNDRY.
UNDEFINED NUMERIC AND UNDEFINED STRING PARAMETERS ARE ALLOWED
IN MANY OF THE COMMANDS. ANY UNDEFINED STRING VARIABLES WILL
BE DIMENSIONED TO THE 4051 BASIC DEFAULT OF 72 CHARACTERS.
ALL DEFINED IMAGE STRINGS MUST HAVE A LENGTH OF 0 MOD 3. THUS
NO PARTIAL (LESS THAN 3 BYTE) VECTORS ARE ALLOWED. ALSO ALL
DEFINED ARGUMENT STRINGS MUST HAVE A LENGTH > 0. THUS NO NULL
ARGUMENT STRINGS ARE ALLOWED.

DISPLAY COUNT :

C >= +0.5 DRAW IMAGE ABS(C) TIMES IN STORAGE MODE.
C = 0.0 DRAW IMAGE ABS(C) TIMES IN NOTEVER MODE.
C <= -0.5 DRAW IMAGE ABS(C) TIMES IN REFRESH MODE.

JOYSTICK COMMANDS :

EXAMPLE : CALL "JCROSS",C,X,Y,K$

CROSS-HAIR JOYSTICK CURSOR WILL BE DRAWN C TIMES OR
UNTIL A KEYBOARD KEY IS PRESSED WHICHEVER OCCURS FIRST.
AFTER JOYSTICK COMMAND EXIT : C = ACTUAL DISPLAY COUNT.
X AND Y ARE UPDATED TO THE CURRENT JOYSTICK POSITION.
ANY PENDING KEYBOARD KEYS ARE SAVED IN K$.

INPUT COMMANDS :

EXAMPLE : CALL "AINPUT",S$,C,X,Y,K$

PRINT STRING S$ AND ANY KEYBOARD CHARACTERS C TIMES OR
UNTIL CARRIAGE RETURN IS PRESSED WHICHEVER OCCURS FIRST.
IF MORE THAN 28 CHARACTERS ARE ENTERED THEN BEEP BELL.
AFTER INPUT COMMAND EXIT : C = ACTUAL DISPLAY COUNT.
PRINT CHARACTER STRINGS STARTING AT LOCATION X,Y
ANY PENDING KEYBOARD KEYS ARE SAVED IN K$.

POINT COMMANDS :

EXAMPLE : CALL "APOINT",I$,N,X,Y

FIND IMAGE I$ POINT N & X,Y POSITION NEAREST POSITION X,Y.
AT CALL ENTRY N = BEGINNING VECTOR NUMBER TO START SEARCH AT.
AT CALL EXIT SIGN(N) INDICATES IF VECTOR IS A MOVE OR A DRAW.

SCALE COMMANDS :

NEGATIVE VALUES FOR H AND/OR V RESULT IN A MIRROR IMAGE.
-65.535 <= SCALE FACTOR RANGE =< +65.535 (17 BIT RESOLUTION).

SHEAR COMMANDS :

ALL ANGLES ARE MEASURED COUNTER CLOCK-WISE.
ALL ANGLES ARE IN CURRENT TRIGONOMETRIC UNITS.

TAPER COMMANDS :

NEGATIVE VALUES FOR H AND/OR V RESULT IN A MIRROR TAPER.
-65.535 <= TAPER FACTOR RANGE =< +65.535 (17 BIT RESOLUTION).

ROTATE COMMANDS :

```
    ALL ANGLES ARE MEASURED COUNTER CLOCK-WISE.
    ALL ANGLES ARE IN CURRENT TRIGONOMETRIC UNITS.

SYNTAX ABBREVIATIONS :

  THE FOLLOWING ABBREVIATIONS ARE USED TO DESCRIBE THE
  SYNTAX OF ALL FAST GRAPHICS COMMANDS.

    ARY:ARG = A COMPLETELY DEFINED NUMERIC ARRAY
    NUM:ARG = A NUMERIC EXPRESSION OR A DEFINED SIMPLE NUMERIC
    IMG:ARG = A DEFINED IMAGE STRING (LENGTH > 0 & = 0 MOD 3)
    STR:ARG = A LITERAL OR A DEFINED STRING (LENGTH > 0)
    ARY:RES = A DIMENSIONED NUMERIC ARRAY (MAY BE UNDEFINED)
    NUM:RES = A SIMPLE NUMERIC RESULT (MAY BE UNDEFINED)
    IMG:RES = A IMAGE STRING RESULT (MAY BE UNDEFINED)
    STR:RES = A STRING RESULT (MAY BE UNDEFINED)
    ARY:ARG&RES = A COMPLETELY DEFINED NUMERIC ARRAY
    NUM:ARG&RES = A DEFINED SIMPLE NUMERIC RESULT
    IMG:ARG&RES = A DEFINED IMAGE STRING (LENGTH > 0 & = 0 MOD 3)
    STR:ARG&RES = A DEFINED STRING RESULT (LENGTH > 0)

IMAGE BOUNDS :

  CALL "BOUNDS",I$,H,V,X,Y

  CALL "BOUNDS",IMG:ARG,NUM:RES,NUM:RES,NUM:RES,NUM:RES

  FIND IMAGE I$ MIN X, MIN Y, MAX X, & MAX Y BOUNDS

JOYSTICK LOCATION :

  CALL "LOCATE",C,X,Y,K$

  CALL "LOCATE",NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

  PRINT JOYSTICK CURSOR LOCATION AT GRAPHIC CURSOR POSITION
  SEE "JOYSTICK COMMANDS :"

ABSOLUTE GIN :

  CALL "AGIN",X,Y

  CALL "AGIN",NUM:RES,NUM:RES

  FIND GRAPHIC CURSOR X,Y POSITION

RELATIVE GIN :

  CALL "RGIN",X,Y

  CALL "RGIN",NUM:ARG&RES,NUM:ARG&RES

  FIND GRAPHIC CURSOR X,Y POSITION RELATIVE TO X,Y

GRAPHICS GIN :

  CALL "GGIN",X,Y

  CALL "GGIN",NUM:ARG&RES,NUM:ARG&RES

  FIND X,Y POSITION AT X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK GIN :
```

```
   CALL "JGIN",T,X,Y,K$

   CALL "JGIN",NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

   WAIT A MAXIMUM OF T MILLISECONDS FOR KEYBOARD KEY
   FIND JOYSTICK CURSOR X,Y POSITION & KEYBOARD KEYS K$
   SEE "JOYSTICK COMMANDS :"

ABSOLUTE POINT :

   CALL "APOINT",I$,N,X,Y

   CALL "APOINT",IMG:ARG,NUM:ARG&RES,NUM:ARG&RES,NUM:ARG&RES

   FIND IMAGE I$ POINT N & X,Y NEAREST ABSOLUTE POSITION X,Y
   SEE "POINT COMMANDS :"

RELATIVE POINT :

   CALL "RPOINT",I$,N,X,Y

   CALL "RPOINT",IMG:ARG,NUM:ARG&RES,NUM:ARG&RES,NUM:ARG&RES

   FIND IMAGE I$ POINT N & X,Y NEAREST RELATIVE POSITION X,Y
   SEE "POINT COMMANDS :"

GRAPHICS POINT :

   CALL "GPOINT",I$,N,X,Y

   CALL "GPOINT",IMG:ARG,NUM:ARG&RES,NUM:ARG&RES,NUM:ARG&RES

   FIND IMAGE I$ POINT N & X,Y NEAREST X,Y RELATIVE GRAPHIC CURSOR
   SEE "POINT COMMANDS :"

JOYSTICK POINT :

   CALL "JPOINT",I$,C,N,X,Y,K$

   CALL "JPOINT",IMG:ARG,NUM:ARG&RES,NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

   FIND IMAGE I$ POINT N & X,Y POSITION NEAREST JOYSTICK CURSOR
   SEE "JOYSTICK COMMANDS :"
   SEE "POINT COMMANDS :"

DASHED GRID :

   CALL "DASHED",H,V,X0,Y0,X1,Y1

   CALL "DASHED",NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

   DRAW DASHED GRID WITH HORIZONTAL PITCH H AND VERTICAL PITCH V
   THE DASHED GRID STARTS AT X0,Y0 AND ENDS AT X1,Y1 (OR BEFORE)

DOTTED GRID :

   CALL "DOTTED",H,V,X0,Y0,X1,Y1

   CALL "DOTTED",NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

   DRAW DOTTED GRID WITH HORIZONTAL PITCH H AND VERTICAL PITCH V
   THE DOTTED GRID STARTS AT X0,Y0 AND ENDS AT X1,Y1 (OR BEFORE)
```

KABCOM :

```
CALL "KABOOM",C,X,Y

CALL "KABOOM",NUM:ARG,NUM:ARG,NUM:ARG
```

DRAW EXPLOSION C TIMES AT POSITION X,Y

RUBBER BAND LINE :

```
CALL "RUBBER",C,X,Y,K$

CALL "RUBBER",NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES
```

DRAW RUBBER BAND LINE FROM JOYSTICK CURSOR TO GRAPHIC CURSOR
SEE "JOYSTICK COMMANDS :"

VERTEX :

```
CALL "VERTEX",I$,C,X,Y,K$

CALL "VERTEX",IMG:ARG,NUM:ARG&RES,NUMRES,NUM:RES,STR:RES
```

DRAW RUBBER BAND LINES FROM JOYSTICK CURSOR TO IMAGE I$ POINTS
SEE "JOYSTICK COMMANDS :"

ABSOLUTE CROSS :

```
CALL "ACROSS",C,X,Y

CALL "ACROSS",NUM:ARG,NUM:ARG,NUM:ARG
```

DRAW CROSSHAIR CURSOR C TIMES AT POSITION X,Y

RELATIVE CROSS :

```
CALL "RCROSS",C,X,Y

CALL "RCROSS",NUM:ARG,NUM:ARG,NUM:ARG
```

DRAW CROSSHAIR CURSOR C TIMES AT X,Y RELATIVE TO CENTER SCREEN

GRAPHICS CROSS :

```
CALL "GCROSS",C,X,Y

CALL "GCROSS",NUM:ARG,NUM:ARG,NUM:ARG
```

DRAW CROSSHAIR CURSOR C TIMES AT X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK CROSS :

```
CALL "JCROSS",C,X,Y,K$

CALL "JCROSS",NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES
```

SEE "JOYSTICK COMMANDS :"

ABSOLUTE DRAW :

```
CALL "ADRAW",I$,C,X,Y

CALL "ADRAW",IMG:ARG,NUM:ARG,NUM:ARG,NUM:ARG
```

DRAW IMAGE I$ C TIMES AT POSITION X,Y

RELATIVE DRAW :

CALL "RDRAW",I$,C,X,Y

CALL "RDRAW",IMG:ARG,NUM:ARG,NUM:ARG,NUM:ARG

DRAW IMAGE I$ C TIMES AT RELATIVE POSITION X,Y

GRAPHICS DRAW :

CALL "GDRAW",I$,C,X,Y

CALL "GDRAW",IMG:ARG,NUM:ARG,NUM:ARG,NUM:ARG

DRAW IMAGE I$ C TIMES AT X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK DRAW :

CALL "JDRAW",I$,C,X,Y,K$

CALL "JDRAW",IMG:ARG,NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

DRAW IMAGE I$ C TIMES AT JOYSTICK CURSOR
SEE "JOYSTICK COMMANDS :"

ABSOLUTE DOTS :

CALL "ADOTS",I$,C,X,Y

CALL "ADOTS",IMG:ARG,NUM:ARG,NUM:ARG,NUM:ARG

DRAW IMAGE I$ DOTS C TIMES AT POSITION X,Y

RELATIVE DOTS :

CALL "RDOTS",I$,C,X,Y

CALL "RDOTS",IMG:ARG,NUM:ARG,NUM:ARG,NUM:ARG

DRAW IMAGE I$ DOTS C TIMES AT RELATIVE POSITION X,Y

GRAPHICS DOTS :

CALL "GDOTS",I$,C,X,Y

CALL "GDOTS",IMG:ARG,NUM:ARG,NUM:ARG,NUM:ARG

DRAW IMAGE I$ DOTS C TIMES AT X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK DOTS :

CALL "JDOTS",I$,C,X,Y,K$

CALL "JDOTS",IMG:ARG,NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

DRAW IMAGE I$ DOTS C TIMES AT JOYSTICK CURSOR
SEE "JOYSTICK COMMANDS :"

ABSOLUTE PRINT :

CALL "APRINT",C$,C,X,Y

```
  CALL "APRINT",STR:ARG,NUM:ARG,NUM:ARG,NUM:ARG

  PRINT CHARACTERS C$ C TIMES AT POSITION X,Y

RELATIVE PRINT :

  CALL "RPRINT",C$,C,X,Y

  CALL "RPRINT",STR:ARG,NUM:ARG,NUM:ARG,NUM:ARG

  PRINT CHARACTERS C$ C TIMES AT X,Y RELATIVE TO CENTER SCREEN

GRAPHICS PRINT :

  CALL "GPRINT",C$,C,X,Y

  CALL "GPRINT",STR:ARG,NUM:ARG,NUM:ARG,NUM:ARG

  PRINT CHARACTERS C$ C TIMES AT X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK PRINT :

  CALL "JPRINT",C$,C,X,Y,K$

  CALL "JPRINT",STR:ARG,NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

  PRINT STRING C$ C TIMES AT JOYSTICK CURSOR
  SEE "JOYSTICK COMMANDS :"

ABSOLUTE INPUT :

  CALL "AINPUT",C$,C,X,Y,K$

  CALL "AINPUT",STR:ARG,NUM:ARG&RES,NUM:ARG,NUM:ARG,STR:RES

  PRINT STRING C$ AND ANY KEYBOARD KEYS C TIMES AT POSITION X,Y
  SEE "INPUT COMMANDS :"

RELATIVE INPUT :

  CALL "RINPUT",C$,C,X,Y,K$

  CALL "RINPUT",STR:ARG,NUM:ARG&RES,NUM:ARG,NUM:ARG,STR:RES

  PRINT STRING C$ AND ANY KEYBOARD KEYS C TIMES AT X,Y RELATIVE TO CENTER SCREEN
  SEE "INPUT COMMANDS :"

GRAPHICS INPUT :

  CALL "GINPUT",C$,C,X,Y,K$

  CALL "GINPUT",STR:ARG,NUM:ARG&RES,NUM:ARG,NUM:ARG,STR:RES

  PRINT STRING C$ AND ANY KEYBOARD KEYS C TIMES AT X,Y RELATIVE TO GRAPHIC CURSOR
  SEE "INPUT COMMANDS :"

JOYSTICK INPUT :

  CALL "JINPUT",C$,C,X,Y,K$

  CALL "JINPUT",STR:ARG,NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

  PRINT STRING C$ AND ANY KEYBOARD KEYS C TIMES AT JOYSTICK CURSOR
  SEE "JOYSTICK COMMANDS :"
```

SEE "INPUT COMMANDS :"

IMAGES :

```
CALL "IMAGES",I$

CALL "IMAGES",IMG:RES
```

INPUT MAG TAPE ASCII FILE INTO IMAGE STRING I$
EXIT OCCURS WHEN EOF IS ENCOUNTERED OR IMAGE STRING IS FULL
NO CHARACTERS ARE LOST IF STRING IS FULL AND CALL IS REPEATED

CHANGE :

```
CALL "CHANGE",F,S$

CALL "CHANGE",ARY:ARG,IMG:RES
```

CHANGE FLOATING POINT IMAGE ARRAY F INTO IMAGE STRING S$
NEGATIVE VALUES IN ARRAY F INDICATE MOVES

```
CALL "CHANGE",S$,F

CALL "CHANGE",IMG:ARG,ARY:RES
```

CHANGE IMAGE STRING S$ INTO FLOATING POINT IMAGE ARRAY F
NEGATIVE VALUES IN ARRAY F INDICATE MOVES

DEFINE POINT :

```
CALL "DEFINE",I$,N,X,Y

CALL "DEFINE",IMG:ARG&RES,NUM:ARG&RES,NUM:ARG,NUM:ARG
```

DEFINE IMAGE I$ POINT N X,Y POSITION
IMAGE I$ POINT N MUST EXIST PRIOR TO THE CALL
SIGN(N) INDICATES IF VECTOR IS A MOVE OR A DRAW.

LOCATE POINTS :

```
CALL "POINTS",I$,N,X,Y

CALL "POINTS",IMG:ARG,NUM:ARG&RES,NUM:RES,NUM:RES
```

FIND IMAGE I$ POINT N X,Y POSITION
SIGN(N) INDICATES IF VECTOR IS A MOVE OR A DRAW.

TOGGLE MOVE FLAG :

```
CALL "TOGGLE",I$,N

CALL "TOGGLE",IMG:ARG&RES,NUM:ARG&RES
```

TOGGLE IMAGE STRING I$ POINT N MOVE FLAG
SIGN(N) INDICATES IF VECTOR IS A MOVE OR A DRAW.
A DRAW IS CHANGED TO A MOVE AND A MOVE IS CHANGED TO A DRAW

ABSOLUTE MOVE :

```
CALL "AMOVE",I$,X,Y

CALL "AMOVE",IMG:ARG&RES,NUM:ARG,NUM:ARG
```

MOVE IMAGE I$ TO POSITION X,Y

RELATIVE MOVE :

CALL "RMOVE",I$,X,Y

CALL "RMOVE",IMG:ARG&RES,NUM:ARG,NUM:ARG

MOVE IMAGE I$ TO RELATIVE POSITION X,Y

GRAPHICS MOVE :

CALL "GMOVE",I$,X,Y

CALL "GMOVE",IMG:ARG&RES,NUM:ARG,NUM:ARG

MOVE IMAGE I$ TO X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK MOVE :

CALL "JMOVE",I$,C,X,Y,K$

CALL "JMOVE",IMG:ARG&RES,NUM:ARG&RES,NUM:RES,NUM:RES,STR:RES

MOVE IMAGE I$ TO JOYSTICK CURSOR
SEE "JOYSTICK COMMANDS :"

ABSOLUTE SCALE :

CALL "ASCALE",I$,H,V,X,Y

CALL "ASCALE",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

SCALE IMAGE I$ BY FACTORS H,V AROUND POSITION X,Y

RELATIVE SCALE :

CALL "RSCALE" I$,H,V,X,Y

CALL "RSCALE",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

SCALE IMAGE I$ BY FACTORS H,V AROUND RELATIVE POSITION X,Y

GRAPHICS SCALE :

CALL "GSCALE" I$,H,V,X,Y

CALL "GSCALE",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

SCALE IMAGE I$ BY FACTORS H,V AROUND X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK SCALE :

CALL "JSCALE" I$,C,H,V,X,Y,K$

CALL "JSCALE",IMG:ARG&RES,NUM:ARG&RES,NUM:ARG,NUM:ARG,NUM:RES,NUM:RES,STR:RES

SCALE IMAGE I$ BY FACTORS H,V AROUND JOYSTICK CURSOR
SEE "JOYSTICK COMMANDS :"

ABSOLUTE SHEAR :

CALL "ASHEAR",I$,H,V,X,Y

CALL "ASHEAR",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

SHEAR IMAGE I$ BY ANGLES H,V AROUND POSITION X,Y

RELATIVE SHEAR :

CALL "RSHEAR" I$,H,V,X,Y

CALL "RSHEAR",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

SHEAR IMAGE I$ BY ANGLES H,V AROUND RELATIVE POSITION X,Y

GRAPHICS SHEAR :

CALL "GSHEAR" I$,H,V,X,Y

CALL "GSHEAR",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

SHEAR IMAGE I$ BY ANGLES H,V AROUND X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK SHEAR :

CALL "JSHEAR" I$,C,H,V,X,Y,K$

CALL "JSHEAR",IMG:ARG&RES,NUM:ARG&RES,NUM:ARG,NUM:ARG,NUM:RES,NUM:RES,STR:RES

SHEAR IMAGE I$ BY ANGLES H,V AROUND JOYSTICK CURSOR
SEE "JOYSTICK COMMANDS :"

ABSOLUTE TAPER :

CALL "ATAPER",I$,H,V,X,Y

CALL "ATAPER",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

TAPER IMAGE I$ BY FACTORS H,V AROUND POSITION X,Y

RELATIVE TAPER :

CALL "RTAPER",I$,H,V,X,Y

CALL "RTAPER",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

TAPER IMAGE I$ BY FACTORS H,V AROUND RELATIVE POSITION X,Y

GRAPHICS TAPER :

CALL "GTAPER" I$,H,V,X,Y

CALL "GTAPER",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG,NUM:ARG

TAPER IMAGE I$ BY FACTORS H,V AROUND X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK TAPER :

CALL "JTAPER" I$,C,H,V,X,Y,K$

CALL "JTAPER",IMG:ARG&RES,NUM:ARG&RES,NUM:ARG,NUM:ARG,NUM:RES,NUM:RES,STR:RES

TAPER IMAGE I$ BY FACTORS H,V AROUND JOYSTICK CURSOR
SEE "JOYSTICK COMMANDS :"

ABSOLUTE ROTATE :

CALL "AROTATE",I$,R,X,Y

```
  CALL "AROTATE",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG

  ROTATE IMAGE I$ BY ANGLE R AROUND POSITION X,Y

RELATIVE ROTATE :

  CALL "RROTATE",I$,R,X,Y

  CALL "RROTATE",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG

  ROTATE IMAGE I$ BY ANGLE R AROUND RELATIVE POSITION X,Y

GRAPHICS ROTATE :

  CALL "GROTATE",I$,R,X,Y

  CALL "GROTATE",IMG:ARG&RES,NUM:ARG,NUM:ARG,NUM:ARG

  ROTATE IMAGE I$ BY ANGLE R AROUND X,Y RELATIVE TO GRAPHIC CURSOR

JOYSTICK ROTATE :

  CALL "JROTATE",I$,C,R,X,Y,K$

  CALL "JROTATE",IMG:ARG&RES,NUM:ARG&RES,NUM:ARG,NUM:RES,NUM:RES,STR:RES

  ROTATE IMAGE I$ BY ANGLE R AROUND JOYSTICK CURSOR
  SEE "JOYSTICK COMMANDS :"

PRINTS :

  CALL "PRINTS",S$

  CALL "PRINTS",STR:ARG

  PRINT STRING S$ CHARACTERS (PRINTABLE CONTROL CHARACTERS)

INPUTS :

  CALL "INPUTS",S$

  CALL "INPUTS",STR:ARG

  INPUT STRING S$ CHARACTERS INTO KEYBOARD BUFFER (28 CHAR MAX)

STRING :

  CALL "STRING",F,S$

  CALL "STRING",ARY:ARG,STR:RES

  CHANGE FLOATING POINT ARRAY F INTO ASCII STRING S$
  ASCII STRING WILL BE VALID ASCII CHARACTERS (ASCII 0..127)

  CALL "STRING",S$,F

  CALL "STRING",STR:ARG,ARY:RES

  CHANGE ASCII STRING S$ INTO FLOATING POINT ARRAY F
  FLOATING POINT ARRAY F WILL BE VALID ASCII CHARACTER CODES

SOUNDS :
```

CALL "SOUNDS",S$

CALL "SOUNDS",STR:ARG

OUTPUT ZERO CROSSING PERIODS IN ASCII STRING S$ TO SPEAKER

SAMPLE PERIOD = (N+1) * 38.4 US
00HEX = 26042 HZ EDGE RATE (FREQ = 13KHZ)
7FHEX = 203.5 HZ EDGE RATE (FREQ = 100HZ)
SAMPLE INTERVAL = 32 CYCLES AT 1.2 US/CYCLE

MUZAKT :

CALL "MUZAKT",S$

CALL "MUZAKT",STR:ARG

OUTPUT MUSICAL NOTES IN ASCII STRING S$ TO SPEAKER

MUZAKT STRING SYNTAX :

```
<WORD> ::= <TEMP>
         +  <REST>
         +  <NOTE>
         +  <NOTE><DOT>
         +  <NOTE><OCT>
         +  <NOTE><OCT><DOT>
         +  <NOTE><OCT><LEN>
         +  <NOTE><OCT><LEN><DOT>
<NOTE> ::= <A + B + C + D + E + F + G>
         +  <A + B + C + D + E + F + G><FLAT>
         +  <A + B + C + D + E + F + G><SHRP>
<OCT>  ::= <0..7> (ZERO = FIRST OCTAVE ON STANDARD KEYBOARD)
<LEN>  ::= <1..99> (RECIPROCAL NOTE LENGTH : 1,2,4,8,16,32,64)
<REST> ::= <R><1..99> (RECIPROCAL REST LENGTH : 1,2,4,8,16,32,64)
<TEMP> ::= <T><0..9> (SEE MUZAKT TEMPO TABLE BELOW)
<DOT>  ::= <.> (PERIOD)
<SHRP> ::= <#> (POUNDS)
<FLAT> ::= <@B> (SMALL B)
```

TO ENTER THIRDS , FIFTHS ... USE THREE , FIVE ... TIMES LENGTH ETC.

TEMPO	RATE (BEATS/MINUTE)
0	80
1	90
2	100
3	110
4	120
5	130
6	140
7	150
8	160
9	170

MUZAKT STRING EXAMPLE :

T4E316G332B4EGB5EG58.G516G54F#R

NOTE	OCT	LEN
TEMP	4	
E	3	1/16
G	3	1/32

B	4	1/32
E	4	1/32
G	4	1/32
B	5	1/32
E	5	1/32
G	5	DOTTED EIGHTH
G	5	1/16
G	5	1/4
F#	5	1/4
REST		1/4

ANY INVALID WORDS OR CHARACTERS ARE IGNORED

BENCHMARKS :

1,500 VECTOR IMAGE :			VECTORS/SECOND	SPEED(X)
CHANGE FLOATING POINT TO STRING :	2.8	SECONDS	536	
CHANGE STRING TO FLOATING POINT :	28.0	SECONDS	54	

3,200 VECTOR IMAGE (3-D DOTS) :			VECTORS/SECOND	SPEED(X)
BASIC USING "MOVE/DRAW" :	175	SECONDS	18	
BASIC USING "PRINT AT:" :	110	SECONDS	29	
BASIC USING "MOVE/DRAW" :	79.0	SECONDS (4052)	41	
EXTENDED BASIC "QDRAW " :	36.0	SECONDS	89	
FAST GRAPHICS "RDRAW " :	12.8	SECONDS	250	13.7 (6.2)
FAST GRAPHICS "RDOTS " :	2.8	SECONDS	1145	125 (55)
FAST GRAPHICS "RMOVE " :	1.9	SECONDS	1684	
FAST GRAPHICS "APOINT" :	5.9	SECONDS	542	
FAST GRAPHICS "ASCALE" :	7.3	SECONDS	438	
FAST GRAPHICS "BOUNDS" :	1.3	SECONDS	2462	

3,300 VECTOR IMAGE (DEATH STAR) :			VECTORS/SECOND	SPEED(X)
BASIC USING "MOVE/DRAW" :	185	SECONDS	18	
BASIC USING "PRINT AT:" :	125	SECONDS	26	
BASIC USING "MOVE/DRAW" :	89.4	SECONDS (4052)	37	
EXTENDED BASIC "QDRAW " :	24.5	SECONDS	135	
FAST GRAPHICS "RDRAW " :	12.7	SECONDS	260	14.6 (7.0)
FAST GRAPHICS "RDOTS " :	2.9	SECONDS	1132	127 (61)
FAST GRAPHICS "RMOVE " :	2.0	SECONDS	1650	
FAST GRAPHICS "APOINT" :	6.1	SECONDS	541	
FAST GRAPHICS "ASCALE" :	7.4	SECONDS	446	
FAST GRAPHICS "BOUNDS" :	1.3	SECONDS	2538	

4,400 VECTOR IMAGE (GOTHIC FONT) :			VECTORS/SECOND	SPEED(X)
BASIC USING "MOVE/DRAW" :	80.0	SECONDS (4052)	55	
EXTENDED BASIC "QDRAW " :	26.4	SECONDS	165	
FAST GRAPHICS "RDRAW " :	17.8	SECONDS	246	(4.5)
FAST GRAPHICS "RDOTS " :	3.25	SECONDS	1112	(40)

10,000 VECTOR IMAGE (TIGER) :			VECTORS/SECOND	SPEED(X)
BASIC USING "MOVE/DRAW" :	FOREVER		...	
BASIC USING "PRINT AT:" :	FOREVER MINUS A LITTLE		...	
EXTENDED BASIC "QDRAW " :	50.0	SECONDS	200	
FAST GRAPHICS "RDRAW " :	36.6	SECONDS	273	
FAST GRAPHICS "RDOTS " :	8.5	SECONDS	1177	
FAST GRAPHICS "RMOVE " :	5.5	SECONDS	1818	
FAST GRAPHICS "APOINT" :	18.0	SECONDS	556	
FAST GRAPHICS "ASCALE" :	21.5	SECONDS	465	

```
    FAST GRAPHICS  "BOUNDS" :  3.4 SECONDS          2941

  ASCII CHARACTERS (3200 CHARACTERS) :       CHARACTERS/SECOND

    CHARACTER PRINTING RATE   :                      >250
    CHANGE ASCII TO CHARACTER :  2.1 SECONDS         1524
    CHANGE CHARACTER TO ASCII :  4.1 SECONDS          780

  SEVERAL DIFFERENT FAST GRAPHICS DEMO TAPES ARE AVAILABLE FROM
      M. D. CRANFORD  MS 50-370  X6131  (SEND BLANK TAPES).

CALL PARAMETER CODES :

  80 : NUMERIC ARRAY ARGUMENT
  40 : NUMERIC SIMPLE ARGUMENT
  20 : NUMERIC CONSTANT ARGUMENT
  10 : STRING ARRAY ARGUMENT
  08 : NUMERIC ARRAY RESULT
  04 : NUMERIC SIMPLE RESULT
  02 : STRING CONSTANT ARGUMENT
  01 : STRING ARRAY RESULT
  00 : NONE (CALL TAG)

CALL SYNTAX FORMS :              CALL PARAMETER TYPES :

 CALL "BOUNDS",I$,X0,Y0,X1,Y1     10 04 04 04 04
 CALL "LOCATE",C,X,Y,K$           44 04 04 01
 CALL "AGIN  ",X,Y                04 04
 CALL "RGIN  ",X,Y                44 44
 CALL "GGIN  ",X,Y                44 44
 CALL "JGIN  ",T,X,Y,K$           44 04 04 01
 CALL "APOINT",I$,N,X,Y           10 44 44 44
 CALL "RPOINT",I$,N,X,Y           10 44 44 44
 CALL "GPOINT",I$,N,X,Y           10 44 44 44
 CALL "JPOINT",I$,C,N,X,Y,K$      10 44 44 04 04 01
 CALL "DASHED",H,V,X0,Y0,X1,Y1    60 60 60 60 60 60
 CALL "DOTTED",H,V,X0,Y0,X1,Y1    60 60 60 60 60 60
 CALL "KABOOM",C,X,Y              60 60 60
 CALL "RUBBER",C,X,Y,K$           44 04 04 01
 CALL "VERTEX",I$,C,X,Y,K$        10 44 04 04 01
 CALL "ACROSS",C,X,Y              60 60 60
 CALL "RCROSS",C,X,Y              60 60 60
 CALL "GCROSS",C,X,Y              60 60 60
 CALL "JCROSS",C,X,Y,K$           44 04 04 01
 CALL "ADRAW ",I$,C,X,Y           10 60 60 60
 CALL "RDRAW ",I$,C,X,Y           10 60 60 60
 CALL "GDRAW ",I$,C,X,Y           10 60 60 60
 CALL "JDRAW ",I$,C,X,Y,K$        10 44 04 04 01
 CALL "ADOTS ",I$,C,X,Y           10 60 60 60
 CALL "RDOTS ",I$,C,X,Y           10 60 60 60
 CALL "GDOTS ",I$,C,X,Y           10 60 60 60
 CALL "JDOTS ",I$,C,X,Y,K$        10 44 04 04 01
 CALL "APRINT",C$,C,X,Y           12 60 60 60
 CALL "RPRINT",C$,C,X,Y           12 60 60 60
 CALL "GPRINT",C$,C,X,Y           12 60 60 60
 CALL "JPRINT",C$,C,X,Y,K$        12 44 04 04 01
 CALL "AINPUT",C$,C,X,Y,K$        12 44 60 60 01
 CALL "RINPUT",C$,C,X,Y,K$        12 44 60 60 01
 CALL "GINPUT",C$,C,X,Y,K$        12 44 60 60 01
 CALL "JINPUT",C$,C,X,Y,K$        12 44 04 04 01
 CALL "IMAGES",I$                 01
 CALL "CHANGE",F,S$               80 01
 CALL "CHANGE",S$,F               10 08
 CALL "DEFINE",I$,N,X,Y           11 44 60 60
```

```
CALL "POINTS",I$,N,X,Y            10 44 04 04
CALL "TOGGLE",I$,N                11 44
CALL "AMOVE ",I$,X,Y              11 60 60
CALL "RMOVE ",I$,X,Y              11 60 60
CALL "GMOVE ",I$,X,Y              11 60 60
CALL "JMOVE ",I$,C,X,Y,K$         11 44 04 04 01
CALL "ASCALE",I$,H,V,X,Y          11 60 60 60 60
CALL "RSCALE",I$,H,V,X,Y          11 60 60 60 60
CALL "GSCALE",I$,H,V,X,Y          11 60 60 60 60
CALL "JSCALE",I$,C,H,V,X,Y,K$     11 44 60 60 04 04 01
CALL "ASHEAR",I$,H,V,X,Y          11 60 60 60 60
CALL "RSHEAR",I$,H,V,X,Y          11 60 60 60 60
CALL "GSHEAR",I$,H,V,X,Y          11 60 60 60 60
CALL "JSHEAR",I$,C,H,V,X,Y,K$     11 44 60 60 04 04 01
CALL "ATAPER",I$,H,V,X,Y          11 60 60 60 60
CALL "RTAPER",I$,H,V,X,Y          11 60 60 60 60
CALL "GTAPER",I$,H,V,X,Y          11 60 60 60 60
CALL "JTAPER",I$,C,H,V,X,Y,K$     11 44 60 60 04 04 01
CALL "AROTATE",I$,R,X,Y           11 60 60 60
CALL "RROTATE",I$,R,X,Y           11 60 60 60
CALL "GROTATE",I$,R,X,Y           11 60 60 60
CALL "JROTATE",I$,C,R,X,Y,K$      11 44 60 04 04 01
CALL "PRINTS",S$                  12
CALL "INPUTS",S$                  12
CALL "STRING",F,S$                80 01
CALL "STRING",S$,F                10 08
CALL "SOUNDS",S$                  12
CALL "MUZAKT",S$                  12
```

ACKNOWLEDGEMENTS :

KURT KRUEGER WROTE THE ORIGINAL VERSION OF MUZAKT IN 1978.
THE FAST GRAPHICS VERSION HAS SEVERAL ADDITIONAL FEATURES.

CARL HOVEY DEMONSTRATED THE FEASIBILITY OF 4051 WRITE-THRU
(NON-STORE) GRAPHICS IN A PINBALL GAME IN 1978.

THE FOLLOWING PEOPLE PROVIDED CONSIDERABLE SUPPORT :

TIMOTHY SPIRO
CLARK P. FOLEY
CALVIN D. DILLER
WILLIAM C. VESSER
FERROUS C. STEINKA