AC97 Controller

# Overview

* The AC97 controller incorporates Digilent’s AC97 controller to communicate with the LM4550 codec device. The AC97 controller maps the LM4550’s register set into the memory map of the system. It accomplishes this using a shadow registers set and by recording which registers are ‘dirty’ and need to be updated in the LM4550. As AC97 frames become available, they are used to update the LM4550, which causes the dirty bit to be reset to clear
* To the system it looks like the LM4550 resides directly in the memory map. Writes and reads to the controller seem to take place instantly. However in the background the controller is managing updates to the LM4550. It’s a good idea to check overall dirty status available from register h68 to be able to tell when all updates to the LM4550 have taken place.
* The controller uses unused register #0x68 to return the update status of the LM4550.

# Clocks

There are two clocks in use – a system bus clock and an AC97 timing clock. The clocks are independent.

# Registers

The AC97 controller has the same register set as the LM4550 device which it is shadowing. Refer to LM4550 device specs for a description of the registers.

There are two exceptions. The status register h26 is read by first writing to it, to force a read cycle in the AC97 controller. The value may then be read from the status register once it has been updated in the controller. This means that low power settings cannot be set in the status register. (The write cycle doesn’t update the LM4550, it triggers a read instead). The second exception is that register h68 is used to reflect the overall dirty status of the AC97 controller. This register is unused in the LM4550.

# 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 |  |
| cyc\_i | 1 | I | cycle active |  |
| stb\_i | 1 | I | data strobe |  |
| ack\_o | 1 | O | data transfer acknowledge |  |
| we\_i | 1 | I | write cycle |  |
| sel\_i | 2 | I | byte lane selects |  |
| adr\_i | 64 | I | decode / register address |  |
| dat\_i | 16 | I | data input |  |
| dat\_o | 16 | O | data output |  |
|  |  |  |  |  |
| PSGout | 18 | I | this input bus is the sound generator output |  |
|  |  |  |  |  |
| BIT\_CLK | 1 | I | provides timing for the AC97 interface |  |
| SYNC | 1 | O | indicates the start of an AC97 frame |  |
| SDATA\_IN | 1 | I | serial data input |  |
| SDATA\_OUT | 1 | O | serial data output |  |
| RESET | 1 | O | active low reset to the AC97 compatible device (LM4550) |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# WISHBONE Compatibility Datasheet

The AC97 core may be directly interfaced to a WISHBONE compatible bus.

|  |  |  |
| --- | --- | --- |
| WISHBONE Datasheet  WISHBONE SoC Architecture Specification, Revision B.3 | | |
|  |  | |
| Description: | Specifications: | |
| General Description: | AC97Controller | |
| 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: |  | |
| 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: |  | |