PokeyMAX developers guide (for core version v1.17)

The PokeyMAX may contain several audio chips, depending on the version

Memory map

$D200-$D20F	Pokey1
$D210-$D21F	Pokey2/Configuration
$D220-$D22F	Pokey3
$D230-$D23F	Pokey4
$D240-$D25F	SID1
$D260-$D27F	SID2
$D280-$D283	COVOX
$D284-$D29F	SAMPLE
$D2A0-$D2AF	PSG1
$D2B0-$D2BF	PSG2
$D2C0-$D2FF	RESERVED

Reduced memory maps

Mono (or any PokeyMAX when in mono mode)

$D200-$D20F	Pokey1
$D210-$D2FF	Shadows of Pokey1

Stereo

$D200-$D20F	Pokey1
$D210-$D21F	Pokey2/Configuration
$D220-$D2FF	Shadows of Pokey1, then Pokey2, then Pokey1 etc

Stereo with covox

$D200-$D20F	Pokey1
$D210-$D21F	Pokey2/Configuration
$D220-$D27F	Shadows of Pokey1, then Pokey2, then Pokey1 etc
$D280-$D29F	COVOX
$D2A0-$D2FF	Shadows of Covox

Quad

$D200-$D20F	Pokey1
$D210-$D21F	Pokey2/Configuration
$D220-$D22F	Pokey3
$D230-$D23F	Pokey4
$D240-$D2FF	Shadows of Pokey1, then Pokey2, then Pokey3, then Pokey4 etc

Quad with covox

$D200-$D20F	Pokey1
$D210-$D21F	Pokey2/Configuration

$D220-$D22F	Pokey3
$D230-$D23F	Pokey4
$D240-$D27F	Shadows of Pokey1, then Pokey2, then Pokey3, then Pokey4 etc
$D280-$D29F	COVOX
$D2A0-$D2FF	Shadows of Covox

Quad with sid

$D200-$D20F	Pokey1
$D210-$D21F	Pokey2/Configuration
$D220-$D22F	Pokey3
$D230-$D23F	Pokey4
$D240-$D25F	SID1
$D260-$D27F	SID2
$D280-$D2FF	Shadow of all above

Pokey registers

Please see wikipedia , Altirra hardware reference manual or Mapping the Atari etc

https://en.wikipedia.org/wiki/POKEY

http://www.virtualdub.org/downloads/Altirra%20Hardware%20Reference%20Manual.pdf

https://www.atariarchives.org/mapping/

IRQ is by default connected only on pokey1, though there is a register that allows all pokeys to be connected.

Pots/keyboard are only connected to pokey1.

Pokey 1 and pokey 3 go to the left channel (and internal channel).

Pokey 2 and pokey 4 go to the right channel.

SID registers

Please see the C64 wiki

https://www.c64-wiki.com/wiki/SID

Sid 1 goes to the left channel
Sid 2 goes to the right channel
Sid is clocked at about 1MHz.

PSG registers

Please see the YM2149 data sheet or other PSG implementation. Note that the registers here are directly memory mapped.

i.e. you can simply read/write $D2A0 for R0 and $D2AF for RF

The PSG is clocked at 1MHz, 2MHz or 1.7MHz.

The number of steps in the envelope can be switched between 16 and 32.

Also the stereo modes can be adjusted.

Mono: All channels to left and right

Polish (default): A+B to left, B+C to right. The same for both chips.

Czech mode: A+C to left, B+C to right. The same for both chips.

Max mode: PSG 1 to left, PSG 2 to right

COVOX registers

$D280 – VOLONLYCH1 – R/W

128	64	32	16	8	4	2	1
V	V	V	V	V	V	V	V

Default: 0

V

Unsigned(0-255) – Current channel volume. For cpu manual sample playing. Left channel.

$D281 – VOLONLYCH2- R/W

128	64	32	16	8	4	2	1
V	V	V	V	V	V	V	V

Default: 0

V

Unsigned(0-255) – Current channel volume. For cpu manual sample playing. Right channel.

$D282 – VOLONLYCH3 – R/W

128	64	32	16	8	4	2	1
V	V	V	V	V	V	V	V

Default: 0

V

Unsigned(0-255) – Current channel volume. For cpu manual sample playing. Right channel.

$D283 – VOLONLYCH4- R/W

128	64	32	16	8	4	2	1
V	V	V	V	V	V	V	V

Default: 0

V

Unsigned(0-255) – Current channel volume. For cpu manual sample playing. Left channel.

SAMPLE registers

The sample player is an extended version of covox. It plays via DMA from 42KiB of internal ram. The 6502 must manually load this internal ram, after which samples may be repeatedly played back from it.

There are 3 data formats:

1. 4 bit signed
2. 8 bit signed
3. 4 bit ima adpcm, giving ~13bit quality. This may be created using for example sox under linux:
    - sox -t wav some.wav -t ima -e ima-adpcm some.ima trim 0s 16384s

Sample rates up to 48KHz are supported (must higher is possible, but memory space is limited).

Samples are played by specifying a memory address, period, length and volume. The address and length are buffered, the period and volume are applied immediately.

When samples come to the end an irq is raised, which allows the next sample to be loaded into address and length. If a new sample has not been loaded the previous one is played.

The core clock speed is 2*phi2. So ~3.6MHz and varies with PAL/NTSC.

$D280-D283: See covox. Note that these are also updated by DMA from block ram, if enabled.

$D284 – RAMADDRL– R/W

128	64	32	16	8	4	2	1
L7	L6	L5	L4	L3	L2	L1	L0

Default: 0

$D285 – RAMADDRH– R/W

128	64	32	16	8	4	2	1
L15	L14	L13	L12	L11	L10	L9	L8

Default: 0

L0-15

Address in block ram, to read or write data

$D286 – RAMDATA– R/W

128	64	32	16	8	4	2	1
D	D	D	D	D	D	D	D

D

Read/write data from/to the internal memory, at the address specified by RAMADDR

$D287 – RAMDATAINC – W

128	64	32	16	8	4	2	1
D	D	D	D	D	D	D	D

D

Write data to the internal memory, at the address specified by RAMADDR. After each write the RAMADDR is incremented by 1.

$D288 – CHANSEL – W

128	64	32	16	8	4	2	1
X	X	X	X	X	C	C	C

Default:0

C

Select the channel to set the address/period/length/volume of

1: channel 1

2: channel 2

3: channel 3

4: channel 4

Other: reserved.

$D289 – SAMADDRL – W

128	64	32	16	8	4	2	1
L	L	L	L	L	L	L	L

$D28A – SAMADDRG – W

128	64	32	16	8	4	2	1
H	H	H	H	H	H	H	H

Default:0

L/H

Set the address of the sample for the channel selected in CHANSEL. This register is applied at the end of the current sample, or when dma state is changed.

$D28B – SAMLENL – W

128	64	32	16	8	4	2	1
L	L	L	L	L	L	L	L

$D28C – SAMLENG – W

128	64	32	16	8	4	2	1
H	H	H	H	H	H	H	H

Default:0

L/H

Set the length of the sample for the channel selected in CHANSEL. This register is applied at the end of the current sample, or when dma state is changed.

This length is specified in number of samples.

Length = L+H*256 +1

$D28D – SAMPERL – W

128	64	32	16	8	4	2	1
L	L	L	L	L	L	L	L

$D28E – SAMPERG – W

128	64	32	16	8	4	2	1
H	H	H	H	H	H	H	H

Default:0

L/H

Set the period of the sample for the channel selected in CHANSEL. This register is applied at the end of the current period.

The next sample is played after this period expires. 2*PHI2/(L+H*256).

i.e. this sets the rate of sample playback

$D28F – SAMVOL– W

128	64	32	16	8	4	2	1
X	X	V	V	V	V	V	V

Default:0

V

Set the volume from 0 (min) to 63 (max) for the channel selected in CHANSEL. The volume is linear.

$D290 – SAMDMA– W

128	64	32	16	8	4	2	1
X	X	X	X	D	D	D	D

Default:0

D

Enable DMA on a per channel basis.

XXX0 (0) = Channel 1 dma off

XXX1 (1) = Channel 1 dma on

XX1X (2) = Channel 2 dma on

X1XX (4) = Channel 3 dma on

1XXX (8) = Channel 4 dma on

$D291 – SAMIRQEN– R/W

128	64	32	16	8	4	2	1
X	X	X	X	I	I	I	I

Default:0

I

Enable IRQ on a per channel basis.

XXX0 (0) = Channel 1 irq off

XXX1 (1) = Channel 1 irq on

XX1X (2) = Channel 2 irq on

X1XX (4) = Channel 3 irq on

1XXX (8) = Channel 4 irq on

$D292 – SAMIRQACT – R/W

128	64	32	16	8	4	2	1
X	X	X	X	I	I	I	I

I

Read if IRQ is active per channel

Write 0 to clear IRQ on a per channel basis.

XXX1 (1) = Channel 1 irq active

XX1X (2) = Channel 2 irq active

X1XX (4) = Channel 3 irq active

1XXX (8) = Channel 4 irq active

$D293 – SAMCFG – R/W

128	64	32	16	8	4	2	1
B	B	B	B	A	A	A	A

Default: 0xf0

A

Enable IMA-ADPCM mode (16-bit sample encoded into 4-bits, approx 12-bit quality)

XXX1 (1) = Channel 1 adpcm mode

XX1X (2) = Channel 2 adpcm mode

X1XX (4) = Channel 3 adpcm mode

1XXX (8) = Channel 4 adpcm mode

B

The channels can be 8-bit (1) or 4-bit (0) quality.

XXX1 (16) = Channel 1 8-bit (0 for 4-bit)

XX1X (32) = Channel 2 8-bit

X1XX (64) = Channel 3 8-bit

1XXX (128) = Channel 4 8-bit

Configuration registers

Configuration registers may be accessed by bank switching it into the $D210-$D21F range. This is done by writing a special value of 0x3c to $D20c.

Reserved locations (X) must be written as 0. Reserved locations will read as 0.

The values to set are shown as X(Y) where X is the bit values (must be shifted into place) and Y is the decimal value.

$D210 –MODE – R/W

128	64	32	16	8	4	2	1
X	X	X	MonoDet	IRQEna	ChannelMode	X	Saturate

Miscellaneous settings

Default: 17

[Pokey] ChannelMode

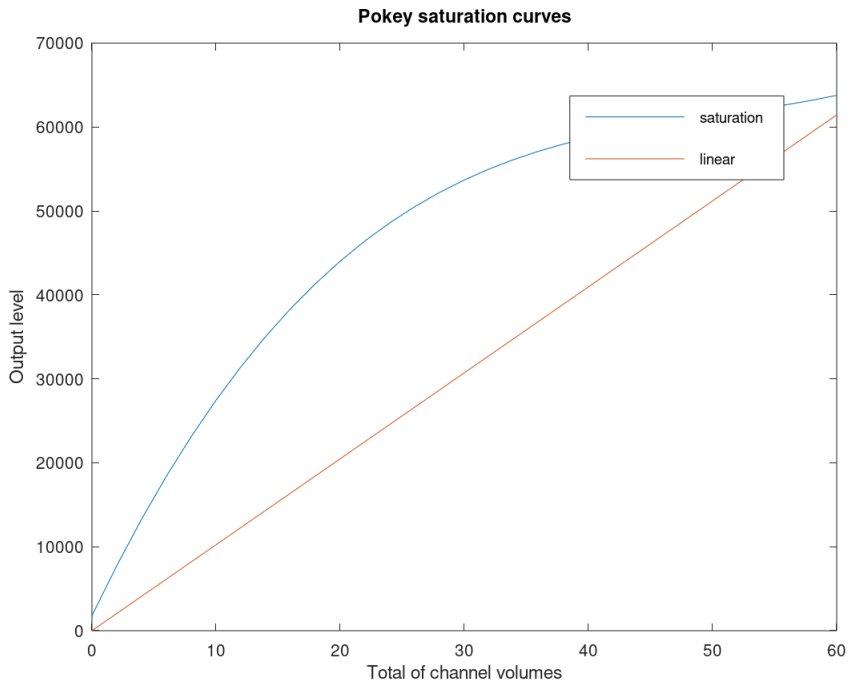
0 (0) = Each pokey outputs as a separate ‘chip’

1 (4) = Output each pokey channel separately. E.g. channel 1 on AUD, channel 2 on audio pin 1, channel 3 on audio pin 2 and channel 4 on audio pin 3. In the case of multiple pokey chips, the channels are summed.

[Pokey] Saturate

0 (0) = Linear saturation curve

1(1) = Pokey saturation curve



In future firmware revisions we anticipate allowing custom saturation curves.

IRQEna

0 (0) = Only pokey 1 irq is enabled

1(8) = All IRQs are enabled

MonoDet

0 (0) = Left channel to left, right channel to right

1(16) = When right channel is silent, play left on both left and right.

Also there will be some settings for other chips, e.g. SID 6581/8580 filter option *maybe*.

$D211 – CAPABILITY - RO

128	64	32	16	8	4	2	1
X	FLASH	SAMPLE	COVOX	PSG	SID	Pokey	Pokey

Default: device specific

Pokey

00 (0) = Device contains a single pokey

01 (1) = Device contains two pokeys

10 (2) = Device contains four pokeys

SID

0(0) = Device does not contain a SID

1(4) = Device contains two SID chips

PSG

0(0) = Device does not contain a PSG

1(8) = Device contains two PSG chips

COVOX

0(0) = Device does not contain a COVOX

1(16) = Device contains two 8-bit COVOX manual volume registers

SAMPLE

0(0) = Device does not contain a sample player

1(32) = Device contains a block ram based sample player, with 32KB of memory

FLASH
0(0) = Device does not support reading or writing flash contents

1(64) = Device supports flash memory access (UFM and CFM)

$D212 – POSTDIVIDE – RW

128	64	32	16	8	4	2	1
CH3	CH3	CH2	CH2	CH1	CH1	CH0	CH0

Default: 160

PokeyMAX natively outputs 0-5V on each audio output. A single audio chip at max volume will be 5V and at min volume 0V.

For internal output into the Atari pin 37 this is correct.

For line level output we want closer to 1V, this register allows us to achieve this by dividing selected channels by 2, 4 or 8, giving 0-2.5V, 0-1.25V or 0-0.0675V.

NB: When multiple devices are outputting at once, we could have a range of 0-10V, 0-20V or even 0-50V! Note that the pokeymax can natively only output 0-5V so anything above 5V will saturate. By using the /4 you will be able to have 4 devices at maximum volume with no distortion.

CH0
00(0) = /1

01(1) = /2

10(2) = /4

11(3) = /8

CH1
00(0) = /1

01(4) = /2

10(8) = /4

11(12) = /8

CH2
00(0) = /1

01(16) = /2

10(32) = /4

11(48) = /8

CH3
00(0) = /1

01(64) = /2

10(128) = /4

11(192) = /8

$D213 – GTIAEN - RW

128	64	32	16	8	4	2	1
X	X	X	X	CH3	CH2	CH1	CH0

Default: 12

This register allows mixing gtia into the output channels.

Typically this is done for the channels actual to phono connectors and not for channels directly outputting into the system, since the motherboard already mixes gtia.

CH0
0(0)=gtia not included in channel 0

0(1)=gtia included in channel 0

CH1
0(0)=gtia not included in channel 1

1(2)=gtia included in channel 1

CH2
0(0)=gtia not included in channel 2

1(4)=gtia included in channel 2

CH3
0(0)=gtia not included in channel 3

1(8)=gtia included in channel 3

$D214 – VERSION – R/ VERSIONLOC –W

W

128	64	32	16	8	4	2	1
X	X	X	X	X	LOC	LOC	LOC

Default: 0

LOC
Write a value from 0-7 for the byte to read from the version string

R

128	64	32	16	8	4	2	1
CHAR	CHAR	CHAR	CHAR	CHAR	CHAR	CHAR	CHAR

Default: NA (depends on version!)

CHAR
Read a single character from the version string.

e.g. write 0, then read character 0, write 1 then read character 1 etc.

$D215 – PSGMODE - RW

128	64	32	16	8	4	2	1
X	VOLP	VOLP	ENV16	STEREO	STEREO	FREQ	FREQ

Default: 4

This register allows changing settings for the PSG chips

FREQ
00(0)=2MHz

01(1)=1MHz

10(2)=1.7MHz

11(3)=reserved

STEREO
00(0)=mono. All channels of both chips to left and right

01(4)=Polish standard. A+B to left, B+C to right. For both chips.

10(8)=Czech standard. A+C to left, B+C to right. For both chips

11(12)=chip 1 to left, chip 2 to right.

ENV16
0(0)=32 step envelope

1(16)=16 step envelope

VOLP
00(0)=Log volume

11(96)=Linear volume

Space reserved for YM/AY3 curves.

$D216 – SIDMODE - RW

128	64	32	16	8	4	2	1
X	X	X	T2	X	X	X	T1

Default: 4

This register allows changing settings for the SID chips. For now this only changes the filter curve linear to approximate the very non-linear 6581 curve. Mixed waveform corruption and filter distortion is planned to be added in a future core, subject to space.

T1

0(0)=8580 (linear filter)

1(1)=6581

T2

0(0)=8580 (linear filter)

1(16)=6581

$D217-$D21A RESERVED

Do not read or write

$D20C/$D21C – ID –R

$D20C/$D21C - CONFIG - W

W

128	64	32	16	8	4	2	1
CFGEN	CFGEN	CFGEN	CFGEN	CFGEN	CFGEN	CFGEN	CFGEN

CFGEN

00111111(63) –Map configuration into the $D210-$D21F memory area

Others values - $D210-$D21F is unchanged

R

128	64	32	16	8	4	2	1
ID	ID	ID	ID	ID	ID	ID	ID

Default: 1

ID

Returns 1 if pokeymax installed

FLASH

The following registers are only present if there is flash support. These are for Retronics use for updates. In 10M02 devices these are not supported. In 10M04 and 10M08 devices they are supported. Cores may be flashed or register defaults changed using the pokeycfg.xex program.

The flash is used for storing the FPGA core and register default contents. In future it will also be used for SID 'corrupt' mixed waveforms, volume curve data, SID filter curves and possibly some standard samples for the sample player.

The flash has a 32-bit parallel interface. See the Max10 flash guide for further details.

$D21B – FLASHOP –RW

128	64	32	16	8	4	2	1
X	X	X	A15	A14	CFG	REQ	R/W_N

Default:0

R/W_N

0(0)=write to flash data or cfg (must erase first)

1(1)=read from flash data or cfg

REQ

0(0)=Do not make a 32-bit wide request to the flash controller.

1(2)=Make a new 32-bit wide request to the flash controller. Once it is complete, REQ is cleared. For reads you can expect data to be available in the 32-bit FLASHDAT register by the next cycle. Writes/erases may takes longer. Writes takes data from the 32-bit FLASHDAT register.

CFG

0(0)=Access data area (i.e. main flash data).

1(4)=Access cfg area. This consists of two special registers in the max 10 for status, write protect and page/sector erase.

$D21D – FLASHADL -RW

128	64	32	16	8	4	2	1
A5	A4	A3	A2	A1	A0	WIN	WIN

Default:0

Read/write to bits 7 downto 0 of the flash address register.

WIN: Window into FLASHDAT (when combined with A15-A0 32-bit address effectively makes a virtual A17-A0 8-bit address)

A15-A0: Flash memory address

$D21E – FLASHADH - RW

128	64	32	16	8	4	2	1
A13	A12	A11	A10	A9	A8	A7	A6

Default:0

Read/write to bits 15 downto 8 of the flash address register.

$D21F – FLASHDAT -RW

Default:0

Read/write from/to 8-bit window into 32-bit configuration register.

The address in the register is specified by the 2 low bits of FLASHADL.

Contents are changed either by the CPU or by a flash read operation.

On a flash write operation the contents of this register is written.