VECTREX

PROGRAMMER'S

MANUAL

VOLUME I

GENERAL INFORMATION

EXECUTIVE STOREAGE UTILIZATION

REV	DATE	PROG	COMMENT(S)
-	MM/DD/YY	JJH	Initial Release

[FDT: Revision 1.1 - 12/6/99]

Vectrex Hardware Description

The Vectrex consists of a 6809 microprocessor, 1k x 8 RAM, 8k x 8 resident ROM, 6522 interface adapter, D / A converter (+2.5 to –2.5 volts), 3 sample / holds (active ground, ‘Z’ and ‘Y’ axises), a programmable sound generator and 2 player controllers (one joystick and 4 buttons each).

The ROM contains the resident game ‘Mine Storm’ and the Executive. The Executive consists of power-up / reset handler and a large selection of subroutines for drawing, calculation, game logic and / or hardware maintenance.

The 6522 interface adapter provides all the interface necessary between the microprocessor and the analog electronics (joysticks, D / A converter, sample / holds and programmable sound generator). The interface adapter provides two 8-bit parallel ports, two 16-bit counters (timers) and an 8-bit shift register.

The 6522’s shift register is used by the dashing and raster drawing routines.

Vectrex Memory Map

Address Range	Description
$000 - $7FFF	Open for game ROM
$DXX0 – $DXXF	Vectrex control registers (6522)
$E000 - $EFFF	Mine storm
$F000 - $FFFF	Executive

6522 Interface Adapter

Register Access Map

Addr	Description
$D000	8-bit port ‘B’ (Control bits) [CNTRL]
$D001	8-bit port ‘A’ (DAC and PSC data) [DAC]
$D002	Port ‘B’ direction control [DCNTRL]
$D003	Port ‘A’ direction control [DDAC]
$D004-7	Timer #1 [T1LOLC, T1HOC, T1LOL, T1HOL]
$D008-9	Timer #2 [T2LOLC, T2HOC]
$D00A	Shift register [SHIFT]
$D00B	Auxiliary control register [ACNTRL]
$D00C	Peripheral control register [PCNTRL]
$D00D	Interrupt flag register [IFLAG]
$D00E	Interrupt enable register [IENABL]
$D00F	ORA

Port ‘A’ / ‘B’ Direction Registers [DDAC & DCNTRL]

The ‘A’ direction register [DDAC] controls the direction of data flow for the ‘A’ data port [DAC].

The ‘B’ direction register [DCNTRL] controls the direction of data flow for the ‘B’ data port [CNTRL].

Data flow direction is determined on a bit-by-bit basis. A 0 in a bit of the data direction register causes the corresponding data port pin to act as an input. A 1 causes the pin to act as an output.

Port ‘A’ Data – DAC and PSG data [DAC]

When outputting data to the DAC, the data byte place in the port ‘A’ data register is in a two’s complement format (bit 6 of the data byte is the high order bit with bit 0 being the lowest. Bit 7 of the data byte is used for the sign). Therefore, the minimum hex value to the DAC is $80; $00 is the center point and $7F is the maximum.

When passing data with the programmable sound generator, the data direction will have to be set according to the type of transfer.

Port ‘B’ Data – Vectrex Hardware Control [CNTRL]

Bit	Description
0	Sample / Hold strobe
1	Analog multiplexer select, bit 0
2	Analog multiplexer select, bit 1
3	Sound generator, term ‘BC1’
4	Sound generator, term ‘DBIR’
5	Analog compare term (Input to 6522)
6	.
7	Integrator ramp

Auxiliary Control Register [ACRTRL]

Bit	Description
0	Refer to Port ‘A’ description
1	Refer to Port ‘B’ description
4-2	Refer to Shift register description
5	Refer to Timer #2 description
7-6	Refer to Timer #1 description

Peripheral Control Register [PCNTRL]

Bit	Description
3-0	Refer to Port ‘A’ handshake controls
7-4	Refer to Port ‘B’ handshake controls

Port ‘A’ Handshake Controls (CA1 / 2)

PCNTRL	
Bit 0	Description
0	The CA1 interupt flag (IFR1) will be set on a negative transition of the CA1 input pin.
1	The CA1 interrupt flag (IFR1) will be set on a positive transition of the CA1 input pin.
	The CA1 interrupt flag (IFR1) will be cleared on a read or write of the Port ‘A’ data register.

PCNTRL bits	
3 2 1	Description
0 0 0	Interrupt input mode – Set CA2 interrupt flag (IFR0) on a negative transition of the CA2 input pin. Clear the interrupt flag (IFR0) on a read or write of the Port ‘A’ data register.
0 0 1	Independent interrupt input mode – Set CA2 interrupt flag (IFR0) on a negative transition of the CA2 input pin. Reading or writing of the Port ‘A’ data register does not affect the status of the interrupt flag (IFR0).

0 1 0 Interrupt input mode – Set CA2 interrupt flag (IFR0) on a positive transition of the CA2 input pin. Clear the interrupt flag (IFR0) on a read or write of the Port ‘A’ data register.

0 1 1 Independent interrupt input mode – Set CA2 interrupt flag (IFR0) on a positive transition of the CA2 input pin. Reading or writing of the Port ‘A’ data register does not affect the status of the interrupt flag (IFR0).

1 0 0 Handshake output mode – Set CA2 output pin low on a read or write of the Port ‘A’ data register. Reset CA2 pin to high with an active transition of the CA1 input pin.

1 0 1 Pulse output mode – CA2 goes low for one cycle following a read or write of the Port ‘A’ data register.

1 1 0 Manual output mode – The CA2 output is held low in this mode.

1 1 1 Manual output mode – The CA2 output is held high in this mode.

Port ‘B’ Handshake Controls (CB1 / 2)

PCNTRL
Bit 4 Description

0 The CB1 interrupt flag (IFR4) will be set on a negative transition of the CB1 input pin.

1 The CB1 interrupt flag (IFR4) will be set on a positive transition of the CB1 input pin.

The CA1 interrupt flag (IFR1) will be cleared on a read or write of the Port ‘A’ data register.

PCNTRL bits
7 6 5 Description

0 0 0 Interrupt input mode – Set CB2 interrupt flag (IFR3) on a negative transition of the CB2 input pin. Clear the interrupt flag (IFR3) on a read or write of the Port ‘B’ data register.

0 0 1 Independent interrupt input mode – Set CB2 interrupt flag (IFR3) on a negative transition of the CB2 input pin. Reading or writing of the Port ‘B’ data register does not affect the status of the interrupt flag (IFR3).

0 1 0 Interrupt input mode – Set CB2 interrupt flag (IFR3) on a positive transition of the CB2 input pin. Clear the interrupt flag (IFR3) on a read or write of the Port ‘B’ data register.

0 1 1 Independent interrupt input mode – Set CB2 interrupt flag (IFR3) on a positive transition of the CB2 input pin. Reading or writing of the Port ‘B’ data register does not affect the status of the interrupt flag (IFR3).

1 0 0 Handshake output mode – Set CB2 output pin low on a read or write of the Port ‘B’ data register. Reset CB2 pin to high with an active transition of the CB1 input pin.

1 0 1	Pulse output mode – CB2 goes low for one cycle following a read or write of the Port ‘B’ data register.
1 1 0	Manual output mode – The CB2 output is held low in this mode.
1 1 1	Manual output mode – The CB2 output is held high in this mode.

Timer #1

Addr	Label	Description
$D004	T1LOLC	Write into low order latch
		Read from low-order counter and reset interrupt
$D005	T1HOC	Write into high-order latch, transfer latches to counter and reset interrupt
		Read from high-order counter
$D006	T1LOL	Write into low-order latch
		Read from low-order latch
$D007	T1HOL	Write into high-order latch and reset timer #1 interrupt
		Read from high-order latch

ACNTRL bits

7 6	Description
0 0	Generate a single time-out interrupt each time timer #1 is loaded. Output pin PB7 is disabled.
0 1	Generate continuous interrupts. Output pin PB7 is disabled.
1 0	Generate a single interrupt and an output pulse on pin PB7 for timer #1 load operation.
1 1	Generate continuous interrupts and a square wave output on pin PB7.

One-shot Mode

The one-shot mode allows generation of a single interrupt each time a load operation occurs. In addition to generating a single interrupt, Timer #1 can be programmed to produce a single negative pulse on the PB7 peripheral pin. With the output enabled (ACNTRL7 = 1), a write to location $D005 [T1HOC] will cause PB7 to go low. PB7 will return high when Timer #a counts down to zero. For PB7 to act as the output of Timer #1, both the direction register for Port ‘B’ (bit 7) and the auxiliary control register (bit 7) must be set (e.g. – DCNTRL7 = 1 and ACNTRL7 = 1).

It is necessary to assure that the low order latch contains the proper data before initiating the count-down by writing to location $D005 [T1HOC]. When starting the down-count, the interrupt flag is cleared, the contents of the low-order latch will be transferred into the low-order counter and the timer will begin to decrement at the system clock rate. If the PB7 output is enabled, this signal will go low on the phase two following the write operation.

When the counter reaches zero, the timer #1 interrupt flag will be set, the IRQ pin will go low and the signal on PB7 will go high. The counter will still continue to decrement, this will allow the time since the interrupt to be determined (note: the timer #1 interrupt will not be set again unless it has previously cleared).

Free-running Mode

The most important advantages associated with the latches are the ability to produce a continuous series of evenly spaced interrupts and the ability to produce a square wave on PB7 whose frequency is not affected by variations in the processor interrupt response time.

In the free-running mode (ACNTRL6 = 1), the interrupt flag is set and the signal on PB7 is inverted each time the counter reaches zero. However, instead of continuing to decrement from zero after a time-out, the timer automatically transfers the contents of the latch into the counter and continues to decrement from there.

Timer #2

Addr	Label	Description
$D008	T1LOLC	Write into low order latch Read from low-order counter and reset interrupt
$D009	T1HOC	Write into high-order latch, transfer latches to counter and reset interrupt Read from high-order counter

ACNTRL Bit 5	Description
0	.
1	.

Interval Timer Mode

In interval timer mode, Timer #2 operates in the ‘one-shot’ mode similar to timer #1. Timer #2 issues an interrupt for each timer load operation. After timing out, the counter will continue to decrement.

Pulse-counting Mode

In the pulse counting mode, Timer #2 serves primarily to count a predetermined number of negative transitions on PB6. Writing into $D009 [T2HOC] clears the interrupt flag and allows the counter to decrement for each negative transition applied to PB6. The interrupt flag will be set when Timer #2 reaches zero.

At this time the counter will continue to decrement with each negative transition on PB6. The pulse must be low on the leading edge of phase two.

Shift Register [SHIFT]

The shift register performs serial data transfers into and out of the CB2 pin under control of an internal modulo-8 counter. Shift pulses can be applied to the CB1 pin from an external source or, with the proper mode selection, shift pulses generated internally will appear on the CB1 pin for controlling shifting in external devices.

The control bits which allow control of the various shift register operating modes are located in the Auxiliary control register [ACNTRL]:

ACNTRL bits

4 3 2	Descriptions
0 0 0	Shift register disabled
0 0 1	Shift-in under control of Timer #2
0 1 0	Shift-in at system clock rate
0 1 1	Shift-in under control of external pulses
1 0 0	Shift-out under control of Timer #2
1 0 1	Shift out under control of Timer #1
1 1 0	Shift out at system clock rate
1 1 1	Shift-out under control of external pulses

The data shifted-out of the shift register appears from bit 7 (note: at the same time, bit 7 is shifted-in at bit 0).

The first two shift-out modes differ in that the second mode (1 0 1) places the shift pulses on the CB1 pin.

The shift-in modes are not used with the standard Vectrex hardware and are described only for reference.

Interrupt enable register [IENABL]

Enabling flags – When writing to the Interrupt enable register ($D00E) and bit 7 is set, then each 1 in bits 6 through 0 sets the corresponding bit in the Interrupt enable register.

Disabling flags – When writing to the Interrupt enable register ($D00E) and bit 7 is cleared, then each 1 in bits 6 through 0 clears the corresponding bit in the Interrupt enable register.

Bit	Description
0	Enable / disable ‘CA2’ flag
1	Enable / disable ‘CA1’ flag
2	Enable / disable Shift register done flag
3	Enable / disable ‘CB2’ flag
4	Enable / disable ‘CB1’ flag
5	Enable / disable Timer #2 flag
6	Enable / disable Timer #1 flag
7	Set / clear control

Interrupt flag register [IFLAG]

Bit Description

0 Set by an active transition of ‘CA2’ signal

Cleared by the reading or writing of the Port ‘A’ data register (register address $D001). Reading or writing the Port ‘A’ data at register address $D00F does not affect the setting of this bit.

1 Set by an active transition of ‘CA1’ signal

Cleared by the reading or writing of the Port ‘A’ data register (register address $D001). Reading or wriging the Port ‘A’ data at register address $D00F does not affect the setting of this bit.

2 Shift register done

Cleared by the reading or writing of the shift register (register address $D00A).

3 Set by an active transition of ‘CB2’ signal

Cleared by the reading or writing of the Port ‘B’ data register (register address $D000).

4 Set by an active transition of ‘CB1’ signal

Cleared by the reading or writing of the Port ‘B’ data register (register address $D000).

5 Timer #2 has down-counted to zero

Cleared by reading Timer #2’s low-order counter (register address $D008) or writing Timer #2’s high-order counter (register address $D009).

6 Timer #1 has down-counted to zero

Cleared by reading Timer #1’s low-order counter (register address $D004) or writing Timer #1’s high-order counter (register address $D005).

7 Logic expression of (IFR6 and IER6) or (IFR5 and IER5) or (IFR4 and IER4) or (IFR3 and IER3) or (IFR2 and IER2) or (IFR1 and IER1) or (IFR0 and IER0)

Programmable Sound Generator

Register Access Map

Reg	Description
$00	Channel A: Fine tone period
$01	Channel A: Course tone period
$02	Channel B: Fine tone period
$03	Channel B: Course tone period
$04	Channel C: Fine tone period
$05	Channel C: Course tone period
$06	Noise period
$07	Tone / Noise enables
$08	Channel A: Amplitude
$09	Channel B: Amplitude
$0A	Channel C: Amplitude
$0B	Fine envelope period
$0C	Course envelope period
$0D	Envelope shape / cycle
$0E	I/O Port

Tone / Noise Enables

<This area was empty>

Course / Fine Tone Period

The frequency of each square wave generated by the three Tone generators is obtained in the PSG by first counting down the input clock by 16, then by further counting down the result by the programmed 12-bit tone period value. The 12-bit tone period is formed by the lower 4-bits of the course tone period (the upper 4-bits of the course tone period are not used) and the full byte of the fine tone period.

Note that the 12-bit value programmed in the combined course and fine tone registers is a 'period' value – the higher the value in the registers, the lower the resultant tone frequency. (The lowest period value is $001).

Channel Amplitude

The amplitudes of the signals generated by the three D/A converters (one for each channel) is determined by the contents of the lower 5 bits (B4 – B0) of register $08, $09 and $0A.

The amplitude mode (bit 4) selects either fixed level amplitude (M = 0) or variable level amplitude (M = 1). It follows then that bit 3 thru 0 are only active when M = 0. When fixed level amplitude is selected, it is 'fixed' only in the sense that the amplitude level is under the direct control of the system processor. Varying the amplitude when in this 'fixed' amplitude mode requires in each instance the direct intervention of the system processor via an address latch / write data sequence to modify the amplitude setting.

When M = 1, the amplitude of each channel is determined by the envelope pattern as defined by the envelope generator's 4-bit output. The amplitude mode (bit 'M') should be thought of as an envelope enable bit.

Noise Period

The frequency of the noise source is obtained in the PSG by first counting down the input clock by 16, then by further counting down the result by the programmed 5-bit noise period value. This 5-bit value consists of the lower 5 bits of register $06.

Note that the 5-bit value in register $06 is a period value – the higher the value in the register, the lower the resultant noise frequency (the lowest period is $01).

Envelope Period

The frequency of the envelope is obtained in the PSG by first counting down the input clock by 256, then by further counting down the result by the programmed 16-bit envelope period value. This 16-bit value is formed by the course and fine envelope period registers. Note that the 16-bit value formed by the course and fine envelope period registers is a period value – the higher the value in the registers, the lower the resultant envelope frequency (the lowest period is $0001).

Envelope Shape / Cycle

<This area was empty>

I/O Port

<This area was empty>

General executive description

Drawing format descriptions

‘Diffy’ Description

A ‘Diffy’ style list contains a counted collection of relative (Y:X) coordinate pairs. When processing one of these, the drawing functions will **draw** a line from the current pen position to the first point in the list. A line is then drawn to the next relative coordinate, until no more points remain.

Depending upon the function processing the list, the first byte may be expected to contain the ‘Vector count –1’, or this value may need to be stored into RAM.

Depending upon the function processing the list, the second byte may be expected to contain the scale factor to be used when processing the list, or this value may need to be stored into RAM.

A sample ‘Diffy’ list might look like the following:

byte 0	- Vector count – 1 [optional]
byte 1	- Scale factor [optional]
bytes 2 / 3	- ‘Y:X’ for coordinate 1
bytes n / n+1	- ‘Y:X’ for coordinate ‘n’

‘Duffy’ Description

A ‘Duffy’ style list is identical to a ‘Diffy’ style list. The only difference appears to be in the way that it is processed. When processing one of these, the drawing functions will **move** to the first point in the list. It will then draw a line to the next relative coordinate, until no more points remain.

‘Packet’ Description

A ‘Packet’ style list is an uncounted list of (mode:Y:X) triplets. This type of packet is useful if you need to mix **move** and **draw** requests within the same list. The end of the list is indicated by the presence of a list terminator ($01).

Depending upon the function processing the list, the first byte may be expected to contain the scale factor to be used when processing tlist, or this value may need to be stored into RAM.

A sample ‘Packet’ list might look like the following:

byte 0	- Scale factor
bytes 1 / 2 / 3	- ‘mode:Y:X’ for coordinate 1
bytes n / n+1 / n+2	- ‘mode:Y:X’ for coordinate ‘n’
$01	- packet terminator

where ‘mode’ can assume one of the following values:

$00	- Move to the specified point

$FF - Draw a line to the specified point

Power-up / reset characteristics

<This area was empty>

Executive storage description

ABSX

Location: $C835

Description:

Working storage for ‘CMPASS’

Modified by:

CLREX, CMPASS
INTALL, INTMSC

ABSY

Location: $C834

Description:

Working storage for ‘CMPASS’

Modified by:

CLREX, CMPASS
INTALL, INTMSC

ACTPLY

Location: $C89B

Description:

Mine Storm: Currently active player ($00 or $02)

Modified by:

CLREX

Accessed by:

SCRMES
WAIT

ANGLE

Location: $C836

Description:

Location for parameter passing

Modified by:

CLREX, CMPASS
DROT
INTALL, INTMSC
LNROT, LROT90
MLTY8, MLTY16
PROT

Accessed by:

ADROT, ALNROT, APROT
BDROT
SINCOS

BACON

Location: $C876

Description:

Allocated but not used by Mine Storm or Executive

Modified by:

CLREX
INTALL, INTMSC

B1FREQ

Location: $C86B - $C86C (2 bytes)

Description:

Allocated but not used by Mine Storm or Executive

Modified by:

CLREX
INTALL, INTMSC

B2FREQ

Location: $C86D - $C86E (2 bytes)

Description:

Allocated but not used by Mine Store or Executive

Modified by:

CLREX
INTALL, INTMSC

DASH

Location: $C829

Description:
Dash pattern for drawing routines

Modified by:
CLREX
INTALL, INTMSC

Accessed by:
DASHDF, DASHPK, DSHDF, DSHDF1

DOREMI

Location: $C84D - $C84E (2 bytes)

Description:
Note table pointer

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
SPLAY

Accessed by:
REPLAY
TPLAY

DWELL

Location: $C828

Description:
Dot ‘ON’ time

Modified by:
CLREX
DOTTIM
INTALL, INTMSC

Accessed by:

ADOT
DDOT, DIFDOT, DOT, DOTAB, DOTPCK, DOTX

EDGE (HEDGES)

Location: $C811

Description:

Used by button handlers

Modified by:

CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

EPOT0 (DPOT0)

Location: $C81F

Description:

Controller #1: Right / left joystick pot enable (must be $00 or $01)

Modified by:

CLREX
INTALL, INTMSC
WAIT

Accessed by:

JOYBIT, JOYSTK

EPOT1 (DPOT1)

Location: $C820

Description:

Controller #1: Up / down joystick pot enable (must be $00 or $03)

Modified by:

CLREX
INTALL, INTMSC
WAIT

Accessed by:

JOYBIT, JOYSTK

EPOT2 (DPOT2)

Location: $C821

Description:

Controller #2: Right / left joystick #2 enable (must be $00 or $05)

Modified by:

CLREX
INTALL, INTMSC
WAIT

Accessed by:

JOYBIT, JOYSTK

EPOT3 (DPOT3)

Location: $C822

Description:

Controller #2: Up / down joystick #2 enable (must be $00 or $07)

Modified by:

CLREX
INTALL, INTMSC
WAIT

Accessed by:

JOYBIT, JOYSTK

ETMP1

Location: $C883

Description:

Mine Storm: Temporary working storage

Modified by:

CLREX

ETMP2

Location: $C884

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP3

Location: $C885

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP4

Location: $C886

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP5

Location: $C887

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP6

Location: $C888

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP7

Location: $C889

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP8

Location: $C88A

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP9

Location: $C88B

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

ETMP10

Location: $C88c – C88E (3 bytes)

Description:
Mine Storm: Temporary working storage

The last two bytes ($C88D - $C88E) serve as a guard against the sloppy programmer and are not used by existing software.

Modified by:
CLREX

FADE

Location: $C84F - $C850 (2 bytes)

Description:
Working storage for ‘SELOPT’ and tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
REPLAY
SELOPT, SPLAY
TPLAY

Accessed by:
XPLAY

FADEA

Location: $C85E

Description:
Working storage for ‘SELOPT’ and tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
REPLAY
SELOPT, SPLAY
TPLAY

Accessed by:
XPLAY

FADEB

Location: $C85F

Description:
Working storage for ‘SELOPT’ and tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
REPLAY
SELOPT, SPLAY
TPLAY

Accessed by:
XPLAY

FADEC

Location: $C860

Description:
Working storage for ‘SELOPT’ and tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
REPLAY
SELOPT, SPLAY
TPLAY

Accessed by:
XPLAY

FEAST

Location: $C871

Description:
Allocated but not used by Mine Storm or Executive

Modified by:

CLREX
INTALL, INTMSC

FRAME

Location: $C825 - $C826 (2 bytes)

Description:
Frame counter

Modified by:
CLREX
FRWAIT
INTALL, INTMSC
SELOPT
WAIT

FRMTIM (XMSEC)

Location: $C83D – C83E (2 bytes)

Description:
Frame rate

Modified by:
CLREX
INTALL, INTMSC

Accessed by:
FRWAIT
INTALL, INTPIA
SELOPT
WAIT

F1FREQ

Location: $C86F - $C870 (2 bytes)

Description:
Allocated but not used by Mine Store or Executive

Modified by:
CLREX
INTALL, INTMSC

GAP

Location: $C86A

Description:
Allocated but not used by Mine Storm or Executive

Modified by:
CLREX
INTALL, INTMSC

HISCOR (HEIGH)

Location: $C8EB - $C8F1 (7 bytes)

Description:
Contains ASCII high score

KEY0

Location: $C812

Description:
Controller #1 – Key #1: De-bounced key depressed flag

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

KEY1

Location: $C813

Description:
Controller #1 – Key #2: De-bounced key depressed flag

Modified by:
CLREX
DBNCE

INPUT, INTALL, INTMSC
SELOPT
WAIT

KEY2

Location: $C814

Description:
Controller #1 – Key #3: De-bounced key depressed flag

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

KEY3

Location: $C815

Description:
Controller #1 – Key #4: De-bounced key depressed flag

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

KEY4

Location: $C816

Description:
Controller #2 – Key #1: De-bounced key depressed flag

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

KEY5

Location: $C817

Description:
Controller #2 – Key #2: De-bounced key depressed flag

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

KEY6

Location: $C818

Description:
Controller #2 – Key #3: De-bounced key depressed flag

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

KEY7

Location: $C819

Description:
Controller #2 – Key #4: De-bounced key depressed flag

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC
SELOPT
WAIT

K1FREQ

Location: $C874 – $C875 (2 bytes)

Description:
Allocated but not used by Mine Storm or Executive

Modified by:
CLREX
INTALL, INTMSC

LAG

Location: $C83C

Description:
Used by transformation functions

Modified by:
ADROT, ALNROT, APROT
BDROT
CLREX
DROT
INTALL, INTMSC
LCSINE, LNROT, LROT90, LSINE
MCSINE, MLTY8, MLTY16, MSINE
PROT
SELOPT

LASRAM

Location: $C880

Description:
First memory location available for use by a game

LATUS

Location: $C868

Description:
Allocated but not used by Mine Storm or Executive

Modified by:

CLREX
INTALL, INTMSC

LEG

Location: $C83B

Description:

Used by transformation functions

During startup, or after a reboot, controls whether a high score is displayed:

= $00 – Display high score
!= $00 – Don't display high score

Modified by:

ADROT, ALNROT, APROT
BDROT
CLREX
DROT
INTALL, INTMSC
LNROT, LROT90
MCSINE, MLTY8, MLTY16, MSINE
PROT

Accessed by:

LCSINE, LSINE

LIST

Location: $C823

Description:

Number of vectors to be drawn

Modified by:

ADROT, APROT
BDROT, BYTADD
CLREX
DASHDF, DASHPK, DIFDOT, DIFFAB, DIFFAX, DIFFY, DIFLST, DIFTIM, DROT, DSHDF, DSHDF1, DUFFAB, DUFFAX, DUFFY, DUFLST, DUFTIM
INTALL, INTMSC
JOYBIT, JOYSTK
LDIFFY, LDUFFY
PROT
SCRADD, SELOPT, STKADD
TDIFFY, TDUFFY
WAIT

MESAGE

Location: $C82C - $C82D (2 bytes)

Description:

Used by string display functions to hold pointer to message to be displayed.

Modified by:

ASMESS
CLREX
DSHIP
INTALL, INTMSC
MSSPOS
RASTER, RSTPOS, RSTSIZ
SCRBTH, SCRMES, SELOPT, SHIPX, SMESS
TXTPOS, TXTSIZ
WAIT

Accessed by:

MRASTR

NEDGE

Location: $C873

Description:

Allocated but not used by Mine Storm or Executive

Modified by:

CLREX
INTALL, INTMSC

NEWGEN

Location: $C855

Description:

Working storage for tune player subroutines

Modified by:

ASPLAY
CLREX
INTALL, INTMSC
REPLAY
SPLAY

TPLAY

OPTION (GAMZ)

Location: $C87A

Description:
Number of player options ($01 - $09)

Modified by:
CLREX
INTALL, INTMSC
SELOPT

PEDGE

Location: $C872

Description:
Allocated but not used by Mine Storm or Executive

Modified by:
CLREX
INTALL, INTMSC

PLAYRS (PLAYRZ)

Location: $C879

Description:
Number of players ($01 - $09)

Modified by:
CLREX
INTALL, INTMSC
SELOPT

Accessed by:
SCRBTH

POT0

Location: $C81B

Description:

Controller #1: Right / left joystick pot value

Modified by:

CLREX
INTALL, INTMSC
JOYBIT, JOYSTK
WAIT

POT1

Location: $C81C

Description:

Controller #1: Up / down joystick pot value

Modified by:

CLREX
INTALL, INTMSC
JOYBIT, JOYSTK
WAIT

POT2

Location: $C81D

Description:

Controller #2: Right / left joystick pot value

Modified by:

CLREX
INTALL, INTMSC
JOYBIT, JOYSTK
WAIT

POT3

Location: $C81E

Description:

Controller #2: Up / down joystick pot value

Modified by:

CLREX
INTALL, INTMSC
JOYBIT, JOYSTK
WAIT

POTRES

Location: $C81A

Description:
Joystick resolution limit

Modified by:
CLREX
INTALL, INTMSC

Accessed by:
JOYBIT, JOYSTK
WAIT

RAMMES

Location: $CA00

Description:
xxxxxx

RANCID

Location: $C87D – C87F (3 bytes)

Description:
Working storage for random number generators

Modified by:
CLREX, CONE
INTALL, INTMSC
RAND3, RANDOM, RANPOS

RATEA

Location: $C858

Description:
Working storage for ‘EXPLOD’

Modified by:
CLREX
EXPLOD
INTALL, INTMSC

RATEB

Location: $C85A

Description:
Working storage for ‘EXPLOD’

Modified by:
CLREX
EXPLOD
INTALL, INTMSC

RATEC

Location: $C85C

Description:
Working storage for ‘EXPLOD’

Modified by:
CLREX
EXPLOD
INTALL, INTMSC

REG0

Location: $C800

Description:
Channel A: Fine tone period

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG1

Location: $C801

Description:
Channel A: Course tone period

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG2

Location: $C802

Description:
Channel B: Fine tone period

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG3

Location: $C803

Description:
Channel B: Course tone period

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG4

Location: $C804

Description:

Channel C: Fine tone period

Modified by:

CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG5

Location: $C805

Description:

Channel C: Course tone period

Modified by:

CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG6

Location: $C806

Description:

Noise period

Modified by:

CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG7

Location: $C807

Description:

Tone / Noise enables

Modified by:

CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG8

Location: $C808

Description:
Channel A: Amplitude

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REG9

Location: $C809

Description:
Channel B: Amplitude

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REGA

Location: $C80A

Description:
Channel C: Amplitude

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REGB

Location: $C80B

Description:
Fine envelope period

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REGC

Location: $C80C

Description:
Course envelope period

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REGD

Location: $C80D

Description:
Envelope shape / cycle

Modified by:
CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REGE

Location: $C80E

Description:

I/O port data register

Modified by:

CLREX
INTALL, INTMSC, INTPSG
REQOUT
WRREG

REQ0

Location: $C83F

Description:

Envelope shape / cycle

Modified by:

ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

REQ1

Location: $C840

Description:

Course envelope period

Modified by:

ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

REQ2

Location: $C841

Description:
Fine envelope period

Modified by:
ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:
REQOUT

REQ3

Location: $C842

Description:
Channel C: Amplitude

Modified by:
ASPLAY
CLREX
EXPLOD
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SETAMP, SPLAY
TPLAY
XPLAY

Accessed by:
REQOUT

REQ4

Location: $C843

Description:
Channel B: Amplitude

Modified by:
ASPLAY

CLREX
EXPLOD
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SETAMP, SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

REQ5

Location: $C844

Description:

Channel A: Amplitude

Modified by:

ASPLAY
CLREX
EXPLOD
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SETAMP, SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

REQ6

Location: $C845

Description:

Tone / noise enables

Modified by:

ASPLAY
CLREX
EXPLOD
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

REQ7

Location: $C846

Description:
Noise period

Modified by:
ASPLAY
CLREX
EXPLOD
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:
REQOUT

REQ8

Location: $C847

Description:
Channel C: Course tone period

Modified by:
ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:
REQOUT

REQ9

Location: $C848

Description:
Channel C: Fine tone period

Modified by:
ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:
REQOUT

REQA

Location: $C849

Description:
Channel B: Course tone period

Modified by:
ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:
REQOUT

REQB

Location: $C84A

Description:
Channel B: Fine tone period

Modified by:
ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

REQC

Location: $C84B

Description:

Channel A: Course tone period

Modified by:

ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

REQD

Location: $C84C

Description:

Channel A: Fine tone period

Modified by:

ASPLAY
CLREX
INTALL, INTMSC, INTPSG, INTREQ
REPLAY
SPLAY
TPLAY
XPLAY

Accessed by:

REQOUT

RESTC

Location: $C857

Description:

Working storage for tune player subroutines

Modified by:

ASPLAY
CLREX
INTALL, INTMSC
REPLAY
SPLAY
TPLAY
XPLAY

SATUS

Location: $C867

Description:

Working storage for ‘EXPLOD’

Modified by:

CLREX
EXPLOD
INTALL, INTMSC

SBTN

Location: $C880

Description:

xxxxxx

Modified by:

CLREX

Accessed by:

WAIT

SCOR1

Location: $C8A8 - $C8AE (7 bytes)

Description:

Holds player 1’s score

Modified by:

CLREX

SCOR2

Location: $C8AF – C8B5 (7 bytes)

Description:

Holds player 2’s score

Modified by:

CLREX

SEED

Location: $C87B - $C87C (2 bytes)

Description:

Seed used by the random number generator

Modified by:

CLREX
INTALL, INTMSC

SIZRAS

Location: $C82A - $C82B (2 bytes)

Description:

Used by the string display functions to hold the height and width to be used when displaying the string

Modified by:

CLREX
INTALL, INTMSC
RSTSIZ
SCRBTH, SCRMES, SELOPT, SMESS
WAIT

Accessed by:

ASMESS
DSHIP
MRASTR, MSSPOS
RASTER, RSTPOS
SHIPX

TXTPOS

SJOY

Location: $C881

Description:
Joystick mask (used by MineStorm)

Modified by:
CLREX

Accessed by:
WAIT

SPEKT

Location: $C878

Description:
Allocated but not used by Mine Store or Executive

Modified by:
CLREX
INTALL, INTMSC

TENSTY

Location: $C827

Description:
Contains the last value used for the intensity setting

Modified by:
CLREX
INT1Q, INT2Q, INT3Q, INTALL, INTENS, INTMAX, INTMSC
SCRBTH, SCRMES, SELOPT
WAIT

TEMP1

Location: $C88F

Description:

Mine Storm: Temporary working storage

Modified by:

CLREX

TEMP2

Location: $C890

Description:

Mine Storm: Temporary working storage

Modified by:

CLREX

TEMP3

Location: $C891

Description:

Mine Storm: Temporary working storage

Modified by:

CLREX

TEMP4

Location: $C892

Description:

Mine Storm: Temporary working storage

Modified by:

CLREX

TEMP5

Location: $C893

Description:

Mine Storm: Temporary working storage

Modified by:
CLREX

TEMP6

Location: $C894

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

TEMP7

Location: $C895

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

TEMP8

Location: $C896

Description:
Mine Storm: Temporary working storage

Modified by:
CLREX

TEMP9

Location: $C897

Description:
Mine Storm: Temporary working storage

Modified by:

CLREX

TEMP10

Location: $C898 - $C89A (3 bytes)

Description:

Mine Storm: Temporary working storage

The last two bytes *$C899 - $C89A) serve as a guard against the sloppy programmer and are not used by existing software.

Modified by:

CLREX

TMR1

Location: $C89C - $C89E (3 bytes)

Description:

This is a 1-byte countdown counter followed by a 2-byte function address. When the counter counts down to zero, the function is called. This is used by MineStorm; the counter is decremented each pass through the main loop.

Modified by:

CLREX
WAIT

TMR2

Location: $C89F - $C8A1 (3 bytes)

Description:

This is a 1-byte countdown counter followed by a 2-byte function address. When the counter counts down to zero, the function is called. This is used by MineStorm; the counter is decremented each pass through the main loop.

Modified by:

CLREX
WAIT

TMR3

Location: $C8A2 – C8A4 (3 bytes)

Description:

This is a 1-byte countdown counter followed by a 2-byte function address. When the counter counts down to zero, the function is called. This is used by MineStorm; the counter is decremented each pass through the main loop.

Modified by:

CLREX
WAIT

TMR4

Location: $C8A5 - $ C8A7 (3 bytes)

Description:

This is a 1-byte countdown counter followed by a 2-byte function address. When the counter counts down to zero, the function is called. This is used by MineStorm; the counter is decremented each pass through the main loop.

Modified by:

CLREX
WAIT

TONEA

Location: $C861 - $C862 (2 bytes)

Description:

Working storage for 'SELOPT' and tune player subroutines

Modified by:

CLREX
INTALL, INTMSC
SELOPT

Accessed by:

ASPLAY
REPLAY
SPLAY
TPLAY
XPLAY

TONEB

Location: $C863 - $C864 (2 bytes)

Description:
Working storage for 'SELOPT' and tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
SELOPT, SPLAY
TPLAY

Accessed by:
REPLAY
XPLAY

TONEC

Location: $C865 - $C866 (2 bytes)

Description:
Working storage for tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
SPLAY
TPLAY

Accessed by:
REPLAY
XPLAY

TRIGGR

Location: $C80F - $C810 (2 bytes)

Description:
Current controller button status. ('TRIGGR + 1' contains an image of 'TRIGGR' from the last time that the buttons were read).

Modified by:
CLREX
DBNCE
INPUT, INTALL, INTMSC

SELOPT
WAIT

TSTAT

Location: $C856

Description:
Working storage for tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
REPLAY
SPLAY
TPLAY
XPLAY

TUNE

Location: $C853 - $C854 (2 bytes)

Description:
xxxxxx

Modified by:
ASPLAY
CLREX
EXPLOD
INTALL, INTMSC
REPLAY
SPLAY
TPLAY
XPLAY

VIBA

Location: $C859

Description:
Working storage for ‘EXPLOD’

Modified by:
CLREX

EXPLOD
INTALL, INTMSC

VIBB

Location: $C85B

Description:
Working storage for ‘EXPLOD’

Modified by:
CLREX
EXPLOD
INTALL, INTMSC

VIBC

Location: $C85D

Description:
Working storage for ‘EXPLOD’

Modified by:
CLREX
EXPLOD
INTALL, INTMSC

Accessed by:

VIBE

Location: $C851

Description:
Working storage for tune player subroutines

Modified by:
ASPLAY
CLREX
INTALL, INTMSC
SPLAY
TPLAY

Accessed by:

REPLAY

WCSINE (COSINE)

Location: $C839 – C83A (2 bytes)

Description:
Location for parameter passing. Generally contains the last cosine value calculated.

Modified by:
ADROT, ALNROT, APROT
BDROT
CLREX
DROT
INTALL, INTMSC
LNROT, LROT90
MLTY8, MLTY16
PROT
SINCOS

Accessed by:
LCSINE
MCSINE

WSINE (SINE)

Location: $C837 - $C838 (2 bytes)

Description:
Location for parameter passing. Generally contains the last sine value calculated

Modified by:
ADROT, ALNROT, APROT
BDROT
CLREX
DROT
INTALL, INTMSC
LNROT, LROT90
MLTY8, MLTY16
PROT
SINCOS

Accessed by:
LSINE
MSINE

XACON

Location: $C877

Description:
Working storage for ‘EXPLOD’

Modified by:CLREX
EXPLOD
INTALL, INTMSC

XATUS

Location: $C869

Description:
Allocated but not used by Mine Storm or Executive

Modified by:
CLREX
INTALL, INTMSC

XTMR0 (X0)

Location: $C82E

Description:
xxxxxx

Modified by:
CLREX
D3TMR, DECTMR
INTALL, INTMSC
SELOPT

XTMR1 (X1)

Location: $C82F

Description:
Countdown timer

Modified by:
CLREX
D3TMR, DECTMR

INTALL, INTMSC
SELOPT

XTMR2 (X2)

Location: $C830

Description:
Countdown timer

Modified by:
CLREX
D3TMR, DECTMR
INTALL, INTMSC
SELOPT

XTMR3 (X3)

Location: $C831

Description:
Countdown timer

Modified by:
CLREX
DECTMR
INTALL, INTMSC

XTMR4 (X4)

Location: $C832

Description:
Countdown timer

Modified by:
CLREX
DECTMR
INTALL, INTMSC

XTMR5 (X5)

Location: $C833

Description:

Countdown timer

Modified by:

CLREX
DECTMR
INTALL, INTMSC

ZSKIP (ZGO)

Location: $C824

Description:

Flag controlling whether integrators will be zeroed:

= $00 = Skip integrator zeroing
!= $00 – Zero integrators

Modified by:

CLREX
INTALL, INTMSC

Accessed by:

APACK
CZERO
DASHDF, DASHPK, DIFFAB, DIFFAX, DIFFY, DIFLST, DIFTIM, DPACK, DSHDF, DSHDF1, DUFFAB, DUFFAX, DUFFY, DUFLST, DUFTIM
LDIFFY, LDUFFY, LPACK
PACK1X, PACK2X, PACKET
TDIFFY, TDUFFY, TPACK

Condensed Executive Parameter Requirements

Subroutine	Entry Values	Return Values
ABSAB	A, B	A, B
ABSB	B	B
ACTGND	DP	A, B
ADOT	Y, DP, DWELL	Same as entry values
ADROT	X, U, ANGLE, LIST	A, B, X, U, LIST
ALNROT	A, DP, ANGLE	A, B
APACK	B, X, Y, DP, ZSKIP	Same as entry values
APROT	X, U, ANGLE	A, B, X, U, LIST
ASMESS	Y, U, DP, SIZRAS	Same as entry values
ASPLAY	X, U, DP	A, B, X, Y, U
BCLR	B, X	A, B
BDROT	B, X, U, ANGLE	A, B, X, U
BLKFIL	A, B, X	B
BLKMOV	A, X, U	A, B
BLKMV1	A, X, U	A, B
BXTEST	A, B, X, Y	CC
BYTADD	A, X, LIST	A, B, U
CLRBLK	A, B, X	A, B
CLREX	None required	A, B, X
CLR80	B, X	A, B
CLR256	X	A, B
CMPASS	A, B, DP	A, B, ANGLE
CONE	SEED	B
COSINE	A	A, B, X
CZERO	DP, ZSKIP	A, B
D3TMR	None required	B, X
DASHDF	X, DP, DASH, LIST	A, B, X, LIST, T1LOLC, ZSKIP
DASHPK	X, DP, DASH, T1LOLC, ZSKIP	A, B, X, LIST
DBNCE	A, DP	A, B, X, EDGE, KEY0 – KEY7, TRIGGR, TRIGGR + 1
DDOT	Y, DP, DWELL	Same as entry values
DECBIT	A	A, X
DECTMR	None required	B, X
DEFLOK	DP	A, B, X
DEL	B	B
DEL13	None required	Same as entry values
DEL20	None required	B
DEL28	None required	B
DEL33	None required	B
DEL38	None required	B
DIFDOT	X, DP, DWELL, LIST, T1LOLC	A, B, X, LIST
DIFFAB	A, B, DP, LIST, T1LOLC, ZSKIP	A, B, X
DIFFAX	X, DP, T1LOLC, ZSKIP	A, B, X, LIST

DIFFY	X, DP, LIST, T1LOLC, ZSKIP	A, B, X, LIST
DIFLST	X, DP, ZSKIP	A, B, X, LIST
DIFTIM	B, X, DP, LIST, ZSKIP	A, B, X, LIST
DOT	DP, DWELL	A, B
DOTAB	A, B, DP, DWELL, T1LOLC	A, B
DOTPCK	X, DP, DWELL, T1LOLC	A, B, X
DOTTIM	B, X, DP, T1LOLC	A, B, X
DOTX	X, DP, DWELL, T1LOLC	A, B, X
DPACK	B, X, Y, DP	Same as entry values
DPIO	None required	A, DP
DPRAM	None required	A, DP
DROT	A, B, X, U	A, B, X, U, LIST
DSHDF	A, X, DP, DASH, T1LOLC, ZSKIP	A, B, X, LIST
DSHDF1	A, X, DP, DASH, T1LOLC, ZSKIP	A, B, X, LIST
DSHIP	A, B, X, DP, SIZRAS	A, B, X, U
DUFFAB	A, B, DP, LIST, T1LOLC, ZSKIP	A, B, X
DUFFAX	X, DP, T1LOLC, ZSKIP	A, B, X, LIST
DUFFY	X, DP, LIST, T1LOLC, ZSKIP	A, B, X, LIST
DUFLST	X, DP, ZSKIP	A, B, X, LIST
DUFTIM	B, X, DP, ZSKIP	A, B, X, LIST
DZERO	None required	A, B, DP
EXPLOD	U, DP	A, B, X, XACON
FRWAIT	FRMTIM	A, B, X, DP`
HISCR	X, U	A, B, X, U
INPUT	DP	A, B, X
INT1Q	DP	A, B
INT2Q	DP	A, B
INT3Q	DP	A, B
INTALL	None required	A, B, X, DP
INTENS	A, DP	A, B
INTMAX	DP	A, B
INTMSC	None required	A, B, X, DP
INTPIA	FRMTIM	A, B, X, DP
INTPSG	DP	A, B, X
INTREQ	None required	A, B, X
JOYBIT	DP, EPOTx, LIST, POTRES	A, B, X, LIST, POTx
JOYSTK	DP, EPOTx, POTRES	A, B, X, LIST, POTx
LCSINE	DP, LEG, WCSINE	A, B
LDIFFY	A, X, DP, T1LOLC, ZSKIP	A, B, X, LIST
LDUFFY	A, X, DP, T1LOLC, ZSKIP	A, B, X, LIST

LNROT	A, B, DP	A, B
LPACK	X, DP, ZSKIP	A, B, X
LROT90	A, B, DP	A, B
LSINE	DP, LEG, WSINE	A, B
MCSINE	A, DP, WCSINE	A, B, LEG
MLTY8	A, B, DP	X, Y
MLTY16	A, B, DP	X, Y
MRASTR	DP, MESAGE, SIZRAS	A, B, X, U`
MSINE	A, DP	A, B, LEG
MSSPOS	A, B, U, DP, SIZRAS	A, B, X, U
OFF1BX	A, B, X, Y, U	CC
OFF2BX	A, B, X, Y, U	CC
PACKET	X, DP, T1LOLC, ZSKIP	A, B, X
PACK1X	X, DP, ZSKIP	A, B, X
PACK2X	X, DP, ZSKIP	A, B, X
POSIT1	X, DP	A, B, X
POSIT2	X, DP	A, B, X
POSITB	B, X, DP	A, B, X
POSITD	A, B, DP	A, B
POSITN	A, B, DP, T1LOLC	A, B
POSITX	X, DP, T1LOLC	A, B, X
POSWID	X, DP	A, B
PROT	A, X, U	A, B, X, U, LIST
PSGLST	U, DP	A, B, X, U
PSGMIR	X, U, DP	A, B, U
RAND3	SEED	A
RANDOM	SEED	A
RANPOS	SEED	A, B
RASTER	U, DP, SIZRAS	A, B, X, U
REQOUT	DP, REQ0 – REQD	A, B, X, U
REPLAY	U, DP	A, B, X, Y, U
RSTPOS	U, DP, SIZRAS	A, B, X, U
RSTSIZ	U, DP	A, B, X, U
SCLR	X	A, B
SCRADD	A, B, X, LIST	A, B
SCRBTH	DP, PLAYRS, SCOR1, SCOR2	A, B, Y, U
SCRMES	DP, ACTPLY, SCOR1, SCOR2	A, B, Y, U
SELOPT	A, B	PLAYRS, OPTION
SETAMP	B, DP	A, X
SHIPX	A, B, X, DP, SIZRAS	A, B, X, U
SINCOS	DP, ANGLE	A, B, WCSINE, WSINE
SINE	A	A, B, X
SMESS	U, DP	Same as entry values
SPLAY	U, DP	A, B, X, Y, U
STKADD	S, LIST	A, B, S
TDIFFY	A, B, X, DP, ZSKIP	A, B, X, LIST
TDUFFY	A, B, X, DP, ZSKIP	A, B, X, LIST
TIMER		
TPACK	B, X, DP, ZSKIP	A, B, X

TPLAY	U, DP	A, B, X, Y, U
TXTPOS	U, DP, SIZRAS	A, B, X, U
TXTSIZ	U, DP	A, B, X, U
WAIT	None required	
WINNER	X, U	A, B, X, U
WRPSG	A, B, X, DP	B
WRREG	A, B, DP	B, X
XPLAY	DP	A, B, X, TSTAT
ZERGND	DP	A, B
ZERO	DP	A, B

Ram And Function Classifications

ASPLAY, Working storage, FADE
ASPLAY, Working storage, FADEA
ASPLAY, Working storage, FADEB
ASPLAY, Working storage, FADEC
ASPLAY, Working storage, NEWGEN
ASPLAY, Working storage, RESTC
ASPLAY, Working storage, TONEA
ASPLAY, Working storage, TONEB
ASPLAY, Working storage, TONEC
ASPLAY, Working storage, TSTAT
ASPLAY, Working storage, VIBE

Absolute value ABSAB (ABSVAL)
Absolute value, ABSB (AOK)

Active-ground, ACTGND (ZEREF)
Active-ground, CZERO (ZEGO)
Active-ground, DEFLOK
Active-ground, DZERO (ZERO.DP)
Active-ground, FRWAIT (FRAM20)
Active-ground, WAIT
Active-ground, ZERGND (ZEROIT)
Active-ground, ZERO (ZERO.)

Arithmetic subroutines, General, ABSAB (ABSVAL)
Arithmetic subroutines, General, ABSB (AOK)
Arithmetic subroutines, General, DECBIT (BITE)

Arithmetic subroutines, Random number, CONE
Arithmetic subroutines, Random number, RAND3
Arithmetic subroutines, Random number, RANDOM
Arithmetic subroutines, Random number, RANPOS

Arithmetic subroutines, Raster Score, BYTADD (SHADD)
Arithmetic subroutines, Raster Score, HISCR (HIGHSCR)
Arithmetic subroutines, Raster Score, SCLR
Arithmetic subroutines, Raster Score, SCRADD (SADD)
Arithmetic subroutines, Raster Score, STKADD (SADD2)
Arithmetic subroutines, Raster Score, WINNER

Arithmetic subroutines, Rotation, DIFFY, ADROT (DIFROT)
Arithmetic subroutines, Rotation, DIFFY, BDROT (DISROT)
Arithmetic subroutines, Rotation, DIFFY, DROT (DANROT)

Arithmetic subroutines, Rotation, DUFFY, ADROT (DIFROT)
Arithmetic subroutines, Rotation, DUFFY, BDROT (DISROT)
Arithmetic subroutines, Rotation, DUFFY, DROT (DANROT)

Arithmetic subroutines, Rotation, LINE, ALNROT (ROTAR)
Arithmetic subroutines, Rotation, LINE, LNROT (ROTOR)
Arithmetic subroutines, Rotation, LINE, LROT90 (RATOR)
Arithmetic subroutines, Rotation, LINE, MLTY16

Arithmetic subroutines, Rotation, LINE, MLTY8

Arithmetic subroutines, Rotation, PACKET, APROT (POTATE)
Arithmetic subroutines, Rotation, PACKET, PROT (POTATA)

Arithmetic subroutines, Trigonometric, ALNROT (ROTAR)
Arithmetic subroutines, Trigonometric, CMPASS (COMPAS)
Arithmetic subroutines, Trigonometric, COSINE (COSGET)
Arithmetic subroutines, Trigonometric, LCSINE (RCOS)
Arithmetic subroutines, Trigonometric, LNROT (ROTOR)
Arithmetic subroutines, Trigonometric, LROT90 (RATOR)
Arithmetic subroutines, Trigonometric, LSINE (RSIN)
Arithmetic subroutines, Trigonometric, MCSINE (RCOSA)
Arithmetic subroutines, Trigonometric, MLTY16
Arithmetic subroutines, Trigonometric, MLTY8
Arithmetic subroutines, Trigonometric, MSINE (RSINA)
Arithmetic subroutines, Trigonometric, SINCOS
Arithmetic subroutines, Trigonometric, SINE (SINGET)

Block initialization, BCLR (CLRSOM)
Block initialization, BLKFIL (FILL)
Block initialization, CLR256
Block initialization, CLR80 (NEGSOM)
Block initialization, CLRBLK (GILL)
Block initialization, CLREX (CLRMEM)

Block transfer, BLKMOV (STFAUX)
Block transfer, BLKMV1 (BAGAUX)

Button related RAM, EDGE (HEDGES)
Button related RAM, KEY0
Button related RAM, KEY1
Button related RAM, KEY2
Button related RAM, KEY3
Button related RAM, KEY4
Button related RAM, KEY5
Button related RAM, KEY6
Button related RAM, KEY7
Button related RAM, SBTN
Button related RAM, TRIGGR

Button related subroutines, DBNCE (ENPUT)
Button related subroutines, INPUT
Button related subroutines, SELOPT (OPTION)
Button related subroutines, WAIT

CMPASS, Working Storage, ABSY
CMPASS, Working Storage, ABSX

CONE, Working storage, RANCID

Collision subroutines, BXTEST (FINEBOX)
Collision subroutines, OFF1BX (OFF1BOX)
Collision subroutines, OFF2BX (OFF2BOX)

Controller related RAM, EDGE (HEDGES)

Controller related RAM, EPOT0 (DPOT0)
Controller related RAM, EPOT1 (DPOT1)
Controller related RAM, EPOT2 (DPOT2)
Controller related RAM, EPOT3 (DPOT3)
Controller related RAM, KEY0
Controller related RAM, KEY1
Controller related RAM, KEY2
Controller related RAM, KEY3
Controller related RAM, KEY4
Controller related RAM, KEY5
Controller related RAM, KEY6
Controller related RAM, KEY7
Controller related RAM, POT0
Controller related RAM, POT1
Controller related RAM, POT2
Controller related RAM, POT3
Controller related RAM, POTRES
Controller related RAM, SBTN
Controller related RAM, SJOY
Controller related RAM, TRIGGR

Controller related subroutines, DBNCE (ENPUT)
Controller related subroutines, INPUT
Controller related subroutines, JOYBIT (PBANG4)
Controller related subroutines, JOYSTK (POTS4)
Controller related subroutines, WAIT

Cosine subroutines, COSINE (COSGET)
Cosine subroutines, LCSINE (RCOS)
Cosine subroutines, MCSINE (RCOSA)
Cosine subroutines, SINCOS

DASH format, DASHDF (DASHY)
DASH format, DASHPK (DASHY3)
DASH format, DSHDF (DASHEL)
DASH format, DSHDF1 (DASHE)

DIFFY format, ADROT (DIFROT)
DIFFY format, BDROT (DISROT)
DIFFY format, DASHDF (DASHY)
DIFFY format, DIFDOT
DIFFY format, DIFFAX
DIFFY format, DIFFY
DIFFY format, DIFLST (DIFFX)
DIFFY format, DIFTIM
DIFFY format, DROT (DANROT)
DIFFY format, DSHDF (DASHEL)
DIFFY format, DSHDF1 (DASHE)
DIFFY format, LDIFFY (DIFLST)
DIFFY format, TDIFFY (DIFTLS)

DOT format, ADOT
DOT format, DDOT
DOT format, DIFDOT
DOT format, DOT
DOT format, DOTAB

DOT format, DOTPCK (DOTPAK)
DOT format, DOTTIM
DOT format, DOTX

DUFFY format, ADROT (DIFROT)
DUFFY format, BDROT (DISROT)
DUFFY format, DIFDOT
DUFFY format, DROT (DANROT)
DUFFY format, DUFFAX
DUFFY format, DUFFY
DUFFY format, DUFLST
DUFFY format, DUFTIM
DUFFY format, LDUFFY (DUFLST)
DUFFY format, TDUFFY (DUFTLS)

Delay subroutines, DEL
Delay subroutines, DEL13
Delay subroutines, DEL20
Delay subroutines, DEL28,
Delay subroutines, DEL33
Delay subroutines, DEL38

Drawing formats, ‘Diffy’
Drawing formats, ‘Duffy’
Drawing formats, ‘Packet’

Drawing subroutines, DASHED, DIFFY format, DASHDF (DASHY)
Drawing subroutines, DASHED, DIFFY format, DSHDF (DASHEL)
Drawing subroutines, DASHED, DIFFY format, DSHDF1 (DASHE)

Drawing subroutines, DASHED, DUFFY format, DASHDF (DASHY)
Drawing subroutines, DASHED, DUFFY format, DSHDF (DASHEL)
Drawing subroutines, DASHED, DUFFY format, DSHDF1 (DASHE)

Drawing subroutines, DASHED, PACKET format, DASHPK (DASHY3)

Drawing subroutines, DIFFY DOT format, DIFDOT
Drawing subroutines, DIFFY DOT format, DOTTIM
Drawing subroutines, DIFFY DOT format, DOTX

Drawing subroutines, DIFFY format, DIFFAX
Drawing subroutines, DIFFY format, DIFFY
Drawing subroutines, DIFFY format, DIFLST (DIFFX)
Drawing subroutines, DIFFY format, DIFTIM
Drawing subroutines, DIFFY format, LDIFFY (DIFLST)
Drawing subroutines, DIFFY format, TDIFFY (DIFTLS)

Drawing subroutines, DOT format, ADOT
Drawing subroutines, DOT format, DDOT
Drawing subroutines, DOT format, DOT
Drawing subroutines, DOT format, DOTAB

Drawing subroutines, DUFFY format, DUFFAX
Drawing subroutines, DUFFY format, DUFFY
Drawing subroutines, DUFFY format, DUFLST (DUFFX)
Drawing subroutines, DUFFY format, DUFTIM

Drawing subroutines, DUFFY format, LDUFFY (DUFLST)
Drawing subroutines, DUFFY format, TDUFFY (DUFTLS)

Drawing subroutines, MARKER format, DSHIP (SHIPSHO)
Drawing subroutines, MARKER format, SHIPX (SHIPSAT)

Drawing subroutines, PACKET DOT format, DOTPCK (DOTPAK)

Drawing subroutines, PACKET format, APACK
Drawing subroutines, PACKET format, DPACK
Drawing subroutines, PACKET format, LPACK (PACXX)
Drawing subroutines, PACKET format, PACK1X (PAC1X)
Drawing subroutines, PACKET format, PACK2X (PAC2X)
Drawing subroutines, PACKET format, PACKET
Drawing subroutines, PACKET format, TPACK (PACB)

Drawing subroutines, RASTER format, ASMESS
Drawing subroutines, RASTER format, DSHIP (SHIPSHO)
Drawing subroutines, RASTER format, MRASTR (RASTER)
Drawing subroutines, RASTER format, MSSPOS (POSDRAS)
Drawing subroutines, RASTER format, RASTER (RASTUR)
Drawing subroutines, RASTER format, RSTPOS (POSNRAS)
Drawing subroutines, RASTER format, RSTSIZ (SIZPRAS)
Drawing subroutines, RASTER format, SCRBTH
Drawing subroutines, RASTER format, SCRMES
Drawing subroutines, RASTER format, SELOPT (OPTION)
Drawing subroutines, RASTER format, SHIPX (SHIPSAT)
Drawing subroutines, RASTER format, TXTPOS (TEXPOS)
Drawing subroutines, RASTER format, TXTSIZ (TEXSIZ)

Drawing subroutines, SCORE format, SCRBTH
Drawing subroutines, SCORE format, SCRMES
Drawing subroutines, SCORE format, WAIT

EXPLOD, Working storage, RATEA
EXPLOD, Working storage, RATEB
EXPLOD, Working storage, RATEC
EXPLOD, Working storage, SATUS
EXPLOD, Working storage, VIBA
EXPLOD, Working storage, VIBB
EXPLOD, Working storage, VIBC
EXPLOD, Working storage, XACON

Hardware related subroutines, ACTGND (ZEREF)
Hardware related subroutines, CZERO (ZEGO)
Hardware related subroutines, DBNCE (ENPUT)
Hardware related subroutines, DEFLOK
Hardware related subroutines, DEL
Hardware related subroutines, DEL13
Hardware related subroutines, DEL20
Hardware related subroutines, DEL28
Hardware related subroutines, DEL33
Hardware related subroutines, DEL38
Hardware related subroutines, DPIO
Hardware related subroutines, DPRAM
Hardware related subroutines, DZERO (ZERO.DP)

Joystick related RAM, EPOT0 (DPOT0)
Joystick related RAM, EPOT1 (DPOT1)
Joystick related RAM, EPOT2 (DPOT2)
Joystick related RAM, EPOT3 (DPOT3)
Joystick related RAM, POT0
Joystick related RAM, POT1
Joystick related RAM, POT2
Joystick related RAM, POT3
Joystick related RAM, POTRES
Joystick related RAM, SJOY

Joystick related subroutines, JOYBIT (PBANG4)
Joystick related subroutines, JOYSTK (POTS4)
Joystick related subroutines, WAIT

MARKER format, DSHIP (SHIPSHO)
MARKER format, SHIPX (SHIPSAT)

Memory block subroutines, ADROT (DIFROT)
Memory block subroutines, APROT (POTATE)
Memory block subroutines, BCLR (CLRSOM)
Memory block subroutines, BDROT (DISROT)
Memory block subroutines, BLKFIL (FILL)
Memory block subroutines, BLKMOV (STFAUX)
Memory block subroutines, BLKMV1 (BAGAUX)
Memory block subroutines, CLR256
Memory block subroutines, CLR80 (NEGSOM)
Memory block subroutines, CLRBLK (GILL)
Memory block subroutines, CLREX (CLRMEM)
Memory block subroutines, DROT (DANROT)
Memory block subroutines, PROT (POTATA)

PACKET format, APACK
PACKET format, APROT (POTATE)
PACKET format, DASHPK (DASHY3)
PACKET format, DOTPCK (DOTPAK)
PACKET format, DPACK
PACKET format, LPACK (PACXX)
PACKET format, PACK1X (PAC1X)
PACKET format, PACK2X (PAC2X)
PACKET format, PACKET
PACKET format, PROT (POTATA)
PACKET format, TPACK (PACB)

PIA initialization, INTALL (INITALL)
PIA initialization, INTPIA (INITPIA)

PSG Sound mirror, REG0
PSG Sound mirror, REG1
PSG Sound mirror, REG2
PSG Sound mirror, REG3
PSG Sound mirror, REG4
PSG Sound mirror, REG5
PSG Sound mirror, REG6
PSG Sound mirror, REG7

PSG Sound mirror, REG8
PSG Sound mirror, REG9
PSG Sound mirror, REGA
PSG Sound mirror, REGB
PSG Sound mirror, REGC
PSG Sound mirror, REGD
PSG Sound mirror, REGE

PSG Working Storage, REQ0
PSG Working Storage, REQ1
PSG Working Storage, REQ2
PSG Working Storage, REQ3
PSG Working Storage, REQ4
PSG Working Storage, REQ5
PSG Working Storage, REQ6
PSG Working Storage, REQ7
PSG Working Storage, REQ8
PSG Working Storage, REQ9
PSG Working Storage, REQA
PSG Working Storage, REQB
PSG Working Storage, REQC
PSG Working Storage, REQD

PSG initialization, INTALL (INITALL)
PSG initialization, INTPSG (INITPSG)

Positioning subroutines, POSIT1
Positioning subroutines, POSIT2
Positioning subroutines, POSITB
Positioning subroutines, POSITD
Positioning subroutines, POSITN
Positioning subroutines, POSITX
Positioning subroutines, POSWID

Programmable Timers, D2TMR (DEKR3)
Programmable Timers, DECTMR (DEKR)
Programmable Timers, WAIT
Programmable Timers, Executive, XTMR0 (X0)
Programmable Timers, Executive, XTMR1 (X1)
Programmable Timers, Executive, XTMR2 (X2)
Programmable Timers, Executive, XTMR3 (X3)
Programmable Timers, Executive, XTMR4 (X4)
Programmable Timers, Executive, XTMR5 (X5)
Programmable Timers, Mine Storm, TMR1
Programmable Timers, Mine Storm, TMR2
Programmable Timers, Mine Storm, TMR3
Programmable Timers, Mine Storm TMR4

RAM initialization, BCLR (CLRSOM)
RAM initialization, BLKFIL (FILL)
RAM initialization, CLR256
RAM initialization, CLR80 (NEGSOM)
RAM initialization, CLRBLK (GILL)
RAM initialization, CLREX (CLRMEM)
RAM initialization, INTALL (INITALL)
RAM initialization, SCLR

SELOPT, Working storage, FADEC
SELOPT, Working storage, TONEA
SELOPT, Working storage, TONEB

SPLAY, Working storage, FADE
SPLAY, Working storage, FADEA
SPLAY, Working storage, FADEB
SPLAY, Working storage, FADEC
SPLAY, Working storage, NEWGEN
SPLAY, Working storage, RESTC
SPLAY, Working storage, TONEA
SPLAY, Working storage, TONEB
SPLAY, Working storage, TONEC
SPLAY, Working storage, TSTAT
SPLAY, Working storage, VIBE

Sine subroutines, LSINE (RSIN)
Sine subroutines, MSINE (RSINA)
Sine subroutines, SINCOS
Sine subroutines, SINE (SINGET)

Sound related subroutines, EXPLOD (AXE)
Sound related subroutines, INTPSG (INITPSG)
Sound related subroutines, INTREQ (IREQ)
Sound related subroutines, PSGLST (PSGLUP)
Sound related subroutines, PSGMIR (PSGULP)
Sound related subroutines, REQOUT
Sound related subroutines, SETAMP (LOUDIN)
Sound related subroutines, WRPSG (PSG)
Sound related subroutines, WRREG (PSGX)

TPLAY, Working storage, FADE
TPLAY, Working storage, FADEA
TPLAY, Working storage, FADEB
TPLAY, Working storage, FADEC
TPLAY, Working storage, NEWGEN
TPLAY, Working storage, RESTC
TPLAY, Working storage, TONEA
TPLAY, Working storage, TONEB
TPLAY, Working storage, TONEC
TPLAY, Working storage, TSTAT
TPLAY, Working storage, VIBE

Temporary Storage, Mine Storm, ETMP1
Temporary Storage, Mine Storm, ETMP2
Temporary Storage, Mine Storm, ETMP3
Temporary Storage, Mine Storm, ETMP4
Temporary Storage, Mine Storm, ETMP5
Temporary Storage, Mine Storm, ETMP6
Temporary Storage, Mine Storm, ETMP7
Temporary Storage, Mine Storm, ETMP8
Temporary Storage, Mine Storm, ETMP9
Temporary Storage, Mine Storm, ETMP10
Temporary Storage, Mine Storm, TEMP1
Temporary Storage, Mine Storm, TEMP2
Temporary Storage, Mine Storm, TEMP3

Temporary Storage, Mine Storm, TEMP4
Temporary Storage, Mine Storm, TEMP5
Temporary Storage, Mine Storm, TEMP6
Temporary Storage, Mine Storm, TEMP7
Temporary Storage, Mine Storm, TEMP8
Temporary Storage, Mine Storm, TEMP9
Temporary Storage, Mine Storm, TEMP10

Tune player subroutines, ASPLAY (SOPLAY)
Tune player subroutines, REPLAY
Tune player subroutines, REQOUT
Tune player subroutines, SPLAY
Tune player subroutines, TPLAY(YOPLAY)
Tune player subroutines, XPLAY

XPLAY, Working storage, FADE
XPLAY, Working storage, FADEA
XPLAY, Working storage, FADEB
XPLAY, Working storage, FADEC
XPLAY, Working storage, RESTC
XPLAY, Working storage, TONEA
XPLAY, Working storage, TONEB
XPLAY, Working storage, TONEC
XPLAY, Working storage, TSTAT