Ultimate Audio

Register API

Gideon Zweijtzer

Version 0.2, October 26th 2012

Table of Contents

1. Introduction 3

1.1. Context 3

1.2. Features 3

1.3. Purpose of this document 3

2. Registers 4

2.1. Overview 4

2.2. Register Read 4

2.3. Register Write 4

2.4. The Audio Channel 4

2.4.1. Control register 5

2.4.2. Volume register 5

2.4.3. Pan register 5

2.4.4. Sample start address register 6

2.4.5. Sample length register 6

2.4.6. Sample rate register 6

2.4.7. Sample repeat point registers 7

2.4.8. Interrupt clear register 7

1. Introduction

1.1. Context

The 'Ultimate Audio' feature is a new feature for the 1541 Ultimate-II module. It implements multi-channel audio sample playback support from REU memory, and is an exciting addition to the possibilities this cartridge offers.

1.2. Features

'Ultimate Audio' offers you the following features:

- 7 independent channels
- 8 or 16 bit audio sample playback
- sample rate up to 48 kHz
- volume control per channel
- panning control per channel
- "a-b" looping
- interrupt generation

1.3. Purpose of this document

The 'Ultimate Audio' feature is accessible from the cartridge I/O range. In this manual, the programming API is described.

2. Registers

2.1. Overview

The functional block is mapped into C64 I/O space, at the address $DF20 up to $DFFF. Mapping this block into the I/O space is optional, and needs to be turned on in the 'C64 and cartridge settings' menu. Obviously, the I/O space is only mapped, when the selected cartridge emulation allows this, just like the REU registers. Note that the REU registers reside at $DF00-$DF1F, when enabled.

Note, that for the 'Ultimate Audio' feature to generate output on the stereo-jack, the audio output selection should be set correctly as well. These settings can be found in the 'Audio settings' menu. Left Channel Output should be set to "Sampler Left", and Right Channel Output accordingly to "Sampler Right".

2.2. Register Read

Functional registers are *write-only*, just like the SID. Reading the I/O space will yield the interrupt status register or module version (as of writing: V1.0).

Address	Read Data	Default
$DF20	Interrupt Status register	$00
$DF21	'Ultimate Audio' version register	$10

2.3. Register Write

Each of the 7 audio channels occupies 32 consecutive bytes of I/O space:

Channel	Channel Base Address
0	$DF20
1	$DF40
2	$DF60
3	$DF80
4	$DFA0
5	$DFC0
6	$DFE0

2.4. The Audio Channel

The (write only) registers of the audio channel are mapped as follows:

Offset	Register	Default
$00	Control	$00
$01	Volume	$3F
$02	Pan	$07
$04–$07	Sample Start Address	$01.00.00.00
$09-$0B	Sample Length	$01.00.00
$0E–$0F	Sample Rate	283 ($01.1B)
$11–$13	Repeat point A	$00.80.00
$15–$17	Repeat point B	$00.C0.00
$1F	Interrupt clear	$00

The multi-byte fields are in *big endian* format. This means that the high byte comes first, and the low(er) bytes follow. This is unlike 16-bit pointers found in the 6502; as they are *little endian*. Since the interface is 8-bit only, there will be no issue. Each byte gets loaded separately.

2.4.1. Control register

The control register contains the following bits:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Interleave	Mode		–	Interrupt	Repeat	Gate

The gate and repeat bit determine how the sample is played. In simple mode (repeat = 0), the sample starts playing from the beginning when the user sets the 'gate' bit to 1. The sample will then play until the end, or until the user clears the gate bit. Thus, when repeat is off, the sample stops playing immediately when the user sets 'gate' to 0.

In repeat mode (repeat = 1), the sample starts to play from the beginning when the user sets the 'gate' bit. When the sample reaches repeat point B, and the gate bit is still set, it reverts to point A. When the user clears the 'gate' bit, repeat point B is ignored and the sample plays until the end. This behavior is very much like the ADSR behavior of the SID chip, where the release phase starts when 'gate' gets cleared.

The interrupt bit determines whether an IRQ is generated when the end of sample is reached. Each channel has its own interrupt status bit in the interrupt status register.

The mode bits determine the format of the sample data in memory. From the four possible modes, two are currently defined:

Mode	Description
00	8-bit PCM
01	16-bit PCM (little endian)
10	*reserved*
11	*reserved*

The interleave bit can be used to skip 'odd' samples. This is useful when a stereo sample is stored in REU memory. When this bit is set, the channel skips the data that is meant for the other channel. A second channel can be programmed such that it exactly reads the samples in between.

2.4.2. Volume register

The volume register controls the output volume of each channel. The register takes values from 0 to 63. Low volumes might generate some noise. This issue still has to be investigated further.

2.4.3. Pan register

The pan register controls where the channel is audible in the stereo image. A value of 7 or 8 put the channel right in the middle. A value of 0 makes the channel appear completely on the left speaker, while a value of 15 makes it sound on the right side.

2.4.4. Sample start address register

This 4-byte register holds the start address of the sample. The upper address byte should always be $01 for the sampler to read data from REU memory. This register in fact holds the address of the SDRAM memory of the Ultimate-II and the base address of the REU memory is $1 00 00 00. Note that this register is big endian; the most significant byte comes first.

2.4.5. Sample length register

The length of the sample (in bytes) is written into this register. The maximum length of a sample can be 16M, the complete REU memory. Note that the length comparison takes place between each sample read from memory. This means that when you play 16 bit samples, the length should be a multiple of 2, otherwise the end condition is never met and the sample will play forever, including a lot of garbage.

2.4.6. Sample rate register

The sample rate register is actually a divider register. The sample rate is derived from a 6.25 MHz reference. The following table shows some common sample rates and corresponding divider settings:

Sample rate (Hz)	Divider
8000	781
11025	567
16000	391
22050	281
32000	195
44100	142
48000	130

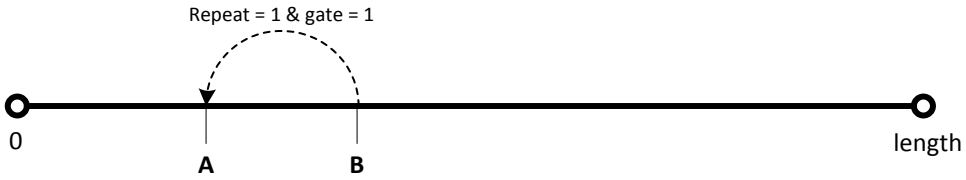
The sample rate register can be safely updated while the sample is playing in order to create all kinds of effects, like modulation.

When an instrument is played, the sample register is used to determine the note. The following example table shows the values to be used when the middle ‘A’ (440 Hz) of an instrument is sampled at 32kHz for various notes, and the error.

Note	Divider	Frequency	Error
A2	391	219.8	-0.1%
A#2	369	232.9	-0.1%
B2	348	246.9	0.0%
C3	328	262.0	0.1%
C#3	310	277.2	0.0%
D3	293	293.3	-0.1%
D#3	276	311.4	0.1%
E3	261	329.3	-0.1%
F3	246	349.3	0.0%
F#3	232	370.4	0.1%
G3	219	392.4	0.1%
G#3	207	415.2	0.0%
A3	195	440.7	0.2%
A#3	184	467.1	0.2%
B3	174	493.9	0.0%
C4	164	524.0	0.1%
C#4	155	554.4	0.0%
D4	146	588.6	0.2%
D#4	138	622.7	0.1%
E4	130	661.1	0.3%
F4	123	698.7	0.0%
F#4	116	740.8	0.1%
G4	110	781.3	-0.3%
G#4	103	834.3	0.4%
A4	98	876.9	-0.4%

2.4.7. Sample repeat point registers

The repeat point registers hold the positions in the sample that are used in repeat mode. Point A is the sample point to which the sequencer returns when reaching point B, as long as both the 'repeat' bit and the 'gate' bit are set in the control register.



2.4.8. Interrupt clear register

The interrupt clear register is used to clear the IRQ condition of the channel. The IRQ can be set when the sample reaches its end. Writing a '1' to bit 0 of this register clears the IRQ of this channel. Writing $FF to this register clears the IRQ of all channels.