PSG16

Table of Contents

[Overview 3](#_Toc325928229)

[Features 3](#_Toc325928230)

[Clocks 3](#_Toc325928231)

[Registers: 4](#_Toc325928232)

[Frequency Register 5](#_Toc325928233)

[Pulse Width Register 5](#_Toc325928234)

[Control Register 5](#_Toc325928235)

[ADSR Register 6](#_Toc325928236)

[I/O Ports 9](#_Toc325928237)

[Operation: 9](#_Toc325928238)

[WISHBONE Compatibility Datasheet 11](#_Toc325928239)

# Overview

PSG16 is an audio interface circuit for use within a programmable system to interface the system to an audio output. It supports four ADSR audio channels with a wavetable option.

# Features

* four ADSR / wave table channels
* programmable frequency and pulse width control
* 0.06 Hz frequency resolution
* attack, decay, sustain and release
* test, ringmod, sync and gate controls
* five voice types: triangle, square, pulse, noise and wave
* exponential decay and release

# Clocks

The PSG16 core uses a single clock for all timing which is the system bus clock. The system bus clock is divided by a pre-scaler to generate an approximately 1MHz clock rate that the audio generator works from. The system bus clock must be at least 8 MHz in order for the core to work properly. A clock prescaling parameter (pClkDivide) must be set for a 1 MHz clock time. If the system bus clock is 25MHz, then the pClkDivide parameter would be set to 25, resulting in a 1MHz working clock rate.

# Registers:

The PSG uses a 64 bit address decode, the register array is internally decoded to appear at $FFFF\_FFFF\_FFD5\_xxxx in the system’s memory. Note that the upper 32 bits of the address are all ‘1’’s to make it simpler to use in a 32-bit address system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| reg | Bits | R/W | Brief |  |
| 0 | nnnnnnnn nnnnnnnn | W | channel 0 frequency |  |
| 1 | ---- nnnn nnnnnnnn | W | channel 0 pulse width |  |
| 2 | trsg – fo- vvvvv -- | W | channel 0 control |  |
| 3 | ssss rrrr aaaa dddd | W | **s**ustain, **r**elease, **a**ttack, and **d**ecay |  |
| 4 | nnnnnnnn nnnnnnnn | W | channel 1 frequency |  |
| 5 | ---- nnnn nnnnnnnn | W | channel 1 pulse width |  |
| 6 | trsg – fo- vvvvv -- | W | channel 1 control |  |
| 7 | ssss rrrr aaaa dddd | W | **s**ustain, **r**elease, **a**ttack, and **d**ecay |  |
| 8 | nnnnnnnn nnnnnnnn | W | channel 2 frequency |  |
| 9 | ---- nnnn nnnnnnnn | W | channel 2 pulse width |  |
| 10 | trsg – fo- vvvvv -- | W | channel 2 control |  |
| 11 | ssss rrrr aaaa dddd | W | **s**ustain, **r**elease, **a**ttack, and **d**ecay |  |
| 12 | nnnnnnnn nnnnnnnn | W | channel 3 frequency |  |
| 13 | ---- nnnn nnnnnnnn | W | channel 3 pulse width |  |
| 14 | trsg – fo- vvvvv -- | W | channel 3 control |  |
| 15 | ssss rrrr aaaa dddd | W | **s**ustain, **r**elease, **a**ttack, and **d**ecay |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 64 | ------------ nnnn | W | master volume |  |
| 65 | nnnnnnn nnnnnnnn | R | osc3 oscillator 3 output |  |
| 66 | ----------- nnnnnnnn | R | env3 envelope 3 output |  |
| 68 | aa------ -------- | W | wave table address bits 15-14 | |
| 69 | aaaaaaaa aaaaaaaaa | W | wave table address bits 31-16 | |
|  |  |  |  |  |
| 80-87 | s---kkkk kkkkkkkk | W | filter coefficients |  |
| 88-96 | reserved for more coefficients | | | |
|  |  |  |  |  |

## Frequency Register

This register sets the tone frequency for the voice. In order to set the frequency specify a value that is a multiple of the base frequency step of 0.06Hz. For example for an 800 Hz tone, 800/0.06 = 13333 would need to be specified

## Pulse Width Register

This register controls the pulse-width when the pulse output waveform is selected. Pulse frequency is controlled by the frequency register.

## Control Register

‘o’ bit enables the output for the voice

‘f’ bit tells the sound generator to route the voice’s output to the filter

‘vvvvv’ sets the output voice type

10000 = triangle wave

01000= square wave

00100 = pulse

00010 = noise

00001 = wave

‘g’ bit ‘gates’ the envelop generator which when set causes it to begin generating the envelope for the voice. When the gate is turned off, the envelope generator enters the release phase.

## ADSR Register

‘a’ - Attack

The attack code controls the attack rate of the sound envelope. The attack slope is triggered when the gate signal is activated. The envelope travels from a zero level to it’s peak during the attack phase.

|  |  |  |
| --- | --- | --- |
| Code | Rate Divider | Attack Time |
| 0 | 8 | 2 ms |
| 1 | 32 | 8 ms |
| 2 | 64 | 16 ms |
| 3 | 96 | 24 ms |
| 4 | 152 | 38 ms |
| 5 | 224 | 56 ms |
| 6 | 272 | 68 ms |
| 7 | 320 | 80 ms |
| 8 | 400 | 100 ms |
| 9 | 955 | 239 ms |
| A | 1998 | 500 ms |
| B | 3196 | 800 ms |
| C | 3995 | 1 s |
| D | 12784 | 3.2 s |
| E | 21174 | 5.3 s |
| F | 31960 | 8 s |

‘d’ = Decay

The decay code controls the decay rate of the sound envelope just after the peak has been reached from the attack phase. The envelop decays from it’s peak value down to the value set by the sustain code.

|  |  |  |
| --- | --- | --- |
| Code | Rate Divider | Decay/Release Time |
| 0 | 24 | 6 ms |
| 1 | 95 | 24 ms |
| 2 | 188 | 48 ms |
| 3 | 282 | 72 ms |
| 4 | 447 | 114 ms |
| 5 | 659 | 168 ms |
| 6 | 800 | 204 ms |
| 7 | 941 | 240 ms |
| 8 | 1176 | 300 ms |
| 9 | 2941 | 750 ms |
| A | 5882 | 1.5 s |
| B | 9412 | 2.4 s |
| C | 11765 | 3.0 s |
| D | 35294 | 9.0 s |
| E | 58824 | 15.0 s |
| F | 94118 | 24.0 s |

‘s’ = Sustain

Sustain sets the signal level at which the signal is ‘sustained’ relative to it’s peak value. There are 16 sustain levels from 0x0 to 0xF with 0x0 being the lowest and 0xF the maximum.

‘r’ = Release

The release code controls the rate at which the signal is ‘released’ after the gate is turned off. When the gate signal is made inactive, the release phase of the ADSR envelope begins. This is an exponential of 2 release.

|  |  |  |
| --- | --- | --- |
| Code | Rate Divider | Decay/Release Time |
| 0 | 24 | 6 ms |
| 1 | 95 | 24 ms |
| 2 | 188 | 48 ms |
| 3 | 282 | 72 ms |
| 4 | 447 | 114 ms |
| 5 | 659 | 168 ms |
| 6 | 800 | 204 ms |
| 7 | 941 | 240 ms |
| 8 | 1176 | 300 ms |
| 9 | 2941 | 750 ms |
| A | 5882 | 1.5 s |
| B | 9412 | 2.4 s |
| C | 11765 | 3.0 s |
| D | 35294 | 9.0 s |
| E | 58824 | 15.0 s |
| F | 94118 | 24.0 s |

# I/O Ports

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Wid | I/O | Description |  |
| rst\_i | 1 | I | This is the active high reset signal |  |
| clk\_i | 1 | I | system bus clock |  |
| s\_cyc\_i | 1 | I | cycle active |  |
| s\_stb\_i | 1 | I | data strobe |  |
| s\_ack\_o | 1 | O | data transfer acknowledge |  |
| s\_we\_i | 1 | I | write cycle |  |
| s\_sel\_i | 2 | I | byte lane selects |  |
| s\_adr\_i | 64 | I | decode / register address |  |
| s\_dat\_i | 16 | I | data input |  |
| s\_dat\_o | 16 | O | data output |  |
|  |  |  |  |  |
| m\_cyc\_o | 1 | O | master cycle valid |  |
| m\_stb\_o | 1 | O | master strobe |  |
| m\_ack\_i | 1 | I | master data transfer acknowledge |  |
| m\_we\_o | 1 | O | master write enable (always inactive) |  |
| m\_sel\_o | 2 | O | master byte lane select (always both selected) |  |
| m\_adr\_o | 64 | O | master address |  |
| m\_dat\_i | 16 | I | master data input |  |
| m\_dat\_o | 16 | O | master data output (always zero) |  |
|  |  |  |  |  |
| o | 18 | O | 18 bit audio output |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Operation:

The PSG uses a harmonic frequency synthesizer with a 24 bit accumulator. This gives the generator a base frequency step of 0.06Hz. (1e6 / 2^24). The upper bits of the accumulator are used as a source for audio waves.

The PSG core can act as a bus master in order to read audio data from a table in memory. The PSG uses the tone generator accumulator to generate addresses from which to read. Bus mastering can be turned on by selecting ‘wave’ as the voice type, in which case the PSG will cycle through a 4k range of addresses for wave data.

# WISHBONE Compatibility Datasheet

The PSG core may be directly interfaced to a WISHBONE compatible bus.

|  |  |  |
| --- | --- | --- |
| WISHBONE Datasheet  WISHBONE SoC Architecture Specification, Revision B.3 | | |
|  |  | |
| Description: | Specifications: | |
| General Description: | PSG16 – programmable ADSR sound generator | |
| Supported Cycles: | SLAVE, READ / WRITE  SLAVE, BLOCK READ / WRITE  SLAVE, RMW | |
| Data port, size:  Data port, granularity:  Data port, maximum operand size:  Data transfer ordering:  Data transfer sequencing | 16 bit  16 bit  16 bit  Little Endian  any (undefined) | |
| Clock frequency constraints: | 8 MHz minimum | |
| Supported signal list and cross reference to equivalent WISHBONE signals | Signal Name:  ack\_o  adr\_i(63:0)  clk\_i  dat\_i(15:0)  dat\_o(15:0)  cyc\_i  stb\_i  we\_i | WISHBONE Equiv.  ACK\_O  ADR\_I()  CLK\_I  DAT\_I()  DAT\_O()  CYC\_I  STB\_I  WE\_I |
| Special Requirements: |  | |