CS4281
Programming Manual

Revision 1.17w

3/7/00

CIRRUS LOGIC®
P.O. Box 17847, Austin, Texas 78760
(512) 445 7222   FAX: (512) 445 7581
http://www.cirrus.com

Contacting Cirrus Logic Support

For a complete listing of Direct Sales, Distributor, and Sales Representative contacts, visit the Cirrus Logic web site at:

http://www.cirrus.com/corporate/contacts/

Preliminary product information describes products which are in production, but for which full characterization data is not yet available. Advance product information describes products which are in development and subject to development changes. Cirrus Logic, Inc. has made best efforts to ensure that the information contained in this document is accurate and reliable. However, the information is subject to change without notice and is provided “AS IS” without warranty of any kind (express or implied). No responsibility is assumed by Cirrus Logic, Inc. for the use of this information, nor for infringements of patents or other rights of third parties. This document is the property of Cirrus Logic, Inc. and implies no license under patents, copyrights, trademarks, or trade secrets. No part of this publication may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc. Items from any Cirrus Logic website or disk may be printed for use by the user. However, no part of the printout or electronic files may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc.Furthermore, no part of this publication may be used as a basis for manufacture or sale of any items without the prior written consent of Cirrus Logic, Inc. The names of products of Cirrus Logic, Inc. or other vendors and suppliers appearing in this document may be trademarks or service marks of their respective owners which may be registered in some jurisdictions. A list of Cirrus Logic, Inc. trademarks and service marks can be found at http://www.cirrus.com.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. DEFINITION OF TERMS

Reserved - Reserved bits may be used in future revisions of the CS4281 or other chips with the same interface; therefore, special rules must be followed to maintain compatibility. For software, reads should mask off reserved bits as they could read 0 or 1. Zeros should always be written to reserved bits.

Stream - An audio path of related data from the PCI Bus to/from the AC-Link. The stream consists of audio samples containing one or more channels of audio data and consists of all the hardware associated with the data transport pipe. For example, one DMAn/FIFOn pair supports a single stream. Therefore, CS4281 can handle four simultaneous streams of audio data (n = 0 - 3).

Sample - All audio data that is sampled at one instant in time or is time related. Stereo samples have two channels of related audio data, mono samples have one channel, and Surround Sound has more than two channels comprising a sample. A related set of samples over time is a stream of audio data.

Channel - Two contexts:

With respect to audio PCM data, a channel is an atomic entity defining one piece of audio data. Stereo data is comprised of two channels, one left and one right. Mono data has only one channel. FIFO memory is two channels wide, comprised of a left and right channel. Both channels must be accessed to move FIFO pointers. A DMA engine can move one stream of audio data that comprises up to two channels between the PCI bus and the FIFOs. The DMA channels are termed Channel 1 and Channel 2 where Channel 1 is moved before Channel 2. Typically, DMA Channel 1 is equivalent to FIFO Left Channel and DMA Channel 2 is equivalent to FIFO Right Channel, but DMRn.SWAPC can swap the channels making DMA Channel 1 the FIFO Right Channel.

With respect to the FM synthesizer, an FM sound or voice is termed a channel and can be generated with up to four FM operators. In four-operator mode, the FM synthesizer supports six 4-operator channels and 6 2-operator channels.

SSC - Sound System Controller. The SSC controls the movement of data between the FIFO and the AC Link. The DMAn engine does a similar function between the PCI bus and the FIFO. The AC Link engine makes requests to the SSC to move data between the AC Link and the FIFO. If the Playback/Record SRC are enabled, their sample clocks make request to the SSC to move data between the SRC and the FIFO. Lastly, if Host Wavetable mode is enabled, the 48 kHz framing clock is used to make requests to move data from a FIFO to the Playback SRC mixing register.

Host Access or Host Software - Host software is software written specifically for the CS4281 using PCI (BA0) memory accesses.

Host Wavetable - Refers to a host stream setup to use the FM mixing engine in the PSRC. The FM synthesis engine outputs data that is mixed with data going through the PSRC before being sent to the AC Link. This architecture supports Sound Blaster operation. If the FM synthesis engine is disabled, host software can use the same mixing engine (48 kHz fixed Fs) by selecting special slot IDs.

Slot IDs - These IDs are used by the SSC to match FIFOs and SRCs, with AC Link engine requests. Slot ID's number from 0 to 31, with 29/30 used for Host Wavetable, and 31 indicating an unused channel. Slot IDs differ from AC Link slots in that Slot ID's are unique whereas AC Link slots for audio data are numbered per signal pin: 3-11 for **ASDOUT**, 3-11 for **ASDIN**, and 3-11 for **ASDIN2**. For example: Slot ID 0 is AC-Link slot 3 for **ASDOUT**; Slot ID 10 is AC Link slot 3 for **ASDIN**; and SLOT ID 20 is AC-Link slot 3 for **ASDIN2**. See Table 37.

Sound System - The Sound System is comprised of the SSC, the SRCs, and enabled for all peripherals. The Sound System controls all logic in the **ABITCLK** domain.

TC - Two contexts.
Terminal Count. Terminal Count is part of a standard 8327 DMA controller and indicates, through interrupts, when a programmed count rolls under. Either stops DMA operation or causes Current Address and Count registers to reload if in auto initialize mode.

Target Codec. Target Codec is a bit the AC-Link interface section, **ACCTL.TC**. Determines which codec (primary or secondary) and associated address and data registers (AC-Link slots 1 and 2) go out on the **ASDOUT** line.

HTC - Half Terminal Count. Half Terminal Count is new to the CS4281 and indicates when the current count (DCCn) counted half way down. Both **TC** and **HTC** interrupts support standard ping-pong buffer architectures.

DMAn/FIFOn - Four DMAn/FIFOn pairs support four independent streams. The suffix ‘n’ designates one of the four pairs, 0 through 3. The pairs cannot be split up. Therefore, the four pairs are DMA0/FIFO0, DMA1/FIFO1, DMA2/FIFO2, DMA3/FIFO3. The DMAn engine supports transfer of data between the corresponding FIFOn and the PCI bus in both DMA mode and Polled FIFO mode. In Polled FIFO mode, the DMAn format register (DMRn) is used to convert the PC memory audio data format to and from the internal 20-bit signed data format.

Formatter - The formatter converts sits between the FIFO memory and the PCI bus. It converts the host memory data format to and from the CS4281-internal/Codec data format of 20-bit little-endian 2’s complement. The DMRn register sets the format for a particular DMAn/FIFOn combination and is utilized in both DMA and Polled FIFO modes.

DMA/Polled FIFO Modes - Audio data can be transferred between host memory and the FIFO memory in two ways: DMA or Polled FIFO. The mode is set in the DMRn register along with the data format. In the DMA mode, the DMAn bus-mastering engine, once enabled and configured, acquires the bus and transfers audio without host software intervention. The DMAn engine transfers playback data from host memory to FIFOn whenever space is available in the FIFO and transfers record data from FIFOn to host memory whenever FIFOn has a sample (trying to keep the FIFO empty). In Polled FIFO mode, host software is responsible for transferring audio data between host memory and FIFOn via the Polled Data Register FPDRn. Generally host software sets up the FIFOn controller to cause an interrupt when a certain FIFO depth is reached (**FSICn.FIC** bits). When interrupted, host software would then fill or empty FIFOn accordingly.

Base Address - Refer to the PCI configuration space memory assigned to the CS4281. The main Base Address is BA0, which defined a 4 k address space that provides access to all CS4281 internal registers used to configure the part. The other rarely used Base Address is BA1, which holds the FIFO memory and other internal memories that are only accessed in debug or test modes.

2. INTRODUCTION

The CS4281 is a PCI AC ‘97 digital controller that provides an interface between a serial AC ‘97 Codec, such as the CS4297, and the parallel PCI bus. The CS4281 contains four bus-mastering DMA controllers, with four FIFOs, that manage up to four independent streams of audio data (up to 8 channels). The CS4281 contains a pair of stereo sample-rate converters that support all Windows sample frequencies using standard AC ‘97 Codecs which have a fixed sample frequency of 48 kHz. The CS4281 also supports Codecs with built-in SRCs, such as the CS4299. The CS4281 can be configured for two codecs on the AC Link supporting applications such as audio/modem, docking station, or 4 to 8 channel audio.

Other features on the CS4281 include an industry-compatible FM synthesizer for low-overhead, low-PCI bandwidth sound support, a MIDI input and output for external wavetable synthesis support, and Hardware Volume control. The CS4281 also contains extended general purpose I/O (EGPIO) that supports hyper-flexible configurability similar to the EGPIO pin definition in the AC ‘97 Codec Spec., Version 2.1, Appendix B - *Modem AFE Extension*. These GPIO pins have programmable polarity, can be sticky, can generate software interrupts, and can generate PME# events.

For power management, the CS4281 supports the PCI power management states as described in the *PCI Bus Power Management Interface Specification*, version 1.1, including **PME#** power-up during full power down, and **CLKRUN#** support for more dynamic power management. Modem-style “wake up” from the AC Link in $D3_{cold}$ is also supported as described in the *AC ‘97 Audio Codec Specification*, version 2.1.

This document specifies the programming manual for the CS4281. This document also describes the CS4281’s system interfaces, configuration methods, programmable options, operational characteristics, and internal architecture. The CS4281 is a 3.3 V part, and is a drop-in replacement for the CS4614 or CS4280-CM although the software is different. The CS4281 comes in a 100-pin MQFP, which is backwards compatible, and a 100-pin TQFP which is unique to the CS4281.

Since this document preceeds the CS4281, risks exist in using the information contained within. Please see the disclaimer on the 2nd page. Although believed to be accurate, this document will change as the design progresses and through testing of the device.

This document does not illustrate how to write drivers for the CS4281. The assumption is made that those using this manual know how to write audio drivers for their operating systemof choice. This manual only provides a register description of the CS4281 product.

This document is a companion to the *CS4281 Data Sheet* which contains the pinout as well as timing specifications.

Figure 1 depicts the internal architecture of the CS4281. The audio data path is divided into two sections: PCI Control and Sound System. On the PCI Control side, the DMA engines move data between the PCI Bus and the FIFO memory (through the Formatter). On the Sound System side, the Sound System Controller moves data between the FIFO memory and the AC Link Interface. The Peripheral block contains the AC-Link engine, the MIDI and Game Ports, and the hardware volume control logic. The Control block contains the Clocking block, internal test control, and the FM synthesizer. The Vaux Logic block is isolated from the other blocks to support removing power from all other blocks while the Vaux block is active to obtain the maximum power savings.

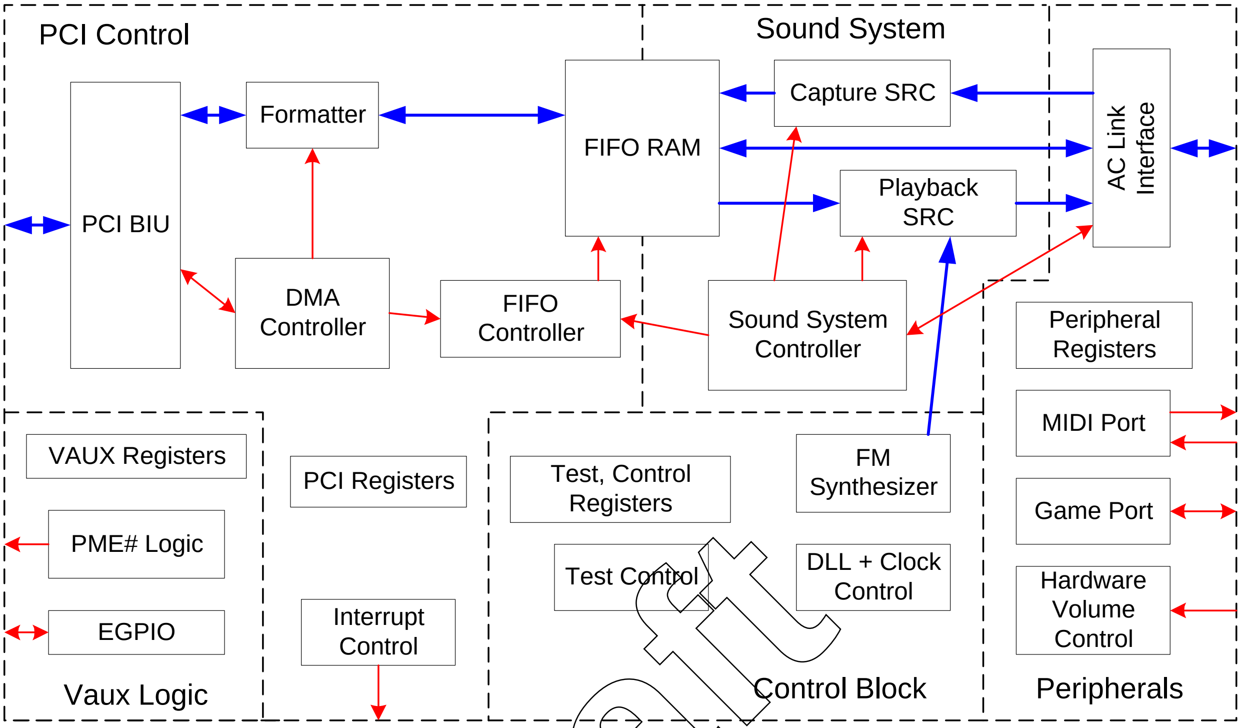


Figure 1. CS4281 Internal Block Diagram

2.1 Typical System Diagrams

The CS4281 is designed to fill a wide variety of end use applications ranging from simple add-in sound card to a complete audio+modem system with docking station support. The following system block diagrams illustrate the device flexibility. All configurations can have a game port and MIDI port if desired. The CS4281 supports one or two Codecs attached. Typical signal hookup diagrams for the AC Link are illustrated in the *AC '97 Configurations* section. The dual Codec connection is designed to support three typical scenarios: a notebook docking station, a Modem Codec for host-based modems, or a secondary audio codec for 4-8 channel support.

Figure 2 illustrates the most typical configuration where the CS4281 is the Controller between the PCI Bus and the serial AC '97 codec. This approach supports two streams of audio data: stereo audio in and stereo audio out. If the CS4297A is replaced by a CS4294 Codec, then up to three streams are supported simultaneously from one AC Codec: stereo audio in, and two stereo audio out (4-channel) streams.

Some modem configurations will use a single Audio/Modem codec, such as the CS4298, as shown in Figure 3. In this scenario, all four streams of the CS4281 are used where two streams support stereo audio in and stereo audio out, and the other two streams support modem in and modem out.

Figure 4 illustrates a portable application where a primary Codec resides in the notebook computer and a secondary codec resides in the docking station. If the primary Codec is a combination Audio/Modem Codec, such as the CS4298, telephony as well as audio functions could be supported. The secondary Codec resides in the docking station and provides audio functionality when the notebook computer is docked. The secondary Codec replaces most of the audio functionality of the primary codec assuming the mic and speaker connections on the docking station should replace the ones on the notebook which become inaccessible.

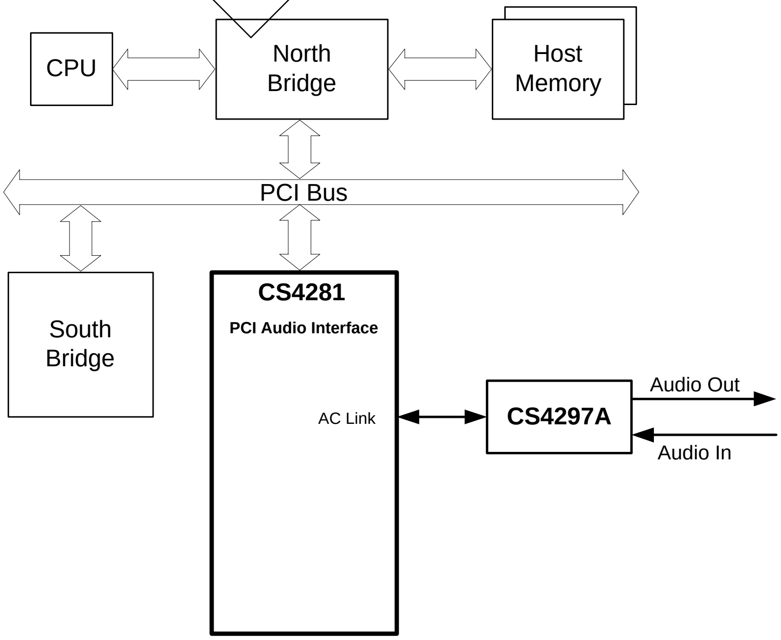


Figure 2. Basic Audio Configuration Diagram

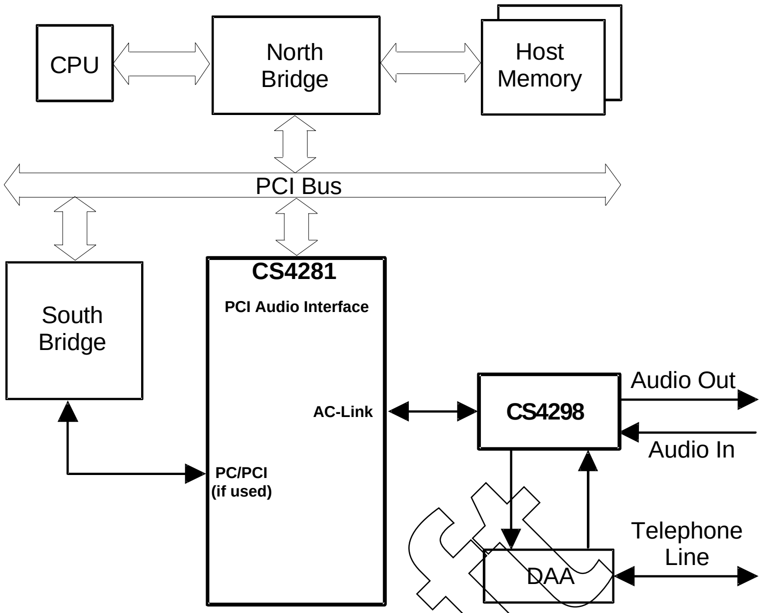


Figure 3. Audio/Modem Codec Host-Modem Block Diagram

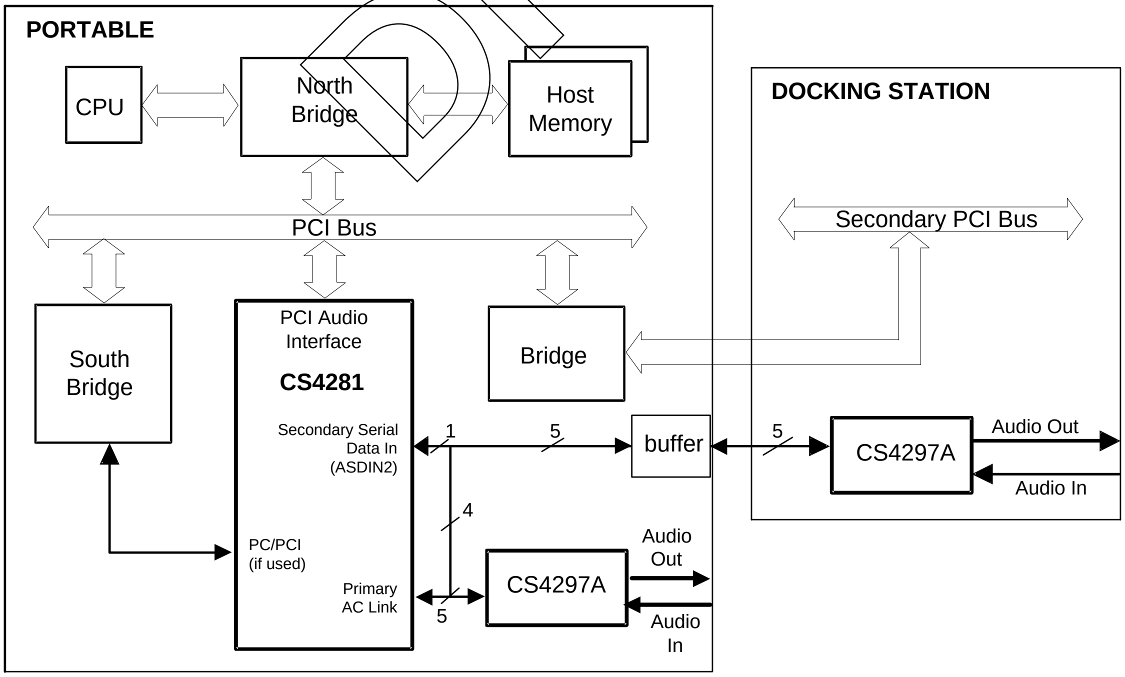


Figure 4. Dual-Codec Docking Station Block Diagram

Figure 5 illustrates an 8-channel application where a primary Codec supports the first four channel and the secondary codec supports the second 8 channels. A stream on the CS4281 supports up to two channels so all four streams are utilized when eight channel out is active. Therefore, since the CS4281 only supports four simultaneous streams, the stereo audio in from either codec could not be used while the eight channel out is active. If only six-channel out is desired, the fourth stream would be available for stereo audio recording.

2.1.1 Surround Sound 5.1 Play

No special modes exist to support 5.1 channel surround sound. Surround sound is supported on the CS4281 in one of two methods. The first is using an S/PDIF renderer to take the stereo compressed 5.1 channel data and, using one stream, send it through the CS4281 to a CS4298/97A/99 that supports an S/PDIF output. The second method is using a filter on the host to decompress the 5.1 streams into 6 channels of PCM data. Then an Audio Surround Sound renderer maps the six channels into three stereo streams for the CS4281. If one of the streams is left/right PCM data, then the fourth CS4281 stream may be used for system sounds or host wavetable support. Since de-compressed surround sound uses three of the four streams supported by the CS4281, simultaneous host modem is not viable. The six-channel out is supported by configuring a CS4297A as the primary Codec and a CS4294 as the secondary codec. The CS4297A supports two-channel in and out while the CS4294 can support the additional 4 channels out.

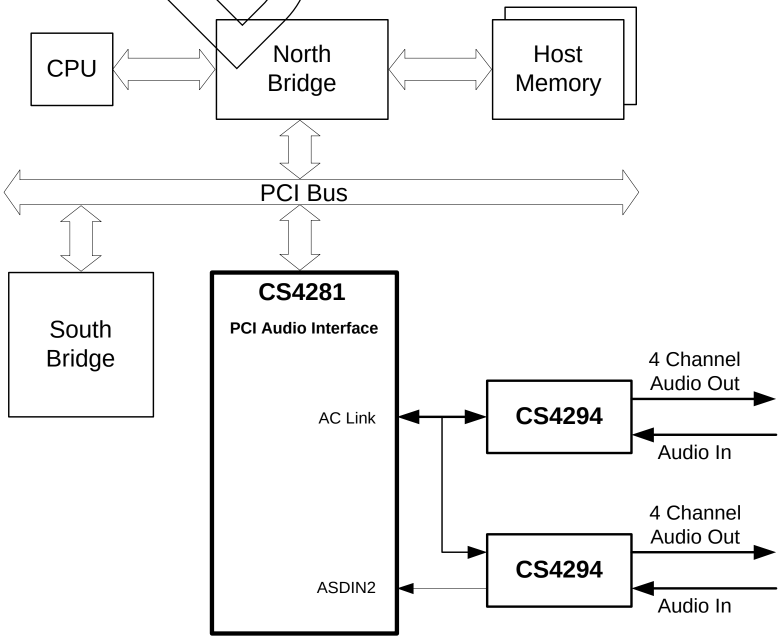


Figure 5. Dual-Codec 8-Channel Audio Out Diagram

2.2 Audio Data Flow Diagrams

Audio data flows between host buffers in PC memory, through the CS4281, and across the AC Link to an AC ‘97 Codec. The audio data path is termed a stream. Inside the CS4281, the audio stream is split into two domains, divided by a FIFO. The two domains are the PCI Bus side and the AC-Link side. The PCI Bus side is controlled by a DMA engine attached to a particular FIFO. The DMA engine is responsible for moving data between the PCI bus and the FIFO. Four DMA/FIFO combinations support four independent streams of audio data. A suffix of ‘n’ is used throughout this document when discussing one DMAn/FIFOn combination, where n is between 0 and 3. On the AC-Link side, the Sound System Controller (SSC) is responsible for moving data between the FIFOs and the AC-Link engine.

The playback data flow is depicted in Figure 6. The audio data goes from the PCI bus, through a formatter, and into FIFOn. The DMAn engine controls the data movement and configures the formatter for the data transfer (from the DMRn register). The formatter converts the PC bus audio data to the internal 20-bit signed format. For playback, FIFOn sends a signal to the DMAn engine indicating FIFOn is not full. The DMAn engine responds by acquiring the PCI bus and moving a sample from host memory to FIFOn. The DMAn engine continues moving samples until FIFOn is full. Anytime FIFOn becomes “not full” the DMAn reacquires the PCI bus and fills FIFOn back up.

On the other side of FIFOn, the Sound System Controller (SSC) is responsible for transferring data from FIFOn to the AC-Link engine which sends the data down the **ASDOUT** pin, in a particular slot, to the Codec DACs. Host software must tell the AC-Link engine which ASDOUT slots need to be transferred (through the ACOSV register). The SSC gets a signal from the AC-Link engine indicating that the AC Link “Needs Slot x”. The SSC then moves data from FIFOn to the AC-Link engine. The SSC knows which FIFOn to connect to Slot x via slot mapping IDs in the FCRn register. If the Playback SRC is configured for the same slot IDs as FIFOn, the audio stream is sent

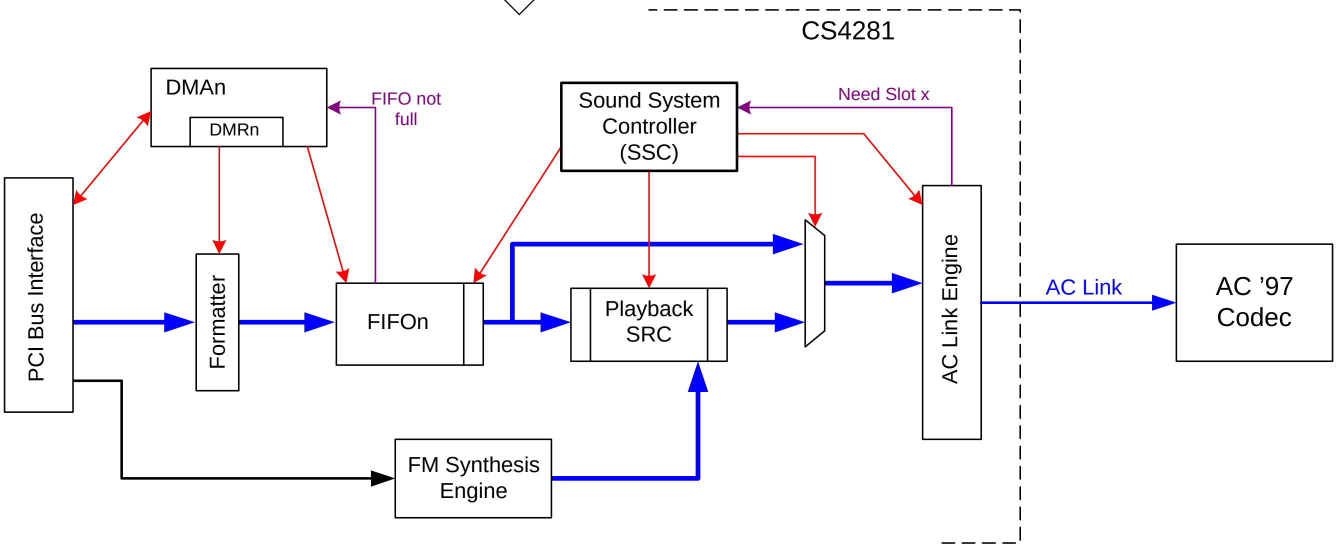


Figure 6. Playback Data Flow Diagram

through the Playback SRC before being sent out the AC Link.

The registers needed to setup a playback sequence are enumerated in the *Start-Up Configuration and Status* section. That section also enumerates global CS4281 chip setup through both the *PCI Start-Up Configuration* and the *Sound-System Start-Up* sections.

If the FM synthesizer is enabled and programmed to generate sounds, the FM digital audio output will be mixed with the data output from the Playback SRC before being sent to the AC-Link engine.

The record data flow is depicted in Figure 7. The audio data comes from the Codec, across the AC Link, where the SSC moves the audio data into FIFOn. The audio data comes from either the **ASDIN** pin for the primary Codec, or from the **ASDIN2** pin if a second Codec is present. The SSC gets a signal from the AC-Link engine indicating that valid data for Slot y is available for transfer. The slot valid bits for the input pins are set by the Codec itself. The SSC knows which FIFOn to connect to Slot y via slot mapping IDs in the FIFOn FCRn register. If the Record SRC is configured for the same slot IDs as FIFOn, the audio stream is sent through the Capture SRC before being loaded into FIFOn.

On the other side of FIFOn, the DMAn engine controls the data movement and configures the formatter for the data transfer. The formatter converts the CS4281 internal 20-bit signed format to what ever format is programmed for PC bus audio data transfers (from the DMRn register). For record, FIFOn sends a signal to the DMAn engine indicating FIFOn is not empty. The DMAn engine responds by acquiring the PCI bus and moving a sample from FIFOn to host memory. The DMAn engine continues moving samples until FIFOn is empty. Anytime FIFOn becomes “not empty” the DMAn reacquires the PCI bus and empties FIFOn. The registers needed to setup a record sequence are enumerated in the *Start-Up Configuration and Status* section.

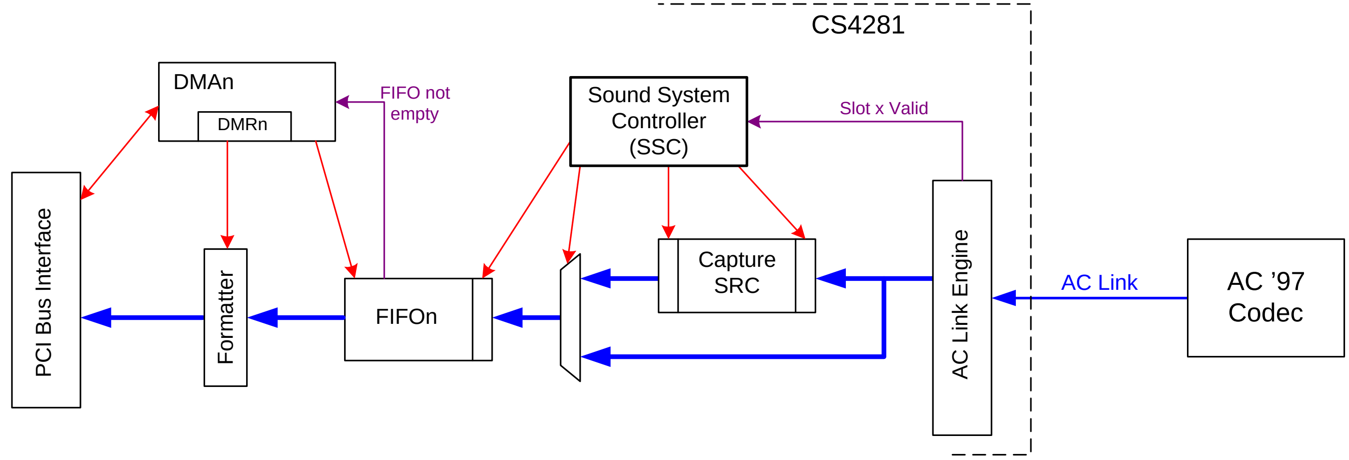


Figure 7. Capture Data Flow Diagram

2.3 AC '97 Configurations

The CS4281 operates as an AC '97 controller capable of driving one CS4297 (an AC '97 1.03-compliant codec), one CS4298 (an AC '97 2.0-compliant codec), or two CS4298s. The CS4281 serial interface is compliant with the *Audio Codec '97 Specification*, version 2.1, and supports the multiple codec extension feature, which defines a method for connecting a second codec to a controller with only one additional input data pin (in theory up to four codecs can be supported by extending this method, but the CS4281 only supports two). The CS4281 also supports one CS4297 codec and one CS4298 codec simultaneously using the method described in the AC '97 2.0 specification.

Figure 8 illustrates the standard connection of one AC '97 codec to the CS4281, where the Codec supplies the master clock (**ABITCLK**) to the CS4281.

Two AC '97 codecs are supported as long as one supports the 2.0 extensions. Figure 9 illustrates two codecs working off one AC Link using two data input pins to the CS4281. This scenario is typically used for modem applications where the second codec is connected to a phone line, docking stations where the primary codec is inside the notebook and the secondary codec is in the dock, or multi-channel support where the second codec is the 3/4 channels or the 5-8 channels when using two CS4294 Quad Codecs.

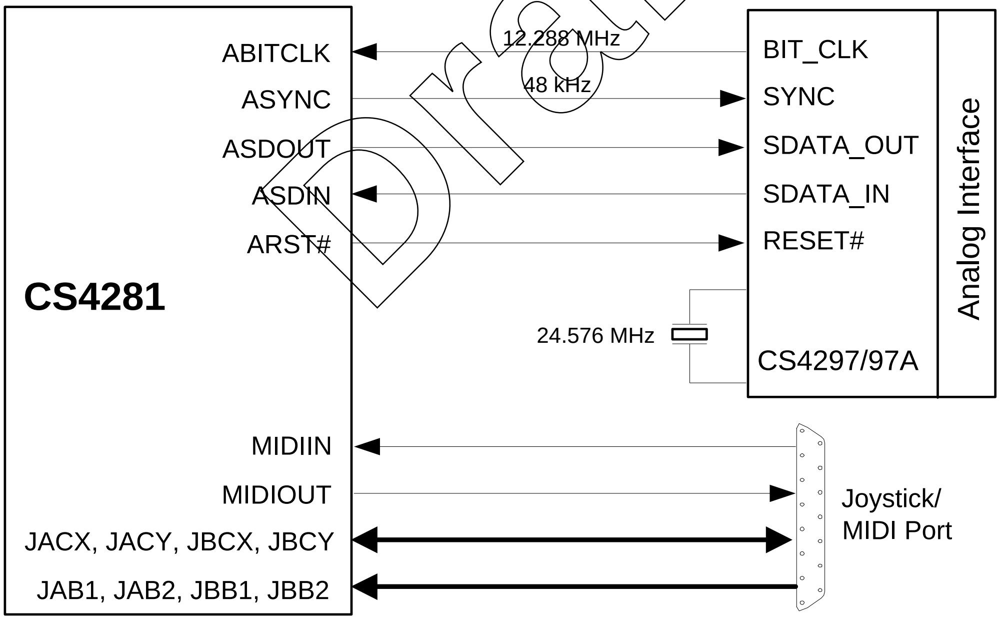


Figure 8. Single Codec Connection Diagram

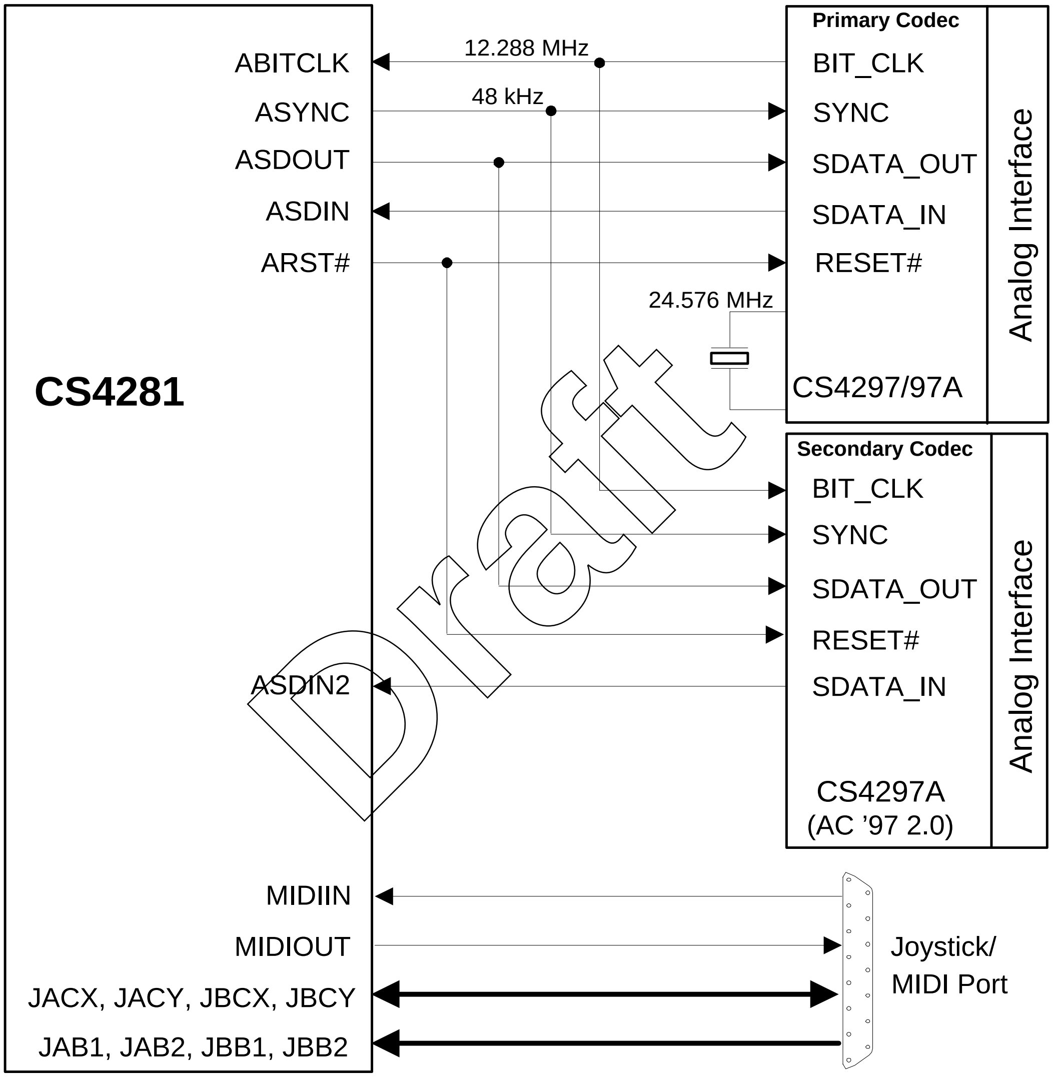


Figure 9. Dual Codec Connection Diagram

3. GENERAL OVERVIEW

As the 4th or 5th generation audio sound system, the CS4281 can take advantage of our learning experiences by simplifying certain things and refining our support of others. This specification will subdivide the chip into many functional units. These units are: PCI interface, Sound Blaster, Sound System Interface, FM Synthesis, Peripheral Devices (MIDI and gameport plus), and Testability Block. Note that the External Peripheral Port and the 8052 processor are gone!

The PCI interface is the physical connection to the PCI bus and provides control and connections to the internal devices. In the mixed signal part, the PCI to local clock sychronization will occur in the PCI Interface. The PCI interface is subdivided into several smaller blocks: PCI configuration and interface, EEPROM interface, DMA controller, interrupt control, and chip control.

The Sound Blaster block is a hardware implementation of the 8052 code that supports all the SB Pro functionality.

The Sound System Interface provides the register access for the codec and mixer controls. This is a simplified version of the combined WSS and Control registers from Brahms/Mahler.

The FM Synthesis is the OPL3 work-a-like from Mahler, with all the bugs fixed. It will sport a new testability section, and use a new gated clock technique to save power.

The Peripheral Devices are the various support stuff. This section includes the AC Link, clock management, MIDI hardware that implements the UART functionality, Game Port.

The Testability Block contains all the normal test functions and the special purpose test features. Here we separate test functions into two categories - those that can be run in the system are the Normal Test Functions, and those that must be in a special test fixture or evaluation board are Special Test Functions.

3.1 Functional Specs

The functional specifications fall into several groups and these are separately enumerated. These specifications are General Specs, Operating Conditions, Absolute Maximum Ratings, I/O Specs, Digital Filter Characteristics, Timing Parameters, and Implied Functional Specs.

3.2 General Specs

Process: A3T - 3 metal, 1 poly. Uses A4M layout with 10 % optical shrink.
Shrunk Die Size: 3.89 x 4.08 mm = 15.87 mm^2
1767 gross die per wafer.
ESD tolerance: HBM - 2 kV minimum, MM - 200 V minimum 6 kV target 400 V Target
Latchup tolerance: ±400 mA at 125° C

The following specifications are data sheet specifications; therefore, the design specs must exceed them by enough margin to be manufacturable over temperature and process variances. The CS4281 must meet both the 5 V and 3 V PCI Bus tables with no change in power supply voltages.

ABSOLUTE MAXIMUM RATINGS

(PCIGND = CGND = CRYGND = 0 V, all voltages with respect to 0 V)

Parameter	Symbol	Min	Typ	Max	Unit
Power Supplies	PCIVDD VAUX CVDD CRYVDD VDD5REF	- - - -	- - - -	4.6 4.6 4.6 4.6 5.5	V V V V V
Total Power Dissipation (Note 1)		-	-	1	W
Input Current per Pin, DC (Except supply pins)		-	-	±10	mA
Output current per pin, DC		-	-	±50	mA
Digital Input voltage (Note 2)		-0.3	-	Vdd+ 0.3	V
Ambient temperature (power applied) (Note 3)		-55	-	125	°C
Storage temperature		-65	-	150	°C

Notes:
1. Includes all power generated by AC and/or DC output loading.
2. The power supply pins are at recommended maximum values.
3. At ambient temperatures above 70° C, total power dissipation must be limited to less than 0.4 Watts.

WARNING: Operation beyond these limits may result in permanent damage to the device. Normal operation is not guaranteed at these extremes.

RECOMMENDED OPERATING CONDITIONS

(PCIGND = CGND = CRYGND = 0 V, all voltages with respect to 0 V)

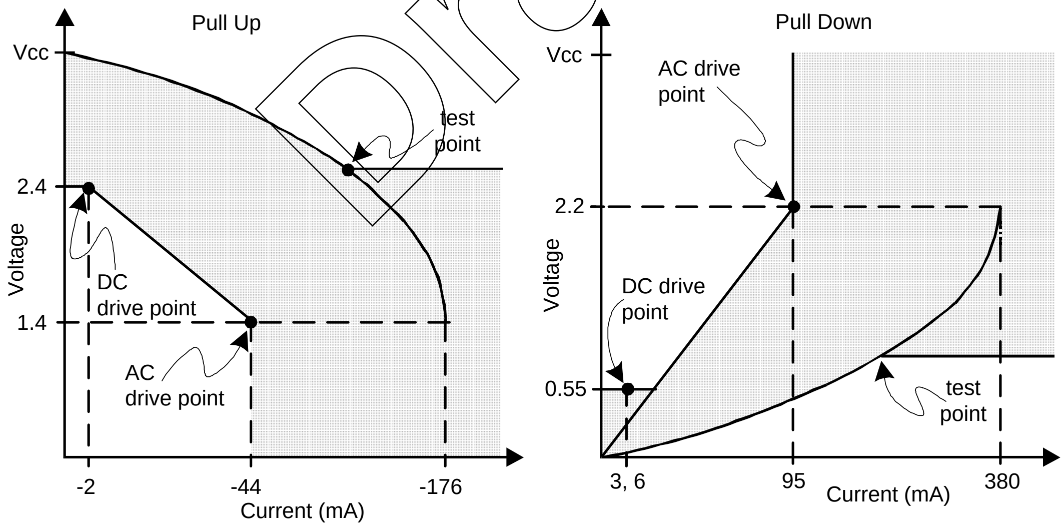
Parameter	Symbol	Min	Typ	Max	Unit
Power Supplies	PCIVDD VAUX CVDD CRYVDD VDD5REF	3 3.135 3 3 4.75	3.3 3.3 3.3 3.3 5	3.6 3.465 3.6 3.6 5.25	V V V V V
Operating Ambient Temperature	T_A	0	25	75	°C

5 V AC CHARACTERISTICS (PCI SIGNAL PINS ONLY)

(T_A = 70° C; PCIVDD = CVDD = VAUX = CRYVDD = 3.3 V; VDD5REF = Vcc = 5 V; PCIGND = CGND = CRYGND = 0 V; Logic 0 = 0 V, Logic 1 = 3.3 V; Reference levels = 1.4 V; unless otherwise noted;(Note 4))

Parameter		Symbol	Min	Max	Unit
Switching Current High (Note 5)	0 < Vout ≤ 1.4	I_{OH}	-44	-	mA
	1.4 < Vout < 2.4		$-44 + \frac{Vout - 1.4}{0.024}$	-	mA
	3.1 < Vout < 3.3		-	(Equation A)	
Switching Current Low (Note 5)	Vout ≥ 2.2	I_{OL}	95	-	mA
	2.2 > Vout > 0.55		Vout/0.023	-	mA
	0.71 > Vout > 0		-	(Equation B)	
Low Clamp Current	-5 < Vin ≤ -1	I_{CL}	$-25 + \frac{Vin + 1}{0.015}$	-	mA
Output rise slew rate	0.4 V - 2.4 V load (Note 6)	slewr	1	5	V/ns
Output fall slew rate	2.4 V - 0.4 V load (Note 6)	slewf	1	5	V/ns

Notes: 4. Specifications guaranteed by characterization and not production testing.
5. Refer to V/I curves in Figure 10. Switching Current High specification does not apply to PME#, CLKRUN#, and INTA# which are open drain outputs.
6. Cumulative edge rate across specified range. Rise slew rates do not apply to open drain outputs.



Equation A:

I_{OH} = 11.9 × (Vout - 5.25) × (Vout + 2.45)
for 3.3 V > Vout > 3.1 V

Equation B:

I_{OL} = 78.5 × Vout × (4.4 - Vout)
for 0 V < Vout < 0.71 V

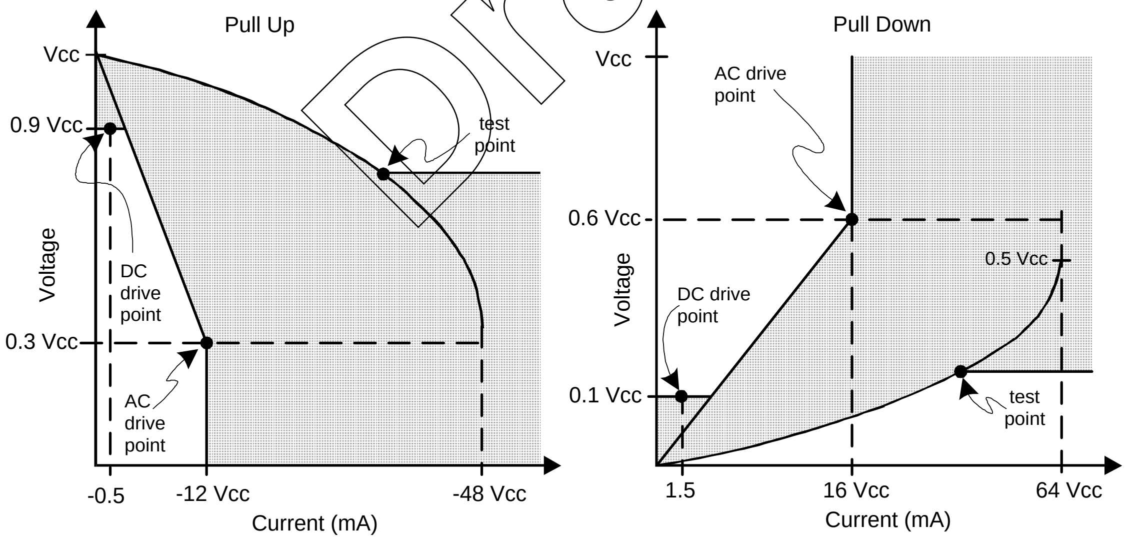
Figure 10. 5 V AC Characteristics

3.3 V AC CHARACTERISTICS (PCI SIGNAL PINS ONLY)

(T_A = 70° C; PCIVDD = CVDD = VAUX = CRYVDD = 3.3 V; VDD5REF = 5 V; PCIGND = CGND = CRYGND = 0 V; Logic 0 = 0 V, Logic 1 = Vcc = 3.3 V; Reference levels = 1.4 V; unless otherwise noted;(Note 4))

Parameter		Symbol	Min	Max	Unit
Switching Current High (Note 7)	0 < Vout ≤ 0.3 Vcc 0.3 Vcc < Vout < 0.9 Vcc 0.7 Vcc < Vout < Vcc	I_{OH}	-12 Vcc -17.1 (Vcc-Vout) -	- - (Equation C)	mA mA mA
Switching Current Low (Note 7)	Vcc > Vout ≥ 0.6 Vcc 0.6 Vcc > Vout > 0.1 Vcc 0.18 Vcc > Vout > 0	I_{OL}	16 Vcc 26.7 Vout -	- - (Equation D)	mA mA mA
Low Clamp Current	-3 < Vin ≤ -1	I_{CL}	$-25 + \frac{Vin + 1}{0.015}$	-	mA
High Clamp Current	Vcc + 4 > Vin ≥ Vcc + 1	I_{CH}	$25 + \frac{Vin - Vcc - 1}{0.015}$	-	mA
Output rise slew rate	0.2 Vcc - 0.6 Vcc load (Note 6)	slewr	1	4	V/ns
Output fall slew rate	0.6 Vcc - 0.2 Vcc load (Note 6)	slewf	1	4	V/ns

Notes: 7. Refer to V/I curves in Figure 11. Vcc is 3.3 V for the Figure below. Switching Current High specification does not apply to PME#, CLKRUN#, and INTA# which are open drain outputs.



Equation C:

I_{OH} = (98.0/Vcc) × (Vout - Vcc) × (Vout + 0.4 Vcc) for Vcc > Vout > 0.7 Vcc

Equation D:

I_{OL} = (256/Vcc) × Vout × (Vcc - Vout) for 0 V < Vout < 0.18 Vcc

Figure 11. 3.3 V AC Characteristics

DC CHARACTERISTICS (T_A = 70° C; PCIVDD = CVDD = VAUX = CRYVDD = 3.3 V; VDD5REF = 5 V; PCIGND = CGND = CRYGND = 0 V; all voltages with respect to 0 V unless otherwise noted)

Parameter	Symbol	Min	Max	Unit
PCI Interface Signal Pins				
High level input voltage	V_{IH}	2	5.75	V
Low level input voltage	V_{IL}	-0.5	0.8	V
High level output voltage Iout = -2 mA	V_{OH}	2.4	-	V
Low level output voltage Iout = 3 mA, 6 mA (Note 8)	V_{OL}	-	0.55	V
High level leakage current Vin = 2.7 V (Note 9)	I_{IH}	-	70	µA
Low level leakage current Vin = 0.5 V	I_{IL}	-	-70	µA
PME# power off input leakage Vin = 5.25 V (Note 10)	I_{OFF}	-	1	µA
Non-PCI Interface Signal Pins (Note 11)				
High level output voltage Iout = -5 mA (Note 12)	V_{oh}	0.9×Vdd	-	V
Low level output voltage Iout = 5 mA	V_{ol}	-	0.1×Vdd	V
High level input voltage (Note 13)	V_{ih}	0.65×Vdd	Vdd+0.3	V
Low level input voltage (Note 13)	V_{il}	–0.3	0.35×Vdd	V
High level leakage current Vin = 3.6 V (Note 13)	I_{ih}	-	10	µA
Low level leakage current Vin = 0 (Note 13)	I_{il}	-	-10	µA

Parameter	Min	Typ	Max	Unit
Power Supply Pins (Outputs Unloaded)				
Power Supply Current: VDD5REF VAUX PCIVDD/CVDD/CRYVDD Total	- - 	0.6 TBD TBD	- 20 TBD	mA mA mA
Low-Power Mode Supply Current	-	TBD	-	mA

Notes: 8. The following signals are tested to 6 mA: FRAME#, TRDY#, IRDY#, DEVSEL#, STOP#, PERR#, and INTA#. All other PCI interface signals are tested to 3 mA.

9. Input leakage currents include hi-Z output leakage for all bi-directional buffers with three-state outputs.

10. Maximum allowable leakage into the PME# open-drain pin when power is removed from VAUX. Assumes no event occurred to drive PME# (idle state).

11. For AC-Link and VOLUP/DN pins, Vdd is VAUX. For all others Vdd is the core supply.

12. For open drain pins, high level output voltage is dependent on external pull-up used and number of attached gates.

13. All inputs that do not include internal pull-ups or pull-downs, must be externally driven for proper operation. If an input is not driven, it should be tied to power or ground, depending on the particular function. If an I/O pin is not driven and programmed as an input, it should be tied to power or ground through its own resistor.

PCI INTERFACE PINS (T$_A$ = 0 to 70° C; PCIVDD = CVDD = VAUX = CRYVDD = 3.3 V; VDD5REF = 5 V; PCIGND = CGND = CRYGND = 0 V; Logic 0 = 0 V, Logic 1 = 3.3 V; Timing reference levels = 1.4 V

Parameter	Symbol	Min	Max	Unit
PCICLK cycle time	t_{cyc}	30	-	ns
PCICLK high time	t_{high}	11	-	ns
PCICLK low time	t_{low}	11	-	ns
PCICLK to signal valid delay - bused signals	t_{val}	2	11	ns
PCICLK to signal valid delay - point to point	$t_{val(p+p)}$	2	12	ns
Float to active delay (Note 14)	t_{on}	2	-	ns
Active to Float delay (Note 14)	t_{off}	-	28	ns
Input Set up Time to PCICLK - bused signals	t_{su}	7	-	ns
Input Set up Time to PCICLK - point to point	$t_{su(p+p)}$	10, 12	-	ns
Input hold time for PCICLK	t_h	0	-	ns
Reset active to output float delay (Notes 14, 15, 16)	$t_{rst\text{-}off}$	-	25	ns

Notes: 14. For Active/Float measurements, the Hi-Z or “off” state is when the total current delivered is less than or equal to the leakage current. Specification is guaranteed by design, not production tested.

15. RST# is asserted and de-asserted asynchronously with respect to PCICLK.

16. All PCI output drivers are asynchronously floated when RST# is active. Note ASDOUT and ASYNC are not affected by RST#.

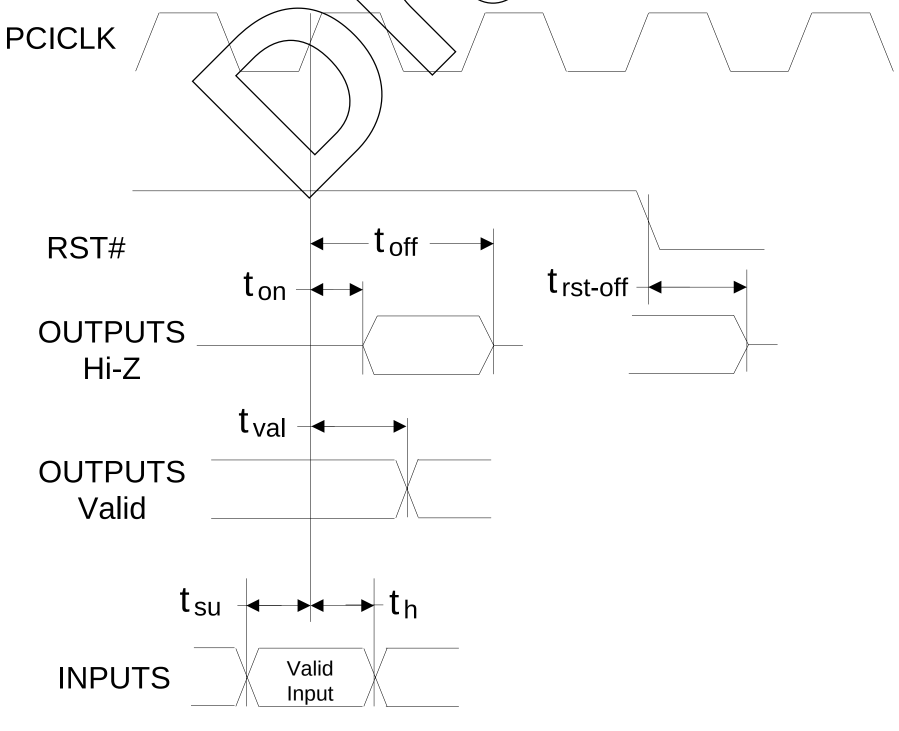


Figure 12. PCI Timing Measurement Conditions

AC ’97 SERIAL INTERFACE TIMING (T$_A$ = 0 to 70° C;

PCIVDD = CVDD = VAUX = CRYVDD = 3.3 V; VDD5REF = 5 V; PCIGND = CGND = CRYGND = 0 V; Logic 0 = V_{ol},V_{il}, Logic 1 = V_{oh},V_{ih}; unless otherwise noted)

Parameter	Symbol	Min	Typ	Max	Unit
ABITCLK input cycle time	t_{clk_period}		81.4	-	ns
ABITCLK input high time	t_{clk_high}	36	40.7	45	ns
ABITCLK input low time	t_{clk_low}	36	40.7	45	ns
ABITCLK input rise/fall time	t_{rfclk}	2	-	6	ns
ASDIN/ASDIN2 valid to ABITCLK falling	t_{setup}	10	-	-	ns
ASDIN/ASDIN2 hold after ABITCLK falling	t_{hold}	10	-	-	ns
ASDIN/ASDIN2 input rise/fall time	t_{rfin}	2	-	6	ns
ABITCLK rising to ASDOUT/ASYNC valid, C_L = 55 pF	t_{co}	2	-	15	ns
ASYNC/ASDOUT rise/fall time, C_L = 55 pF	t_{rfout}	2	-	6	ns

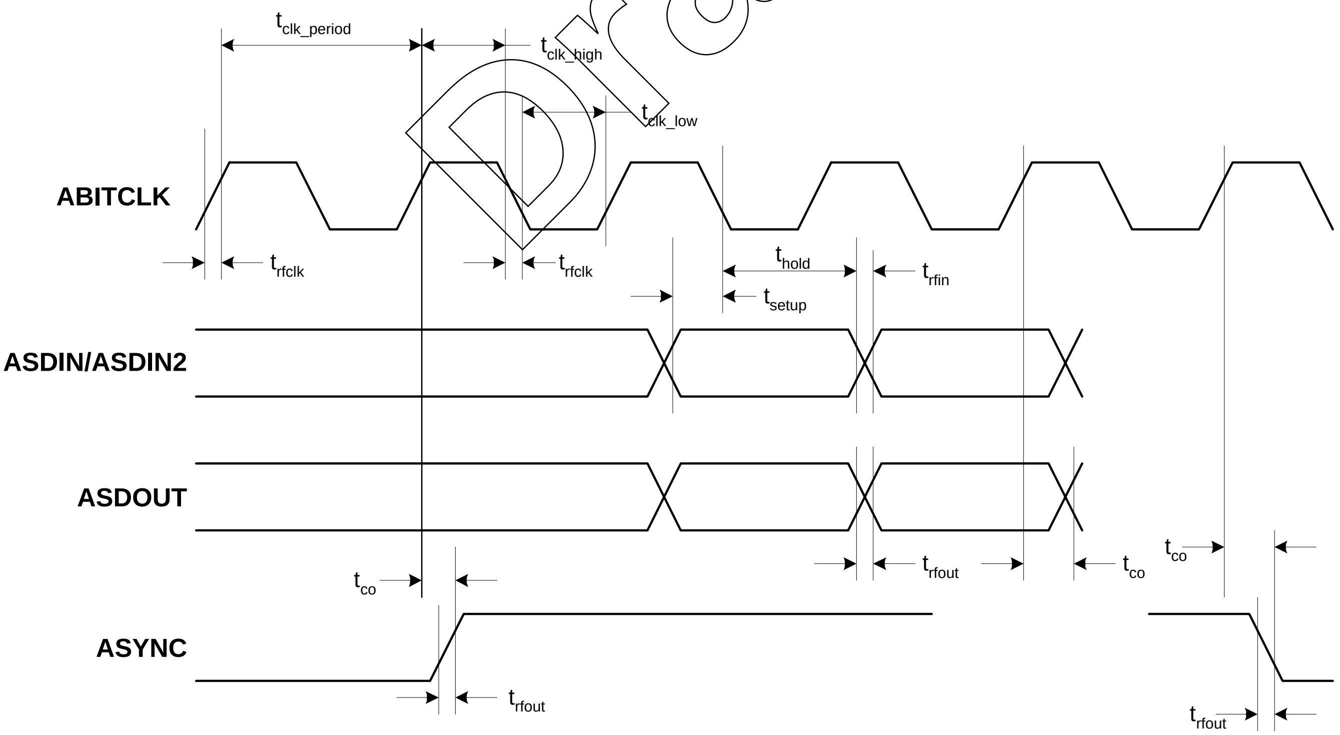


Figure 13. AC ’97 Configuration Timing Diagram

EEPROM TIMING CHARACTERISTICS (T_A = 0 to 70° C; PCIVDD = CVDD = VAUX = CRYVDD = 3.3 V; VDD5REF = 5 V; PCIGND = CGND = CRYGND = 0 V; Logic 0 = 0 V, Logic 1 = 3.3 V; Timing reference levels = 1.4 V; PCI clock frequency = 33 MHz; unless otherwise noted)

Parameter	Symbol	Min	Max	Units
EECLK Low to EEDAT Data Out Valid	t_{AA}	0	7.0	µs
Start Condition Hold Time	$t_{HD:STA}$	5.0	-	µs
EECLK Low	t_{LEECLK}	10	-	µs
EECLK High	t_{HEECLK}	10	-	µs
Start Condition Setup Time (for a Repeated Start Condition)	$t_{SU:STA}$	5.0	-	µs
EEDAT In Hold Time	$t_{HD:DAT}$	0	-	µs
EEDAT In Setup Time	$t_{SU:DAT}$	250	-	ns
EEDAT/EECLK Rise Time (Note 17)	t_R	-	1	µs
EEDAT/EECLK Fall Time	t_F	-	300	ns
Stop Condition Setup Time	$t_{SU:STO}$	5.0	-	µs
EEDAT Out Hold Time	t_{DH}	0	-	µs

Notes: 17. Rise time on EEDAT is determined by the capacitance on the EEDAT line with all connected gates and the required external pull-up resistor.

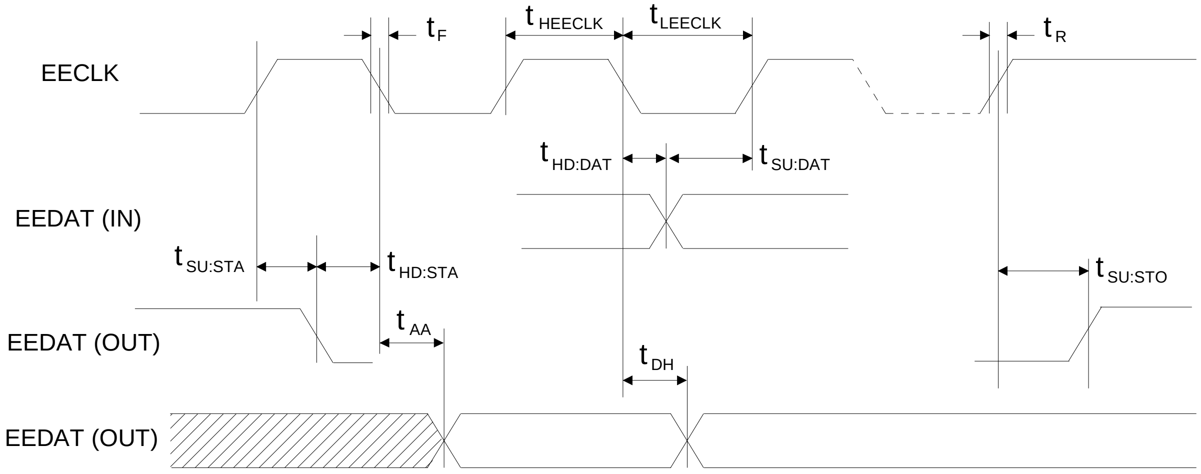


Figure 14. EEPROM Timing

Digital Functionality, but not meeting timing or level specifications

Supply Voltage - Core	2.5	4.1Volts
Temperature	-55	125°C

There are no power supply sequencing limitations on any supply pins. However, care must be taken to ensure that the I/O pins are not driven beyond the Maximum Input Current Limit before power is applied.

The I/O pin currents are DC limits, it is expected that transient conditions will pass more current.

Note that some pins will have internal pullups or pulldowns. Leakage specs for these pins will be measured in the Float All Pins test mode.

3.3 Digital Filter Specs

The digital filter specifications are valid for audio sample rate of 48 kHz. Other sample rates will have somewhat worse performance.

3.4 Implied Functional Specs

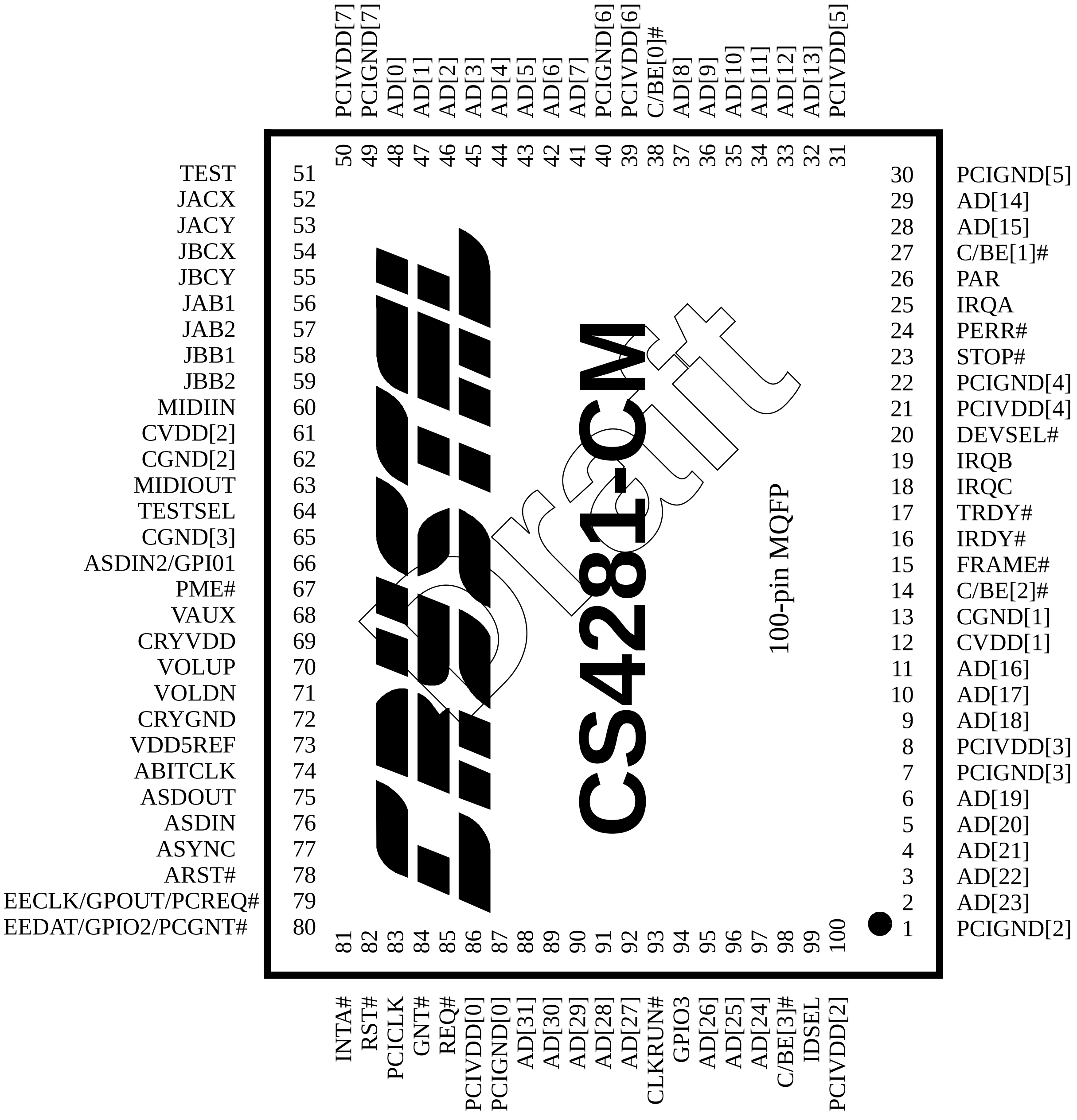
Implied functional specifications are the un-tabulated expectations that a reasonable customer may have. These often are described in the data sheet language and in sales literature. These specs will also come from industry standards that we reference - such as PC ‘98, AC ‘97 etc. Another source is a common expectation of the device being suitable for its intended use.

	Min	Typ	Max
Digital Dynamic Range DA	85	90	dB Fs
Digital Dynamic Range AD	75	80	dB Fs
THD+N (-1 dB Fs)	-80		dB Fs
Freq Response +/-0.25dB	20	19200	Hz
Transition Band	19200	28800	Hz
Stop Band	28800	light	Hz
Stop Band Rejection	-74		dB
Out of Band Rejection	-40		dB
Group delay		1	ms
Spurious Tome Rejection	-100		dB
Attenuation Step Size	-1.5		dB

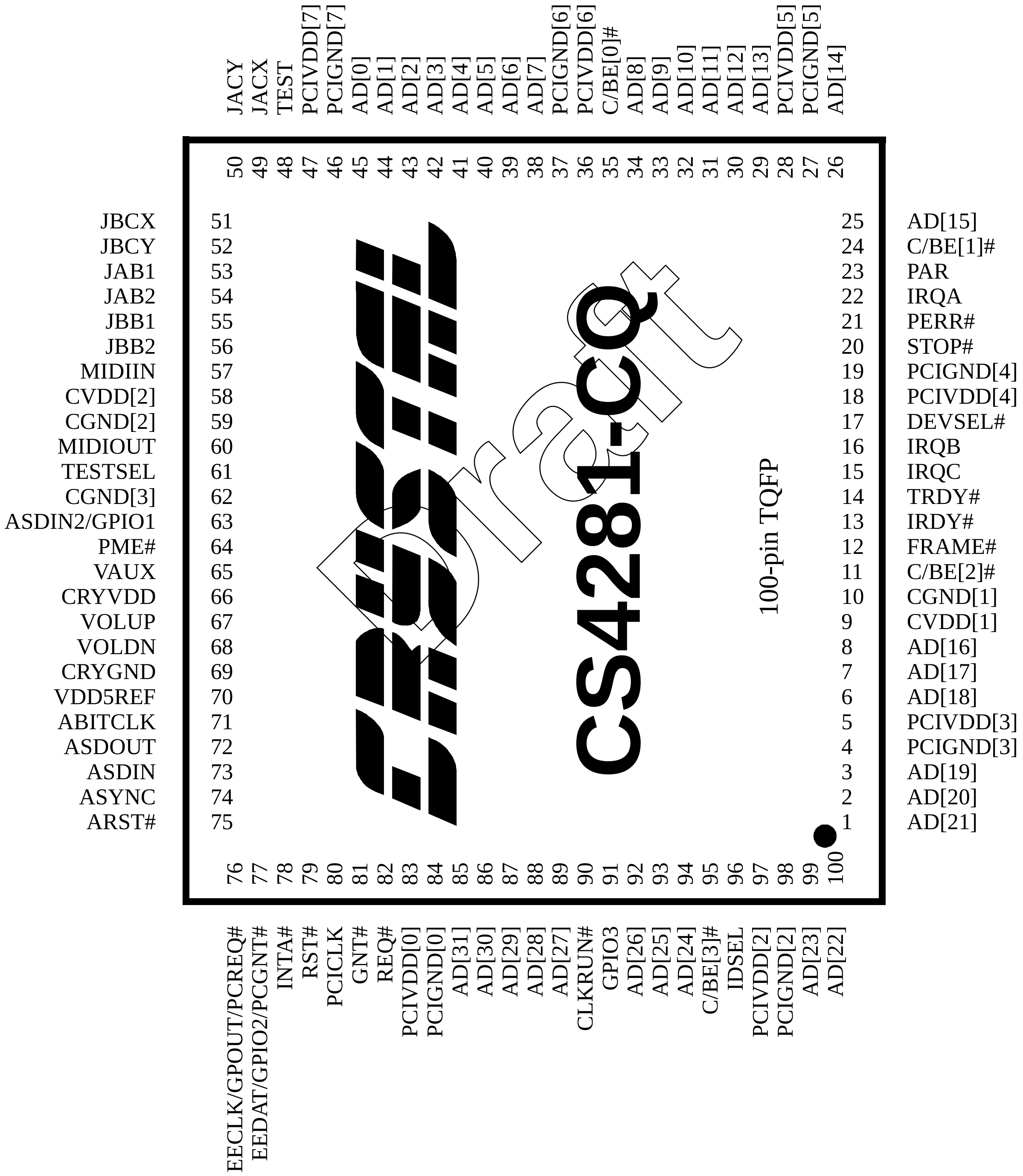
Table 1. Digital Filter Specs

4. PINOUT

4.1 Physical Pin Placement - CS4610/14/CS4280 Compatible 100 Pin MQFP



4.2 Physical Pin Placement - 100 Pin TQFP



4.3 Pin Descriptions

4.3.1 PCI Interface

AD[31:0]

Address/Data Bus
Input/Output
These pins form the multiplexed address/data bus for the PCI interface.

C/BE[3:0]#

Command Type/Byte Enables
Input/Output
These four pins are the multiplexed command/byte enables for the PCI interface. During the address phase of a transaction, these pins indicate cycle type. During the data phases of a transaction, active low byte enable information for the current data phase is indicated. These pins are inputs during slave operation and they are outputs during bus mastering operation.

PAR

Parity
Input/Output, Active High
The Parity pin indicates even parity across **AD[31:0]** and **C/BE[3:0]#** for both address and data phases. The signal is delayed one PCI clock from either the address or data phase for which parity is generated.

FRAME#

Cycle Frame
Input/Output, Active Low
FRAME# is driven by the current master to indicate the beginning and duration of a transaction.

IRDY#

Initiator Ready
Input/Output, Active Low
IRDY# is driven by the current bus master to indicate that as the initiator it is ready to transmit or receive data (complete the current data phase).

TRDY#

Target Ready
Input/Output, Active Low
TRDY# is driven by the current bus target to indicate that as the target device it is ready to transmit or receive data (complete the current data phase).

STOP#

Transaction Stop
Input/Output, Active Low
STOP# is driven active by the current bus target to indicate a request to the master to stop the current transaction.

IDSEL

Initialize Device Select

Input, Active High

IDSEL is used as a chip select during PCI Configuration Space read and write cycles.

DEVSEL#

Device Select

Input/Output, Active Low

DEVSEL# is driven by the target device to indicate that it has decoded the address of the current transaction as its own chip select range.

REQ#

Master Request

Tri-state Output, Active Low

REQ# indicates to the system arbiter that this device is requesting access to the PCI bus. This pin must be tri-stated when **RST#** is active.

GNT#

Master Grant

Input, Active Low

GNT# is driven by the system arbiter to indicate to the device that the PCI bus has been granted.

PERR#

Parity Error

Input/Output, Active Low

PERR# is used for reporting data parity errors on the PCI bus.

INTA#

Host Interrupt A (for SP)

Open Drain, Active Low

INTA# is the level triggered interrupt pin dedicated to servicing internal device interrupt sources.

PCICLK

PCI Bus Clock

Clock Input, Rising Edge

PCICLK is the PCI bus clock for timing all PCI transactions. All PCI synchronous signals are generated and sampled relative to the rising edge of this clock.

RST#

PCI Device Reset

Input, Active Low

RST# is the PCI bus master reset.

VDD5REF

Clean 5V for Pseudo Supply

Power

VDD5REF is the power connection pin for the 5 V PCI pseudo supply for the PCI bus drivers.

PCIVDD[7:2, 0]

PCI Driver Power
Power
PCIVDD pins are the PCI driver power supply pins and must be connected to a 3.3 Volt supply.

PCIGND[7:2, 0]

PCI Driver Ground
Ground
PCIGND pins are the PCI driver ground reference pins.

PME#

PCI Power Management Event
Output, Open Drain, Active Low
PME# signals a PCI Power Management event. This pin powers up high impedance for backwards compatibility. It is also backwards compatible since the previous function was a ground pin. **PME#** functionality is powered from the **VAUX** power supply to support D3$_{cold}$ wake-up from the AC Link.

CLKRUN#

PCI Clock Activity Sense/Request
I/O Open Drain, Active Low
As an input, high indicates that the **PCICLK** is active. The bus controller drives **CLKRUN#** low when it wants to stop the **PCICLK**. As an output, driven low to request that the **PCICLK** be activated or not stopped. If not used, this pin must have a weak pull-down attached to keep low. This pin is backwards compatible since the previous function was a ground pin.

VAUX

PCI Auxiliary Power Supply
Power
Auxiliary 3.3 Volt VDD pin used to maintain limited device functionality when the normal VDD is turned off. This pin is backwards compatible since the previous function was a core power supply pin with the same voltage.

4.3.2 Clock/Miscellaneous

IRQ[A:C]

ISA interrupt pins
Outputs
These pins can be used in DOS legacy-compatibility mode to bypass the PCI interrupt and use an ISA interrupt directly. Using these pins can preserve the multi-drop capability of the PCI interrupt at the expense of another interrupt line. These pins power up high impedance for backwards compatibility.

BONDID[1:0]

Bond Pad ID 1, 0

Inputs

These two pads are bonded to the PCI power or ground and do not appear on external pins. Their logical state is reflected in the **FRR.ID[1:0]** bits and can be used for Marketing part differentiation if needed. The default value is both pads grounded. These bits affect no internal hardware except the register value.

4.3.3 External Interface Pins

TEST

Test Mode Strobe

Input with Pulldown, Edge triggered

This pin is used for test mode entry. A low-to-high transition enters test mode and activates the selected test capability. The part remains in the test mode until the **TEST** pin goes low. This pin has an internal pulldown to ground, and may remain floating. When the **TESTSEL** pin is high, the XOR-tree test mode is entered. If **TESTSEL** is low, then the **MIDIIN** and four joystick button pins determine the test mode. This pin has super-hysteresis input buffer.

TESTSEL

Test Mode Select

Input with Pullup

This pin selects the active test mode while the **TEST** pin is active high. **TESTSEL** is sampled on the rising edge of **TEST** to determine the basic test mode functionality. **TESTSEL** low activates normal factory test functions. **TESTSEL** high activates the I/O pin boundary XOR tree test mode.

EEDAT/GPIO2/PCGNT#

External EEPROM Data/PC/PCI Grant

Input/Output, Open Drain Serial I/O

For expansion card designs, a data line for external serial EEPROM containing device configuration data. For motherboard designs using PC/PCI, the PC/PCI serialized grant input signal. This pin can also be a general purpose I/O pin.

EECLK/GPOUT/PCREQ#

External EEPROM Clock/PC/PCI Request

Output, Serial Output

For expansion card designs, clock for external serial EEPROM containing configuration data. For motherboard designs using PC/PCI, the PC/PCI serialized request signal or general purpose output.

GPIO3

General Purpose IO Pin 3

Input/Output

A general purpose I/O pin that is powered off the PCI power supply. Therefore, this pin does not support **PME#** control from the AC link during D3$_{cold}$. This pin powers up in a high impedance state providing backward compatibility.

VOLUP

Volume Control Up Input

Input with Pullup, Active Low

Volume up button control input. This pin has an internal pullup and is internally debounced. **VOLUP** and **VOLDN** together going low cause the master volume to toggle between mute and unmute.

VOLDN

Volume Control Up Input

Input/Output with Pullup, Active Low input

Volume down button control input. This pin has an internal pullup and is internally debounced. **VOLUP** and **VOLDN** together going low cause the master volume to toggle between mute and unmute.

4.3.4 Clock/Miscellaneous

CRYVDD

DLL Power

Power

A 3.3 Volt power pin for internal delay locked loop.

CRYGND

DLL Ground

Ground

Ground pin for internal delay locked loop.

J[A:B]C[X:Y]

Joystick Coordinate Input

Analog Input/Open Drain Output

These pins are the 4 axis analog inputs for the joystick port. These pins are clamped to ground until a write occurs to the joystick port when they are released and start charging based on the joystick potentiometers.

JAB1

Joystick A Button 1

Input with Pullup, Active Low

Button 1 input pin for joystick A.

JAB2

Joystick A Button 2
Input with Pullup, Active Low
Button 2 input pin for joystick A.

JBB1

Joystick B Button 1
Input with Pullup, Active Low
Button 1 input pin for joystick B.

JBB2

Joystick B Button 2
Input with Pullup, Active Low
Button 2 input pin for joystick B.

MIDIIN

MIDI Data Input, internal 20 kΩ Pullup
This is the serial input pin for the internal MIDI port.

MIDIOUT

MIDI Data Output
This is the serial output pin for the internal MIDI port.

CVDD[2:1]

Core Power
Power
3.3 Volt core power pins.

CGND[3:1]

Core Ground
Ground
Core ground reference pins.

4.3.5 Serial Codec Interface

ABITCLK

AC 97 Bit Rate Clock
Input
Master timing clock for serial audio data. This pin is an input which drives the timing for the AC Link interface, along with providing the source clock for the DLL.

ASYNC

AC Link Frame Sync
Output
Framing clock for serial audio data. This pin is an output which indicates the 48 kHz framing for the AC Link. High during slot 0 and low for slots 1 through 12. Also used for a warm reset of the AC Link.

ASDOUT

AC Link Data Out
Output, Active High

Serial output data. Output from the CS4281 and provides a register interface and audio data to both the Primary and Secondary Codecs.

ASDIN

AC Link Data In
Input, Active High

Serial input data for the first / Primary AC ‘97 Codec. This pin is powered from the **VAUX** power pin to support wake-up events and drive **PME#**.

ASDIN2/GPIO1

Secondary AC Link Input/General Purpose IO Pin 1
Input/Output

Serial data input for Secondary AC ‘97 2.0-compatible codec or general purpose input, selected via the **SPMC.ASDI2E** bit. This pin is powered from the **VAUX** power pin to support wake-up events and drive **PME#**. As a general purpose I/O pin, it supports extended capability and **PME#** (powered from **VAUX**). If this pin is not used, it should have an external resistor of 50 kΩ or greater attached to ground (not power).

ARST#

AC Link and Codec Reset
Output, Active Low

AC ‘97 link reset pin. This pin is the logical OR or the PCI reset pin **RST#** and the software controlled **SPMC.RSTN** bit. When low, forces any Codecs attached to the AC Link into a Cold Reset.

5. PCI PROPERTIES AND CONFIGURATION

The following table specifies all supported data transfers across the CS4281 PCI interface

Initiator	Target	Type	PCI Dir
Host	Registers (BA0)	Memory Write	In
Host	Registers (BA0)	Memory Read	Out
Host	Memories (BA1)	Memory Write	In
Host	Memories (BA1)	Memory Read	Out
Host	Config Space 1	Config Write	In
Host	Config Space 1	Config Read	Out
DMA	Host System	Memory Write	Out
DMA	Host System	Memory Read Multiple	In
DMA	South Bridge	I/O Read	In

Table 2. All Supported Data Transfers Across the CS4281 PCI Interface

5.1 Slave Interface Properties

5.1.1 Supported Cycles

- Memory Read
- Memory Write
- Configuration Read
- Configuration Write
- Memory Read Multiple (aliased to Memory Read)
- Memory Read Line (aliased to Memory Read)
- Memory Write and Invalidate (aliased to Memory Write)
- I/O Read
- I/O Write

5.1.2 Unsupported Cycles

- Interrupt Acknowledge
- Special Cycles
- Dual Address Cycle

5.1.3 Error Conditions

There are no “target abort” error conditions signaled.

5.1.4 Protocol Support

<u>Configuration Space Accesses</u>

- Byte enables are supported (byte/word/doubleword transfers are valid)
- No bursting (disconnect after first data phase if a burst access is attempted)

Register Area Accesses

- Byte enables are supported
- No bursting (disconnect after first data phase if a burst access is attempted)

Memory Area Accesses

- Byte enables are supported
- No bursting (disconnect after first data phase if a burst access is attempted)

5.2 Master Interface Properties

5.2.1 Generated Cycles

- Memory Read
- Memory Write
- I/O Read

5.2.2 Cycles Not Generated

- Memory Read Multiple
- I/O Write
- Configuration Read
- Configuration Write
- Memory Read Line
- Memory Write and Invalidate
- Interrupt Acknowledge
- Special Cycles
- Dual Address Cycle

5.3 Configuration Space

The following table specifies the primary PCI configuration space for the CS4281.

Byte 3	Byte 2	Byte 1	Byte 0	Offset
Device ID: R/O, 6005h for CS4281 6XXXh range is reserved for Crystal		Vendor ID: R/O, 1013h (Cirrus Logic's PCI ID)		00h
Status Register, bits 15-0: Bit 15: Detected Parity Error: Error Bit Bit 14: Signalled SERR: R/O: 0 Bit 13: Received Master Abort: Error Bit Bit 12: Received Target Abort: Error Bit Bit 11: Signalled Target Abort: R/O 0 Bit 10-9: DEVSEL Timing: R/O, 01 (medium) Bit 8: Data Parity Error Detected: Error Bit Bit 7: Fast Back to Back Capable: R/O 0 Bit 6: User Definable Features: R/O 0 Bit 5: 66MHz Bus: R/O 0 Bit 4: New Capabilities: R/O 1 Bit 3-0: Reserved: R/O 0000 Reset Status State: 0210h Write of 1 to any error bit position clears it.		Command Register, bits 15-0: Bit 15-10: Reserved, R/O 000000 Bit 9: Fast B2B Enable: R/O 0 Bit 8: SERR Enable: R/O 0 Bit 7: Wait Control: R/O 0 Bit 6: Parity Error Response: R/W, default 0 Bit 5: VGA Palette Snoop: R/O 0 Bit 4: MWI Enable: R/O 0 Bit 3: Special Cycles: R/O 0 Bit 2: Bus Master Enable: R/W, default 0 Bit 1: Memory Space Enable: R/W, default 0 Bit 0: I/O Space Enable: R/O 0		04h
Class Code: R/O 040100h Class 04h (multimedia device), Sub-class 01h (audio), Interface 00h			Revision ID: R/O 01h	08h
BIST: R/O 0	Header Type: Bit 7: R/O 0 Bit 6-0: R/O 0 (type 0)	Latency Timer: Bit 7-3: R/W,default 0 Bit 2-0: R/O 0	Cache Line Size: R/O 0	0Ch
Base Address Register 0 Device Control Register space, memory mapped. 4 kByte size Bit 31-12: R/W, default 0. Compare address for register space accesses Bit 11 - 4: R/O 0, specifies 4 kByte size Bit 3: R/O 0, Not Prefetchable (Cacheable) Bit 2-1: R/O 00, Location Type - Anywhere in 32 bit address space Bit 0: R/O 0, Memory space indicator				10h
Base Address Register 1 Device Memory Array mapped into host system memory space, 64 kByte size Bit 31-16: R/W, default 0. Compare address for memory array accesses Bit 15 - 4: R/O 0, specifies 64 kByte size Bit 3: R/O 0, Not Prefetchable (Cacheable) Bit 2-1: R/O 00, Location Type - Anywhere in 32 bit address space Bit 0: R/O 0, Memory space indicator				14h
Base Address Register 2: R/O 00000000h, Unused				18h
Base Address Register 3: R/O 00000000h, Unused				1Ch
Base Address Register 4: R/O 00000000h, Unused				20h
Base Address Register 5: R/O 00000000h, Unused				24h

Table 3. PCI Configuration Space

Byte 3	Byte 2	Byte 1	Byte 0	Offset
Cardbus CIS Pointer: R/O 00000000h, Unused				28h
Subsystem ID R/O 0000h default, see Subsystem ID section		Subsystem Vendor ID R/O 0000h default, see Subsystem ID section		2Ch
Expansion ROM Base Address: R/O 00000000h, Unused				30h
Reserved: R/O 000000h			Cap_Ptr: R/O, 40h	34h
Reserved: R/O 00000000h				38h
Max_Lat: R/O 18h 24 x 0.25 µS = 6 µS	Min_Gnt: R/O 04h 4 x 0.25 µS = 1 µS	Interrupt Pin: R/O 01h, INTA used	Interrupt Line: R/W, default 0	3Ch
PMC Bit 15: PME# from D3$_{cold}$: R/O 0 default Bit 14: PME# from D3$_{hot}$: R/O 1 Bit 13: PME# from D2: R/O 1 Bit 12: PME# from D1: R/O 1 Bit 11: PME# from D0: R/O 1 Bit 10: D2 support: R/O 1 Bit 9: D1 support: R/O 1 Bit 8-6: Vaux Current: R/O 000 default Bit 5: Device specific init: R/O 1 Bit 4: Auxiliary power: R/O 0 default Bit 3: PME# clock: R/O 0 Bit 2-0: Version: R/O 010 (Rev. 1.1 compliant)		Next Item Ptr: R/O 00h	Capability ID: R/O 01h	40h
Data: R/O 0	PMCS_BSE R/O 0	PMCS Bit 15: PME# status: R/W 0 Bit 14-13: Data scale: R/O 00 Bit 12-9: Data select: R/O 0000 Bit 8: PME_En: R/W 0 Bit 7-2: Reserved: R/O 000000 Bit 1-0: Power state: R/W 00		44h
Reserved: R/O 00000000h				48h-DFh
Configuration Write Protect (CWPR)				E0h
Extended PCI Power Management Control (EPPMC)				E4h
GPIO Pin Interface (GPIOR)				E8h
Serial Port Power Management & Control (SPMC)				ECh
Configuration Load Register (CFLR)				F0h
BIOS Config. Flags	ISA IRQC R/W	ISA IRQB R/W	ISA IRQA: R/W	F4h
Reserved				F8h
Subsystem ID R/W 0000h default, see Subsystem ID section		Subsystem Vendor ID R/W 0000h default, see Subsystem ID section		FCh

Table 3. PCI Configuration Space (Continued)

5.3.1 Configuration Write Protect Register (CWPR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWP15	CWP14	CWP13	CWP12	CWP11	CWP10	CWP9	CWP8	CWP7	CWP6	CWP5	CWP4	CWP3	CWP2	CWP1	CWP0

Address: BA0: 3E0h, Read-Write
PCI CFG:0E0h, Read-Write

Default: 0000h

Definition: Vaux powered. Protects vendor defined configuration space from inadvertent writes. This register must be an exact value to allow the other vendor defined configuration space registers to be written. This register is always read/writable, resides in the PCI config space, and is only reset by a Vaux POR circuit. CWPR is unaffected by the PCI **RST#** signal. The default value is set by a Vaux POR signal. Host software should initialize this register before use.

Bit Descriptions:

CWP[15:0] Configuration Write Protect. Configuration space addresses 0E4h through 0FFh are read-only unless the **CWP** bits are set to 4281h. This provides some protection of the vendor defined configuration space from inadvertent writes. The **CWP** bits have no affect on the BA0 address access to these registers (3E4h-3FFh) which are always read-write.

4281h - Vendor defined configuration space between 0E4h - 0FFh is writable in addition to being readable.

not 4281h - If the CWP bits are not set to 4281h, then the vendor defined configuration space between 0E4h and 0FFh is read only.

5.4 Subsystem ID and Vendor ID Field Initialization

These configuration fields can be loaded in two different ways. The first method is by automatic load from an external EEPROM. The second method is to write the Subsystem ID and Subsystem Vendor ID at Configuration offset 0FCh. This method allows BIOS code to write the ID's and have it reflected at 02Ch, which is read-only per Microsoft specifications. This location lets the driver know who the design is for in case custom software support is needed.

5.4.1 Subsystem Vendor ID/Subsystem ID Pre-load Register (SSVID)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SID15	SID14	SID13	SID12	SID11	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	SID2	SID1	SID0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VID15	VID14	VID13	VID12	VID11	VID10	VID9	VID8	VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0

Address: BA0:3FCh, Read-Write
PCI CFG:0FCh, Read-Write if CWPR configured, otherwise Read-Only
PCI CFG:02Ch, Read Only

Default: 00000000h

Definition: Vaux powered. The Subsystem Vendor ID/Subsystem ID Pre-load register provides a host port for initializing the PCI configuration space subsystem ID's located at CFG:02Ch - which are read-only per Microsoft requirements. This register is unaffected by the PCI RST# signal. The default value is set by a Vaux POR signal. This register is loaded with EEPROM data if present.

Bit Descriptions:

VID[15:0] This bit field contains the PCI subsystem Vendor ID.

SID[15:0] This bit field contains the PCI Subsystem ID.

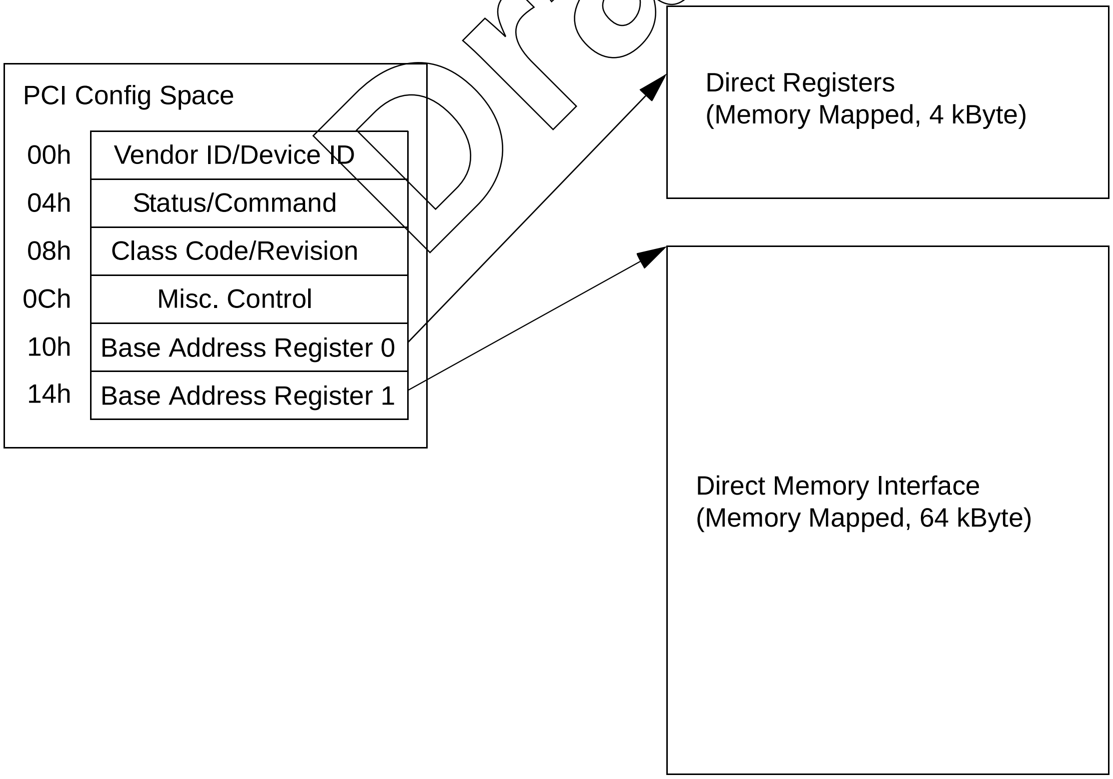
6. PCI INTERFACE

The PCI interface serves as a master/target PCI device. A two base address registers provide access to the chip operation registers and internal memory blocks. The chip will be a PCI master for DMA activity. Four separate DMA controllers will support simultaneous capture, play, modem transmit and receive, and various loopback paths. All other PCI access will be as a target.

The PCI Interface Block contains several distinct functional units. These are the PCI Bus Interface, the Data Formatter, the DMA Controller, and PCI FIFO controller.

7. HOST-INTERFACE PROGRAMMING MODEL

The CS4281 host interface is comprised of two separate interface blocks which are memory mapped into host address space. Both interface blocks can be located anywhere in 32-bit physical address space as defined by the address programmed into the two base address registers in PCI configuration space. These base addresses are normally set up by the system's Plug and Play BIOS. The first interface block (at Base Address 0) contains the general purpose configuration, control, and status registers for the device. The second interface block (at Base Address 1) maps the device RAMs and ROMs into host memory space. This allows the host to directly peek and poke FIFO RAM locations on the device for device testability.



PCI Config Space

00h	Vendor ID/Device ID
04h	Status/Command
08h	Class Code/Revision
0Ch	Misc. Control
10h	Base Address Register 0
14h	Base Address Register 1

Figure 15. PCI Memory Block Diagram

7.1 BA0 Register Map

All registers powered from the Core are set to their default state by **RST#** and by a generic chip POR power-up circuit. The Vaux-controlled registers indicate whether **RST#** resets them or not. All registers are read/write unless specified otherwise.

Address (BA0 Offset)	Name	Description	Power From	Int. Loc.	Page
000h	HISR	Host Interrupt Status Register	Core	PCI	80
004h					
008h	HICR	Host Interrupt Control Register	Core	PCI	81
00Ch	HIMR	Host Interrupt Mask Register	Core	PCI	82
010h	IIER	ISA Interrupt Enable Register	Core	PCI	83
014h - 0ECh					
0F0h	HDSR0	Host DMA Engine 0 Status Register	Core	PCI	104
0F4h	HDSR1	Host DMA Engine 1 Status Register	Core	PCI	104
0F8h	HDSR2	Host DMA Engine 2 Status Register	Core	PCI	104
0FCh	HDSR3	Host DMA Engine 3 Status Register	Core	PCI	104
100h					
104h - 10Ch					
110h	DCA0	DMA Engine 0 Current Address Register	Core	PCI	98
114h	DCC0	DMA Engine 0 Current Count Register	Core	PCI	100
118h	DBA0	DMA Engine 0 Base Address Register	Core	PCI	97
11Ch	DBC0	DMA Engine 0 Base Count Register	Core	PCI	99
120h	DCA1	DMA Engine 1 Current Address Register	Core	PCI	98
124h	DCC1	DMA Engine 1 Current Count Register	Core	PCI	100
128h	DBA1	DMA Engine 1 Base Address Register	Core	PCI	97
12Ch	DBC1	DMA Engine 1 Base Count Register	Core	PCI	99
130h	DCA2	DMA Engine 2 Current Address Register	Core	PCI	98
134h	DCC2	DMA Engine 2 Current Count Register	Core	PCI	100
138h	DBA2	DMA Engine 2 Base Address Register	Core	PCI	97
13Ch	DBC2	DMA Engine 2 Base Count Register	Core	PCI	99
140h	DCA3	DMA Engine 3 Current Address Register	Core	PCI	98
144h	DCC3	DMA Engine 3 Current Count Register	Core	PCI	100
148h	DBA3	DMA Engine 3 Base Address Register	Core	PCI	97
14Ch	DBC3	DMA Engine 3 Base Count Register	Core	PCI	99
150h	DMR0	DMA Engine 0 Mode Register	Core	PCI	101
154h	DCR0	DMA Engine 0 Command Register	Core	PCI	103
158h	DMR1	DMA Engine 1 Mode Register	Core	PCI	101
15Ch	DCR1	DMA Engine 1 Command Register	Core	PCI	103
160h	DMR2	DMA Engine 2 Mode Register	Core	PCI	101
164h	DCR2	DMA Engine 2 Command Register	Core	PCI	103
168h	DMR3	DMA Engine 3 Mode Register	Core	PCI	101
16Ch	DCR3	DMA Engine 3 Command Register	Core	PCI	103
170h		Reserved			
174h		Reserved			
178h-17Ch		Reserved			

Table 4. BA0 Register Map

Address (BA0 Offset)	Name	Description	Power From	Int. Loc.	Page
180h	FCR0	FIFO Control Register 0	Core	Ctrl	105
184h	FCR1	FIFO Control Register 1	Core	Ctrl	105
188h	FCR2	FIFO Control Register 2	Core	Ctrl	105
18Ch	FCR3	FIFO Control Register 3	Core	Ctrl	105
190h	FPDR0	FIFO Polled Data Register 0	Core	PCI	106
194h	FPDR1	FIFO Polled Data Register 1	Core	PCI	106
198h	FPDR2	FIFO Polled Data Register 2	Core	PCI	106
19Ch	FPDR3	FIFO Polled Data Register 3	Core	PCI	106
200h - 208h		Reserved			
20Ch	FCHS	FIFO Channel Status	Core	PCI	108
210h	FSIC0	FIFO Status and Interrupt Control Register 0	Core	PCI	106
214h	FSIC1	FIFO Status and Interrupt Control Register 1	Core	PCI	106
218	FSIC2	FIFO Status and Interrupt Control Register 2	Core	PCI	106
21Ch	FSIC3	FIFO Status and Interrupt Control Register 3	Core	PCI	106
220h - 2FCh					
300h - 340h		PCI Configuration Space Echo, offsets 00h - 42h, RO	Core	Ctrl	47
344h	PMCS	Power Management Control/Status	Vaux	Vaux	69
348h - 3E0h		Reserved (PCI config space expansion)			
3E0h	CWPR	Configuration Write Protect Register	Vaux	Vaux	49
3E4h	EPPMC	Extended PCI Power Management Control	Vaux	Vaux	71
3E8h	GPIOR	GPIO Pin Interface Register	Vaux	Vaux	186
3ECh	SPMC	Serial Port Power Management Control (& ASDIN2 enable)	Vaux	Vaux	212
3F0h	CFLR	Configuration Load Register (EEPROM or BIOS)	Vaux	Vaux	62
3F4h	IISR	ISA Interrupt Select Register	Vaux	Vaux	84
3F8h	TMS	Test Register - Reserved			
3FCh	SSVID	Subsystem ID register (read-only version at 32Ch)	Vaux	Vaux	50
400h	CLKCR1	Clock Control Register 1	Core	Ctrl	183
404h		Reserved			
408h - 40Ch		Reserved			
410h	FRR	Feature Reporting Register	Core	Ctrl	62
414h - 418h		Reserved			
41Ch	SLT12O	Slot 12 GPIO Output Register for AC Link	Core	Peri	220
420h	SERMC	Serial Port Master Control Register	Core	Peri	211
424h		Reserved			
428h	SERC1	Serial Port Configuration Register 1 - RO	Core	Peri	214
42Ch	SERC2	Serial Port Configuration Register 2 - RO	Core	Peri	214
430h - 458h		Reserved			
45Ch	SLT12M	Slot 12 Monitor Register for Primary AC Link	Core	Peri	221
460h	ACCTL	AC ‘97 Control Register	Core	Peri	215
464h	ACSTS	AC ‘97 Status Register	Core	Peri	216
468h	ACOSV	AC ‘97 Output Slot Valid Register	Core	Peri	216
46Ch	ACCAD	AC ‘97 Command Address Register	Core	Peri	218
470h	ACCDA	AC ‘97 Command Data Register	Core	Peri	218

Table 4. BA0 Register Map (Continued)

Address (BA0 Offset)	Name	Description	Power From	Int. Loc.	Page
474h	ACISV	AC ‘97 Input Slot Valid Register	Core	Peri	219
478h	ACSAD	AC ‘97 Status Address Register	Core	Peri	219
47Ch	ACSDA	AC ‘97 Status Data Register	Core	Peri	220
480h	JSPT	Joystick Poll/Trigger Register	Core	Peri	199
484h	JSCTL	Joystick Control Register	Core	Peri	200
488h - 48Ch		Reserved			
490h	MIDCR	MIDI Control Register	Core	Peri	195
494h	MIDCMD	MIDI Command Register - WO	Core	Peri	196
494h	MIDSR	MIDI Status Register - RO	Core	Peri	196
498h	MIDWP	MIDI Write Port	Core	Peri	197
49Ch	MIDRP	MIDI Read Port - RO	Core	Peri	197
4A0h - 4A4h		Reserved			
4A8h	AODSD1	AC ‘97 On-Demand Slot Disable for primary link - RO	Core	Peri	217
4ACh	AODSD2	AC ‘97 On-Demand Slot Disable for secondary link - RO	Core	Peri	217
4B0h	CFGI	Configuration Interface Register (EEPROM interface)	Core	Ctrl	223
4B4h		Reserved			
4BCh - 4D8h		Reserved			
4DCh	SLT12M2	Slot 12 Monitor Register 2 for Secondary AC Link	Core	Peri	222
4E0h		Reserved			
4E4h	ACSTS2	AC 97 Status Register 2	Core	Peri	222
4E8h - 4F0h		Reserved			
4F4h	ACISV2	AC 97 Input Slot Valid Register 2	Core	Peri	223
4F8h	ACSAD2	AC 97 Status Address Register 2	Core	Peri	223
4FCh	ACSDA2	AC 97 Status Data Register 2	Core	Peri	224
500h		Reserved			
504h		Reserved			
508h		Reserved			
50Ch		Reserved			
510h - 528h		Reserved			
52Ch		Reserved			
530h		Reserved			
534h - 53Ch		Reserved			
540h - 55Ch		Reserved			
560h - 57Ch		Reserved			
580h - 588h		Reserved			
58Ch		Reserved			
590h - 59Ch		Reserved			
600h		Reserved			
604h		Reserved			
608h		Reserved			
60Ch		Reserved			
610h - 6FCh		Reserved			
700h		Reserved			154

Table 4. BA0 Register Map (Continued)

Address (BA0 Offset)	Name	Description	Power From	Int. Loc.	Page
704h		Reserved			154
708h		Reserved			155
70Ch		Reserved			155
710h		Reserved			156
710h		Reserved			156
714h		Reserved			157
718h-72Ch		Reserved			
730h	FMSR	FM Synthesis Status Register - RO	Core	Ctrl	179
730h	B0AP	FM Bank 0 Address Port - WO	Core	Ctrl	179
734h	FMDP	FM Data Port	Core	Ctrl	180
738h	B1AP	FM Bank 1 Address Port	Core	Ctrl	180
73Ch	B1DP	FM Bank 1 Data Port	Core	Ctrl	180
740h	SSPM	Sound System Power Management	Core	Ctrl	167
744h	DACSR	DAC Sample Rate - Playback SRC	Core	Ctrl	168
748h	ADCSR	ADC Sample Rate - Capture SRC	Core	Ctrl	168
74Ch	SSCR	Sound System Control Register	Core	Ctrl	169
750h		Reserved			
754h	FMLVC	FM Synthesis Left Volume Control	Core	Ctrl	170
758h	FMRVC	FM Synthesis Right Volume Control	Core	Ctrl	170
75Ch	SRCSA	SRC Slot Assignments	Core	Ctrl	172
760h	PPLVC	PCM Playback Left Volume Control	Core	Ctrl	171
764h	PPRVC	PCM Playback Right Volume Control	Core	Ctrl	171
768h		Reserved			
76Ch		Reserved			
770h - 7ECh		Reserved			
7F0h - 7F8h		Reserved			
7FCh		Reserved			
800h-FFCh		Reserved			

Table 4. BA0 Register Map (Continued)

Notes:See separate functional description sections for register bit assignments.

7.2 BA1 Direct Memory Map

Address (BA1 Offset)	Name	Description
0000h - 03FFh	FIFO RAM	Audio Sample RAM Memory Block - FIFOs Logical Size: 256 x 32 bits (1 kbytes stereo double words)
0400h - D51Fh	Reserved	Reserved internal memory
D600h - FFFFh	Reserved	Reserved for future use

Table 5. BA1 Direct Memory Map

7.3 Config Space Map

The standard PCI Config space is mapped out in Table 3 on page 47. The registers below are further described on their listed pages. The vendor defined registers located between addresses E4h and FFh are write protected by the *Configuration Write Protection* register (CWPR) located at address EEh. In this register, the **CWP** bits must equal 4281h for the other vendor-defined registers to be writable. Any other value will make the registers between E4h and FFh read-only. The CWPR register (EEh-FFh) is always read/writable. The PCI config registers are mapped into the register space (BA0) between 300h and 3FFh at the same offset. For all Vaux-powered registers, see the register description for which bits are affected by the PCI **RST#** reset signal.

Offset	Name	Description	Power From	Size	Page
06h	PCIST	PCI Status	Core	word	66
34h	CAPPTR	Capabilities Pointer - RO	Core	byte	67
40h	CAPID	Capabilities ID - RO	Core	byte	67
41h	NIP	Next Item Pointer - RO	Core	byte	67
42h	PMC	Power Management Capabilities - RO	Core	word	68
44h	PMCS	Power Management Control/Status	Vaux	word	69
46h		PCI-PCI Bridge Support Extensions - RO	Core	byte	70
47h		Power Data - RO	Core	byte	70
48h-DFh	Reserved	Reserved			
E0h	CWPR	Configuration Write Protect Register	Vaux	dword	49
E4h	EPPMC	Extended PCI Power Management Control	Vaux	dword	74
E8h	GPIOR	GPIO Pin Interface Register	Vaux	dword	186
ECh	SPMC	Serial Port Power Management Control (& ASDIN2 enable)	Vaux	dword	212
F0h	CFLR	Configuration Load Register (EEPROM or BIOS)	Vaux	dword	62
F4h	IISR	ISA Interrupt Select Register (plus BIOS flags)	Vaux	dword	84
F8h	TMS	Test Register - Reserved			
FCh	SSVID	Subsystem ID register	Vaux	dword	50

Table 6. Config Space Map

8. START-UP CONFIGURATION AND STATUS

8.1 PCI Start-Up Configuration

PCI configuration of the CS4281 is generally handled by standard PCI setup BIOS code or operating system code. This code loads the standard configuration space with BA0 and BA1 address to allocate the memory needed by the CS4281. Other CS4281-specific registers exist in Config Space that BIOS or configuration code must set to let the CS4281 Driver and operating system know that those features exist in the system as configured. Those extra setup issues are enumerated below. Note that all configuration space data at 00-FFh in config space can also be read and, all locations above 44h, written in host memory space BA0:300h - BA0:3FFh. In config space, locations above 44h are write-protected by the CWPR register located at CFG:0EEh, which must be programmed to 4281h before any of the vendor defined configuration registers can be written.

- To write any of the vendor defined configuration registers (above CFG: 044h), write CWPR to 4281h

- Microsoft Win 95/98 OS requires a non-zero SSID and SSVID at PCI config. location 2Ch-2Fh. For the CS4281, this register can be loaded through PCI config locations FCh-FFh which will show up as read-only at 2Ch-2Fh. These can be loaded in 2 ways:
  - EEPROM bytes 1-4. If EEPROM byte 0 is 55h, the CS4281 will auto-load EEPROM bytes 1-4 into PCI config space FCh-FFh.
  - BIOS loads bytes at Config Space location FCh-FFh.

- CFLR bits indicate configuration data. The CS4281 supports the CS4280/4614/4622/4624 (Inky) 2 bytes, plus two additional bytes. The bytes are loaded into the CFLR register at PCI config. location F0h, where F0h and F1h are Inky's CFL1 and CLF2 respectively. Bytes at locations F2h and F3h are new to the CS4281. These four bytes can also be loaded via two methods. Note that the Inky CFL1 and CFL2 registers are combined and moved from their old BA0 locations of 414h and 418h, to the CS4281 CFLR register mapped to BA0:3F0.
  - EEPROM bytes 5-8. If EEPROM byte 0 is 55h, the CS4281 will auto-load EEPROM bytes 5-8 into PCI config space F0h-F4h.
  - BIOS can load the CFLR register (32 bits) at PCI Config location F0h-F4h directly.

- ISA interrupts. BIOS needs to indicate which ISA interrupt pins are connected and are available for use through the IISR register at PCI config. location F4h-F8h. Each ISA interrupt pin has a corresponding four bits that must be programmed to non-zero values if used. The BIOS code should load the actual interrupt number into the respective bits in IISR: **IRQA[3:0]**, **IRQB[3:0]**, **IRQC[3:0]**. The driver uses this information to decide which interrupts are connected. Setting these bits does not actually enable the ISA interrupt (still high impedance). The driver enables the ISA interrupt (to drive the pin) by have these bits non-zero and enabling the interrupt through the IIER register. IISR also contains three generic flags that are maintained through a $D3_{cold}$ transition. Note that Inky uses a coded configuration for ISA interrupt support through the CFL2 bits.

- V_{AUX} power. On the CS4281 V_{AUX} power is not auto-detected. The default configuration assumes that the **VAUX** power pin is attached to a core power supply that is not maintained in a $D3_{cold}$ state. If the **VAUX** pin is attached to a real V_{AUX} 3 V power supply, the following bits must be programmed:
  - PME# from $D3_{cold}$ supported. BIOS should set **IISR.VAUXS** which will show up as **PMC.PMD3C**.

  - V_{AUX} power requirements. BIOS should set **IISR.VAC[2:0]** to indicate how much current is pulled from the V_{AUX} supply in $D3_{cold}$. This current number includes any attached codecs and circuitry and is not just the CS4281.
  - Unknown bit. The **IISR.AUXP** bit is reflected in the **PMC.AUXP** bit. However, exact definition of this bit is unknown at this time.

- When finished writing the vendor defined config space, set CWPR = 0000h (disable further writes)

8.2 Sound System Start-Up Configuration

The following operations should be used in starting up the Sound System side, after PCI Configuration start-up. The following data does not address actually sending data to or from the codec.

- Remove reset from AC Link; **SPMC.RSTN** = 1
- If two codecs present, enable **ASDIN2** and set secondary codec ID; **SPMC.ASDIN2E** = 1, **SERMC.TCID[1:0]**
- Turn on Sound System clocks based on **ABITCLK**; CLKCR1 = 0030h
- Set enables for sections that are needed in the SSPM register:
  - Enable AC Link; **SSPM.ACLEN** = 1
  - Enable SRCs if needed; **SSPM.PSRCEN** = **SSPM.CSRCEN** = 1
  - Enable FM if needed; **SSPM.FMEN** = 1
  - Enable Joystick if needed; **SSPM.JSEN** = 1
- Wait for clock stabilization; **CLKCR1.DLLRDY** = **CLKCR1.CLKON** = 1
- Enable **ASYNC** generation; **ACCTL.ESYN** = 1
- Wait for Codec ready (**ACSTS.CRDY** / **ACSTS2.CRDY2**)
- Enable Valid Frame output (**ACCTL.VFRM**) on ASDOUT
- Wait until Codec calibration is finished; Codec register 26h (lower 4 bits)
- Configure/Initialize Codec mixer registers; both codecs written through ACCAD/ACCDA + ACCTL
  - Primary Codec registers read from ACSAD/ACSDA
  - Secondary Codec registers read from ACSAD2/ACSDA2
- Initialize CS4281 volume registers for 0 dB, unmuted; FMxVC = PPxVC = 0000h
- Initialize any GPIO register use; GPIOR
- Initialize AC-Link features:
  - Variable Sample-Rate support; **SERMC.ODSEN1** / **SERMC.ODSEN2**
  - PME# generation from GP_INT bit (**SLT12M.GP_INT**/**SLT12M2.GP_INT**); **SPMC.GIPPEN** / **SPMC.GISPEN**
  - Interrupt generation from GP_INT bit; **HIMR.GPPIM** / **HIMR.GPSIM** (setting bit disables function)
  - Slot 12 Output bits; SLT12O
- Initialize SSCR register features
  - Initialize **DACZ** and **PSH**
  - Enable Hardware Volume Control if desired; **SSCR.HVC**
  - Setup which Codec and registers affected by hardware volume; **MVCS, MVLD, MVAD, MVMD**
  - If hardware volume enabled, write the Codec master volume register to initialize the CS4281 hardware volume control.
- Maybe, initialize all FIFO sizes and offsets (must be globally managed).

8.2.1 Sound System Playback

The following DMA sequence is needed when setting up a stream for playback. It assumes that the FIFO sizes are pre-setup since size and offset (FCRn register) must be initialized and managed across all FIFOs.

- Find unused DMA/FIFO - check **DMRn.DMA**, **DMRn.POLL**
- Set DMAn Mask bit; **DCRn.MSK**
- Program DMAn for DMA operation and set data format; DMRn
- Program DMAn for Auto Initialize and set transfer type control to Read Transfer; DMRn
- Program DMAn Base Address, and Base Count Register; DBAn, DBCn
- Program AC-Link Slot ID to attach FIFOm to AC-Link; FCRm
- If Playback SRC needed (and unused), enable and program for same slots.
    - Set Playback SRC sample frequency; DACSR
    - Set AC-Link slots assigned to PSRC (same as FIFOm); SRCSA
    - Enable Playback SRC; **SSPM.PSRCEN**
- If needed, make sure codec DACs are powered up; ACCAD, ACCDA, ACCTL - ACSAD, ACSDA
    - If powering up DACs, make sure calibration is finished; Codec 26h, DAC bit
- Setup interrupts; DCRn (TC/half TC/or both for ping-pong buffer), HIMR, HICR
- Enable FIFOn; **FCRn.FEN**
- Enable AC-Link Output Slot Valid Bits; ACOSV

8.2.2 Sound System Record

The following DMA sequence is needed when setting up a stream for recording. It assumes that the FIFO sizes are pre-setup since size and offset (FCRn register) must be initialized and managed across all FIFOs.

- Find unused DMA/FIFO - check **DMRn.DMA**, **DMRn.POLL**
- Set DMAn Mask bit; **DCRn.MSK**
- Program DMAn for DMA operation and data format; DMRn register
- Program DMAn for Auto Initialize and set Transfer Type to Write Transfer; DMRn register
- Program DMAn Base Address, and Base Count Register; DBAn, DBCn
- Setup interrupts; DCRn (TC/half TC/or both for ping-pong buffer), HIMR, HICR
- If needed, make sure codec ADCs are powered up; ACCAD, ACCDA, ACCTL - ACSAD, ACSDA
    - If powering up ADCs, make sure calibration is finished; Codec 26h, ADC bit
- If Capture SRC needed (and unused), enable and program for same slots.
    - Set Capture SRC sample frequency; ADCSR
    - Set AC-Link slots assigned to CSRC (same as FIFOn); SRCSA
    - Enable Capture SRC; **SSPM.CSRCEN**
- Program AC-Link Slot ID to attach FIFOn to AC-Link; FCRn
- Enable FIFOn; **FCRn.FEN**

8.2.3 6-Channel Dual-Codec Playback Setup Example

The following example, illustrated in Figure 16, describes the codec register configuration needed for a dual-codec system where the primary codec is a standard AC ‘97 codec, such as the CS4297/CS4297A, and the secondary codec is a quad-channel CS4294. The primary codec supports the standard stereo output stream and the secondary codec supports the other four channels. The *Sound System Start-Up* and the *Sound System Playback* sections describe how to set-up the CS4281. The CS4281 must be configured to use three streams where AC-Link slots 3 and 4 are left and right channels, slots 7 and 8 are left and right surround channels, and slots 6 and 9 are center and LFE channels. Three CS4281 FIFOs must be mapped to the above mentioned slot pairs and the appropriate six output slot valid bits (ACOSV) must be set when everything else is configured. The following list describes what registers inside the AC ‘97 codecs need to be configured to support playback of 6-channels (3 streams) through the two codecs.

- Make sure primary codec (CS4297/CS4297A) is ready; **ACSTS.CRDY** set
- Make sure calibration finished; Read primary codec Index 26h, **REF** = **ANL** = **DAC** = 1
  - If **ACSTS.VSTS** set, read ACSAD and ACSDA to clear **VSTS**
  - Write Command register index; ACCAD = 0026h
  - Send read request down AC Link; Set **ACCTL.CRW** and **ACCTL.DCV**
  - When **ACSTS.VSTS** set, read ACSAD = 26h (index) and ACSDA = xxxEh/xxxFh (**REF**, **ANL**, **DAC** set)
- Unmute and set volume for Slot 3/4 DACs; Write primary codec Index 18h = 0808 (0 dB)
  - Write Command register index; ACCAD = 0018h
  - Write Data register; ACCDA = 0808h
  - Send write request down AC link; Clear **ACCTL.CRW**, set **ACCTL.DCV**

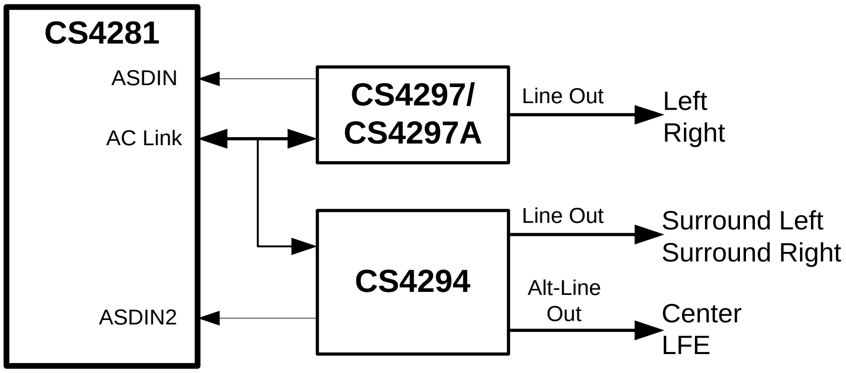


Figure 16. 6-Channel Playback Block Diagram

- Unmute and set Master Line Out Volume; Write primary codec Index 02h = 0000h (0 dB)
  - Write Command register index; ACCAD = 0002h
  - Write Data register; ACCDA = 0000h
  - Send write request down AC link; Clear **ACCTL.CRW**, set **ACCTL.DCV**

- Make sure secondary codec (CS4294) is ready; **ACSTS2.CRDY2** set

- Make sure calibration finished; Read secondary codec Index 26h, **REF** = **ANL** = **DAC** = 1
  - If **ACSTS2.VSTS2** set, read ACSAD2 and ACSDA2 to clear **VSTS2**
  - Write Command register index; ACCAD = 0026h
  - Send read request down AC Link to secondary codec; Set **ACCTL.CRW**, **ACCTL.TC**, and **ACCTL.DCV**
  - When **ACSTS2.VSTS2** set,
    read ACSAD2 = 26h (index) and ACSDA2 = xxxEh/xxxFh (**REF**, **ANL**, **DAC** set)

- Set CS4294 to Mode 3 and bypass mixer (DDM); Write secondary codec Index 5Eh = 0103h
  (In general, the output mixer would not be used on the surround channels)
  - Write Command register index; ACCAD = 005Eh
  - Write Data register; ACCDA = 0103h
  - Send write request down AC link to 2nd codec; Clear **ACCTL.CRW**, and set **ACCTL.TC**, **ACCTL.DCV**

- Unmute and set Left/Right Surround Volume (slots 7/8); Write 2nd codec Index 38h = 0000h (0 dB)
  - Write Command register index; ACCAD = 0038h
  - Write Data register; ACCDA = 0000h
  - Send write request down AC link to 2nd codec; Clear **ACCTL.CRW**, and set **ACCTL.TC**, **ACCTL.DCV**

- Unmute and set LFE and Center Volume (slots 9/6); Write secondary codec Index 36h = 0000h (0 dB)
  - Write Command register index; ACCAD = 0036h
  - Write Data register; ACCDA = 0000h
  - Send write request down AC link to 2nd codec; Clear **ACCTL.CRW**, and set **ACCTL.TC**, **ACCTL.DCV**

8.3 Feature and Configuration Reporting

The Feature Reporting register (FRR) and Configuration Load register (CFLR) contain feedback bits for host software based detection of various device and device family configuration options.

8.3.1 Feature Reporting Register (FRR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
											MM2	MM1	MM0		

Address: BA0: 410h, Read-Only

Default: 00000000h

Definition: Core powered. Provides a host port for reading of device ID.

Bit Descriptions:

MM[2:0] This bit field reports the version for the CS4281.
0 0 0 = Revision A

8.3.2 Configuration Load Register (CFLR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CB37	CB36	CB35	CB34	CB33	CB32	CB31	CB30	CB27	CB26	CB25	CB24	CB23	CB22	CB21	CB20
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CB17	CB16	CB15	CB14	CB13	CB12	CB11	CB10	CB07	CB06	CB05	CB04	CB03	CB02	CB01	CB00

Address: BA0: 3F0h, Read-Write
PCI CFG: 0F0h, Read-Write if CWPR configured, otherwise Read-Only

Default: 00000001h

Definition: Vaux powered. The Configuration Load Register provides a host port for reading of the first byte of device configuration options from EEPROM (or BIOS pre-load). This register is unaffected by the PCI **RST#** signal. The default value is set by a Vaux POR signal. The BIOS can pre-load this register by writing to it in configuration space. The following bit descriptions are for driver information only as these bits have no direct hardware affect.

Bit Descriptions:

CB0[7:0] This bit field returns the first configuration byte. No bits have been assigned yet.

CB1[7:0] This bit field returns the second configuration byte. No bits have been assigned yet.

CB2[7:0] This bit field returns the third configuration byte. No bits have been assigned yet.

CB3[7:0] This bit field returns the fourth configuration byte. No bits have been assigned yet.

9. PCI POWER MANAGEMENT

The CS4281 supports the *PCI Bus Power Management Interface Specification*, version 1.1.

9.1 Power Management States

The PCI power management specification defines four major power states: D0 (fully on), D1, D2, and D3 (fully off). The D3 state is divided into two substates, D3$_{hot}$ and D3$_{cold}$; D3$_{cold}$ differs from D3$_{hot}$ in that the normal PCI bus V$_{cc}$ signals (not to be confused with the V$_{AUX}$ signal) are turned off. The D0 state is also divided into two sub-states, D0$_{active}$ and D0$_{uninitialized}$; the D0$_{unitialized}$ state describes a device that has just received a PCI **RST#** signal and has not yet been programmed (and therefore it is not consuming full power), while the D0$_{active}$ state describes a device that has been programmed and is fully operational (and therefore consumes full power). The CS4281 can be in the D0$_{uninitialized}$ state when initial power is applied, or coming from a D3 state. In D3$_{cold}$, **VAUX** supplies power to allow power management events to occur where the rest of the chip has power removed. The **RST#** signal will be active during this state and cannot reset the registers and logic powered by **VAUX**.

The CS4281 supports all of the power states defined in the PCI power management specification, and provides bits in the EPPMC register that determine what gets powered down when in a particular state. The default is to have the CS4281 do no power management when changing between states; thereby having all power management controlled through software. Assuming the CS4281 is programmed to manage power automatically, the following would be a typical configuration:

The DLL refers to the CS4281 internal clock derived from the AC Link **ABITCLK** signal. See the *Clock Management* section for full details. In D1 the DLL and Joystick are powered off but the AC Link is still active. In D2, host software would also power the AC Link down by programming the Codec into a PR4 state. This would kill ABITCLK and force the Codec into a fairly low-power mode. In D3$_{hot}$, the CS4281 is placed in its lowest power state where the entire chip, except for Config space accesses and the BIU) is powered down and all registers are reset to their default state. The Codec is also held in reset using the **ARST#** pin. In D3$_{cold}$, VDD power is removed from the PCI bus and **VAUX** power may - **optionally** - be available. The CS4281 and Codec are held in reset by the PCI **RST#** pin. By default, the power management bits indicating support for D3$_{cold}$ and **VAUX** are false.

State	EPPMC Register Bits	Interpretation
D0$_{active}$	not applicable	DLL running: Normal operation
D0$_{unintialized}$	not applicable	DLL off, AC Link down: reset defaults
D1	D1DLL, D1JSD	DLL off: maintain context
D2	D2JSD, PR4*	DLL off, AC Link powered down: maintain context
D3$_{hot}$	D3FPD, D3ART	Chip in full reset (except config space): context lost
D3$_{cold}$	doesn’t matter	V$_{cc}$ off, V$_{AUX}$ active: context lost

* PR4 requires Host software to write a Codec Register to bring AC Link down. Brought back up through **SPMC.ASYN**.

Table 7. Power Management Typical Configuration

BIOS code must initialize these bits, through the IISR register, if VAUX power is available.

The following diagram is taken from the PCI power management specification. The transitions between states have been labeled with the actions taken by the CS4281 as mentioned in the above Table, where the CS4281 is automatically controlling the power to the CS4281.

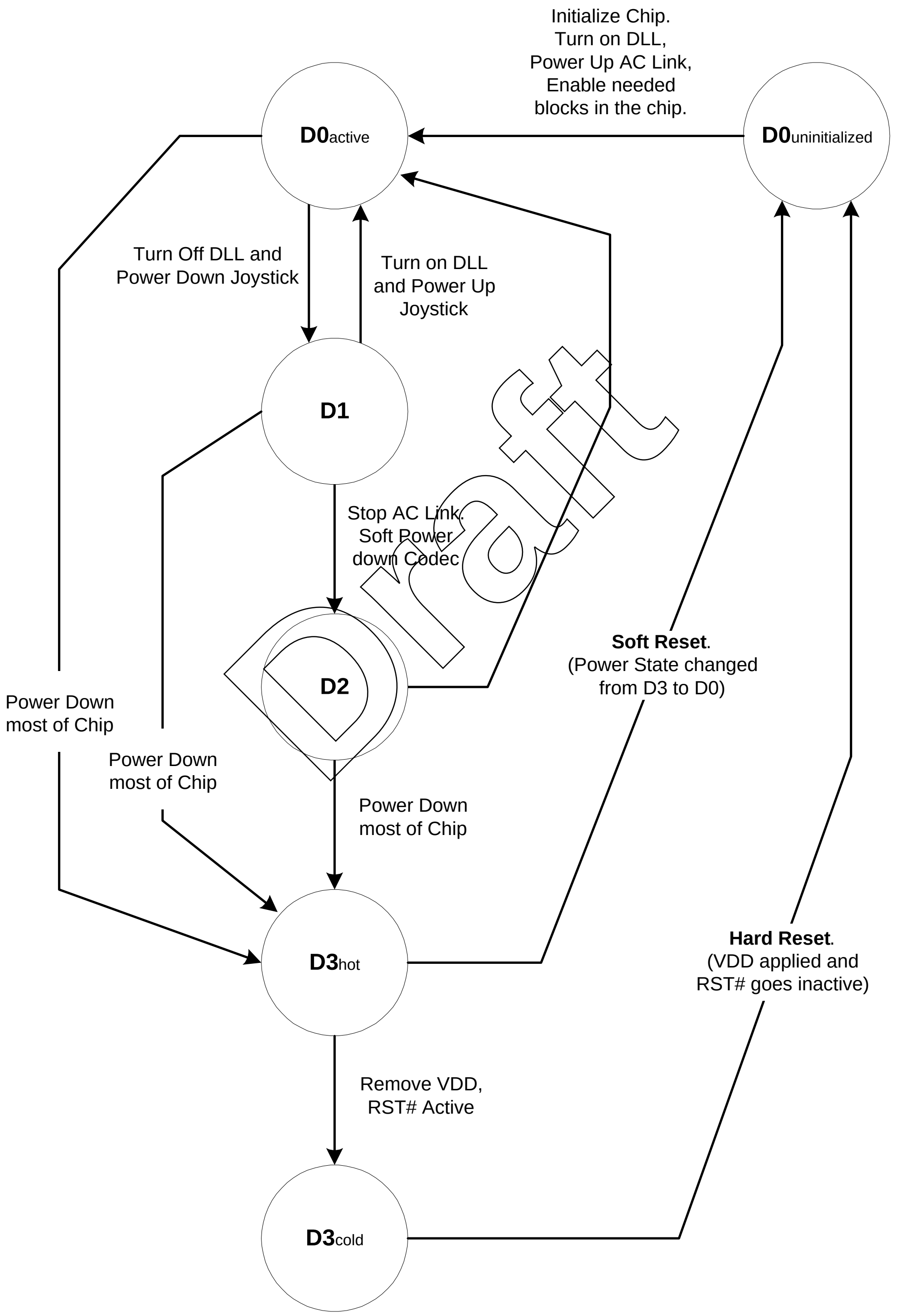


Figure 17. Typical Power State Diagram

The PCI clock can be stopped any time there is no PCI bus activity. There are rules governing the start-up and shutdown of the **PCICLK** signal relative to the beginning and end of any PCI bus transaction. The PCI clock control is accomplished through the **CLKRUN#** signal. The DLL running statement means that the DLL will be active and locked to the AC Link **ABITCLK** signal. It is possible, and likely, that the **ABITCLK** will be turned off in the D1 and D2 states. If this happens, the DLL will automatically powerdown until the **ABITCLK** resumes. Software controlling the power management functions must allow adequate time for the clock system to stabilize following any power up event. The DLL will stabilize within 30 **ABITCLK** periods and a stable DLL is reflected in **CLKCR1.DLLRDY**.

9.2 Modem Wakeup

This section defines how the CS4281 can respond to a wakeup event from the modem. Depending on the operating configuration, the CS4281 can notify the host using either the **PME#** pin or PCI interrupt. The AC ‘97 Codec will notify the CS4281 by using either Slot 12 status bits when the AC Link is active, or the **ASDIN** and **ASDIN2** pins when the AC Link is powered down. See the *PME# Assertion* section for additional information.

If the CS4281 and the Codec are fully programmed and powered ($D0_{active}$), the CS4281 can monitor Slot 12 and shadow the status bits in two registers: SLT12M for the **ASDIN** line, and SLT12M2 for the **ASDIN2** line. The GPIOR register logic can trap the **GP_INT** status from the slot 12 registers above. When **GP_INT** changes from a 0 to a 1, the GPIOR logic can store the change in two status bits: **GPIOR.GPPS** for the primary codec and **GPIOR.GPSS** for the secondary codec. These bits can generate a PCI interrupt which is maskable through the HICR register. The interrupt is cleared by writing a 0 to the **GPPS**/**GPSS** bit causing the interrupt. This only clears the local (CS4281) interrupt. The actual condition causing the interrupt must be cleared through the GPIO Sticky register (Index 50h) on the particular AC ‘97 Codec, which will then clear the **GP_INT** bit the slot 12. The host driver is free to use either the interrupt or poll the slot 12 shadow register. The discussions below assume the PME enable bit, **PMEEN**, in PCI config space at 0x44, is set. **GP_INT** going from 0 to 1 also generates a PME signal. PME generation can be blocked in the D0 (active) state by setting **EPPMC.PMD0D**.

In the D1 power state, the CS4281 can monitor Slot 12 and assert **PME#** when the **GP_INT** goes active (SLT12M or SLT12M2).

In the D2 power state (assuming the AC Link is down), the CS4281 can monitor **ASDIN**, **ASDIN2**, or a CS4281 EGPIO pin and assert **PME#** when either pin goes high.

CS4281 in $D3_{hot}$ with Codec powered down (PR4): The **ASDIN**, **ASDIN2**, or a CS4281 EGPIO pin can signal the wakeup event to the CS4281 and the CS4281 will assert **PME#**.

Assuming V_{AUX} power exists and the CS4281 in $D3_{cold}$ with the codec powered down. The CS4281 wakeup logic is powered by **VAUX** and unaffected by the PCI reset signal **RST#** (which will be active). A rising edge on **ASDIN** or **ASDIN2** will cause the **PME#** pin to be asserted active low and set the PME# status bit, **PMES** (sticky) in the config space at 0x44. (If configured properly, **GPIO1**, **VOLUP**, and **VOLDN** can also cause the assertion of **PME#**.) The **VAUX** supply current can only draw 20 mA maximum. If this is exceeded, when **PMCS.PMEEN** is clear, the PME logic must be disabled to get power under 20 mA (from *Instantly Available PC Design Guide*, pp24).

9.3 Register Interface

The registers controlling PCI power management are located in configuration space, and their contents are fixed by the specification (except for EPPMC).

The first register involved is the standard 16-bit status register at address 0x06:

9.3.1 PCI Status (PCIS)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERR		RMA	RTA	STA	DVSL1	DVSL0	DPED	FBBC	UDF	66MHz	NC	R3	R2	R1	R0

Address: PCI CFG: 06h, Read-Write
BA0: 306h, Read Only

Default: 0210h

Definition: Core powered. This register holds PCI bus interface status information.

Bit Descriptions:

R[3:0] Reserved: Writes are ignored, reads return zeros.

NC New Capabilities: Read-only bit set to 1 to indicate the CS4281 supports the standard configuration space capabilities extension mechanism based on linked lists of register blocks; this extension mechanism is used by the PCI power management specification.

66MHz 66 MHz Operation: Read-only bit set to 0. The CS4281 doesn't support 66 MHz operation.

UDF User Definable Features: Read-only bit set to 0, indicating the CS4281 doesn't support user definable features.

FBBC Fast Back-to-Back Capable: Read-only bit set to 0, indicating the CS4281 doesn't support fast back-to-back transactions.

DPED PERR# Detected: Set to 1 when the CS4281 detects the **PERR#** signal asserted (including when it asserts **PERR#** itself). Writing a 1 to this bit clears it.

DVSL[1:0] **DEVSEL#** Timing: Read-only bits set to 0x1, indicating the CS4281 uses medium **DEVSEL#** timing.

STA Signalled Target Abort: Set to 1 when the CS4281 terminates a transaction with target-abort. Writing a 1 to this bit clears it. Note that this bit will always be 0 since the CS4281 does not generate target abort transactions.

RTA Received Target Abort: Set to 1 when a CS4281 master transaction is terminated with a target-abort. Writing a 1 to this bit clears it.

RMA Received Master Abort: Set to 1 when a CS4281 master transaction is terminated with a master-abort. Writing a 1 to this bit clears it.

PERR Detected Parity Error: Set to 1 when the CS4281 detects a parity error. Writing a 1 to this bit clears it.

The next configuration space register involved is the 8-bit capabilities pointer at offset 0x34:

9.3.2 Capabilities Pointer

7	6	5	4	3	2	1	0
O7	O6	O5	O4	O3	O2	O1	O0

Address: PCI CFG: 34h, Read-Only
BA0: 334h, Read-Only

Default: 40h

Definition: This register holds an offset into the configuration space where the first item in the capabilities linked list is located.

Bit Descriptions:

O[7:0] Offset: Offset into configuration space where the first item in the capabilities linked list is located. Reads return 0x40.

The PCI power management capabilities register block contains six registers. The first register, at configuration space offset 0x40, identifies the register block as a PCI power management block:

9.3.3 Capabilities ID

7	6	5	4	3	2	1	0
ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

Address: PCI CFG: 40h, Read-Only
BA0: 340h, Read-Only

Default: 01h

Definition: This register holds a value identifying the register block as belonging to PCI power management.

Bit Descriptions:

ID[7:0] ID: The value 0x01 identifies a PCI power management register block.

The next register, at configuration space offset 0x41, contains the offset to the next capabilities register block. In the CS4281 there are no additional capabilities register blocks, so this register contains the value zero, flagging the end of the linked list.

9.3.4 Next Item Pointer

7	6	5	4	3	2	1	0
O7	O6	O5	O4	O3	O2	O1	O0

Address: PCI CFG: 41h, Read-Only
BA0: 341h, Read-Only

Default: 00h

Definition: This register holds an offset to the next capabilities register block.

Bit Descriptions:

O[7:0] Offset: Zero signals the end of the capabilities linked list.

The next register, at configuration space offset 0x42, describes the CS4281's power management capabilities:

9.3.5 *Power Management Capabilities (PMC)*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMD3C	PMD3H	PMD2	PMD1	PMD0	D2S	D1S	VAC2	VAC1	VAC0	DSI	AUXP	PMEC	V2	V1	V0

Address: PCI CFG: 42h, Read-Only
BA0: 342h, Read-Only

Default: 7E22h

Definition: Core powered. This register holds power management capabilities information.

Bit Descriptions:

V[2:0] Version: The value 2 indicates compliance with version 1.1 of the *PCI Bus Power Management Interface Specification*.

PMEC PME Clock: Set to 0 to indicate the PCI bus clock can be stopped for the CS4281 to generate **PME#**.

VAC[2:0] Vaux Current. These bits are set from **IISR.VAC[2:0]** which must be initialized by the BIOS to indicate how much current Vaux pulls. Note this is total current and is the combined CS4281 and any attached Codecs and external logic using Vaux.
000 - 0 mA (self powered/don't support Vaux)
001 - 55 mA
010 - 100 mA
011 - 160 mA
100 - 220 mA
101 - 270 mA
110 - 320 mA
111 - 375 mA (spec maximum)

DSI Device Specific Initialization: Set to 1 to indicate the CS4281 needs to be programmed to be useful. Note that Microsoft operating systems don't use this bit.

AUXP Auxiliary Power. This bit is set from **IISR.AUXP**. Not sure what this bit does yet.

D1S D1 Support: Set to 1 to indicate the CS4281 supports the D1 power management device state.

D2S D2 Support: Set to 1 to indicate the CS4281 supports the D2 power management device state.

PMD0 PME# Support: Set to 1. CS4281 can generate **PME#** from D0 power state.

PMD1 PME# Support: Set to 1. CS4281 can generate **PME#** from D1 power state.

PMD2 PME# Support: Set to 1. CS4281 can generate **PME#** from D2 power state.

PMD3H PME# Support: Set to 1. CS4281 can generate **PME#** from $D3_{hot}$ power state.

PMD3C PME# Support: BIOS settable through **IISR.VAUXS**. When set, the CS4281 can generate **PME#** from $D3_{cold}$ power state. Note that **VAUX** must be powered in the $D3_{cold}$ state for this bit to be true.

The next register, at configuration space offset 0x44, controls the CS4281's power management state and indicates its current state.

Power Management Control/Status (PMCS)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMES	DSC1	DSC0	DSL3	DSL2	DSL1	DSL0	PMEEN	R5	R4	R3	R2	R1	R0	PS1	PS0

Address: PCI CFG: 44h, Read-Write
BA0: 344h, Read-Write

Default: 0000h

Definition: Vaux powered. This register controls the power management device state and indicates the current state. The **PS** bits are reset by the PCI VDD POR circuit. All other bits are reset by Vaux POR.

Bit Descriptions:

PS[1:0] Power State: When written, indicates the power state the CS4281 should enter. When read, indicates the CS4281's current power state. Note D3_{cold} is not listed since this register is not readable in a D3_{cold} state. These bits are reset by the PCI VDD POR signal - NOT Vaux POR.
0 0 = D0 (reset default)
0 1 = D1
1 0 = D2
1 1 = D3_{hot}

R[5:0] Reserved. Writes are ignored and reads return zeros.

PMEEN PME# Assertion Enable: When written, enables or disables the capability to assert **PME#**. When read, indicates the current **PME#** assertion capability. When **IISR.VAUXS** = 1**,** the **PMEEN** bit is unaffected by the PCI **RST#** signal. When **IISR.VAUXS** = 0**,** the **PMEEN** bit is reset by the PCI **RST#** signal. The default value is set by a Vaux POR signal.
0 = **PME#** assertion disabled (POR default)
1 = **PME#** assertion enabled

DSL[3:0] Data Select: Not supported on the CS4281. Writes are ignored and reads return 0.

DSC[1:0] Data Scale: Not supported on the CS4281. Writes are ignored and reads return 0.

PMES PME# Status: Indicates whether or not the CS4281 would be asserting **PME#**, regardless of the state of the **PMEEN** bit (if a condition occurs where the CS4281 should assert **PME#**, the **PMES** bit is set but actual assertion of the **PME#** signal is gated by the **PMEEN** bit). Writing a 1 to this bit clears it. When **IISR.VAUXS** = 1**, PMES** is unaffected by the PCI **RST#** signal. When **IISR.VAUXS** = 0**, PMES** is reset by the PCI **RST#** signal. The default value is set by a Vaux POR signal.
0 = **PME#** not asserted (POR default)
1 = **PME#** asserted

The next register applies only to PCI-PCI bridges and is not used by the CS4281.

9.3.6 PCI-PCI Bridge Support Extensions

7	6	5	4	3	2	1	0
BPCCE	B2B3	R5	R4	R3	R2	R1	R0

Address: PCI CFG: 46h, Read-Only
BA0: 346h, Read-Only

Default: 00h

Definition: This register indicates the capabilities of a PCI-PCI bridge. It is not supported by the CS4281.

Bit Descriptions:

R[5:0] Reserved. Reads return zeros.

B2B3 B2/B3 Support for $D3_{hot}$: Reads return zeros.

BPCCE Bus Power/Clock Control Enable: Reads return zeros.

The last PCI standard power management register is part of a facility allowing a device to describe its power consumption in the power management states it supports. The CS4281 doesn’t support this register.

9.3.7 Power Data

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Address: PCI CFG: 47h, Read-Only
BA0: 347h, Read-Only

Default: 00h

Definition: This register is used by devices to return data describing their power consumption in the device states it supports. It is not supported in the CS4281.

Bit Descriptions:

D[7:0] Data. Reads always return zeros.

9.3.8 Extended PCI Power Management Control (EPPMC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D3FMD	D3JSD	D3ALD	D3SWC	D3DLL	D3ART	D3FPD		D2FMD	D2JSD	D2ALD	D2SWC	D2DLL	D2ART		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMD0D	FPDN					PS1	PS0	D1FMD	D1JSD	D1ALD	D1SWC	D1DLL	D1ART	res	GWU

Address: BA0: 3E4h, Read-Write (2 bits read-only)
PCI CGF: 0E4h, Read-Write if CWPR configured, otherwise Read-Only

Default: 00000000h

Definition: Vaux powered. PCI power management interface control for software, full power down, and D1 through D3 power management options. The **D1xxx**, **D2xxx**, and **D3xxx** bits determine how the CS4281 automatically responds to power management events set through the **PMCS.PS[1:0]** bits in the config space. If all these bits are clear (default), the CS4281 does nothing on power event changes; therefore, host software has total control over power management. Reset to default state by Vaux POR. Additionally **FPDN** is reset by the PCI **RST#** signal.

Bit Descriptions:

GWU Generate Wake Up: A 0 to 1 transition in this bit generates a wake-up event, just as the AC '97 and extended GPIO pins generate wake-up events; the **PMES** bit in the power management control/status register will be set, and, if **PME#** assertion is enabled via the **PMEEN** bit in the power management control/status register, the external **PME#** signal will be asserted. **GWU** reads as 1 until system or device driver software clears the **PMES** bit in the power management control/status register. Writes:
0 = Don't generate a wake-up event (reset default)
0 to 1 transition = Generate a wake-up event

D1ART D1 **ARST#** enable. When the **PS[1:0]** bits are in the D1 state and **D1ART** is set, the **ARST#** pin, going to the AC Link, is driven active similar to setting **SPMC.RSTN** = 0.

D1DLL D1 DLL disable. When the **PS[1:0]** bits are in the D1 state and **D1DLL** is set, the DLL is turned off (powered down) similar to **CLKCR1.DLLP** = 0.

D1SWC D1 Software Clock disable. When the **PS[1:0]** bits are in the D1 state and **D1SWC** is set, the Core clock is turned off similar to **CLKCR1.SWCE** = 0.

D1ALD D1 AC Link Disable. When the **PS[1:0]** bits are in the D1 state and **D1ALD** is set, the AC Link logic is powered down similar to **SSCR.ACLEN** = 0.

D1JSD D1 Joystick Disable. When the **PS[1:0]** bits are in the D1 state and **D1JSD** is set, the joystick logic is powered down similar to **SSCR.JSEN** = 0.

D1FMD D1 FM Disable. When the **PS[1:0]** bits are in the D1 state and **D1FMD** is set, the FM block is powered down similar to **SSCR.FMEN** = 0.

PS[1:0] Read Only bits that indicate the CS4281's current power state. These are read only versions of the bits by the same name in the config space 0x44, *Power Management Control/Status*
0 0 = D0 (reset default)
0 1 = D1
1 0 = D2
1 1 = $D3_{hot}$

FPDN Full Power-Down. When set, the entire chip (with the exception of the Configuration Space) is reset. All registers, except for config space, get reset to their power-on default values. This includes all DMA, FIFO, clock control, Serial Port, and Sound System registers. **FPDN** is logically OR'd with the POR circuit for all blocks except the Configuration Space registers. When **FPDN** goes from a 1 to a 0, the EEPROM engine goes out and looks for the existence of an EEPROM identical to a POR condition. Resetting all registers places the CS4281 in its lowest power state, since nothing is enabled. Clearing **SPMC.RSTN** is also recommended to place the Codec on the AC Link in its lowest power state. When **FPDN** is set, reads from any register in any of the powered-down blocks produces indeterminate results and writes go into the bit bucket. Unlike the other bits in this register, **FPDN** is reset by the PCI **RST#** signal.

PMD0D PME# D0 Disable. This bit controls the generation of **PME#** when in the D0 state. **GWU** can always generate a **PME#** event. **PMED0D** blocks **PME#** events in D0 for all other sources: extended GPIO pins, **GP_INT** bits in Slot 12, and AC link powerdown conditions. This bit is cleared by **RST#** which forces the CS4281 into $D0_{\text{uninitialized}}$ state where **PME#** generation is required from all sources. See the *PME# Assertion* section for more details.
0 **- PME#** generated from all sources when in D0 (active) state
1 **- PME#** generated from **GWU** only (software) when in the D0 state.

D2ART D2 **ARST#** enable. When the **PS[1:0]** bits are in the D2 state and **D2ART** is set, the **ARST#** pin, going to the AC Link, is driven active similar to setting **SPMC.RSTN** = 0.

D2DLL D2 DLL disable. When the **PS[1:0]** bits are in the D2 state and **D2DLL** is set, the DLL is turned off (powered down) similar to **CLKCR1.DLLP** = 0.

D2SWC D2 Software Clock disable. When the **PS[1:0]** bits are in the D2 state and **D2SWC** is set, the Core clock is turned off similar to **CLKCR1.SWCE** = 0.

D2ALD D2 AC Link Disable. When the **PS[1:0]** bits are in the D2 state and **D2ALD** is set, the AC Link logic is powered down similar to **SSCR.ACLEN** = 0.

D2JSD D2 Joystick Disable. When the **PS[1:0]** bits are in the D2 state and **D2JSD** is set, the joystick logic is powered down similar to **SSCR.JSEN** = 0.

D2FMD D2 FM Disable. When the **PS[1:0]** bits are in the D2 state and **D2FMD** is set, the FM block is powered down similar to **SSCR.FMEN** = 0.

D3FPD D3 Full Power Down. When the **PS[1:0]** bits are in the D3 state and **D3FPD** is set, the full chip powers down (except for config space) similar to setting **EPPMC.FPDN** = 1.

D3ART D3 **ARST#** enable. When the **PS[1:0]** bits are in the D3 state and **D3ART** is set, the **ARST#** pin, going to the AC Link, is driven active similar to setting **SPMC.RSTN** = 0.

D3DLL D3 DLL disable. When the **PS[1:0]** bits are in the D3 state and **D3DLL** is set, the DLL is turned off (powered down) similar to **CLKCR1.DLLP** = 0.

D3SWC D3 Software Clock disable. When the **PS[1:0]** bits are in the D3 state and **D3SWC** is set, the Core clock is turned off similar to **CLKCR1.SWCE** = 0.

D3ALD D3 AC Link Disable. When the **PS[1:0]** bits are in the D3 state and **D3ALD** is set, the AC Link logic is powered down similar to **SSCR.ACLEN** = 0.

D3JSD D3 Joystick Disable. When the **PS[1:0]** bits are in the D3 state and **D3JSD** is set, the joystick logic is powered down similar to **SSCR.JSEN** = 0.

D3FMD D3 FM Disable. When the **PS[1:0]** bits are in the D3 state and **D3FMD** is set, the FM block is powered down similar to **SSCR.FMEN** = 0.

9.4 PME# Assertion

The PCI power management **PME#** signal is used to signal a generic power management event to the system's chip set. If the Extended GPIO pins are setup to generate a modem wake-up event (**PME#** assertion), the pins must be connected to a *Ring Discriminator* to be compliant with Microsoft; otherwise, false triggering will occur. **PME#** is an open-drain signal. Assertion of **PME#** can theoretically signal a device's desire to be placed in either a higher or lower power management device state, but for all practical purposes (read: Microsoft operating systems), **PME#** signals the chip-set that a device wants to be placed in the D0 state to handle some external wake-up event. The CS4281 can assert **PME#** from all (D0, D1, D2, $D3_{hot}$, and $D3_{cold}$ - when Vaux is available) power management device states in response to the following events:

- A low-to-high transition on the **ASDIN** or **ASDIN2** pin when the AC link is down. Codecs compliant with the AC '97 2.0 specification use this mechanism to signal a wake-up event to the AC '97 controller.
- **GP_INT** being set in SLT12M or SLT12M2 when the AC Link is powered up. See Figure 52 in the *General Purpose Input/Output Pins* section for conceptual logic.
- A GPIO pin configured to generate wake-up events does exactly that. This is intended for use with modem DAA circuits attached to the CS4281's extended GPIO pins. Note that **GPIO3** can only generate a **PME#** event when PCI power is applied while **GPIO1**, **VOLUP**, **VOLDN** can generate a **PME#** event from any power state (powered from **VAUX**). See Figures in the *General Purpose Input/Output Pins* section for conceptual logic.
- **EPMC.GWU** set by host software.

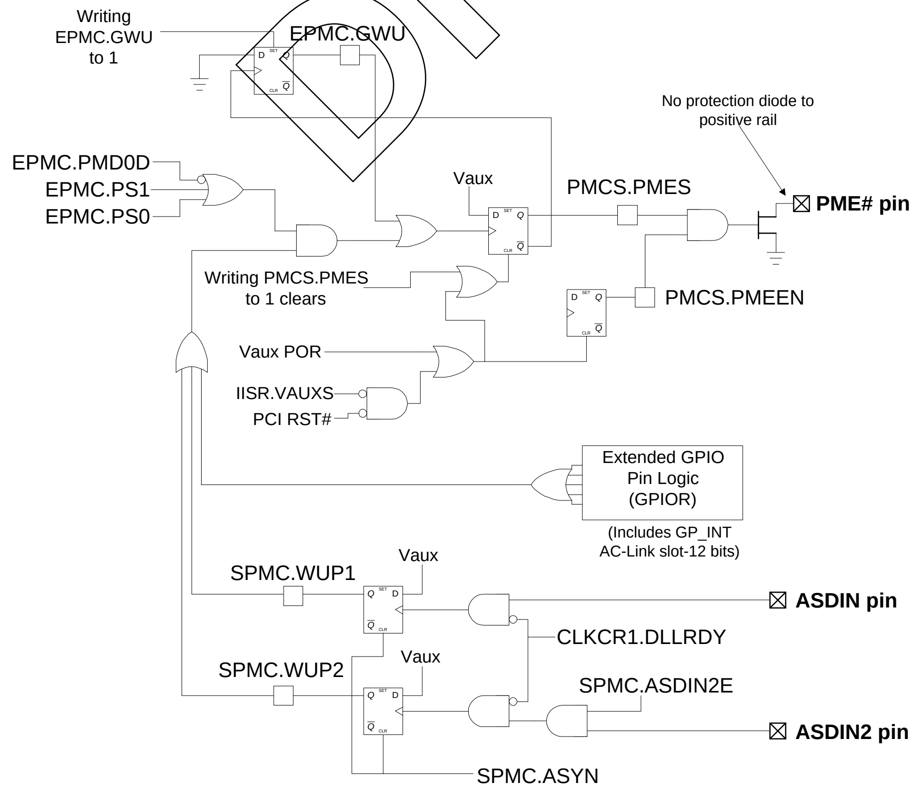


Figure 18. PME# Conceptual Logic

9.5 Vaux Powered Circuitry

The **VAUX** power supply pin may be active when the Core and PCI power pins are NOT powered. As illustrated in Figure 19, there are eight registers that are powered from the **VAUX** supply. One of these registers, EPPMC, has a bit that is reset to its default state by **RST#**. However, the other bits in EPPMC as well as the other registers are not affected by the **RST#** signal. These registers are initialized by a special **VAUX** POR circuit.

VAUX also powers two sets of logic, the Extended GPIO logic, which could cause a **PME#** event, and the **PME#** logic itself. **VAUX** also powers 8 of the pads: the extended GPIO pads, the serial port pads, and the **PME#** pin itself.

9.6 Block-Level Power-down and Reset States

This section describes the affect that the power-down bit **EPPMC.FPDN** and chip reset have on each block within the CS4281. For some internal blocks the affect of these two signals are the same, for other blocks their affect is vastly different. Chip reset is composed of three signals: the PCI **RST#** pin, an internally-generated core power-on reset signal labelled CPOR, and an internally-generated **VAUX** power-on reset signal labelled VPOR. The power-down bit **EPPMC.FPDN** powers down most of the chip, leaving only the PCI Configuration Registers accessible. This power-down bit is the closest thing to an entire chip reset while leaving the register accessible for software control.

The following blocks list the effect of both the power-down bit and chip reset.

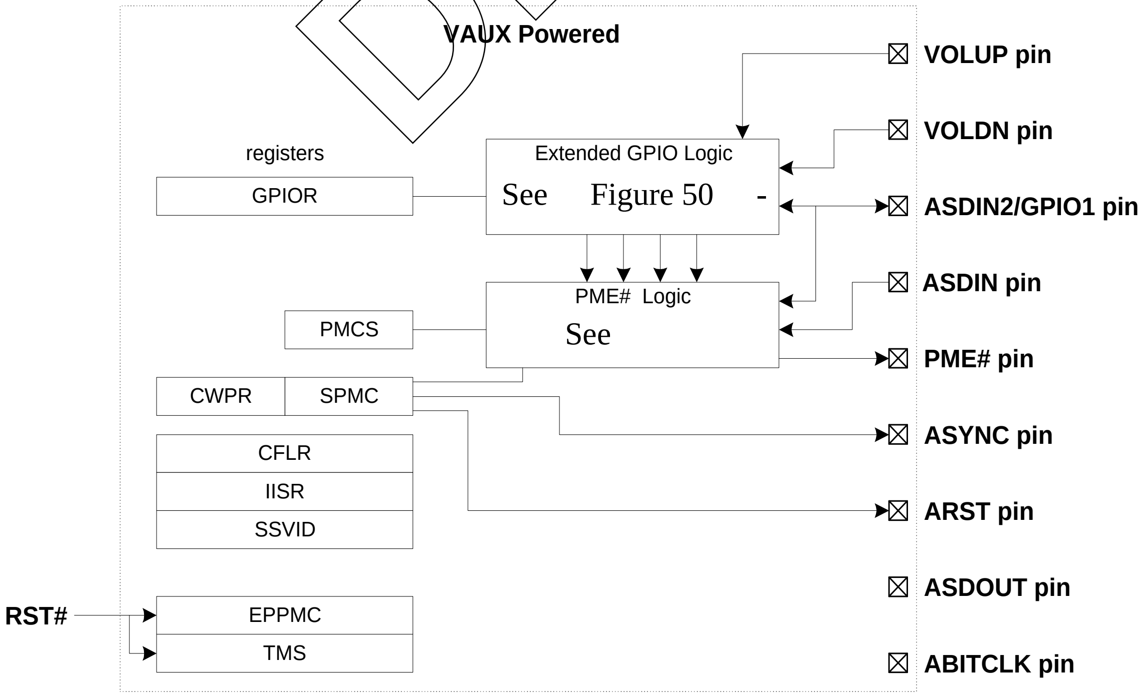


Figure 19. VAUX Powered Circuitry

9.6.1 PCI Bus Interface Unit (BIU) Block

Power-down and reset affect this block differently.

Power-down: from the **EPPMC.FPDN** bit.

- Process PCI CFG register access

Reset: from Core POR or PCI **RST#**.

- Ignore all PCI accesses

9.6.2 Control Block

Power-down and reset affect this block differently.

Power-down: from the **EPPMC.FPDN** bit.

- Reset all BA0 registers

Reset: from Core POR or PCI **RST#**.

- Reset all BA0 and PCI CFG registers (except VAUX registers)

9.6.3 VAUX Block

Power-down and reset affect this block differently. The **EPPMC.FPDN** bit has NO effect on this block.

Reset: PCI **RST#**

- Resets the **EPPMC.FPDN** bit and the TMS test register. All other bits and VAUX registers are not affected by **RST#**.

Reset: Core POR signal

- Only resets the **PMCS.PS[1:0]** and **EPPMC.FPDN** bits. All other VAUX register bits are not affected by the Core POR signal.

Reset: Vaux POR signal

- Resets all VAUX registers.

9.6.4 Sound System/DLL Block

Power-down and reset affect this block in the same way. The Sound System is powered-down when either the **CLKCR1.DLLP** or **CLKCR1.SWE** bits are clear, or when **ABITCLK** is off. These bits are forced clear by the **EPPMC.FPDN** power-down bit, PCI **RST#** signal, or core POR signal. In power-down or reset:

- State machine in reset - includes the Sound System Controller (SSC)
- The DLL clock, **clk512**, is off

When the Sound System and DLL are powered-down, the FM synthesizer, which uses the high frequency clock is powered down. The AC-Link and hardware volume control can still operate for Codec register accesses.

9.6.5 SRC Block

Power-down and reset affect this block in the same way. This block is composed of two sections: the playback SRC (PSRC) and the capture SRC (CSRC). The PSRC is powered down whenever the **SSPM.PSRCEN** bit is clear and the CSRC is powered down whenever the **SSPM.CSRCEN** bit is clear. These bits are both forced clear by the **EPPMC.FPDN** power-down bit, PCI **RST#** signal, or core POR signal. In power-down or reset:

- State machines are held in reset

9.6.6 Joystick Block

Power-down and reset affect this block in the same way. This block is powered down whenever the **SSPM.JSEN** bit is clear. This bit is forced clear by the **EPPMC.FPDN** power-down bit, PCI **RST#** signal, or core POR signal. In power-down or reset:

- Discharge gate off
- Button pull-ups active
- Comparators and reference off
- Registers held in reset
- Register writes ignored

9.6.7 MIDI Block

Power-down and reset affect this block in the same way. This block is powered-down whenever the **MIDCR.TXE** and **MIDCR.RXE** bits are both clear. These bits are forced clear by the **EPPMC.FPDN** power-down bit, PCI **RST#** signal, or core POR signal. In power-down or reset:

- MIDI clocks off
- MIDOUT forced high
- MIDIIN ignored
- Reset state machine
- Registers held in reset and are readable

9.6.8 Hardware Volume Block

Power-down and reset affect this block in the same way. This block is powered-down whenever the **SSCR.HVC** bit is clear. This bit is forced clear by the **EPPMC.FPDN** power-down bit, PCI **RST#** signal, or core POR signal. In power-down or reset:

- Hardware Volume pins controlled by GPIOR register (hardware volume off)
- Reset registers and state machine

9.6.9 AC-Link Engine Block

Power-down and reset affect this block in the same way. This block is powered-down whenever the **SSPM.ACLEN** bit is clear. This bit is forced clear by the **EPPMC.FPDN** power-down bit, PCI **RST#** signal, or core POR signal. This block also internally resets when **ABITCLK** goes away. In power-down or reset:

- ABITCLK don’t care
- ASYNC/ASDOUT forced low
- ASDIN/ASDIN2 ignored
- Reset state machines
- Registers held in reset and are readable

9.6.10 DMA/Registers Block

Power-down and reset affect this block differently.

Power-down: from the **EPPMC.FPDN** bit.

- Process PCI CFG register access
- Ignore BA0 and BA1 register access
- Reset BA0 registers and state machines

Reset: from Core POR or PCI **RST#.**

- Reset PCI CFG registers (except for VAUX registers as mentioned in VAUX section)

9.6.11 EEPROM Block

Power-down and reset affect this block in the same way. This block is powered-down and reset under the following conditions:

- **SPMC.EESPD** bit set, or
- **EPPMC.FPDN** bit set, or
- PCI **RST#** active, or
- Core POR

In power-down/reset, the state machine is held in reset and the block is disconnected from the **EECLK** and **EEDAT** pins.

9.6.12 FM Block

Power-down and reset affect this block in the same way. This block is powered-down whenever the **SSPM.FMEN** bit is clear. This bit is forced clear by the **EPPMC.FPDN** power-down bit, PCI **RST#** signal, or core POR signal. In power-down or reset:

- FM clock is off
- FM data output is zero
- All state machines held in reset
- Register access is ignored

10. INTERRUPT SUBSYSTEM

The CS4281 interrupt controller receives interrupt requests from multiple sources inside the device and presents a single interrupt line (**INTA#**) to the host system. Interrupt controller registers in CS4281 provide the host interrupt service routine (ISR) with source identification and indicating end-of-interrupt (EOI).

Interrupt Sources:

- DMA - half/end transfer count
- FIFO for polled data transfers - FIFO reaches certain number on counts
- MIDI receive port - have data
- Hardware Volume Control - volume change or pin transition
- GPIO pins - pin transition
- AC Link Slot 12 - **GP_INT** bit going high

Registers:

- Host Interrupt Status Register (HISR) - Reports pending interrupt sources (does NOT reset interrupts)
- Host Interrupt Control Register (HICR) - Command register for EOI, mask control, initialization
- Host Interrupt Mask Register (HIMR) - Interrupt source masks
- ISA Interrupt Enable Register (IIER) - supports ISA IRQ pin enables
- Host DMA Status Registers (HDSRn) - status and clearing of DMA interrupts
- DMA Command Registers (DCRn) - interrupt enables for DMA interrupts
- FIFO Status and Interrupt Control (FSICn) - enables, status, and clearing for Polled FIFO interrupts
- GPIO Control Register (GPIOR) - resets GPIO, AC Link, and Hardware Volume interrupts
- Host MIDI Status Register (MIDSR) - reads clear the transmit and receive interrupts.

Interrupts are normally generated on the PCI **INTA#** pin. ISA interrupts are available for mother-board CS4281 devices to support real-mode DOS applications where software does not handle the shared-interrupt concept of PCI interrupts.

10.1 PCI Interrupt Line Control

A master PCI interrupt enable bit, **INTENA**, controls assertion of the CS4281 PCI interrupt line (**INTA#**) at a global level. When **INTENA** is clear, no interrupt will be generated as a result of interrupt sources going active. When **INTENA** is set any interrupt source assertion will generate an external interrupt.

INTENA is set (interrupts are enabled) by the following condition:

- EOI command received (to HICR register)

INTENA is cleared (interrupts are disabled) by any one of the following conditions:

- Host read of HISR (only if any interrupt source bits set, therefore interrupt active)
- Explicit clear by write to HICR command register
- Device hardware reset

10.2 DMA and Polled FIFO Interrupts

There are four DMA engines that are tied to four FIFO controllers where, n = 0 to 3. The FIFO (FIFOn) is connected to the DMA engine (DMAn) based on equivalent n value. The DMA engine and registers are used in both DMA and Polled FIFO modes. DMA mode is enabled by setting **DMRn.DMA**, and Polled FIFO mode is enabled by setting **DMRn.POLL**. These bits are mutually exclusive.

When in DMA mode (**DMRn.DMA** = 1), the DMAn engine can generate an interrupt on the Terminal

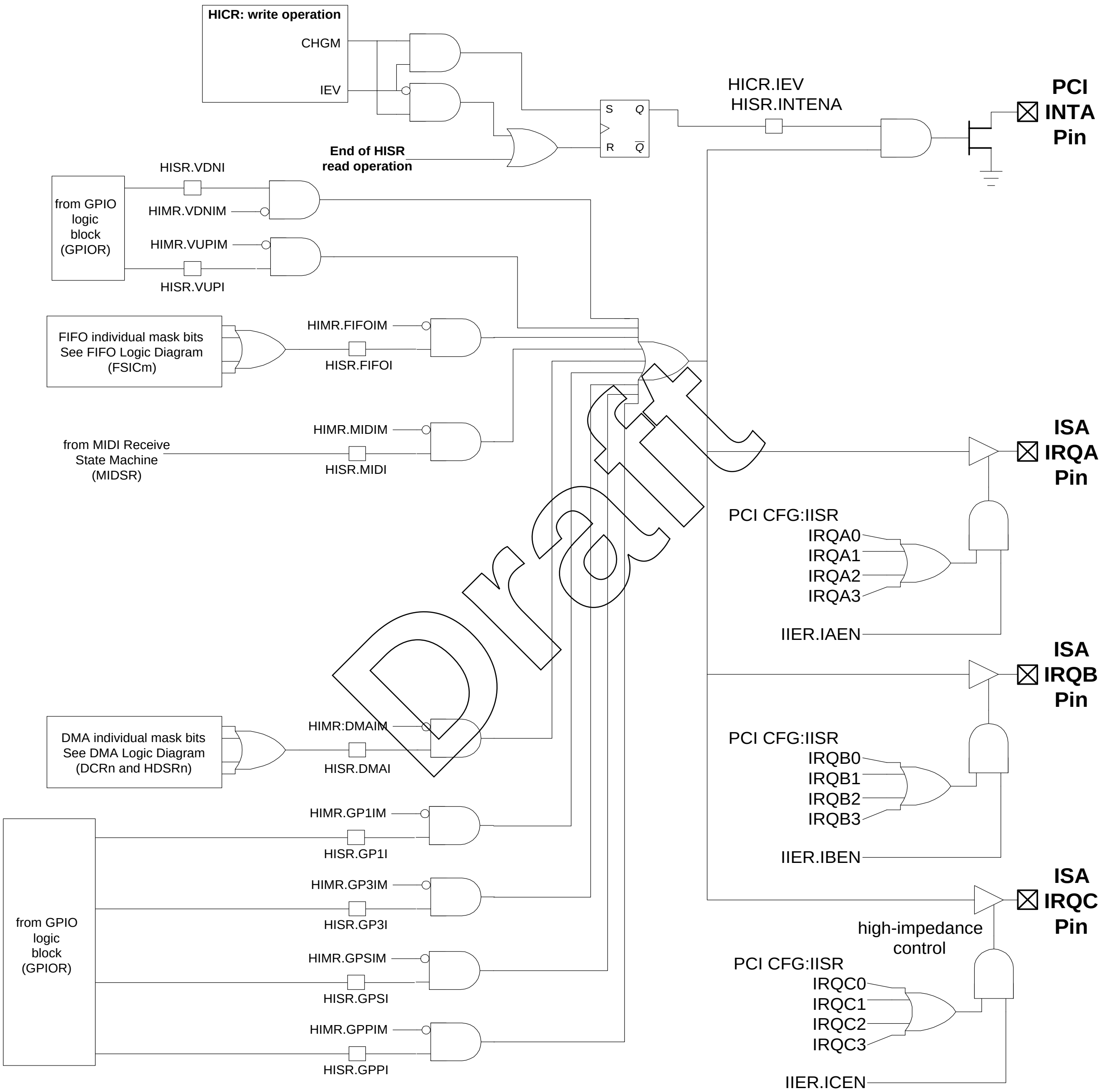


Figure 20. Interrupt Architecture

Count, TC, or Half the Terminal Count, HTC. HISR indicates which DMA is causing the interrupt. The interrupts are enabled through the upper word in the DCRn register. The interrupt condition is cleared by reading the *Host DMA Status Register*, HDSRn, which also indicates which interrupt occurred (TC or HTC). DMA interrupt conceptual logic is illustrated in Figure 21.

When in Polled FIFO mode (attached **DMAn.POLL** = 1), the DMAn engine supports the transfer of data between the host software (FPDRn) and FIFOn. Interrupts are enabled through the upper word of FSICn. HISR indicates which FIFO is causing the interrupt. Interrupts are cleared by reading the lower word of the FSICn register, which also indicates which type of FIFO interrupt occurred. As an example, HISR indicates that FIFO0 caused an interrupt, then the interrupt status and clearing occurs in FSIC0. The FIFO interrupt conceptual logic is illustrated in Figure 22.

10.3 Extended GPIO and Hardware Volume Control Interrupts

The extended GPIO pins can be configured through the GPIOR register to cause interrupts. Four pins have this capability: **GPIO3**, **ASDIN2/GPIO1**, **VOLUP**, **VOLDN**. All GPIO interrupts are cleared by writing a 0 to the appropriate status bit in GPIOR. To configure for interrupts, **GPIO3** and **ASDIN2/GPIO1** must have the following:

- Configured as an input (**GPIOR.GPxOE** = 0, where x is 1 or 3)
- Configured as sticky (**GPIOR.GPxST** = 1)
- Interrupt mask bit cleared (**HICR.GPxIM** = 0)

The **ASDIN2/GPIO1** pin must also NOT be configured for **ASDIN2.** Conceptual logic for GPIO pins as well as hardware volume control pins is illustrated in the *General Purpose Input/Output Pins* section: Figures 49 through 51.

The **VOLUP/VOLDN** pins are always inputs and can generate interrupts either as general purpose inputs or while being used as hardware volume control. For these pins to generate interrupts, the following configuration is used:

- Configured as sticky (**GPIOR.VxxST** = 1, where xx is UP and DN for the two pins)
- Individual mask bit cleared (**HICR.VxxIM** = 0)

When hardware volume is enabled (**SSCR.HVC** = 1), the GPIO logic can be triggered every time the hardware volume is updated (**GPIOR.VxxLT** = 1). This feature enables host software to keep track of master volume changes, or allows host software to totally control the master volume updates.

10.4 Interrupt Reporting Registers

10.4.1 Host Interrupt Status Register (HISR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTENA	res								MIDI	res	FIFOI	res	DMAI	res	res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIFO3	FIFO2	FIFO1	FIFO0	DMA3	DMA2	DMA1	DMA0			GPPI	GPSI	GP3I	GP1I	VUPI	VDNI

Address: BA0: 000h, Read-Only

Default: 00000000h

Definition: Core powered. The Host Interrupt Status register provides interrupt source information to the host interrupt service routine. Reading this register does NOT clear any interrupt. Interrupts are cleared from their respective sources. Each interrupt source has a corresponding mask bit in HIMR register.

Bit Descriptions:

INTENA Current state of internal interrupt enable bit. First read after interrupt will show **INTENA** = 1 (not cleared until end of/after read). Subsequent reads of HISR will show **INTENA** = 0 until EOI issued.

MIDI MIDI port interrupt. Caused by UART receiving data. Cleared by reading the MIDSR register.

FIFOI FIFO polled mode interrupt. Cleared by using **FIFO[3:0]** to select FIFO causing the interrupt. Then reading the respective FCHSn register (reads clear the interrupt condition).

DMAI DMA interrupt, either end of DMA or half of transfer. Cleared by using **DMA[3:0]** to select DMA engine causing the interrupt. Then reading the correct HDSRn register.

FIFO[3:0] FIFOn interrupt status. A 1 indicates the FIFOn that's causing the FIFOI flag. Cleared by reading the associated FCHSn register for the FIFO(s) that caused the interrupt.

DMA[3:0] DMAn engine interrupt status. A 1 indicates the DMAn that's causing the **DMAI** flag. Cleared by reading the associated HDSRn register.

GPPI General Purpose Input pin from Primary AC '97 Link (**ASDIN**) caused an interrupt (Slot 12, **GP_INT** bit). This interrupt is cleared by writing a 0 to the **GPPS** bit in the GPIOR register (see Figure 52); however, since the interrupt condition occurred in the Primary codec, it must be removed by writing to the Primary Codec (**ACCTL.TC** = 0) GPIO Pin Sticky register, Index 54h.

GPSI General Purpose Input pin from Secondary AC '97 Link (**ASDIN2**) caused an interrupt (Slot 12, **GP_INT** bit). This interrupt is cleared by writing a 0 to the **GPSS** bit in the GPIOR register (see Figure 52); however, since the interrupt condition occurred in the Secondary codec, it must be removed by writing to the Secondary Codec (**ACCTL.TC** = 1) GPIO Pin Sticky register, Index 50h.

GP3I GPIO3 pin caused an interrupt. Cleared by writing a 0 to the **GP3S** bit in the GPIOR register.

GP1I The **ASDIN2/GPIO1** (configured as a **GPIO1** input) caused an interrupt. Cleared by writing a 0 to the **GP1S** bit in the GPIOR register.

VUPI The **VOLUP** pin caused an interrupt. Cleared by writing a 0 to the **VUPS** bit in the GPIOR register. If hardware volume control is enabled, an interrupt can be generated every time the hardware volume control engine modifies the volume control or mute settings. To generate an interrupt hardware volume must be configured properly in the GPIOR register.

VDNI The **VOLDN** pin caused an interrupt. Cleared by writing a 0 to the **VDNS** bit in the GPIOR register. If hardware volume control is enabled, an interrupt can be generated every time the hardware volume control engine modifies the volume control or mute settings. To generate an interrupt hardware volume must be configured properly in the GPIOR register.

10.4.2 Host Interrupt Control Register (HICR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														CHGM	IEV

Address: BA0: 008h, Read-Write

Default: 00000000h

Definition: Core powered. The Host Interrupt Control register provides a host write port for EOI and discrete masking of interrupts. Also, it provides a read port for **INTENA** status.

Bit Descriptions:

CHGM **INTENA** Change Mask: This bit, if set to 1 on a write, enables the writing of the **IEV** bit into **INTENA**. If **CHGM** = 0, then **INTENA** is unaffected. On a host read this bit always returns a zero.

IEV **INTENA** Value: On a write, this bit contains the new value of **INTENA** to be stored if **CHGM** = 1. On a host read this bit contains the current state of **INTENA**.

Usage Descriptions:

- EOI command will be a write of 00000003 to HICR.
- Interrupt disable command (clear of **INTENA**) will be a 00000002 write to HICR.
- Read of current value of **INTENA** (without side effects) will be the value in bit 0 from a read of HICR.

10.4.3 Host Interrupt Mask Register (HIMR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	res								MIDIM	res	FIFOIM	res	DMAIM	res	res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F3IM	F2IM	F1IM	F0IM	D3IM	D2IM	D1IM	D0IM			GPPIM	GPSIM	GP3IM	GP1IM	VUPIM	VDNIM

Address: BA0: 00Ch, Read-Write

Default: 00F4FF3Fh

Definition: Core powered. The Host Interrupt Mask register provides a single register means for masking interrupt sources. Writing a 1 in the bit prevents an interrupt from being recognized from that source.

Bit Descriptions:

MIDIM MIDI port interrupt mask. Receive UART port has data available.

FIFOIM FIFO polled mode interrupt mask: either half empty/full, capture sample needed, playback sample available. Masts all FIFO interrupts.

DMAIM DMA interrupt mask, either end of DMA or half of transfer. Masks all DMA interrupts.

F[3:0]IM FIFOn interrupt mask. Masks an individual FIFO.

D[3:0]IM DMA engine interrupt mask. Masks an individual DMA engine.

GPPIM General Purpose Input from Primary AC ‘97 Link (**ASDIN**), slot 12, **GP_INT** interrupt mask.

GPSIM General Purpose Input from Secondary AC ‘97 Link (**ASDIN2**), slot 12, **GP_INT** interrupt mask.

GP3IM **GPIO3** input interrupt mask. **GPIO3** must be configured properly in the GPIOR register for interrupts to be generated.

GP1IM **GPIO1** input interrupt mask. **GPIO1** must be configured properly in the GPIOR register for interrupts to be generated.

VUPIM **VOLUP** input interrupt mask. This interrupt mask is associated with the **VOLUP** pin - either used as a hardware volume control, or as a general purpose input.

VDNIM **VOLDN** input interrupt mask. This interrupt mask is associated with the **VOLDN** pin - either used as a hardware volume control, or as a general purpose input.

10.4.4 ISA Interrupt Enable Register (IIER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ICEN	IBEN	IAEN

Address: BA0: 010h, Read-Write

Default: 00000000h

Definition: Core powered. Supports enables for the ISA interrupt pins. Configuration software can write this register to enable an ISA interrupt in real-mode DOS. This eliminates the PCI interrupt-sharing problem albeit at the expense of another interrupt used. For these bits to work, the corresponding bits in IISR must be non-zero. Note that in backward-compatible sockets, **IRQA** was the PCI signal **SERR#**, **IRQB** was a Core VDD power supply, and **IRQC** was a Core ground pin.

Bit Descriptions:

ICEN **IRQC** ISA interrupt Enable.
0 - **IRQC** pin high-impedance
1 - (along with any bit in **IISR.IRQC[3:0]**) actively drives (enables) the **IRQC** pin.

IBEN **IRQB** ISA interrupt Enable.
0 - **IRQB** pin high-impedance
1 - (along with any bit in **IISR.IRQB[3:0]**) actively drives (enables) the **IRQB** pin.

IAEN **IRQA** ISA interrupt Enable.
0 - **IRQA** pin high-impedance
1 - (along with any bit in **IISR.IRQA[3:0]**) actively drives (enables) the **IRQA** pin.

10.4.5 ISA Interrupt Select Register (IISR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VAUXS	VAC2	VAC1	VAC0	AUXP	BCF2	BCF1	BCF0	GTD				IRQC3	IRQC2	IRQC1	IRQC0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IRQB3	IRQB2	IRQB1	IRQB0					IRQA3	IRQA2	IRQA1	IRQA0

Address: BA0: 3F4h, Read-Write
PCI CFG: 0F4h, Read-Write if CWPR configured, otherwise Read-Only

Default: 00000000h

Definition: Vaux powered. Defines the ISA interrupt associated with a particular pin and relays flags from BIOS to the OS and host software. This register is unaffected by the PCI **RST#** signal. The default value is set by a Vaux POR signal.

Bit Descriptions:

VAUXS Vaux Support. This bit is reflected into the $D3_{cold}$ support bit, **PMC.PMD3C**. BIOS code would generally set this bit (through config space) if VAUX is supported.

VAC[2:0] Vaux Current. These bits are reflected in the **PMC.VAC[2:0]** bits and must be initialized by the BIOS to indicate how much current Vaux pulls. Note this is total current and is the combined CS4281 and any attached Codecs and external logic using Vaux.
000 - 0 mA (self powered/don’t support Vaux)
001 - 55 mA
010 - 100 mA
011 - 160 mA
100 - 220 mA
101 - 270 mA
110 - 320 mA
111 - 375 mA (spec maximum)

AUXP Auxiliary Power. This bit is reflected in the **PMC.AUXP** bit. Not sure what this bit does yet.

BCF[2:0] BIOS Configuration Flags. These bits are read-writable and available in config. and host memory space. They have no direct affect on the operation of the CS4281 and may be used by host software when communicating with the BIOS.

GTD Global Trapping Disable. When set, disables all I/O trapping including overriding the **IOTCR.ITD** bit. When **GTD** is clear, **IOTCR.ITD** controls all I/O trapping.

IRQA[3:0] **IRQA** pin interrupt mapping. A 0 disables (high impedance) the corresponding ISA interrupt pin. A non-zero value (preferably the actual ISA interrupt connected to the pin) allows the pin to be enabled through the ISA interrupt enable bit **IIER.IAEN**.

IRQB[3:0] **IRQB** pin interrupt mapping. A 0 disables (high impedance) the corresponding ISA interrupt pin. A non-zero value (preferably the actual ISA interrupt connected to the pin) allows the pin to be enabled through the ISA interrupt enable bit **IIER.IBEN**.

IRQC[3:0] **IRQC** pin interrupt mapping. A 0 disables (high impedance) the corresponding ISA interrupt pin. A non-zero value (preferably the actual ISA interrupt connected to the pin) allows the pin to be enabled through the ISA interrupt enable bit **IIER.ICEN**.

11. DMA CONTROLLER

Host initiated DMA will be supported by 4 DMA engines that function as a super-set of the 8237 legacy DMA controller. For this discussion, a *stream* is the concept of forming a path from host memory to the AC Link (playback), or from the AC Link to host memory (capture/record). All hardware associated with supporting this data transport pipe is the *stream*. A given stream may transport either stereo or monaural data, as controlled by the data format bits.

11.1 System Operation

The DMA controller will support the 4 data streams as 4 independent DMA engines. The engines can be enabled simultaneously, but only one engine at a time will have access to the PCI bus. Only two DMA operations are supported: transfers from AC Link to host memory, and transfers from host memory to the AC Link. The DMA controller will only do single data transfers, NO bursts.

Host buffer data alignment is controlled by the data channel transfer size. In the CS4281, a channel is atomic (can't be split across PCI bus transactions). Byte channels will naturally be aligned to bytes, word channels must align to word addresses, and 20-bit data must align to double word addresses. Samples, however, are not atomic and can be split across PCI transactions. Mono samples contain only one channel and are atomic by their nature. Stereo samples contain two channels which can be split across PCI bus transfers. For split transactions, the DMA engine will do two PCI transfers to get both channels. 20-bit stereo data always requires 2 PCI accesses. For 8- and 16-bit stereo data, the DMA engine will look at the starting address to decide if split access is required. In Polled FIFO mode, the DMA engine will use **DMRn.TBC** and the program format bits to determine the transfer size.

For 16-bit stereo data, **DBAn.BAL0** must be 0 since channels are atomic. If **DBAn.BAL1** = 0, then the 32-bit sample is 32-bit aligned and the DMA engine will do 1 PCI access per sample (transferring both channels simultaneously). If **DBAn.BAL1** = 1, then the 32-bit sample is mis-aligned and the DMA engine will do two PCI accesses per sample (first transfer for channel 1 and second transfer for channel 2).

For 8-bit stereo data, if **DBAn.BAL0** = 0, then the 16-bit sample is word aligned and the DMA engine will do 1 PCI access per sample (transferring both channels simultaneously). If **DBAn.BAL0** = 1, then the word sample is mis-aligned and the DMA engine will do 2 PCI accesses per sample (first transfer for channel 1 and second transfer for channel 2). Once the DMA engine starts transferring mis-aligned samples using 2 transfers, the DMA engine will continue to do by-channel transfers even when the mis-aligned stereo sample is within a single 32-bit dword.

The DMA engine supports counting by sample or channel. This count is independent of alignment mentioned above. For mono data, one sample equals one channel. For stereo data, one sample equals two channels. The length of the data transfer can be the complete 4 Gbyte address space. There are no restrictions for page or segment boundaries; however, since scatter/gather is not supported, restrictions are placed on the driver that allocates the memory buffer. For generic memory allocation, Windows only guarantees contiguous memory on a Pentium page boundary (4 k). To guarantee a contiguous buffer larger than 4 k, the driver must ask for memory in the crowded lower 16 Mbyte space.

Since the DMA controller is intimately tied to the sample FIFOs, the DMA controller will also provide the engine for polled data transfers. There are no restrictions on mixing DMA and polled access. Each stream can be operated independently of the others in terms of DMA or polled operation. However, polled operation on a stream that has an enabled DMA operation will not be allowed.

The DMRn controller is used in both DMA and Polled FIFO modes, where **DMRn.DMA** indicates that the DMAn/FIFOn combination is in DMA mode, and **DMRn.POLL** indicates that the DMAn/FIFOn connection is in Polled FIFO mode. The **DMA** and **POLL** bits are mutually exclusive. When **DMA** or **POLL** goes from a 0 to a 1, the DMAn engine is set in it's initial state (channel status cleared). For either mode, the **FCRn.FEN** bit must be set to enable the FIFO.

11.1.1 Host Interrupts for DMA Mode

There is a global mask/interrupt bit for DMA generated interrupts and individual mask bits for each engine. DMA interrupts are masked through the *Host Interrupt Mask Register*, HIMR, and the interrupt status is reported in the *Host Interrupt Status Register*, HISR.

When a FIFOn/DMAn pair is setup for DMA (**DMRn.DMA** set), interrupts can be generated from either *Terminal Count* or *Half Terminal Count*. The logic diagram for DMA interrupts is illustrated in Figure 21. The upper word of DCRn provides enable bits for each interrupt. Interrupt status for individual DMAn engines is located in the HDSRn register. Reading the upper word of HDSRn clears the interrupt condition/status (after read). As an example, if the HISR indicates that DMA1 caused the interrupt (**HISR.DMA1**), the interrupt condition is cleared by reading HDSR1.

11.1.2 Host Interrupts in FIFO Polling Mode

Host interrupts can also be generated for Polled FIFO modes. All interrupts are masked through the *Host Interrupt Mask Register,* HIMR, and the interrupt status is reported in the *Host Interrupt Status Register*, HISR. There is a global mask/interrupt bit for Polled FIFO mode and individual mask bits for each FIFO.

The FIFO interrupt logic, shown in Figure 22, can generate interrupts as long as the particular FIFO is enabled (**FCRn.FEN**). Generally, FIFO interrupts would only be enabled in Polled FIFO mode. The **FSICn.FSC[6:0]** bits contain the current sample count for the FIFO (number of samples currently in the FIFO). An interrupt can be generated when the **FSICn.FSC** bits match the host programmed **FSICn.FIC** bits. The **FIFOn.FIC** bits can be programmed to generated an interrupt on any FIFO depth such as:

- FIFO Empty: **FSICn.FIC[6:0]** = 0
- FIFO Not Empty: **FSICn.FIC[6:0]** = 1
- FIFO Half Full: **FSICn.FIC[6:0]** = **FCRn.SZ**/2
- FIFO Not Full: **FSICn.FIC[6:0]** = **FCRn.SZ** - 1
- FIFO Full: **FSICn.FIC[6:0]** = **FCRn.SZ[6:0]**

Two error conditions can also generate interrupts. FIFO Overrun, **FSICn.FOR**, indicates that, pending on FIFO direction, either the Bus Interface side

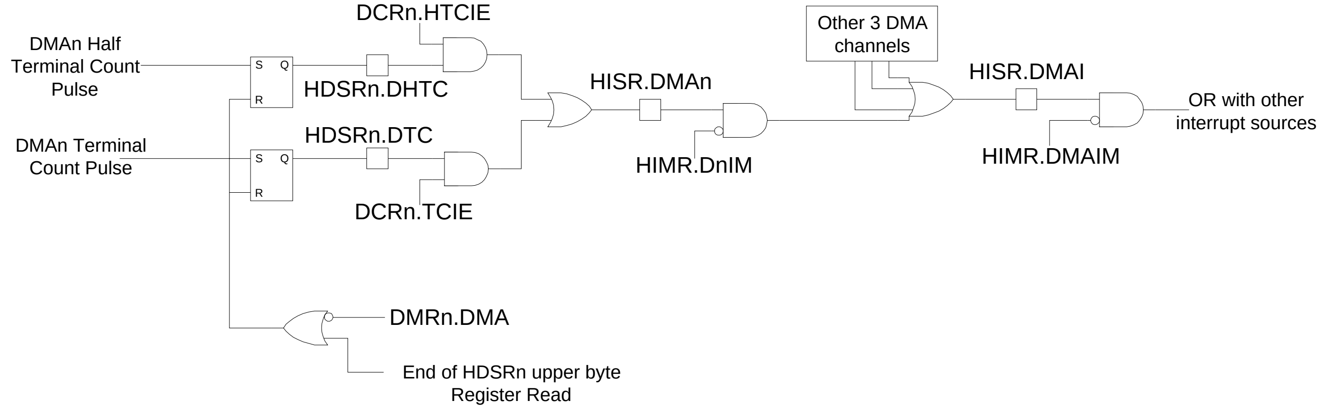

DCRn.HTCIE
DMAn Half Terminal Count Pulse
S
Q
R
HDSRn.DHTC
HISR.DMAn
Other 3 DMA channels
HISR.DMAI
OR with other interrupt sources
DMAn Terminal Count Pulse
HDSRn.DTC
S
Q
R
HIMR.DnIM
HIMR.DMAIM
DCRn.TCIE
DMRn.DMA
End of HDSRn upper byte Register Read

Figure 21. DMA Interrupt Conceptual Logic

(DMAn/Host), or the AC-Link side (SSC), tried writing data into a full FIFO. FIFO Underrun, **FSICn.FUR**, indicates that, pending on FIFO direction, either the Bus Interface side (DMAn/Host), or the AC-Link side, tried reading and empty FIFO.

The upper word of FSICn provides enable bits for each interrupt with the lower word providing status. Reading FSICn clears the interrupt condition/status (after read). For example, if HISR indicates that FIF00 caused the interrupt (**HISR.FIFO0**), the interrupt status is in FSIC0 and the interrupt condition is cleared by the read of FSIC0.

11.2 Data Format Conversions

The DMA controller only supports transferring a maximum of one sample per PCI transaction. The DMA controller/formatter is capable of packing/unpacking stereo data and steering the data being transferred in several different ways. A field in each DMA Mode Register (DMRn) allows special operations on data being transferred to convert between various types of digital audio data. The options available are targeted at transforming several different audio formats, as represented in host memory, into the format used by the AC-97/98 codec (20-bit signed data). The *Host Data Format* refers to how the sample appears in host memory, and the *Codec Data Format* refers to how the data is on the AC Link and in the FIFOs. Internal to the CS4281, audio data is always 20-bit signed (2's-complement) integers.

Audio samples can be either monaural or binaural, that is stereophonic. Monaural audio data is a single channel, whereas stereo data is composed of a left and a right component waveforms that are separately processed, by the left and right channels.

There are three host data sizes supported: 8 bit, 16 bit and 20 bit. The 16- and 20-bit formats can be big endian or little endian. A 20-bit data channel format is represented in one 32-bit host double word, MSB aligned (Host bit 31 is data bit 19, etc). All three data sizes can be signed or unsigned. In an IBM PC-compatible system, the only currently supported formats are 8-bit unsigned and 16-bit signed.

Audio data coming to/from the PCI bus is sent through a *formatter*, shown in Figure 23, that converts the data to the internal 20-bit signed format. The data is then transferred to/from a FIFO which, in turn, sends/receives the data across the AC Link to the Codec. The FIFO's are 40 bits wide, holding a maximum of one 40-bit stereo sample. Data in the FIFOs is always in 20-bit stereo signed format. The

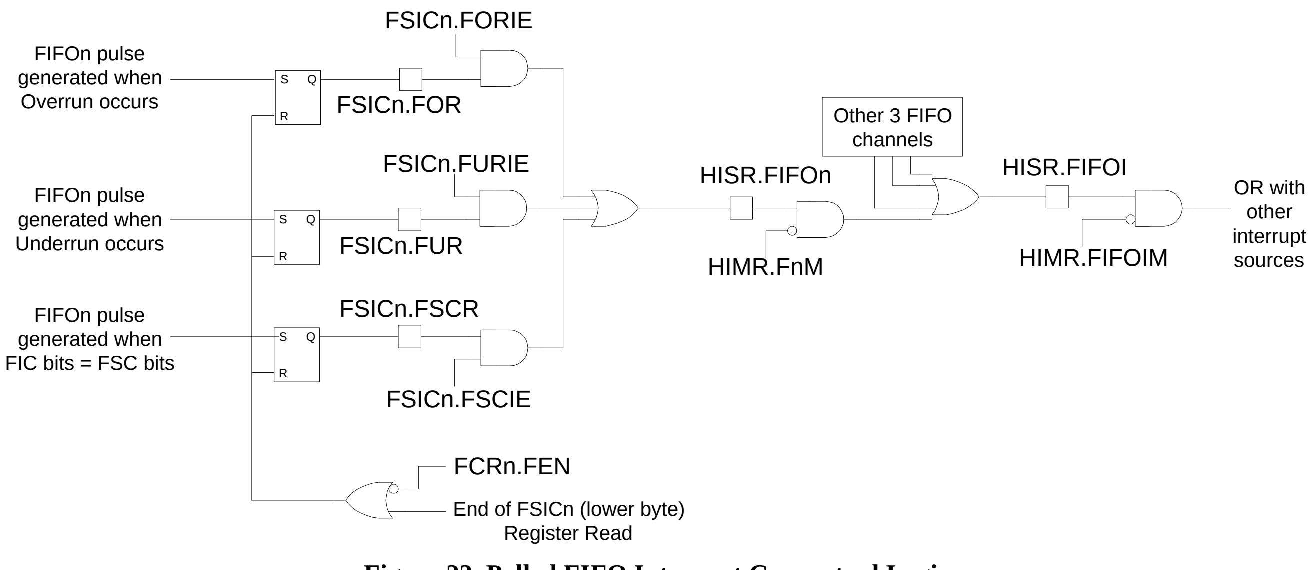

FSICn.FORIE
FIFOn pulse generated when Overrun occurs
S
Q
R
FSICn.FOR
Other 3 FIFO channels
FSICn.FURIE
HISR.FIFOn
HISR.FIFOI
FIFOn pulse generated when Underrun occurs
S
Q
R
FSICn.FUR
HIMR.FnM
HIMR.FIFOIM
OR with other interrupt sources
FSICn.FSCR
FIFOn pulse generated when FIC bits = FSC bits
S
Q
R
FSICn.FSCIE
FCRn.FEN
End of FSICn (lower byte) Register Read

Figure 22. Polled FIFO Interrupt Conceptual Logic

depth is programmable in the FCRn registers. Samples can move to/from the FIFO in 20-bit channels; however both channels in a sample must be read/written for the FIFO pointers to change. The DMAn engine sends signals to the formatter indicating which channels on the bus are valid (channel 1 and channel 2 valid signals). The formatter takes these signals and modifies them based on the **DMRn.MONO** and **DMRn.SWAPC** bits, and sends the modified signals to the FIFOn as Left and Right Channel valid signals. If the **SWAPC** bit is set, the formatter swaps the data on the bus and swaps the channel valid signals going to the FIFO. If the **MONO** bit is set, the DMAn engine will tell the formatter that channel 1 is valid. The formatter will expand the mono data to stereo and change the channel 1 valid signal, from the DMA, to both Left and Right channel valid signals, going to the FIFO.

For PCI host playback data, mono data is duplicated and sent to both FIFO channels. If only one mono channel is needed on the AC-Link side, the FIFO-to-AC Link mapping (**FCRn.LS/RS** bits) can be programmed to throw away an unused channel. For PCI host record data, MONO data generally comes from the left side of the FIFO. The channel used for mono data can be changed using the *Swap Channel* bit **DMRn.SWAPC**. **SWAPC** changes the side of the FIFO attached to the formatter channels and is typically used in Sound Blaster mode where the channels are sent right then left (opposite to normal operation). When the formatter reads mono data from the FIFO, it must read both sides of the FIFO (left and right Channel Valid signals) so the FIFO read pointer moves. The formatter will then place the proper mono data channel on the PC bus and leave the other channel in the formatter's latch unused.

Figure 24 shows 8-bit sample transfers and Figure 25 shows 16-bit transfers between the PCI bus and the FIFO. The left sides illustrate mono data and the right sides stereo. The *Byte swizzler* is part of the *Formatter* block and converts the data to the proper format between the fixed-format FIFO and the variable-format PCI host. The data channels are listed as Ch1 and Ch2 since the *Channel Swap* bit can change the channel order. All data transfer diagrams assume that the channel swap bit (**DMRn.SWAPC**) is 0 for capture data since the Figures depict mono-capture transfers coming from the left side of the FIFO. **SWAPC** is generally cleared for host processing, setting Ch1 as the Left Channel and Ch2 as the Right Channel. The Figures also illustrate the atomic nature of samples (only one sample per PCI transfer) and the 2-PCI cycles needed when samples are mis-aligned in memory. The small number to the right of the 2-cycle transfers indicates that Ch1 is transferred before Ch2. If both channels are sent across the PCI bus, the formatter expands the data to signed 20-bit format and sends the 40-bit stereo sample to the FIFO and the FIFO pointer is changed. If only one channel is transferred at a time, the formatter indicates to the FIFO which channel is read/written. The FIFO pointers do NOT change until both channels are read/written.

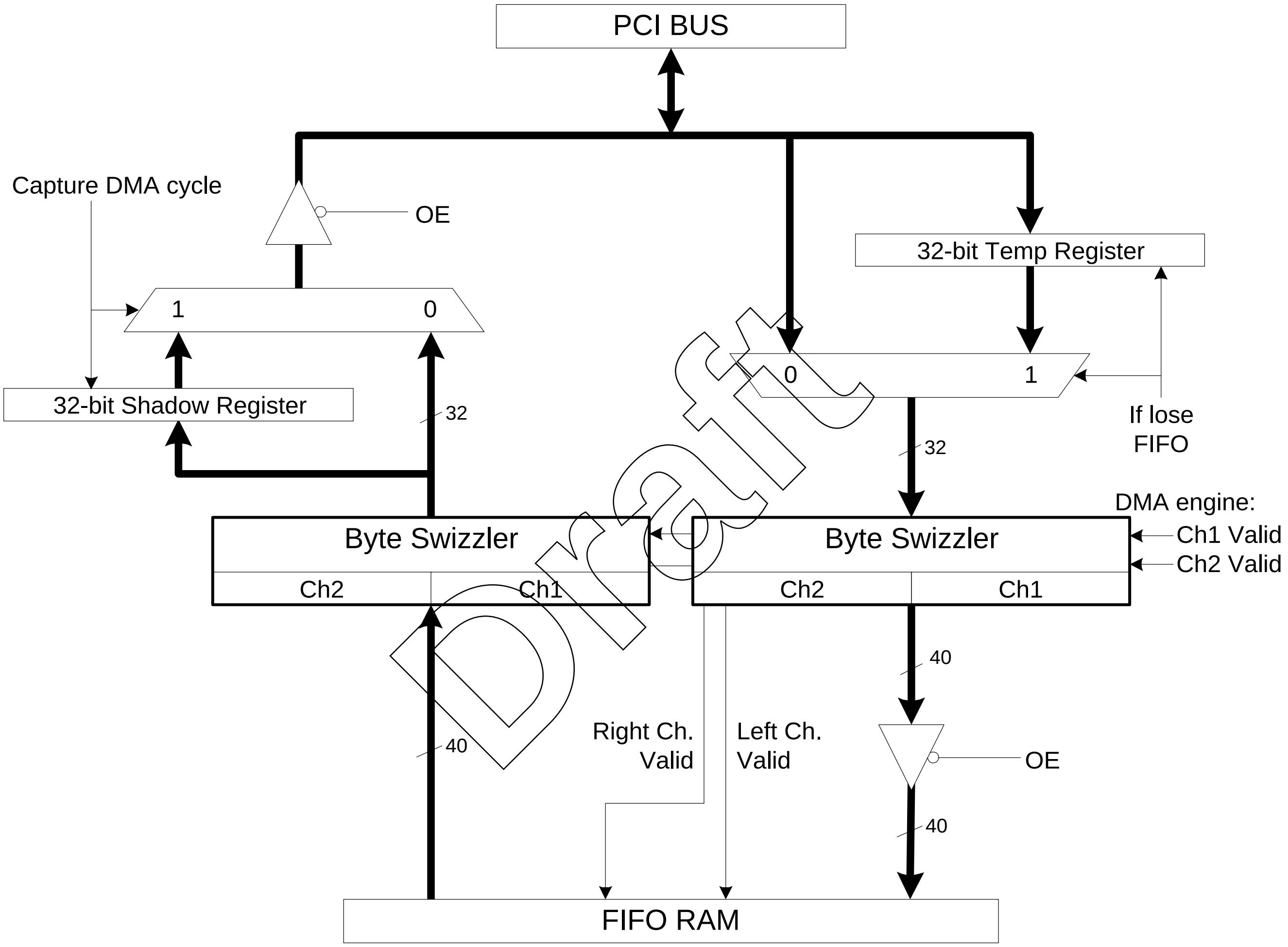


Figure 23. Formatter Block Diagram

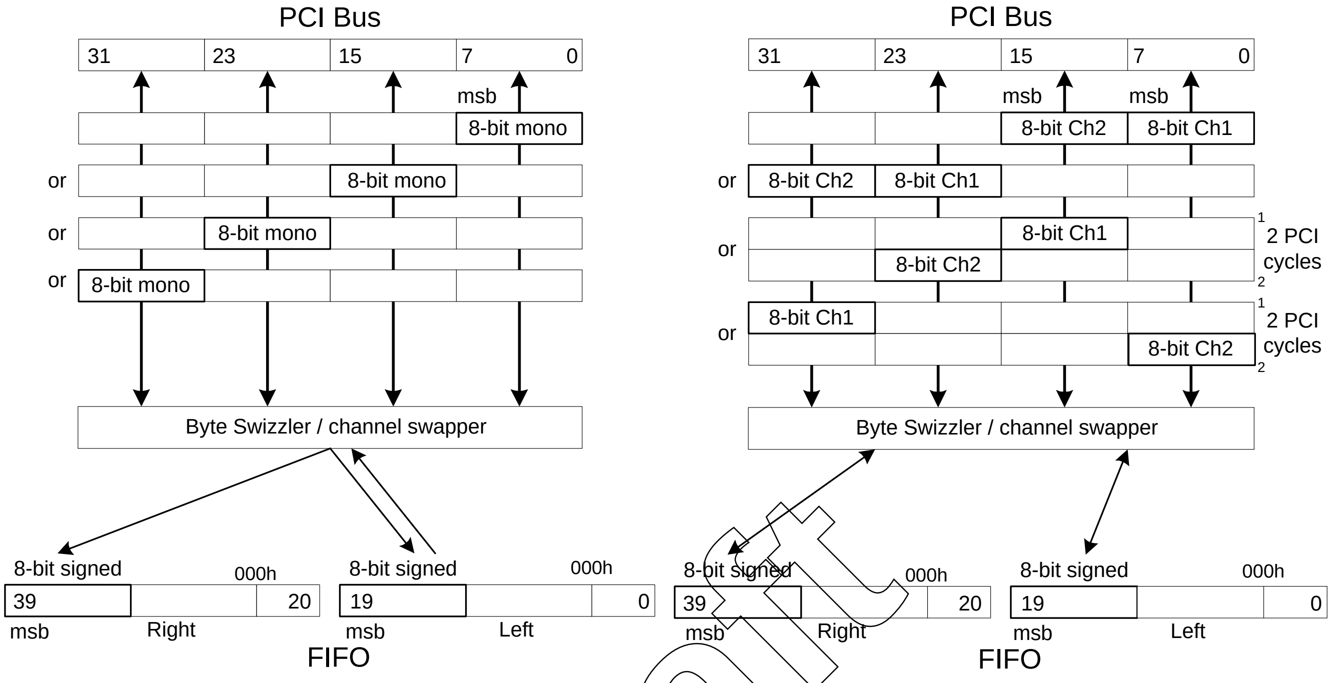


Figure 24. 8-bit PCI Data Transfers

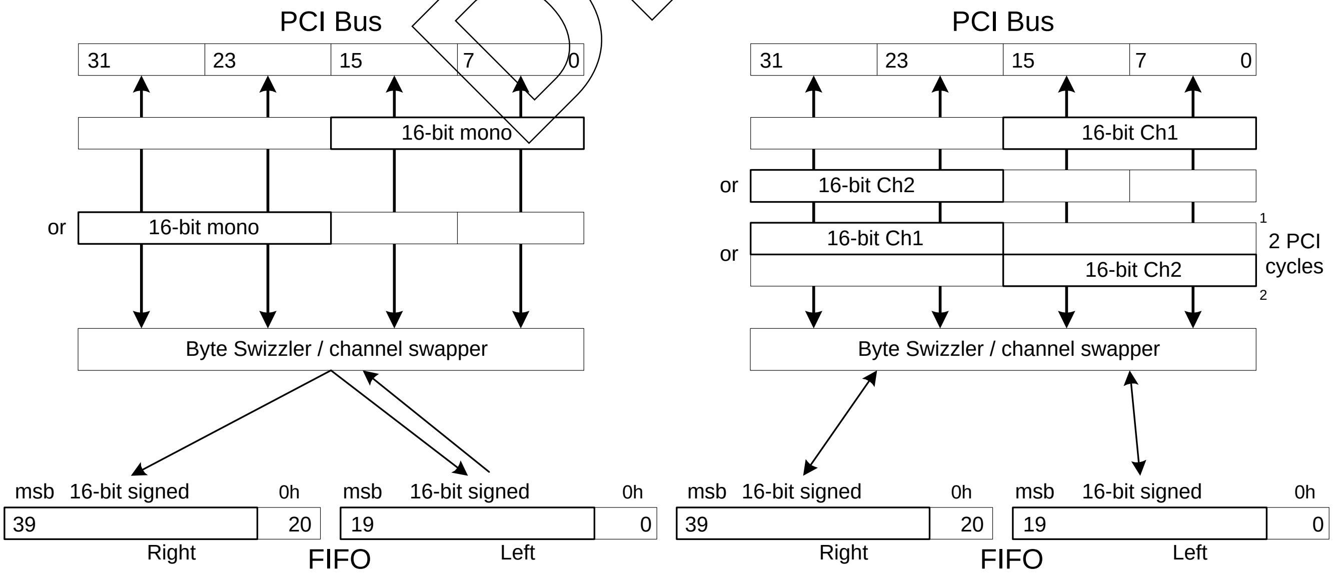


Figure 25. 16-Bit PCI Data Transfers

The PC only supports bytes, words, and double words (dwords). The 20-bit data format is stored as one dword per channel. As shown in Figure 26, 20-bit playback data format is MSB aligned. Stereo 20-bit data always require two PCI transfers to complete, where the DMA engine performs two PCI transfers to get one stereo 20-bit sample. The formatter block transfers data to the FIFOs as it is received.

Since 20-bit data format is stored one dword per channel, the capture data format writes the entire 32-bit dword by placing zeros in the unused LSB's, as illustrated in Figure 27.

The formatter Byte Swizzler for the playback direction is illustrated in Figure 28. The Byte Swizzler takes the PCI bus data and converts it to the proper internal format needed for the FIFOs. For stereo data, a sample could come across the PCI bus a channel at a time. For this case the DMA engine indicates to the formatter that the only Ch1 is valid. The Byte Swizzler formats Channel One properly and sends the data on to the FIFO. When the second channel arrives, the DMA engine indicates that Ch2 is valid, the Byte Swizzler formats Channel Two properly and sends the second channel to the FIFO causing the FIFO to increment its pointer since it now has both channels. The data out of the Byte Swizzler is shown in three sections for each channel x (1 or 2): ChxH is the high byte, ChxL is the low byte, and Chxr is the residual nibble forming the 20-bit word.

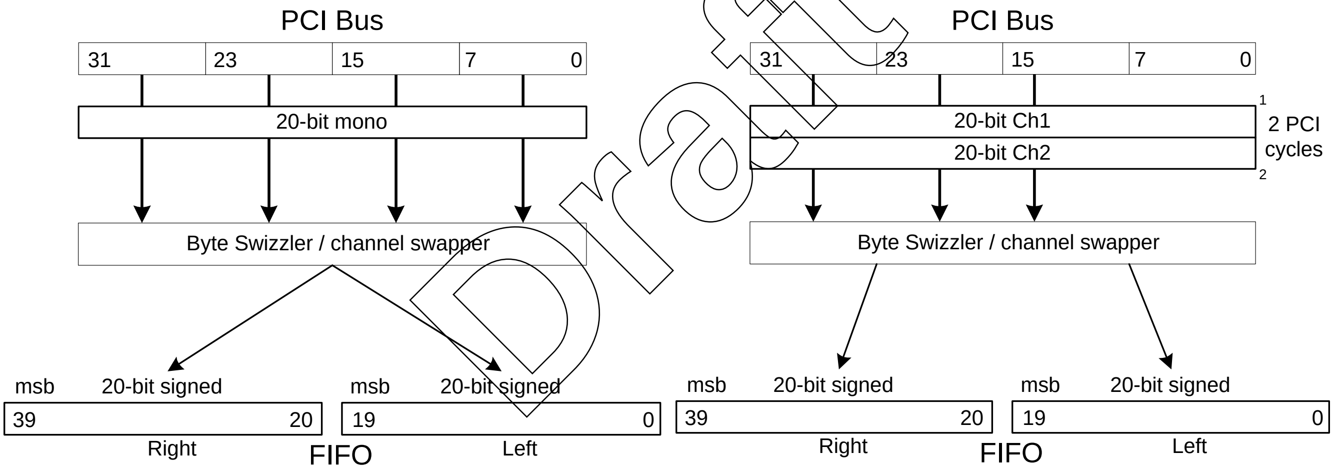


Figure 26. 20-Bit PCI Playback Data Transfers

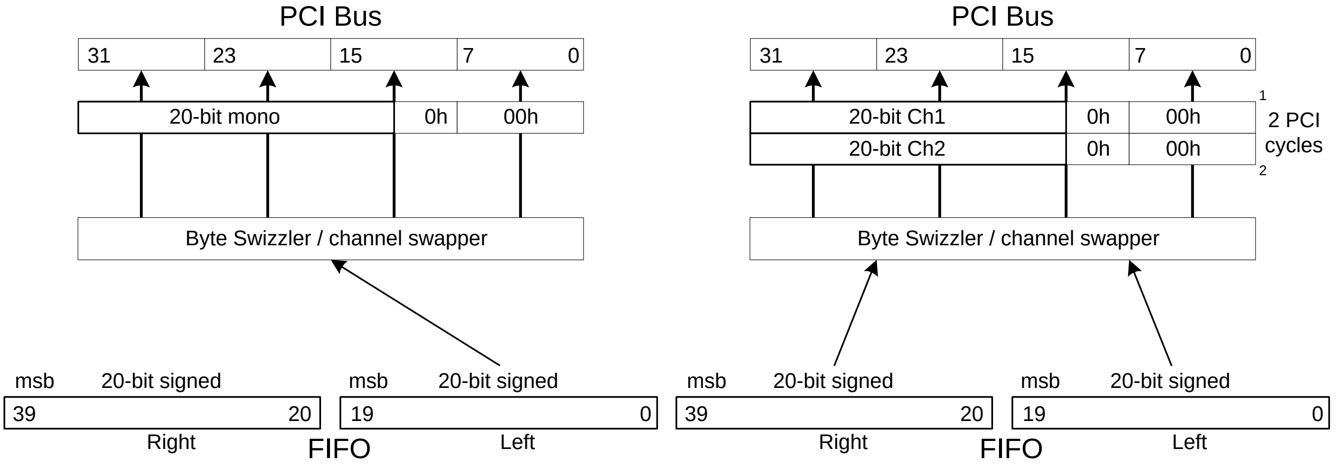


Figure 27. 20-Bit Little-Endian PCI Capture Data Transfers

The Byte Swizzler, illustrated in Figure 29, changes the format stored in host memory to and from the 20-bit signed little-endian format required for the FIFO's/AC Link. The Byte Swizzler re-orders bytes and inverts the MSB (sign bit) if the host data is unsigned (**DMRn.USIGN**). The Byte Swizzler also swaps channels if the **DMRn.SWAPC** is set. The *DMAn Mode Register* (DMRn), upper word, contains the format bits that control the Byte Swizzler and are valid for both DMA and Polled FIFO modes.

When the host writes the FIFO RAM directly, the formatter is internally forced to signed 20-bit little endian. Since this format is the internal data format, the formatter's Byte Swizzler does not modify the data sent by the host. The host can read or write one channel at a time out/into of the FIFO. FIFO read/write pointers are atomic on a sample basis and FIFO data is always stereo. If only one channel is written, the FIFO sets a flag and waits for the other channel before changing the write pointer. If the formatter gets two Ch1 valid signals in a row, the new data replaces the old data in the FIFO and it still waits for the other channel. The same is true when reading the FIFO. With this method, any software that stops transferring stereo samples after the first channel, does not affect other channels or software.

The Channel Swizzler, illustrated in Figure 30, which is part of the Formatter's Byte Swizzler, takes the Channel Valid signals from the DMAn controller, and converts them into Left and Right Channel Valid signals used by the FIFOn to indicate which channel (or both) is valid. If the data format is mono (**DMRn.MONO**), then Ch1_Valid is sent to both Left and Right Channel Valid signals. The **DMRn.SWAPC** controls swapping of channel valid signals.

If host software accesses the FIFO memory directly, the Channel Swizzler converts the bus address bit 2 (**BA1_A2**) to a Left/Right Channel Valid signal for FIFOn memory as shown in Figure 30.

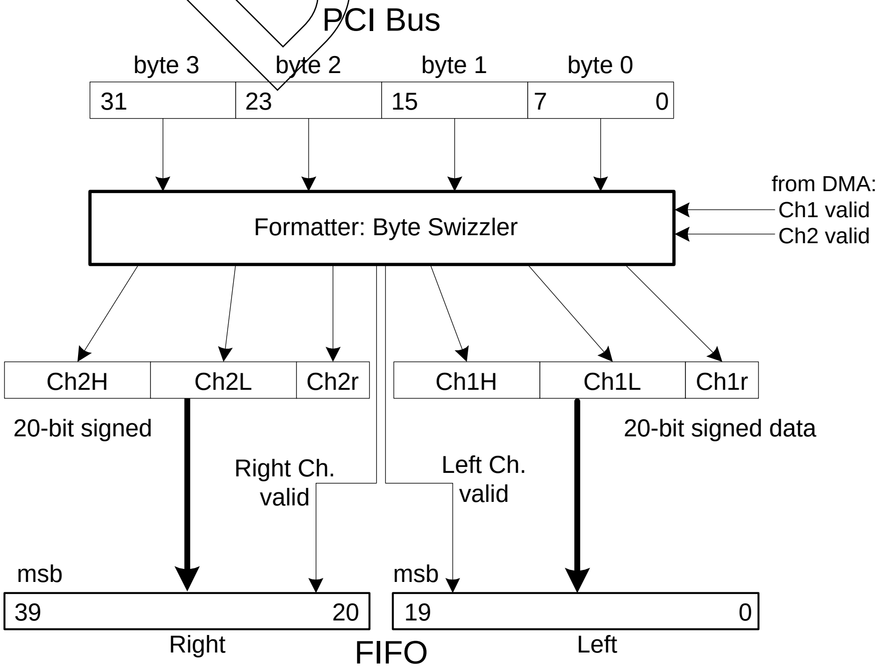


Figure 28. Formatter Playback Flow

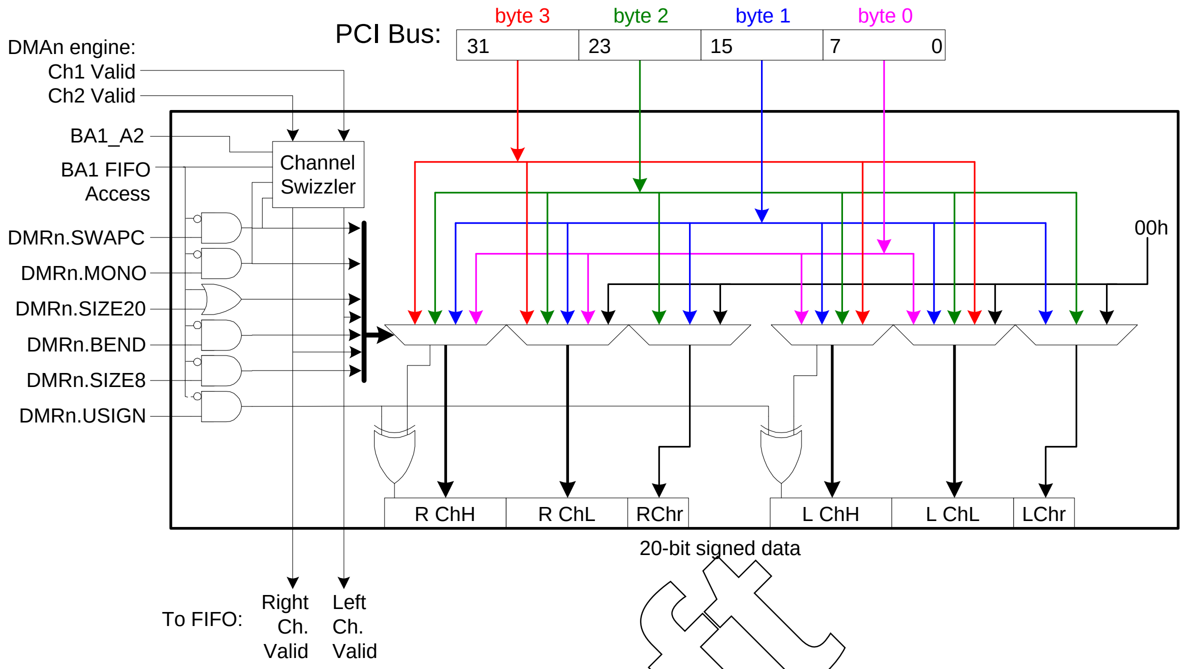


Figure 29. Playback Formatter Byte Swizzler

For 8-bit formats, the Byte Swizzler changes host memory format into FIFO format as depicted in Figure 31. For clarity, the Figure shows the data double-word (dword) aligned; however, the data can also be aligned as shown in Figure 24, which also illustrates a stereo sample taking two PCI bus cycles.

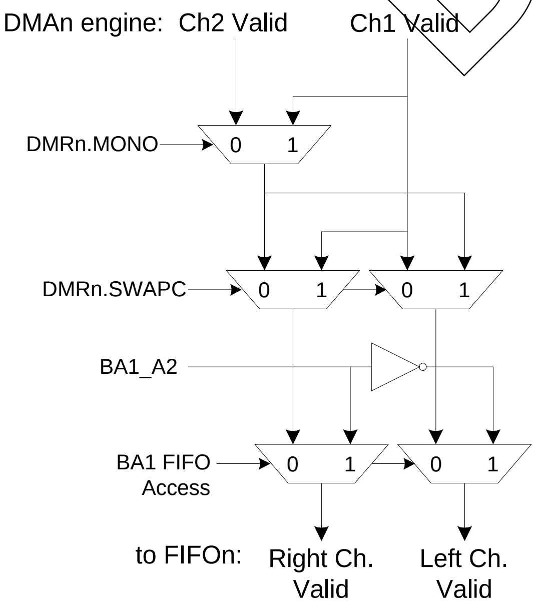


Figure 30. Playback Formatter Channel Swizzler

For 16-bit formats, the Byte Swizzler changes host memory format into FIFO format as depicted in Figure 32 for little endian and Figure 34 for big endian. For clarity, the Figure shows the data double-word (dword) aligned; however, the data can also be aligned as shown in Figure 25, which illustrates a stereo sample taking two PCI bus cycles (samples mis-aligned).

For 20-bit formats, the Byte Swizzler changes host memory format into FIFO format as depicted in Figure 33 for stereo little endian, Figure 35 for stereo big endian, and Figure 36 for mono 20-bit playback. The 20-bit data format is stored in host memory is 32-bit dwords. On 20-bit reads of the FIFO, the formatter drives the unused 12 LSB's to 0 (all 32-bits are driven).

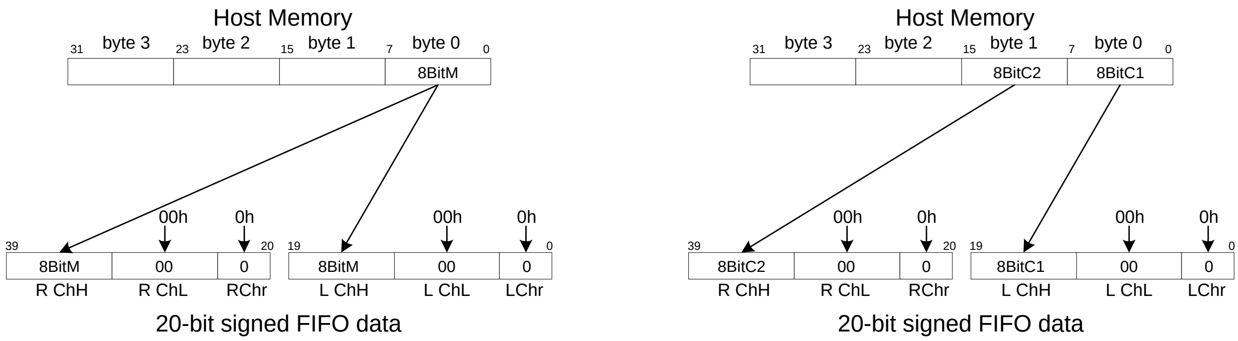


Figure 31. 8-Bit Formats

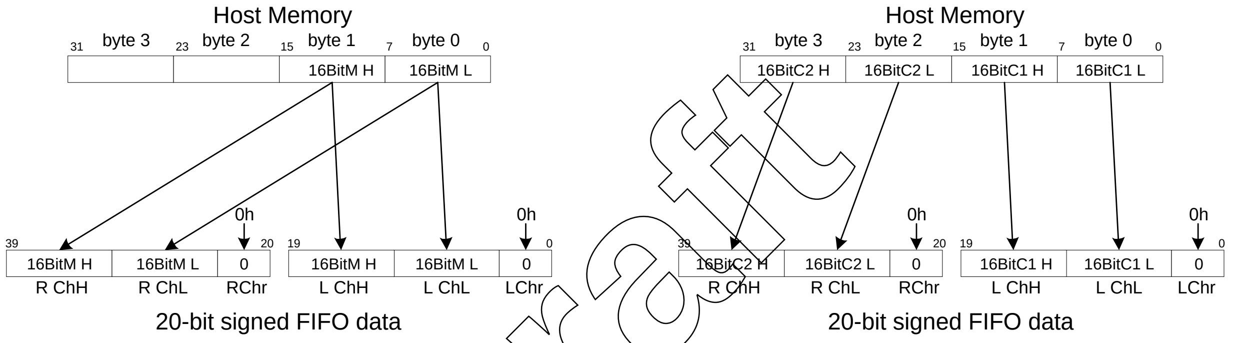


Figure 32. 16-Bit Little-Endian Formats

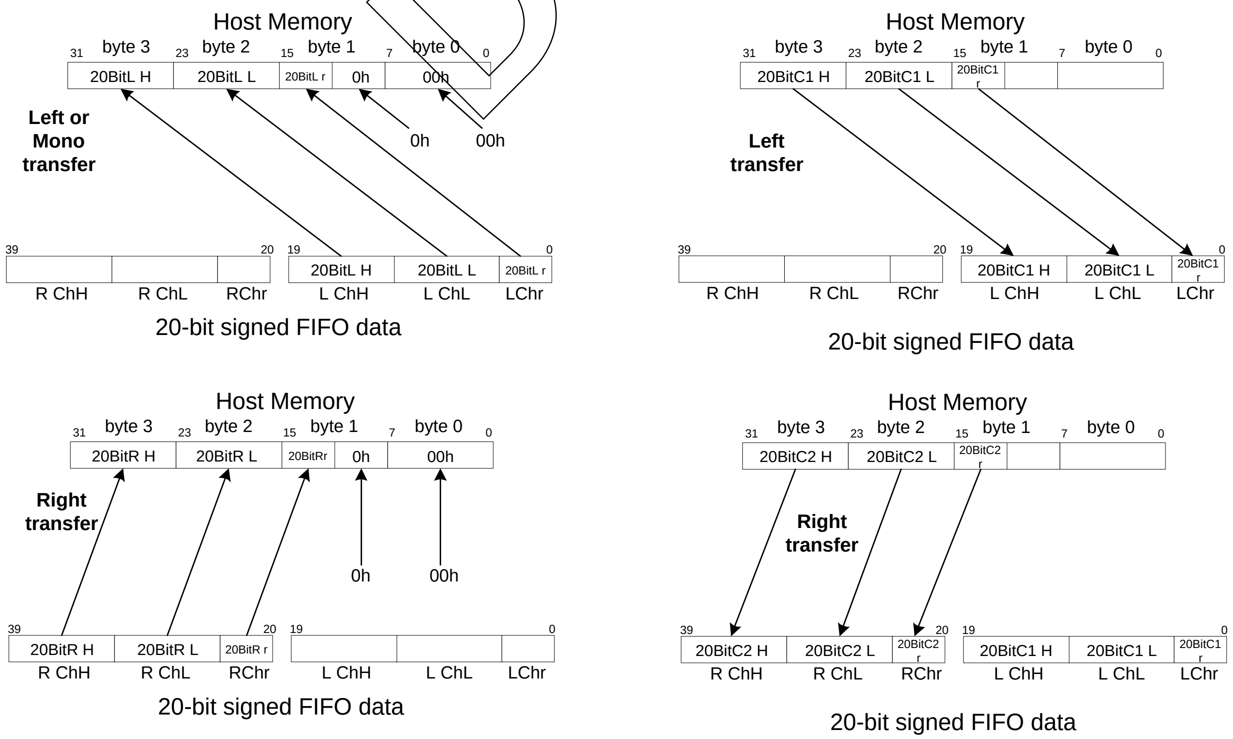


Figure 33. 20-Bit Stereo Little-Endian Formats

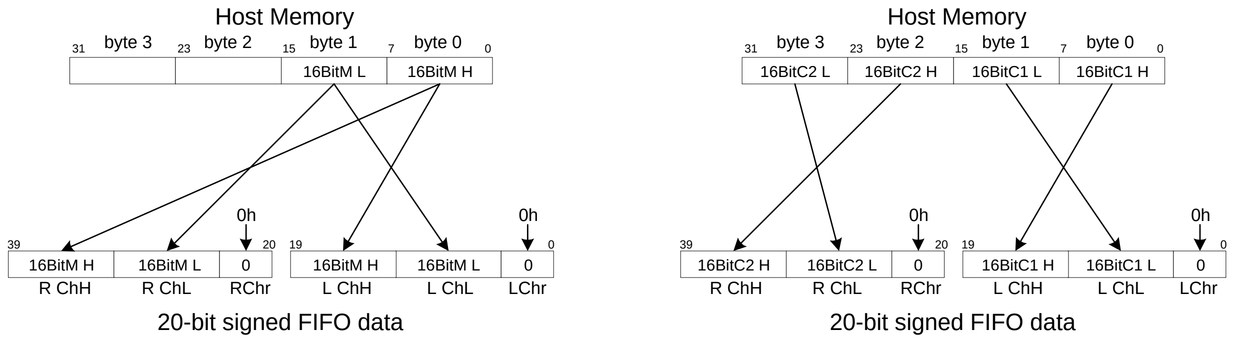


Figure 34. 16-Bit Big-Endian Formats

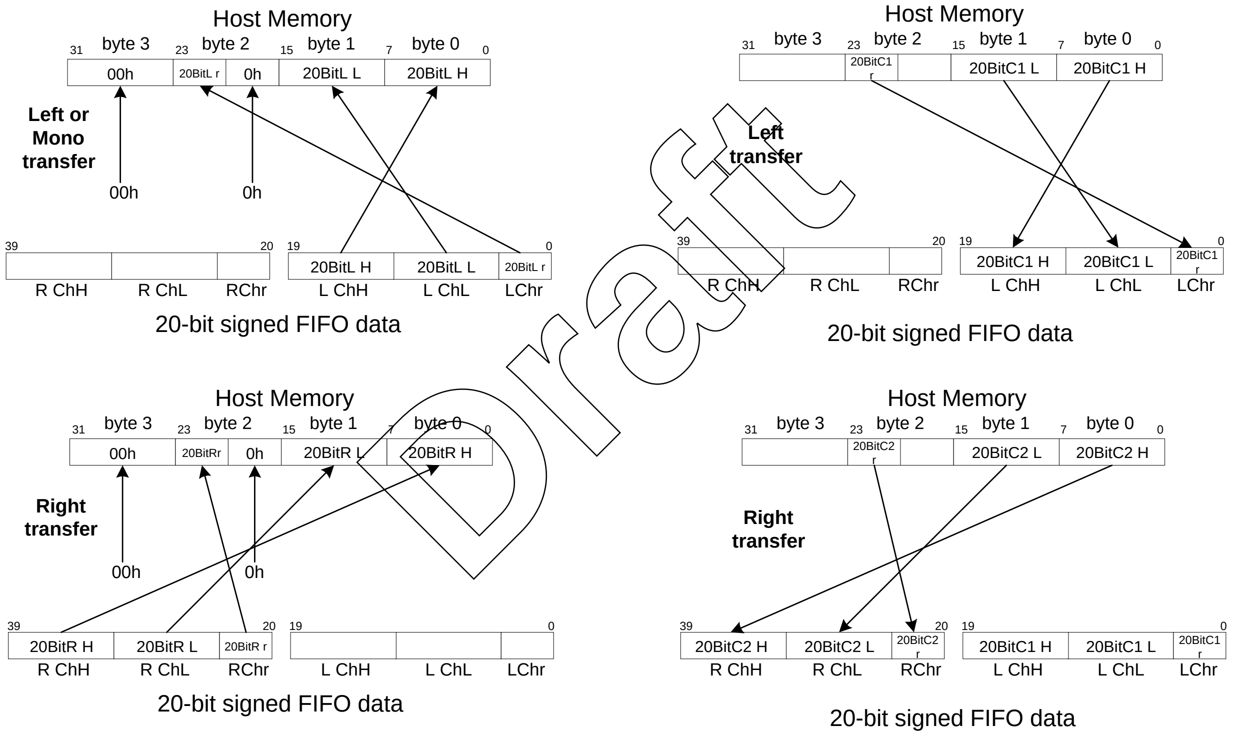


Figure 35. 20-Bit Stereo Big-Endian Formats

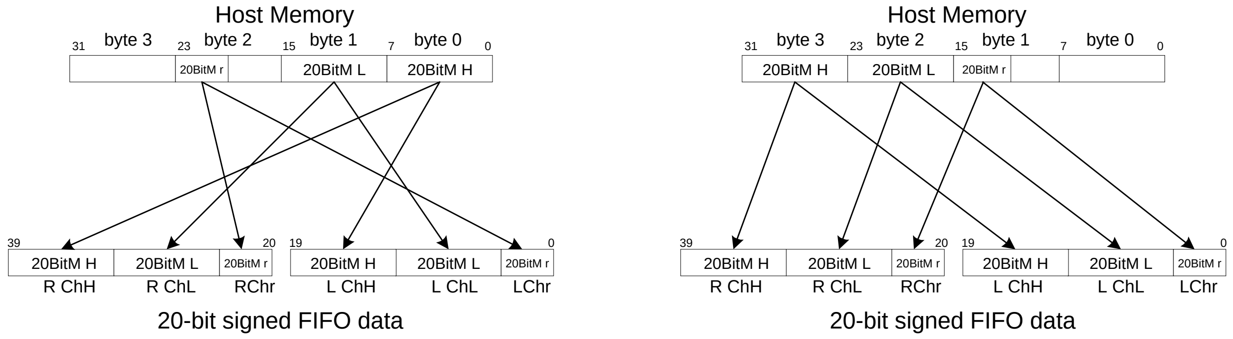


Figure 36. 20-Bit Playback Mono Formats

11.3 Differences from the 8237

Several important differences exist in the CS4281 emulation of the 8237 DMA controller when compared to the stand-alone 8237, or the DMA controller built into various chip sets and ISA bridge chips. Most differences arise from the fact that the CS4281 does not reside on the ISA bus, so that all the ISA specific items disappear. Since there is no ISA bus around, anything related to ISA timing selection, signal polarity selection, or other ISA specific function is not needed. Verify transfer types (as defined in the 8237 DMA controller) are not supported. Cascade operation is not supported. Block and demand transfer modes are supported but are emulated as single transfer operation. The CS4281 DMA will not support any sort of chaining or scatter-gather capability. The 8237 DMA Request Register functionality is not supported.

When the DMA controller becomes bus master, it will do one transfers before giving up the bus, regardless of whether a sample is mis-aligned or not. A maximum of one sample is transferred at a time even if other samples fit within the double-word alignment.

11.4 DMA Control Register Operation

DMA operations are setup as follows:

- The host Address Register is set to point at the data buffer, assuming the buffer has data.
- The Base Count Register is set to the transfer size.
- The Base Address Register is set to the transfer address
- The DMA Mode Register is loaded with format and transfer direction
- The FIFO Control register is setup for the FIFO configuration and AC Link slot selection.
- The AC Link Output Slot valid bits are set if playback operation
- And finally, the DMA Command Register is written to unmask (enable) the DMA engine.

For playback, the FIFO is the timing master for audio data transfers. The DMA engine will try and keep the FIFO full. For capture, the AC Link is the timing master for audio data transfers. The Sound System Controller (SSC) will transfer valid AC Slot data into the appropriate FIFO. The DMA engine will try to keep the FIFO empty by transferring data to system memory as it becomes available in the FIFO.

The DMA control registers mimic those of the 8237. The address pointer is simplified into a single register that provides the source/destination byte address. The concept of pages is ignored, each DMA buffer is assumed to be one linear chunk.

11.5 DMA Control Registers

11.5.1 DMA Base Address n (DBAn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BAUH7	BAUH6	BAUH5	BAUH4	BAUH3	BAUH2	BAUH1	BAUH0	BAUL7	BAUL6	BAUL5	BAUL4	BAUL3	BAUL2	BAUL1	BAUL0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAH7	BAH6	BAH5	BAH4	BAH3	BAH2	BAH1	BAH0	BAL7	BAL6	BAL5	BAL4	BAL3	BAL2	BAL1	BAL0

Address: BA0: 118h, Read-Write, for DMA Engine 0
BA0: 128h, Read-Write, for DMA Engine 1
BA0: 138h, Read-Write, for DMA Engine 2
BA0: 148h, Read-Write, for DMA Engine 3

Default: 00000000h

Definition: Core powered. Holds the host base address of the DMA data buffer. When this register is loaded, the DMA Current Address register is loaded with the same value. This register reloads the DMA Current Address Register during auto-initialize operation. In Host mode, this is a 32-bit linear base address. The DMAn controller can transfer data a sample at a time or a channel at a time. If **DMRn.TBC** is set, data will always be transferred a channel at a time regardless of alignment. If the data is stereo (**DMRn.MONO** = 0), the lower two bits of this register determine the number of bus cycles needed for one sample for 8- and 16-bit stereo formats. For 8-bit stereo format, if **BAL0** = 0, the sample is word aligned and the DMA controller will transfer complete samples in one PCI bus cycle. If **BAL0** = 1, the sample is mis-aligned and the DMA controller will transfer the samples in 2 PCI bus cycles - one for each channel. For 16-bit stereo format, **BAL0** must be 0 since channels are atomic on the CS4281. If **BAL1** = 0, the sample is dword aligned and the DMA controller will transfer complete samples in one PCI bus cycle. If **BAL1** = 1, the sample is mis-aligned and the DMA controller will transfer the samples in 2 PCI bus cycles - one for each 16-bit channel. See Table 2 below.

Bit Descriptions:

BAUH[7:0] Base Address - Upper word, High byte: Most significant byte of the base address in host mode.

BAUL[7:0] Base Address - Upper word, Low byte: Third byte of base address in host mode.

BAH[7:0] Base Address - lower word, High byte.

BAL[7:0] Base Address - lower word, Low byte

DMRn. SIZE20	DMRn .SIZE8	DMRn .MONO	DMRn .TBC	DBAn. BAL1	DBAn. BAL0	XFER Size	XFER Split	Description
-	1	0	0	x	0	16	no	8-bit stereo
-	1	0	1	x	0	8	yes	8-bit stereo; forced channel at a time
-	1	0	x	x	1	8	yes	8-bit stereo; mis-aligned
-	1	1	x	x	x	8	n/a	8-bit mono
0	0	0	0	0	0*	32	no	16-bit stereo
0	0	0	1	0	0*	16	yes	16-bit stereo; forced channel at a time
0	0	0	x	1	0*	16	yes	16-bit stereo; mis-aligned
0	0	1	x	x	0*	16	n/a	16-bit mono
1	0	0	x	0*	0*	32	yes	20-bit stereo; always mis-aligned (PCI bus 32 bits, stereo sample 40 bits)
1	0	1	x	0*	0*	32	n/a	20-bit mono

18. * - must be 0. Mis-aligned channels are not supported in the CS4281
19. x - don’t care

11.5.2 DMA Current Address n (DCAn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAU15	CAU14	CAU13	CAU12	CAU11	CAU10	CAU9	CAU8	CAU7	CAU6	CAU5	CAU4	CAU3	CAU2	CAU1	CAU0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAH7	CAH6	CAH5	CAH4	CAH3	CAH2	CAH1	CAH0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0

Address: BA0: 110h, Read-Write, for DMA Engine 0
BA0: 120h, Read-Write, for DMA Engine 1
BA0: 130h, Read-Write, for DMA Engine 2
BA0: 140h, Read-Write, for DMA Engine 3

Default: 00000000h

Definition: Core powered. Holds the address of the DMA data buffer where data will be read from or written to. It will be auto-incremented or auto-decremented on each DMA transaction to always point at the next sample or channel. This register is automatically loaded with DBA data when the DMA Base Address Register (DBA) is written. The DBA data is also loaded into this register on terminal count (TC) when auto-initialize is set, **DMRn.AUTO**. In Host mode, this register is treated as a 32-bit linear address.

Bit Descriptions:

CAU[15:0] Current Address, Upper word: In Host mode, this word is changed by address increment/decrement and is just the upper word of the linear 32-bit address.

CAH[7:0] Current Address, lower word, Upper byte.

CAL[7:0] Current Address, lower word, Lower byte.

11.5.3 DMA Base Count n (DBCn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCU15	BCU14	BCU13	BCU12	BCU11	BCU10	BCU9	BCU8	BCU7	BCU6	BCU5	BCU4	BCU3	BCU2	BCU1	BCU0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH7	BCH6	BCH5	BCH4	BCH3	BCH2	BCH1	BCH0	BCL7	BCL6	BCL5	BCL4	BCL3	BCL2	BCL1	BCL0

Address: BA0: 11Ch, Read-Write, for DMA Engine 0
BA0: 12Ch, Read-Write, for DMA Engine 1
BA0: 13Ch, Read-Write, for DMA Engine 2
BA0: 14Ch, Read-Write, for DMA Engine 3

Default: 00000000h

Definition: Core powered.Holds the DMA base transfer count. The count should be one less then the number of transfers (channels or samples) since interrupts occur on roll-unders. When any byte of this register is loaded, the same Current Count register byte is loaded. This register also reloads the DMA Current Count Register during auto-initialize operation.

Bit Descriptions:

BCU[15:0] Base transfer Count, Upper word.

BCH[7:0] Base transfer Count, lower word, High byte.

BCL[7:0] Base transfer Count, lower word, Low byte.

11.5.4 DMA Current Count n (DCCn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCU15	CCU14	CCU13	CCU12	CCU11	CCU10	CCU9	CCU8	CCU7	CCU6	CCU5	CCU4	CCU3	CCU2	CCU1	CCU0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCH7	CCH6	CCH5	CCH4	CCH3	CCH2	CCH1	CCH0	CCL7	CCL6	CCL5	CCL4	CCL3	CCL2	CCL1	CCL0

Address: BA0: 114h, Read-Write, for DMA Engine 0
BA0: 124h, Read-Write, for DMA Engine 1
BA0: 134h, Read-Write, for DMA Engine 2
BA0: 144h, Read-Write, for DMA Engine 3

Default: 00000000h

Definition: Core powered. Holds the current DMA transfer count. The count may be channels or samples depending on **DMRn.CBC**. If **CBC** is cleared, then this register is decremented once per sample. If **CBC** is set, then the decrement is once per channel. For mono data, 1 sample = 1 channel. For stereo data, 1 sample = 2 channels (decrement twice if **CBC** set). DCCn will be auto-decremented on each DMA transaction to always hold the number of transfers to go + 1. When the count transitions from 00000000 to FFFFFFFFh, TC for this engine is generated. If auto-initialize is set (**DMRn.AUTO**), then the DMA Base Count register (DBCn), will be loaded into this register and counting will continue. This register is also written when the DBCn register is written. When the count reaches (Base Count n) / 2, **HTC** (half **TC**) for this engine is set. Interrupts can be generated on **TC** and **HTC**. For host access, HDSRn contains the interrupt status, and DCRn contains the interrupt enables.

Bit Descriptions:

CCU[15:0] Transfer count upper word.

CCH[7:0] Transfer count high byte.

CCL[7:0] Transfer count low byte.

11.5.5 DMA Mode Register n (DMRn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DMA	POLL			TBC	CBC		SWAPC		SIZE20	USIGN	BEND	MONO	SIZE8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TYPE1	TYPE0	DEC	AUTO	TR1	TR0		

Address: BA0: 150h, Read-Write, for DMA Engine 0
BA0: 158h, Read-Write, for DMA Engine 1
BA0: 160h, Read-Write, for DMA Engine 2
BA0: 168h, Read-Write, for DMA Engine 3

Default: 00000000h

Definition: Core powered. This register hold the mode and format conversion bit fields for the DMA engine. This register also controls the format for the FIFO block in Polled mode. The DMAn engine is enabled whenever either **POLL** or **DMA** is enabled. If **DMA** = **POLL** = 0, this DMA engine is idle and moves no data. The lower byte corresponds to the 8237 mode register. The upper word controls the data format conversion performed by the engine. See the *Data Format Conversion* section for details of data conversions.

Bit Descriptions:

DMA When set, DMAn is enabled in DMA mode. **DMA** and **POLL** are mutually exclusive. When **DMA** goes from a 0 to a 1, the DMA engine is placed in an initial state with **HDSRn.CH1P** and **HDSRn.CH2P** in their initial state. After **DMA** is set, **DCRn.MSK** is used to control DMAn starting and stopping.

POLL When set, DMAn is enabled and in polled I/O mode. **DMA** and **POLL** are mutually exclusive. When **POLL** is set, DMAn manages the transfer of data between the PCI bus (Host) and FIFOn. **POLL** transferring from 0 to 1 forces the **HDSRn.CH1P** and **HDSRn.CH2P** bits to their initial state and initializes the DMAn controller for polled operation.

TBC Transfer By Channel. When set, this bit forces stereo data to be transferred one channel at a time across the PCI bus for 8- and 16-bit data formats. When clear, the DMA controller decides whether to transfer stereo data as a sample or one channel at a time, based on the Base Address (DBAn) lower two address bits and the sample size (**SIZE8**).

CBC Count By Channel. When clear, the DMA Count Register, DCCn, counts by sample: decrements once per sample in mono or stereo (2 channel) format. When **CBC** is set, DCCn counts by channel: decrements once per sample in mono format, twice per sample in stereo format (once for left channel, once for right channel). Note that this bit is autonomous to the DMA's ability to transfers channels or samples across the PCI bus (which is controlled through data format and Base addresses).

The following bits: **SIZE20**, **USIGN**, **BEND**, **MONO**, and **SIZE8** define the host memory data format for DMA and Polled FIFO mode. They have no effect on host reads and writes directly into FIFO memory.

SWAPC Swap channels. Swaps the two channels interfacing with the PCI bus.

SIZE20 Sample is 20-bit (4 bytes) audio data. This bit and **SIZE8** cannot both be active. For host-to-FIFO transfers, the 20 most significant bits of a 32-bit dword are stored. For FIFO-to-host transfers, the data is MSB-aligned in a 32-bit dword, with the unused lower 12 bits set to 0.

USIGN Sample is unsigned. When set, the MSB will be inverted since the internal format must be signed.

BEND Host data is big endian. **BEND** = 0 is little endian format. This bit affects 16- and 20-bit transfers only.

MONO Sample is monaural. FIFO-to-host transfers use the left channel of FIFO if **SWAPC** is clear. For host-to-FIFO transfers, the mono data is expanded to both channels and input into the FIFO. For AC-Link slots that are only one channel, the other channel (right) can be disabled by setting the channel slot (**FCRn.RS[4:0]** for right) to 31. Slot value 31 disables a FIFO channel on the AC Link side.

SIZE8 Sample is 8 bit. This bit and the **SIZE20** bit cannot both be active. (**SIZE8** takes precedence over **SIZE20**).

TYPE[1:0] Transfer Mode Select. (used in DMA and Polled FIFO modes)
00 - Demand mode select
01 - Single mode select
10 - Block mode select
11 - Cascade mode select - Not supported.

DEC Address Increment or decrement control: 0 - Increment, 1 - Decrement.

AUTO Auto-Initialize control: 0 Auto-initialize disable, 1 Auto-initialize enable

TR[1:0] Transfer Type Control: (used in DMA and Polled FIFO mode)
00 - Verify transfer - Not supported.
01 - Write transfer, data goes from FIFO RAM to host memory - Record
10 - Read transfer, data goes from host memory to FIFO RAM - Playback
11 - Reserved

11.5.6 DMA Command Register n (DCRn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														HTCIE	TCIE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															MSK

Address: BA0: 154h, Read-Write, for DMA Engine 0
BA0: 15Ch, Read-Write, for DMA Engine 1
BA0: 164h, Read-Write, for DMA Engine 2
BA0: 16Ch, Read-Write, for DMA Engine 3

Default: 00000000h

Definition: Core powered. This register holds the mask and interrupt control for the DMA engine. DCRn is NOT the Command register of the legacy 8237. The upper word selects the interrupt functionality of this DMA engine for host operation. Interrupt status and clearing occurs in register HDSRn. If a PCI **REQ#** request is pending when host software sets this bit, the transfer is completed to get the state machine into an idle state.

Bit Descriptions:

HTCIE Half Terminal Count Interrupt Enable. When set, an interrupt is generated on Half TC (midway through buffer), when used in DMA mode. To support ping-pong buffers both **HTCIE** and **TCIE** should be set which will cause interrupts at both the half- and full-buffer locations.

TCIE Terminal Count Interrupt Enable. When set, an interrupt is generated on TC (end of buffer) when used in DMA mode. To support ping-pong buffers both **HTCIE** and **TCIE** should be set which will cause interrupts at both the half- and full-buffer locations.

MSK DMA Mask bit. This bit controls the enabling of the DMAn controller. When **MSK** is clear, the DMA engine starts operating. When **MSK** is set, the DMA engine is paused. Software will generally setup the DMA registers and then clear **MSK** to start operation. If not in auto-initialize mode (**DMRn.AUTO** clear), **MSK** is set on reaching terminal count (DCCn count register rolls under). .

11.5.7 Host DMA Status Register n (HDSRn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						CH1P	CH2P							DHTC	DTC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRUN								RQ							

Address: BA0: 0F0h, Read-Write, for DMA Engine 0
BA0: 0F4h, Read-Write, for DMA Engine 1
BA0: 0F8h, Read-Write, for DMA Engine 2
BA0: 0FCh, Read-Write, for DMA Engine 3

Default: 00000000h

Definition: Holds the status bits for DMA engine n. The **TC[a:d]**, **DTC**, and **DHTC** bits are cleared when this register is read. Interrupts are enabled in the DCRn register. For DMA mode (**DMRn.DMA** = 1), the available status/interrupts are **DTC** and **DHTC**. The **DTC** bit is a repeat of the respective **TC[d:a]** bits, but cleared independently. For host operation, the upper word of this register should be used for clearing DMA related interrupts. For DMA interrupt logic diagram, see Figure 21. **DHTC** and **DTC** are forced clear if **DMRn.DMA** = 0. .

Bit Descriptions:

CH1P Channel 1 Pending. This bit is set when DMAn is requesting the first channel data, mono data, or when stereo samples are transferred as an atomic unit. This bit is active in DMA or POLL modes (DMRn). **CH1P** is generally the left channel, but is the right channel when **DMRn.SWAPC** is set.

CH2P Channel 2 Pending. This bit is set when DMAn is requesting the second channel data. This bit is always clear when sending mono data or when stereo samples are transferred as an atomic unit. This bit is active in DMA or POLL modes (DMRn). **CH1P** is generally the right channel, but is the left channel when **DMRn.SWAPC** is set.

DHTC DMA Half Terminal Count. When set, indicates this DMA engine is half way through the buffer. This bit is cleared by reading this byte. Forced clear when this engine not in DMA mode **(DMRn.DMA** = 0). This interrupt is enabled by **DCRn.HTCIE**.

DTC DMA Terminal Count. When set, indicates this DMA engine is passed the end of the buffer (rolled under). **DTC** is forced clear when this engine not in DMA mode (**DMRn.DMA** = 0). This interrupt is enabled by **DCRn.TCIE**.

DRUN DMA Running. When the DMA engine starts a transaction, this bit is set. When the DMA engine completes the transaction, this bit is clear. If **DRUN** and **RQ** bits are both clear, the DMA engine is idle. When **DRUN** is set, the DMA arbiter will not change to the next DMA engine until **DRUN** is clear.

RQ Set when the DMAn controller has a request pending (is requesting service).

11.5.8 *FIFO Control Register n (FCRn)*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FEN	DACZ	PSH	RS4	RS3	RS2	RS1	RS0				LS4	LS3	LS2	LS1	LS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SZ6	SZ5	SZ4	SZ3	SZ2	SZ1	SZ0		OF6	OF5	OF4	OF3	OF2	OF1	OF0

Address: BA0: 180h, Read-Write, for FIFO 0
BA0: 184h, Read-Write, for FIFO 1
BA0: 188h, Read-Write, for FIFO 2
BA0: 18Ch, Read-Write, for FIFO 3

Default: 1F1F0000h

Definition: Core powered. Contains the enable, starting address offset, size, and AC-Link slot ID mappings for each FIFO. Note that the **SZ** and **OF** bits should only be changed when **FEN** is clear. For host software **RS/LS** bits should also only be changed when **FEN** = 0.

Bit Descriptions:

FEN FIFO Enable bit. When set, FIFOn runs normally. When clear, FIFOn is flushed (read and write pointers reset), status bits in FSICn are cleared, and FIFOn is powered down.

DACZ DAC Zero. This bit controls what the FIFOs output on an underrun condition while the FIFO is operating (**FEN** = 1).
0 - Output last sample when empty FIFO read. If no previous samples exist, zeros are output.
1 - Output zero when empty FIFO read.

PSH Previous Sample Hold. When the FIFO is halted (**FEN** = 0) and a data read is requested, **PSH** is used instead of **DACZ** to decide what data to output.
0 - Play zero's when FIFOn underrun occurs while halted.
1 - Hold last sample when FIFOn underrun occurs while halted

RS[4:0] Right Slot mapping for FIFOn. Connects an AC-Link slot to the right channel of FIFOn. If **RS[4:0]** matches a channel in one of the sample rate converters (SRC), the SRC is placed between this FIFO and the AC Link. Only one FIFO can be linked to an SRC, playback or record (SRCSA). Said another way, the left and right channels of an SRC can only be assigned to one FIFOn (cannot split across FIFOs). See Table 29 for playback (host memory-FIFO-AC Link) and Table 30 for record (AC Link-FIFO-host memory). For recording mono channels from the AC Link, the left side of the FIFO (**LS[4:0]**) must be used to transfer the data with the **RS[4:0]** bits set to 31.

LS[4:0] Left Slot mapping for FIFOn. Connects an AC Link slot to the left channel of FIFOn. If **LS[4:0]** matches a channel in one of the sample rate converters (SRC), the SRC is placed between this FIFO and the AC Link. Only one FIFO can be linked to an SRC, playback or record (SRCSA). Said another way, the left and right channels of an SRC can only be assigned to one FIFOn (cannot split across FIFOs). See Table 37 for playback (host memory-FIFO-AC Link) and Table 30 for record (AC Link-FIFO-host memory).

SZ[6:0] FIFO buffer size, in samples (samples are 40-bit quantities that include up to 2 20-bit words). Can only be changed when **FEN** = 0.

OF[6:0] Buffer starting offset, in samples, from the beginning of FIFO memory. Can only be changed when **FEN** = 0.

11.5.9 FIFO Status and Interrupt Control Register n (FSICn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIC6	FIC5	FIC4	FIC3	FIC2	FIC1	FIC0	FORIE	FURIE						FSCIE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0	FOR	FUR						FSCR

Address: BA0: 210h, Read-Write, for FIFO 0
BA0: 214h, Read-Write, for FIFO 1
BA0: 218h, Read-Write, for FIFO 2
BA0: 21Ch, Read-Write, for FIFO 3

Default: 00000000h

Definition: Core powered. Contains the status bits for FIFO n. For Polled FIFO interrupt logic diagram, see Figure 22. The lower byte of this register is cleared when read. This status byte can generate an interrupt if the respective interrupt enables are set and **HISR.FnIM** is clear and **HISR.FIFOIM** is clear.

Bit Descriptions:

FIC[6:0] FIFO Interrupt Count. When FIFOn has this many samples in it, **FSCI** is set. Host software can set this value to generate an interrupt on any depth. For example, an empty FIFO (**FIC** = 0), a Full FIFO (**FIC** = **FCRn.SZ**), a FIFO not full condition (**FIC** = **FCRn.SZ**-1), or a FIFO half way point (**FIC** = **FCRn.SZ**/2).

FORIE FIFO OverRun Interrupt Enable. When set, an interrupt can be generated on an overrun condition.

FURIE FIFO UnderRun Interrupt Enable. When set, an interrupt can be generated on an underrun condition.

FSCIE FIFO Sample Count Interrupt Enable. When set, a interrupt can be generated when the FIFO Interrupt Count equals the FIFO Sample Count (number of samples in FIFO), **FIC** = **FSC**.

FSC[6:0] FIFO Sample Count. Indicates the number of valid samples currently in the FIFO.

FOR FIFO Overrun. Indicates a write to a full FIFO occurred. The data could be from the AC-Link side or the DMAn controller, based on FIFO direction. A full FIFO cannot be written and the new data goes to the bit bucket.

FUR FIFO Underrun. Indicates the FIFO was read while empty. The read could be from the AC-Link side or the DMAn controller, based on FIFO direction. The value read is based on **FCRn.DACZ**. If **DACZ** set, zero is read from an empty FIFO. If **DACZ** is clear, the last sample is read. If the FIFO has no samples (just turned on), the value read is zero.

FSCR FIFO Sample Count Reached. Set when the FIFO Interrupt Count equals the FIFO Sample Count, **FIC** = **FSC**.

11.5.10 FIFO Polled Data Register n (FPDRn)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDU15	FDU14	FDU13	FDU12	FDU11	FDU10	FDU9	FDU8	FDU7	FDU6	FDU5	FDU4	FDU3	FDU2	FDU1	FDU0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FDH7	FDH6	FDH5	FDH4	FDH3	FDH2	FDH1	FDH0	FDL7	FDL6	FDL5	FDL4	FDL3	FDL2	FDL1	FDL0

Address: BA0: 190h, Read-Write, for FIFO Engine 0
BA0: 194h, Read-Write, for FIFO Engine 1
BA0: 198h, Read-Write, for FIFO Engine 2
BA0: 19Ch, Read-Write, for FIFO Engine 3

Default: 00000000h

Definition: Core powered. This register transfers data to/from the FIFOs in polled data mode from/to the PCI bus. To allow polled FIFO operation, the **DMRn.POLL** bit must be set. The direction of transfer and data format are also set in the same register, **DMRn.TR[1:0]**. Data reads when set for host-to-FIFO direction will read bogus data. Data writes when set for FIFO-to-host direction will just go in the bit bucket (the FIFO controller will ignore the data). Data transfers are atomic based on the attached **DMRn.TBC** bit. If data transfers are atomic on a channel basis, both channels must be written (back-to-back) when in stereo mode for the data to actually get into the FIFO. 20-bit stereo data is transferred as two writes, one for left and one for right and is always atomic on a channel basis.

Bit Descriptions:

FDU[15:0] FIFO Data upper word.

FDH[7:0] FIFO Data lower word, High byte.

FDL[7:0] FIFO Data lower word, Low byte

11.5.11 *FIFO Channel Status (FCHS)*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCO3	LC03	MRP3	FE3	FF3	IOR3	RCI3	LCI3	RCO2	LCO2	MRP2	FE2	FF2	IOR2	RCI2	LCI2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCO1	LCO1	MRP1	FE1	FF1	IOR1	RCI1	LCI1	RCO0	LCO0	MRP0	FE0	FF0	IOR0	RCI0	LCI0

Address: BA0: 20Ch, Read-Only

Default: 18181818h

Definition: Core powered. These bits are generally used for debug only. Each FIFOn has 6 bits that indicate when only one channel of the FIFO is read or written and overrun/underrun internal status. When both channels are read or written, the read/write pointer is incremented and the associated channel status bits are cleared. When **FCRn.FEN** clear, all bits associated with that FIFOn are forced clear, except the **FE**/**FF** bits which are set when the frame-clock synchronized **FEN** is cleared. When **FEN** is set, **FF** going clear indicates that the internal frame-clock synchronized **FEN** is set. Writing to or reading from the FIFO in polled mode should not occur until **FF** is read clear.

Bit Descriptions:

LCI[3:0] Left Channel In. When set, indicates that - only - the left channel has been written at the current FIFO write pointer location. The write pointer will not be moved until the other channel is received.

RCI[3:0] Right Channel In. When set, indicates that - only - the right channel has been written at the current FIFO write pointer location. The write pointer will not be incremented until the other channel is received.

IOR[3:0] Internal Overrun Flag. Set when one channel written to a full FIFO. Used in synchronizing channel-to-channel overrun conditions.

FF[3:0] FIFO Full. Set when FIFO full. Both **FF** and **FE** will be set when **FEN** for a particular FIFO is clear.

FE[3:0] FIFO Empty. Set when FIFO empty. Both **FF** and **FE** will be set when **FEN** for a particular FIFO is clear.

LCO[3:0] Left Channel Out. When set, indicates that - only - the left channel at the current FIFO read pointer location has been read. The read pointer will not be incremented until the other channel is read.

RCO[3:0] Right Channel Out. When set, indicates that - only - the right channel at the current FIFO read pointer location has been read. The read pointer will not be incremented until the other channel is read.

MRP[3:0] Move Read Pointer. Set when FIFO empty and read pointer needs to move the next time the write pointer moves. Supports previous sample data out (**DACZ**) when the FIFO is empty.

LS[4:0]/ RS[4:0]	AC ‘97 Output Slot	AC ‘97 Slot Function (primary & secondary codec)
0	3	Left PCM Playback
1	4	Right PCM Playback
2	5	Phone Line 1 DAC
3	6	Center PCM Playback
4	7	Left Surround PCM Playback
5	8	Right Surround PCM Playback
6	9	LFE PCM Playback
7	10	Phone Line 2 DAC
8	11	HeadSet DAC
29	Left WT	Data sent to CS4281 Left FM/WT digital mixer
30	Right WT	Data sent to CS4281 Right FM/WT digital mixer
31	None	FIFO channel is not used (data thrown away)

20. 1. DMRn.TR[1:0] must equal 10 (transfer from host memory to FIFO).

21. 2. SRC channels must both come from one FIFO. SRC left and right cannot come from different FIFOs and cannot be swapped.

LS[4:0]/ RS[4:0]	AC ‘97 Input Slot	AC ‘97 Codec	AC ‘97 Slot Function
10	3	Primary	Left PCM Record
11	4	Primary	Right PCM Record
12	5	Primary	Phone Line 1 ADC
13	6	Primary	Mic ADC
14	7	Primary	reserved
15	8	Primary	reserved
16	9	Primary	reserved
17	10	Primary	Phone Line 2 ADC
18	11	Primary	HeadSet ADC
20	3	Secondary	Left PCM Record
21	4	Secondary	Right PCM Record
22	5	Secondary	Phone Line 1 ADC
23	6	Secondary	Mic ADC
24	7	Secondary	reserved
25	8	Secondary	reserved
26	9	Secondary	reserved
27	10	Secondary	Phone Line 2 ADC
28	11	Secondary	HeadSet ADC
31*	none		Channel is not used, data returned is fixed at 0

22. * The unused channel must be RS[4:0] when recording a mono stream from the AC Link.

23. 1. 1. DMRn.TR[1:0] must equal 01 (transfer from FIFO to host memory).

24. 2. SRC channels must both come from one FIFO. SRC left and right cannot come from different FIFOs and cannot be swapped.

11.6 FIFO Hardware Implementation

The DMA engine configuration creates a 1 to 1 correspondence between host buffers and FIFOs. The FIFO configuration extends the correspondence to the AC-Link data slots. The FIFO RAM starting address is mapped into PCI memory space by the PCI configuration Base Address 1. The FIFO RAM is accessible from the PCI bus anytime that DMA is not the bus master. This lets host software snoop the DMA operation for debug and test purposes. The FIFO-to-slot assignments are controlled by the FIFO control registers. Each FIFO is composed of a left and a right half that can be separately routed to/from the AC-Link slots. The FIFO-to-slot assignment **FCRn.LS/RS** bits determine the direction of the FIFO half connected to the AC-link slots. The **DMRn.TR[1:0]** bits determine the direction of transfers between the PCI bus and the FIFO for both Polled and DMA modes. Both sides of the FIFO **must be** programmed for the same FIFO direction (PC bus to FIFO to AC Link, or AC link to FIFO to PC bus).

In the CS4281, the starting addresses and sizes are adjustable, so host software can fine tune the FIFO depth for different applications. The FIFO depth and offset can only be changed when the FIFO is inactive (**FCRn.FEN** = 0). Each FIFO is defined by a starting offset and length in stereo samples. The default design will allocate 32 samples for each stream. Each sample will be capable of holding stereo 40-bit audio data, so the physical RAM size is: (see Figure 37)

Logically (to host software), each sample will appear as two double words in the host address space. The PCI side will be able to control exactly which channel of the sample pair gets read or written. FIFO pointers do not move when the host accesses the FIFO RAM directly. The *formatter* converts the 20-bit internal FIFO data to the PCI-bus 32-bit double-word little-endian format. The logical RAM size is: (see Figure 38)

The FIFO RAM architecture is illustrated in Figure 39.

The FCRn registers contain the Physical Address Offset, **FCRn.OF[6:0]** and FIFO Size, **FCRn.SZ[6:0]**. Seven bits for each covers the entire FIFO RAM ending at location 127. These bits can only be changed when the FIFO is disabled (**FCRn.FEN** clear).

The FIFO data is accessed by the Sound System Controller (SSC) connected to the AC Link, or the DMAn controller/formatter that transfers data to/from the PCI bus. The SSC writes to FIFO's based on the AC-Link slot assignments (which determine direction). The DMAn writes to FIFO's based on the **DMRn.TR[1:0]** transfer direction bits.

$$20\ \frac{\text{bits}}{\text{channel}} \times 2\ \frac{\text{channels}}{\text{sample}} \times 32\ \frac{\text{samples}}{\text{stream}} \times 4\ \text{streams}\ =\ 5120\ \text{bits}$$

$$3\ \frac{\text{bytes}}{\text{channel}} \times 2\ \frac{\text{channels}}{\text{sample}} \times 32\ \frac{\text{samples}}{\text{stream}} \times 4\ \text{streams}\ =\ 768\ \text{bytes}$$

Figure 37. FIFO Hardware Physical RAM Size equation

$$4\ \frac{\text{bytes}}{\text{channel}} \times 2\ \frac{\text{channels}}{\text{sample}} \times 32\ \frac{\text{samples}}{\text{stream}} \times 4\ \text{streams}\ =\ 1024\ \text{bytes}$$

Figure 38. FIFO Hardware Logical RAM Size equation

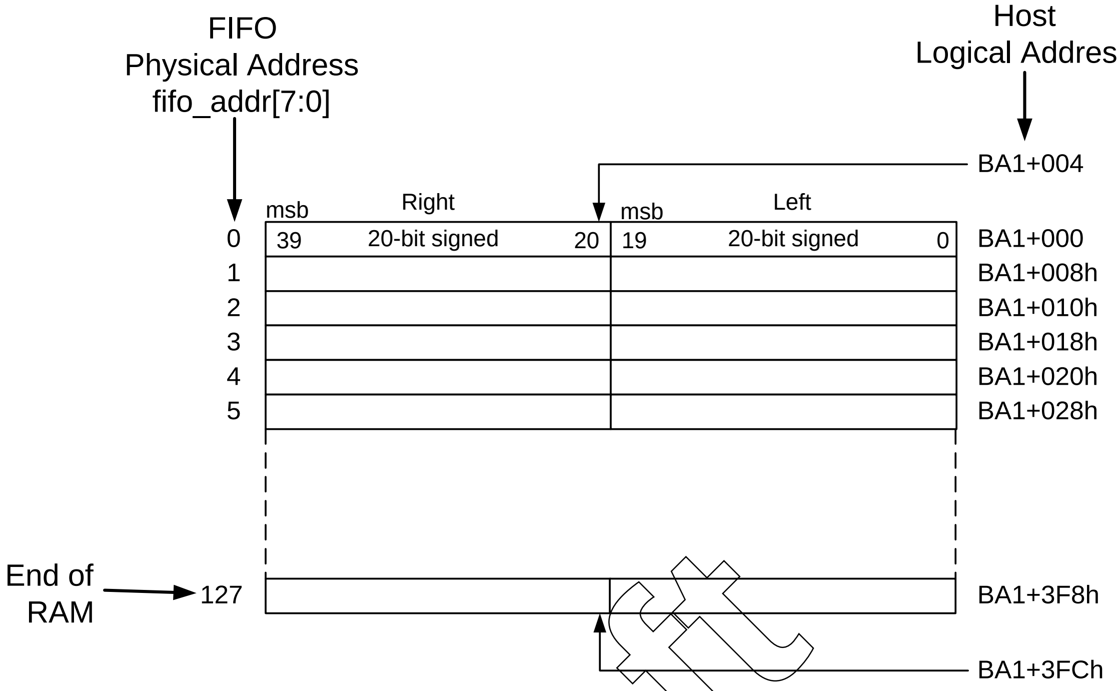


Figure 39. FIFO Memory Architecture

The number of samples currently available in a FIFO are found in **FSICn.FSC[6:0]**. A FIFO arbiter controls whether the SSC data or DMAn controller data (from the Formatter) gets access to the FIFO RAM.

The FIFOn controllers maintain status for their particular FIFOs. Three conditions are flagged and can generate interrupts to the host. The FIFOn controller generates a pulse and sets the *FIFO Sample Count Reached*, **FSICn.FSCR**, bit when the **FSICn.FIC** count matches the number of samples currently available in the FIFO (**FSICn.FSC** bits). Host software can set the **FSICn.FIC** bits to generate an interrupt/status on an empty FIFO, half/way through the FIFO, a full FIFO, or anywhere in-between. This bits can generate an interrupt if unmasked and are cleared by reading the corresponding FSICn register.

If a FIFOn is full and new data is sent to FIFOn, the FIFOn controller blocks the write forcing the data in the bit bucket. The FIFOn write pointer is NOT changed. The FIFO controller then generates an overrun condition/pulse by setting the **FSICn.FOR** bit. This bit remains set until the host reads FSICn.

A FIFO could be empty under two conditions, if it was just turned on or if the FIFO is active and data was read from the FIFO faster than data is written into the FIFO. If FIFOn is empty and no data was ever in the FIFO, the FIFOn controller sends a signal to the data bus buffer forcing zero to be read. If FIFOn is empty and old data exists in the FIFO, the FIFOn controller responds based on **FCRn.DACZ**. If **DACZ** (DAC Zero) is set, the FIFOn controller sends a signal to the data bus buffer forcing zero to be read. If **DACZ** is clear, the FIFOn controller allows a read to occur from the last location that was read with valid data. The read pointer is NOT changed. On reading an empty FIFOn, the FIFOn controller generates an underrun condition/pulse by setting the **FSICn.FUR** bit. This bit remains set until the host reads FSICn.

The FIFO controller maintains read and write pointers for sample pairs only; however, a mechanism is provided that allows a single channel to be read/written at a time. The FIFO controller has two signals from each side (DMA controller and Sound System Controller) on the read/write operation indicating which channel is being read/written: Left, Right, or Both. Whenever a channel is written or read, a bit is set indicating the channel has been accessed. These bits are located in the **FCHS** register for debug purposes. Each FIFO pointer has a bit for each channel (four bits per FIFO). When both channels are accessed, the FIFO pointer moves and the status bits are reset. As an example:

- The DMA engine writes the left channel of FIFO2 - which sets **FCHS.LCI2**
- On the next PCI bus cycle, the DMA engine writes the right channel of FIFO2 - which sets **FCHS.RCI2**
- When both channel bits are set (**LCI2/RCI2**), the FIFO2 controller moves the FIFO2 write pointer and clears **LCI2** and **RCI2**.

Another example:

- The DMA engine writes mono data into the formatter - Channel 1 valid
- The formatter expands the mono data to stereo data
  - sends the FIFO both a left channel and a right channel valid signal
- The FIFO loads both channels, both channel bits are set (**LCI2/RCI2**)
- The FIFO moves the write pointer since both channel bits are set and then resets the channel bits.

A third example:

- The Sound System Controller (SSC) gets AC Link primary slot 3 and sends it to FIFO1 left channel
  - FIFO1 controller sets **LCI1**.
- The SSC gets AC-Link secondary slot 3 and sends it to FIFO3 left channel
  - FIFO3 controller sets **LCI3**
- The SSC gets AC-Link primary slot 4 and sends it to FIFO1 right channel
  - FIFO1 controller sets **RCI1**. Now both bits are set, write pointer moved, both status bits cleared.
- The SSC gets AC-Link secondary slot 4 and sends it to FIFO3 right channel
  FIFO3 controller sets **RCI3**. Now both bits are set, write pointer moved, both status bits cleared.

The FIFO location is considered empty until both channels are entered and the FIFO write pointer is moved. If the DMA engine writes the same FIFO channel twice, the new data overwrites the old and the FIFO still waits for the other channel to increment the write pointer. The write pointer always points to the next sample to be written.

Special cases (and internal flags) are needed to handle underrun and overrun conditions now that the FIFO can be accessed one channel at a time; otherwise, the FIFO could get out of sync. with the rest of the system. A FIFO write pointer points to an empty location unless the FIFO is full. The write pointer ONLY moves when both *valid channel* signals are received. The FIFO keeps flags to indicate which channel (or both) the FIFO is waiting for. On overrun conditions, **the valid channel flags are set regardless of the fact that the data is not stored**. If host software or the AC Link try and write the left channel into a full FIFO, the FIFO sets the overrun flag **and sets the left channel valid flag**. The FIFO now waits until the right channel is written (which will also set the overrun flag), before updating the FIFO write pointer - even if the FIFO is read before the right channel is written thereby opening up a location in the FIFO (not full). The same is true if the channel order received is reversed. As an example:

- AC Link writes both left and right channels into FIFO0
    - FIFO0 is full, **FCHS.FF0** set
- AC Link tries to write data into the right channel
    - FIFO0 sets right channel valid flag **FCHS.RCI0**
    - FIFO0 sets overrun flag **FSIC0.FOR**, plus internal overrun flag **FCHS.IOR0**, and discards data
    - FIFO0 write pointer not incremented (only one channel received)
- DMA engine reads a sample out of FIFO0 - now FIFO0 not full, **FCHS.FF0** clear
- AC Link tries to write data into the left channel
    - FIFO0 sets left channel valid flag **FCHS.LCI0**
    - Since **FCHS.IOR0** set, FIFO0 sets **FSIC0.FOR** overrun flag and discards data
    - FIFO0 has both **FCHS.LCI0** and **FCHS.RCI0** channel valid flags set
      - both channel valid flags are cleared, **FCHS.LCI0** = **FCHS.RCI0** = 0
      - Since **FCHS.IOR0** set, FIFO0 does not increment write pointer
      - FIFO0 clears **FCHS.IOR0**

If **FCRn.FEN** is clear, data written to a FIFO is discarded.

If **FCRn.FEN** is clear, the FIFO is off and zeros are always read. The FIFO read pointer usually points to the next location to be read. This allows for fast response time on PC bus accesses. When an empty FIFO is read, the data generally sent out is the previous data (based on **DACZ**). This requires the FIFO read pointer to NOT be incremented when the FIFO is empty. Then the read pointer is pointing to the last sample read. A flag is used to manage the read pointer when a FIFO is empty. The *Move Read Pointer* **FCHS.MRPn** flag, stores the fact that the read pointer is one location behind where is should be. When new data is placed in an empty FIFO, the write pointer is moved, and if **FCHS.MRPn** is set, the read pointer is also moved. If one side of an empty FIFO is read, the read pointer is not moved until the other side is read, thereby keeping sample synchronization throughout the system.

Host software can only access one channel of FIFO RAM at a time. The channel accessed by the host is read or written, no status is changed, and no pointers are moved. Since FIFO memory is double word aligned, the channel accessed is based on the address bit 2 as illustrated on the right side of the FIFO memory in Figure 39.

11.6.1 Using FM/Wavetable Digital Stream from Host

To use the Wavetable host feature in lieu of the FM digital mixing stream, one of the DMA/FIFO pairs above (not associated with the SRC's) must be allocated. Only four streams are allowed at once, although one stream can be Modem record, Modem playback, or host wavetable. The digital mixing stream is available when the FM synthesizer is powered down (**SSPM.FMEN** = 0). Then the stream can be directed to use the FM digital mixer by setting the FIFO-to-Slot mapping bits (**FCRn.LS[4:0]**/ **FCRn.RS[4:0]**) to slot IDs 29/30 and setting the direction for the associated DMA to host-to-FIFO, **DMRn.TR[1:0]** = 10. This host wavetable stream has an associated volume control in the FMLVC/FMRVC registers.

12. DIRECT PCI TRANSACTION GENERATION

The Sound Blaster hardware in the CS4281 needs to generate PCI I/O transactions to emulate the 8237 DMA controller registers. In the most common scenario, the hardware will use DMA engine 1 for Sound Blaster digital audio emulation. Host software will read the 8237 status register at I/O address 0x08 to check whether a DMA operation is still in progress or whether it has reached its terminal count. Unfortunately, the status register at 0x08 includes bits for DMA engines 0, 1, 2, and 3, while the Sound Blaster hardware is emulating only DMA engine 1; the status information from the real 8237 (for engines 0, 2, and 3) needs to be bitwise ORed with the DMA engine 1 status information and the result returned to host software. Therefore the Sound Blaster hardware not only has to trap an access to I/O address 0x08, but it must also generate a new I/O read transaction targeted to the real 8237 to retrieve its status information; the result of the second I/O read transaction is bitwise ORed with the status of DMA engine 1 and the result is used to complete the original I/O read transaction.

13. PCI I/O TRAPPING

The control logic behind the CS4281 legacy support is implemented in hardware. The hardware traps transactions that would otherwise be claimed by a south bridge and forwarded over an ISA bus to a legacy sound card. There are two kinds of trap hardware requirements - one that is configurable and one that is custom to support legacy DMA operation. There are three configurable I/O trapping registers to support Sound Blaster, FM, and the Gameport.

The custom I/O trapping is designed to support three different types of legacy DMA: DDMA, PC/PCI, and CrystalClear Legacy (CCLS). When configured for DDMA, the IOTDMA register sets the base address, DmaBase, and address DmaBase+00h to DmaBase+0Fh are trapped. The supported trapped registers are mapped into the CS4281 DMA engine that’s programmed to support legacy DMA.

For PC/PCI operation, the host DMA PC/PCI controller supports all the DMA I/O operations and the data is transferred via I/O reads and writes to location 00/04. On the CS4281, the legacy DMA trapping (IOTDMA) is turned off, and the PC/PCI I/O trapping register (IOTPCP) is configured to trap the PC/PCI I/O locations 00 and 04 (or 00 - 04).

CrystalClear Legacy (CCLS) is defined as CS4281 support for directly trapping the I/O accesses that were destined for the legacy 8237 controller and responding accordingly. Due to the custom nature of this type of DMA support, extra control bits (IOTCC) are provided to allow host software more flexibility in how CCLS responds to different legacy 8237 I/O accesses. When using CCLS, the DmaBase in IOTDMA should be set to 0, which is the legacy address for the 8237. Then IOTCC defines how and which addresses are trapped.

13.1 Typical I/O Trap Ranges

The software emulating legacy hardware will be interested in the following I/O port ranges:

- 0x200 - 0x207 Game port
- 0x220 - 0x22F Sound Blaster digital audio
- 0x388 - 0x38B Adlib FM
- 0x002 - 0x003 8237 DMA engine 1 base address, current address, base count, current count
- 0x008 - 0x00F 8237 shared registers
- 0x083 DMA engine 1 page register

DMA engine 1 is listed as an example — any one of the 8-bit DMA engines should be supported. The Sound Blaster addresses listed are the “standard” ones — if necessary to avoid conflict with another device, these can be adjusted. The CS4281 will support 3 relocatable I/O trap locations, and implement a general method of trapping only the necessary DMA I/O locations.

13.2 Trap Actions

The CS4281 handles trapped I/O write transactions by claiming the transaction, capturing the data written, completing the PCI transaction. The CS4281 handles I/O read transactions by returning data from the internal hardware that services that specific legacy function. The CS4281 handles snooped I/O transactions by not claiming the transaction, but capturing a copy of the data as the transaction occurs on the PCI bus. All legacy I/O transactions will complete without retries with the exception of DMA registers shared across all legacy DMA engines (such as the Status register). Since x86 processors only support 16 address bits of I/O space, the lower 16 bits of address are all the control needed. The upper 16-bits are ignored.

Emulating a 8237 DMA controller introduces additional trapping requirements. Three 8237 I/O locations are unique to each DMA engine, and can be handled using the standard method outlined in the previous paragraph. The shared 8237 registers are problematic. Most of these registers are write-only, and the CS4281 handles them by eavesdropping (snooping) on I/O write transactions and storing the captured data in a register, just as it does for a normal write trap (the CS4281 does not, however, claim the I/O write transaction). There are two shared 8237 registers (the status register at 0x08 and the mask register at 0x0F) that can be read. The DMA hardware will handle them by trapping the PCI read I/O transaction, generating an I/O read transaction targeted to the actual 8237, and combining the data read from the 8237 with its own status information before depositing the final result in the PCI interface (which completes the original I/O read transaction when it's retried).

13.3 I/O Trapping Registers

Each configurable I/O range to be trapped, has one 32-bit combined address and control register, and the CS4281 supports four such ranges which are locked to specific legacy functions. The trapped I/O address must begin on an even byte address if it is more than one byte long, a modulo two address if it is two bytes long, a modulo four address if it is four bytes long, a modulo eight address if it is eight bytes long, and a modulo 16 address if it is 16 bytes long. Each combined address/control register can describe an I/O port range of at most 16 contiguous addresses, longer ranges are not supported.

13.3.1 I/O Trap Game Port (IOTGP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RE	WE	res	res	res	IODC1	IODC0		DLY3	DLY2	DLY1	DLY0		MSK2	MSK1	MSK0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

Address: BA0: 500h, Read-Write

Default: 00000000h

Definition: Trapping I/O for the Game port. All trapped addresses are channeled to one internal register, JSPT (BA0:480h).

Bit Descriptions:

SA[15:0] GameBase Start Address: Starting PCI I/O space address of the trap range. Bits above the I/O Decode Control bits must be set to 0. For example, if **IODC[1:0]** = 10, **SA15-SA10** must be 0.

MSK[2:0] Address Mask: Logically, these bits are used to selectively mask off the lower four bits of address accompanying an I/O transaction. After the mask operation is performed, the resulting address is compared against SA[15:0]; if they are equal, the original I/O transaction lies within the trap range. The cycle is trapped and connected to register JSPT.

DLY[3:0] Delay PCI Cycle for this number of PCICLKs. TRDY# is delayed this number of clocks on both read and write transactions. The **DLY** number does not include the normal clocks required by the bus interface logic. For example, if the but interface logic normally takes 5 PCICLKs to complete the transaction and **DLY[3:0]** = 3, then the entire PCI cycle will take 8 cycles. Up to 15 clocks can be added.

IODC[1:0] I/O Decode Control: These two bits control how many I/O address bits are decoded for the trap range.
0 0 = Decode **SA15** down to **MSK** bits of I/O address (no aliasing, reset default)
0 1 = Decode **SA11** down to **MSK** bits of I/O address (**SA15-SA12** alias)
1 0 = Decode **SA9** down to **MSK** bits of I/O address (**SA15-SA10** alias)
1 1 = Reserved

WE Write Enable: This bit controls whether or not write trapping is enabled for the trap range.
0 = Write trapping disabled (reset default)
1 = Write trapping enabled

RE Read Enable: This bit controls whether or not read trapping is enabled for the trap range.
0 = Read trapping disabled (reset default)
1 = Read trapping enabled

13.3.2 I/O Trap - Sound Blaster (IOTSB)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RE	WE	res	res	res	IODC1	IODC0		DLY3	DLY2	DLY1	DLY0	1	1	1	1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	0	0	0	0

Address: BA0: 504h, Read-Write

Default: 000F0000h

Definition: This Sound Blaster I/O Trap is set for 16 I/O addresses to decode. The decoding to Sound Blaster registers is listed in Table 20 on page 153.

Bit Descriptions:

SA[15:4] SbBase Start Address: Starting PCI I/O space address of the trap range. Bits above the I/O Decode Control bits must be set to 0. For example, if **IODC[1:0]** = 10, **SA15-SA10** must be 0.

Bits 16 - 19 Address Mask: Logically, these bits are used to selectively mask off the lower four bits of address accompanying an I/O transaction. After the mask operation is performed, the resulting address is compared against **SA[15:4]**; if they are equal, the original I/O transaction lies within the trap range. Set to all ones for Sound Blaster.

DLY[3:0] Delay PCI Cycle for this number of PCICLKs. TRDY# is delayed this number of clocks on both read and write transactions. The **DLY** number does not include the normal clocks required by the bus interface logic. For example, if the but interface logic normally takes 5 PCICLKs to complete the transaction and **DLY[3:0]** = 3, then the entire PCI cycle will take 8 cycles. Up to 15 clocks can be added.

IODC[1:0] I/O Decode Control: These two bits control how many I/O address bits are decoded for the trap range.
0 0 = Decode **SA15** down to **SA4** bits of I/O address (no aliasing, reset default)
0 1 = Decode **SA11** down to **SA4** bits of I/O address (**SA15-SA12** alias)
1 0 = Decode **SA9** down to **SA4** bits of I/O address (**SA15-SA10** alias)
1 1 = Reserved

WE Write Enable: This bit controls whether or not write trapping is enabled for the trap range.
0 = Write trapping disabled (reset default)
1 = Write trapping enabled

RE Read Enable: This bit controls whether or not read trapping is enabled for the trap range.
0 = Read trapping disabled (reset default)
1 = Read trapping enabled

13.3.3 I/O Trap - FM Synthesis (IOTFM)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RE	WE	res	res	res	IODC1	IODC0		DLY3	DLY2	DLY1	DLY0	0	0	1	1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	0	0

Address: BA0: 508h, Read-Write

Default: 00030000h

Definition: The mapping of these registers to the FM synthesis block is listed in Table 32.

Bit Descriptions:

SA[15:2] FmBase Start Address: Starting PCI I/O space address of the trap range. Bits above the I/O Decode Control bits must be set to 0. For example, if **IODC[1:0]** = 10, **SA15-SA10** must be 0.

Bits 19-16 Address Mask: Logically, these bits are used to selectively mask off the lower two bits of address accompanying an I/O transaction. After the mask operation is performed, the resulting address is compared against **SA[15:2]**; if they are equal, the original I/O transaction lies within the trap range. Forced to 3 for FM.

DLY[3:0] Delay PCI Cycle for this number of PCICLKs. TRDY# is delayed this number of clocks on both read and write transactions. The **DLY** number does not include the normal clocks required by the bus interface logic. For example, if the but interface logic normally takes 5 PCICLKs to complete the transaction and **DLY[3:0]** = 3, then the entire PCI cycle will take 8 cycles. Up to 15 clocks can be added.

IODC[1:0] I/O Decode Control: These two bits control how many I/O address bits are decoded for the trap range.
0 0 = Decode **SA15** down to **SA2** bits of I/O address (no aliasing, reset default)
0 1 = Decode **SA11** down to **SA2** bits of I/O address (**SA15-SA12** alias)
1 0 = Decode **SA9** down to **SA2** bits of I/O address (**SA15-SA10** alias)
1 1 = Reserved

WE Write Enable: This bit controls whether or not write trapping is enabled for the trap range.
0 = Write trapping disabled (reset default)
1 = Write trapping enabled

RE Read Enable: This bit controls whether or not read trapping is enabled for the trap range.
0 = Read trapping disabled (reset default)
1 = Read trapping enabled

As an example, the legacy emulation software may specifying the following I/O trap ranges using these registers:

- Game port (0x200 - 0x207) using IOTGP:
  Start address = 0x200, mask = 0x07, I/O decode control = 0x2, RE = WE = 1
- Sound Blaster digital audio (0x220 - 0x22F) using IOTSB:
  Start address = 0x220, I/O decode control = 0x2, RE = WE = 1
- Adlib FM (0x388 - 0x38B) using IOTFM:
  Start address = 0x388, I/O decode control = 0x2, RE = WE = 1

13.3.4 I/O Trap - DMA (IOTDMA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	IODC1	IODC0									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	res	res	res	res

Address: BA0: 50Ch, Read-Write

Default: 00000000h

Definition: Legacy DMA trapping register that contains the starting and ending addresses constituting an I/O trap range. This register is used when CCLS or DDMA is enabled. For DDMA operation, addresses DmaBase+00h to DmaBase+0Fh are trapped on reads and writes. For CCLS operation, DmaBase should be set to 0, and trapping is controlled through the IOTCC register. In CCLS operation, Start Addresses should never use SA7 or below (always set to 0) as the legacy Page registers use these address bits.

Bit Descriptions:

SA[15:4] DmaBase Start Address: Starting PCI I/O space address of the trap range. For Crystal DMA, this address should be set to 0. For DDMA, this address should be set similar to the motherboard chipset for the trapped DMA engine. Bits above the I/O Decode Control bits must be set to 0. For example, if **IODC[1:0]** = 10, **SA15-SA10** must be 0.

IODC[1:0] I/O Decode Control: These two bits control how many I/O address bits are decoded for the trap range.
0 0 = Decode **SA15** down to **SA4** bits of I/O address (no aliasing, reset default)
0 1 = Decode **SA11** down to **SA4** bits of I/O address (**SA15-SA12** alias)
1 0 = Decode **SA9** down to **SA4** bits of I/O address (**SA15-SA10** alias)
1 1 = Reserved

13.3.5 I/O Trap - PCPCI (IOTPCP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RE	WE	res	res	res	IODC1	IODC0						res	MSK2	MSK1	MSK0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

Address: BA0: 52Ch, Read-Write

Default: 00000000h

Definition: I/O Trapping register for the PC/PCI mechanism. The addresses needed are 0 and 4, where 0 transfers data in and out, and 4 also transfers data - but also indicating terminal count. Although double words are supported, this interface only uses 0/4 since Sound Blaster only supports byte samples. **The I/O access is only trapped if PCGNT# is low.** For capture, the I/O Trap logic redirects host PC/PCI agent reads (from 0/4) to the Legacy DMA controller. For playback, the I/O Trap logic redirects host PC/PCI agent writes (to 0/4) to the Legacy DMA controller.

Bit Descriptions:

SA[15:0] PC/PCI Start Address: Starting PCI I/O space address of the trap range. Set to 0. Bits above the I/O Decode Control bits must be set to 0. For example, if **IODC[1:0]** = 10, **SA15-SA10** must be 0.

MSK[2:0] Address Mask: Logically, these bits are used to selectively mask off the lower four bits of address accompanying an I/O transaction. After the mask operation is performed, the resulting address is compared against **SA[15:0]**; if they are equal, the original I/O transaction lies within the trap range. Set to 4 for PC/PCI.

IODC[1:0] I/O Decode Control: These two bits control how many I/O address bits are decoded for the trap range.
0 0 = Decode **SA15** down to **MSK** bits of I/O address (no aliasing, reset default)
0 1 = Decode **SA11** down to **MSK** bits of I/O address (**SA15-SA12** alias)
1 0 = Decode **SA9** down to **MSK** bits of I/O address (**SA15-SA10** alias)
1 1 = Reserved

WE Write Enable: This bit controls whether or not write trapping is enabled for the trap range.
0 = Write trapping disabled (reset default)
1 = Write trapping enabled

RE Read Enable: This bit controls whether or not read trapping is enabled for the trap range.
0 = Read trapping disabled (reset default)
1 = Read trapping enabled

13.3.6 I/O Trap CCLS Control Register (IOTCC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C1RE	C1WE	C1WSE								C2WSE	C2RSE				BPFF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		C3WSE						C4RE		C4WSE					

Address: BA0: 530h, Read-Write

Default: 00000000h

Definition: Provides I/O Trapping control for CCLS legacy 8237 trapping when enabled (**IOTCR.EDMA[1:0]** = 01). When any **C1xxx or C2xxx** bit is set, write snooping to DmaBase+0Ch is enabled. Once enabled, any write to DmaBase+0Ch clears **BPFF**. There is no initialization associated with setting bits in this register; therefore, this register can be dynamically managed by software wanting access to the real legacy DMA controller.

Bit Descriptions:

C1RE CCLS 1 Read Enable: When set, read trapping is enabled for the CCLS assigned legacy DMAn controller (**IOTCR.DMA**) Current Address, Current Count, and Address Page registers. When **C1RE** is set, **BPFF** is toggled on every read and write access to the Base/Current Count and Base/Current Address DMA registers (not the Address Page register).
0 = Read trapping disabled (reset default)
1 = Read trapping enabled

C1WE CCLS 1 Write Enable: When set, write trapping is enabled for the CCLS assigned legacy DMAn controller (**IOTCR.DMA**) Base Address, Base Count, and Address Page registers. This bit is mutually exclusive with **C1WSE**. When **C1WE** is set, **BPFF** is toggled on every read and write access to the Base/Current Count and Base/Current Address DMA registers (not the Address Page register).
0 = Write trapping disabled (reset default)
1 = Write trapping enabled

C1WSE CCLS 1 Write Snoop Enable: When set, the CCLS I/O Trap logic snoops the writes to the CCLS-assigned legacy DMAn controller (**IOTCR.DMA**) Base Address, Base Count, and Address Page registers. Instead of claiming the I/O write transaction, the CS4281 captures data written to the snooped I/O range and stores it in the appropriate register, just as if the I/O location was actually trapped. The actual legacy 8237 will claim these transactions. This bit is mutually exclusive with **C1WE**. When **C1WE** is set, **BPFF** is toggled on every read and write access to the Base/Current Count and Base/Current Address DMA registers (not the Address Page register).
0 = Write snooping disabled (reset default)
1 = Write snooping enabled

C2WSE CCLS 2 Write Snoop Enable: When set, the CCLS I/O Trap logic snoops writes to the addresses between DmaBase+0 through DmaBase+7 not claimed or snooped by **C1xxx** bits. When a write occurs to these registers, **BPFF** is XOR'd.

C2RSE CCLS 2 Read Snoop Enable: When set, the CCLS I/O Trap logic snoops reads to the addresses between DmaBase+0 through DmaBase+7 not claimed by **C1RE**. When a read occurs to these registers, **BPFF** is XOR'd.

BPFF Byte Pointer flip-flop. This bit keeps track of upper vs. lower byte in DMA CCLS address and count values. This bit is reset by write snoops to DmaBase+0Ch when any **C1xxx** or **C2xxx** bit is set. **BPFF** is XORed (flipped) by any access claimed or snooped by **C1xxx** bits (except the Address Page location), and the other registers in the DmaBase+0 to DmaBase+7 range snooped by **C2WSE** or **C2RSE**.

C3WSE CCLS 3 Write Snoop Enable: When set, the CCLS I/O Trap logic snoops the writes to the Single Mask register (DmaBase+0Ah) and the Mode register (DmaBase+0Bh). The lower two bits of these snoops indicate which DMA the data is for. If the lower two bits are set to the legacy DMAn controller (**IOTCR.DMA** bits), then the data is sent to the appropriate DMAn register. Instead of claiming the I/O write transaction, the CS4281 captures data written to the snooped I/O range and stores it in the appropriate register, just as if the I/O location was actually trapped. The actual legacy 8237 will claim these transactions.
0 = Write snooping disabled (reset default)
1 = Write snooping enabled

C4RE CCLS 4 Read Enable: When set, read trapping is enabled for the CCLS assigned legacy DMAn controller (**IOTCR.DMA**) Status register at DmaBase+08h and Multi-engine mask register at DmaBase+0Fh. Setting this bit causes a complicated series of steps which starts with telling the host to retry the cycle. Then, for the Status register, the CS4281 reads the real 8237 location 08h, combines the other engine status with the proper status bits for the legacy DMAn engine located in HDSRn, and stores the data in DLSR. When the host retries the read to 08h, the DLSR register is read. Note that this host-retry mechanism cannot interfere with any DMA cycles that may occur during this procedure. The Multi-Engine mask register read responds similarly except that the host address read is 0Fh. The current legacy DMAn mask register is DCRn, and the combined legacy mask register that is finally read by the host when retried is DLMR. **Note that any DMA activity has to be supported while the re-try is in progress.**
0 = Read trapping disabled (reset default)
1 = Read trapping enabled

C4WSE CCLS 4 Write Snoop Enable: When set, the CCLS I/O Trap logic snoops the writes to the Multi-Engine Mask register (DmaBase+0Fh) and the Clear-All Masks register (DmaBase+0Eh). For the Multi-Engine Mask, the legacy DMAn controllers bit is redirected to bit 0 of DMRn. Instead of claiming the I/O write transaction, the CS4281 captures data written to the snooped I/O range and stores it in the appropriate register, just as if the I/O location was actually trapped. The actual legacy 8237 will claim these transactions. For the Clear-All Mask register, any write to this address clears **DMRn.MSK**. The actual data for DmaBase+0Eh write snoops is irrelevant.
0 = Write snooping disabled (reset default)
1 = Write snooping enabled

13.3.7 I/O Trap Control Register (IOTCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DMA1	DMA0	EDMA1	EDMA0		res	res	res	res	res	res	ITE

Address: BA0: 58Ch, Read-Write

Default: 00000000h

Definition: Global trapping control, which DMA controller is attached to the legacy trapping, and the type of legacy support enabled. Global trapping control can also be disabled through the **IISR.GTD** bit which overrides the **ITD** bit.

Bit Descriptions:

ITE I/O Trap Enable: Setting this bit enables I/O trapping globally (this avoids having to write each IOTxxx register to disable each trap range). For trapping to be enabled **ITE** must be set and **IISR.GTD** must be clear.
0 = Global I/O trapping disabled (reset default)
1 = Global I/O trapping enabled - if **IISR.GTD** is clear

EDMA[1:0]Enable DMA Trapping: Defines the DMA trapping logic behavior.
00 = No DMA trapping, use new DMA support (reset default)
01 = CCLS - Legacy DMA - proprietary 8237 trapping enabled via IOTCC.
10 = PC/PCI DMA operation (SB engine supports directly - DMA controller does nothing). Trapping enabled via IOTPCP.
11 = DDMA Operation and trapping of DmaBase+00h through DmaBase+0Fh enabled.

DMA[1:0] DMA Engine Select: binary value of these two bits selects the DMA engine assigned to the legacy support. Since the DMAn/FIFOm link is inseparable, the **DMA** bits are active in Polled FIFO and DMA mode if **SSCR.SB** is true.
00 = DMA Engine 0 (reset default)
01 = DMA Engine 1
10 = DMA Engine 2
11 = DMA Engine 3

13.4 Interface to CCLS

The CrystalClear Legacy Support, CCLS, interface is Cirrus-custom in that legacy trapping of 8237 registers is controlled through a separate IOTCC register by functional grouping. When CCLS is enabled (**IOTCR.EDMA[1:0]** = 01), the I/O Trap mechanism and DMA controller assigned to legacy support (**IOTCR.DMA[1:0]**) respond as if they were one DMA engine of the real legacy 8237.

There are four sets of legacy registers that have unique responses:

- R/W Address, Count, and Page registers that have unique addresses based on legacy DMA engine.
- Write location 0Ch - which clears **IOTCC.BPFF** and is always snooped when CCLS is enabled
- Write locations 0Ah/0Bh - Single Mask & Mode - which are shared among all legacy DMA engines. The lower two bits determine the mapping to the appropriate CS4281 MDAn registers
- Read locations 08h/0Fh - Status & All Masks - which are also shared among all legacy DMA engines. These require retrying the host, getting data from real 8327. combining, and host retry getting data.

The first set, listed in Table 8, consists of the Base/Current Address registers and the Page register.

The second set of legacy DMA registers is only one register at address 0C - Byte Pointer Flip-Flop clear. When anything is written to DmaBase+0Ch (write snooped by the CS4281), **IOTCC.BPFF** is cleared. All accesses to DmaBase+0 to

I/O Loc.	Access	Name	Host Reg.	Comment
00h	write	Ch. 0 Base (+Current) Address	[3]DBA0	16-bit, 1 byte at a time, IOTCC.BPFF determines
00h	read	Ch. 0 Current Address	[3]DCA0	16-bit, 1 byte at a time, IOTCC.BPFF determines
01h	write	Ch. 0 Base (+Current) Count	[3]DBC0	16-bit, 1 byte at a time, IOTCC.BPFF determines
01h	read	Ch. 0 Current Count	[3]DCC0	16-bit, 1 byte at a time, IOTCC.BPFF determines
87h	r/w	Ch. 0 Page Register	[3]DBA0	Upper word of Base (+ Current) Address
02h	write	Ch. 1 Base (+Current) Address	[3]DBA1	16-bit, 1 byte at a time, IOTCC.BPFF determines
02h	read	Ch. 1 Current Address	[3]DCA1	16-bit, 1 byte at a time, IOTCC.BPFF determines
03h	write	Ch. 1 Base (+Current) Count	[3]DBC1	16-bit, 1 byte at a time, IOTCC.BPFF determines
03h	read	Ch. 1 Current Count	[3]DCC1	16-bit, 1 byte at a time, IOTCC.BPFF determines
83h	r/w	Ch. 1 Page Register	[3]DBA1	Upper word of Base (+ Current) Address
04h	write	Ch. 2 Base (+Current) Address	[3]DBA2	16-bit, 1 byte at a time, IOTCC.BPFF determines
04h	read	Ch. 2 Current Address	[3]DCA2	16-bit, 1 byte at a time, IOTCC.BPFF determines
05h	write	Ch. 2 Base (+Current) Count	[3]DBC2	16-bit, 1 byte at a time, IOTCC.BPFF determines
05h	read	Ch. 2 Current Count	[3]DCC2	16-bit, 1 byte at a time, IOTCC.BPFF determines
81h	r/w	Ch. 2 Page Register	[3]DBA2	Upper word of Base (+ Current) Address
06h	write	Ch. 3 Base (+Current) Address	[3]DBA3	16-bit, 1 byte at a time, IOTCC.BPFF determines
06h	read	Ch. 3 Current Address	[3]DCA3	16-bit, 1 byte at a time, IOTCC.BPFF determines
07h	write	Ch. 3 Base (+Current) Count	[3]DBC3	16-bit, 1 byte at a time, IOTCC.BPFF determines
07h	read	Ch. 3 Current Count	[3]DCC3	16-bit, 1 byte at a time, IOTCC.BPFF determines
82h	r/w	Ch. 3 Page Register	[3]DBA3	Upper word of Base (+ Current) Address

[3] Only the DMA engine set up for legacy responds to these registers (IOTCR.DMA[1:0]).

Table 8. CCLS DMA Engine Specific I/O Map

DmaBase+7 are snooped and, when read or written, **IOTCC.BPFF** is toggled. In addition, one set of legacy DMA registers in the same address range can be trapped or snooped based on the IOTCC bits **C1RE**, **C1WE**, and **C1WSE**. Only one of the four sets of registers responds based on the **IOTCR.DMA[1:0]** bits. Using legacy DMA3 for example, a typical setup scenario would be:

- Game software writes xxh to address 0Ch
  - DmaBase+0Ch snooped and **IOTCC.BPFF** cleared.
- Game software then writes lower address byte to Base address register - 06h
  - DmaBase+06h trapped/snooped and (since **BPFF** = 0) data written to **DBA3.BAL[7:0]** and **DCA3.CAL[7:0].**
  - **BPFF** toggled (XOR'd) - now **BPFF** = 1
- Game software then writes upper address byte to Base address register - 06h
  - DmaBase+06h trapped/snooped and (since **BPFF** = 1) data written to **DBA3.BAH[7:0]** and **DCA3.CAH[7:0].**
  - **BPFF** toggled (XOR'd) - now **BPFF** = 0
- Game software then writes lower count byte to Base count register - 07h
  - DmaBase+07h trapped/snooped and (since **BPFF** = 0) data written to **DBC3.BCL[7:0]** and **DCC3.CCL[7:0].**
  - **BPFF** toggled (XOR'd) - now **BPFF** = 1
- Game software then writes upper count byte to Base count register - 07h
  - DmaBase+07h trapped/snooped and (since **BPFF** = 1) data written to **DBC3.BCH[7:0]** and **DCC3.CCH[7:0].**
  - **BPFF** toggled (XOR'd) - now **BPFF** = 0
- Game software then writes DMA3 page register - 82h
  DmaBase+82h trapped/snooped and data written to **DBA3.BAU[15:0]** and **DBA3.CAU[15:0].** Note that the upper byte is forced to 0 and the data written goes into the lower byte.
- Since this example uses DMA3, addresses between 0 and 5 are snooped. Any reads or writes toggle **BPFF.**

The third set of legacy DMA registers consist of two write-only registers, *Single Mask* and *Mode*; where the lower two bits determine the mapping to CS4281 DMAn registers DCRn and DMRn, respectively. Trapping for these registers is controlled through the **IOTCC.C3WSE** bit, which only allows write snooping. Write trapping is not supported since these registers are shared by all legacy DMA controllers. Read trapping is not supported since these registers are write-only. Figure 40 illustrates the logic decoding for the *Mode* register. The lower two data bits are compared to the **IOTCR.DMA** bits. If they match (and CCLS enabled), the data is written into the lower byte of the DMRn register assigned to legacy support.

Figure 41 illustrates the conceptual logic for the *Single Mask* register. The lower two data bits are compared to the **IOTCR.DMA** bits. If they match (and CCLS enabled), then *Single Mask* bit 2 is redirected to **DCRn.MSK** in the bit 0 position. **DCRn.MSK** is in the 0th bit position to support DDMA operation.

The fourth set of legacy DMA registers consist of one read-only register, one read-write register, and one write-only register, *Status*, *All Mask*, and *Clear-All Mask*; where reading two of these registers requires the CS4281 to force a host retry. Then the CS4281 must go set status from the real 8237 and combine that with the CS4281's legacy engine. Then when the host retries, the CS4281 returns the combined status. Trapping for these registers is controlled through the **IOTCC.C4xxx** bits, which support read trapping and write snooping. Write trapping is not supported since these registers are shared by all legacy DMA controllers and snooping can accomplish the same task. Read trapping (for Status and All Mask registers) requires host retries and assumes the only other device interested in legacy DMA is a subtractive bridge. Note that this method of trapping will not work with positive decode bridges (such as the

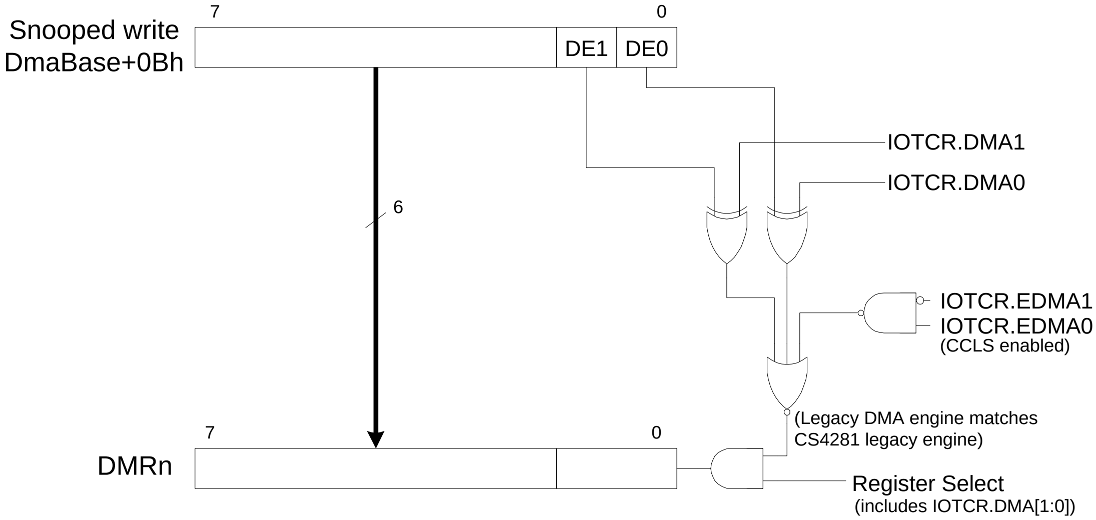


Figure 40. CCLS Mode Register Conceptual Logic

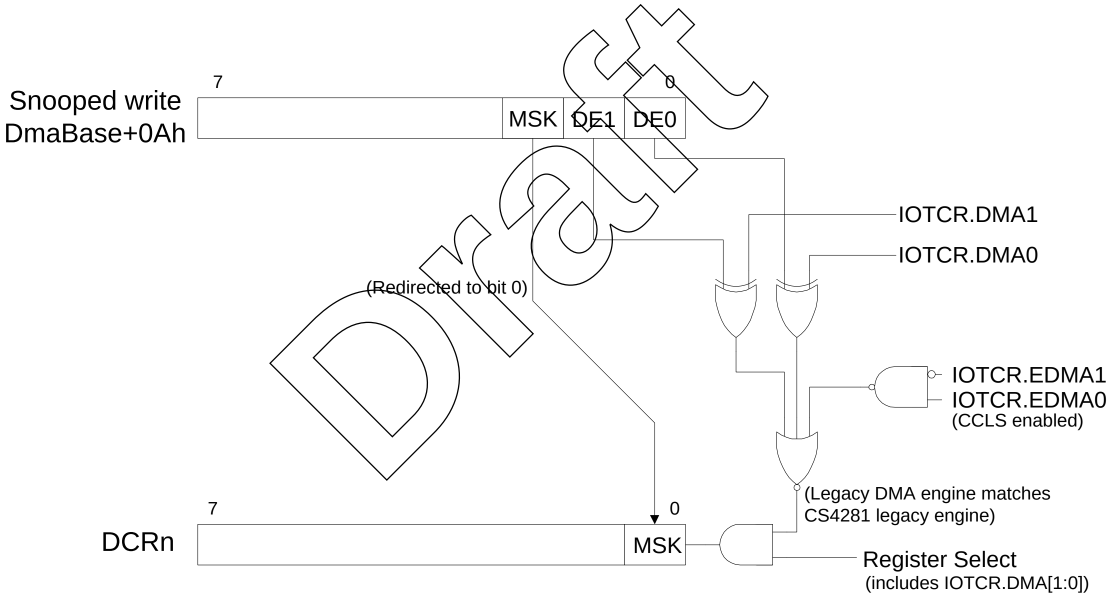


Figure 41. CCLS Single-Mask Register Conceptual Logic

PIX 4 chipset) wherein only snooping is available. The Clear-All Mask register is only write-snooped when **IOTCC.C4WSE** is set. **IOTCC.RE** has no effect on the Clear-All Mask register since it is write-only.

The *Status* register is read-only; therefore, read-trapping for location DmaBase+08h is only enabled when **IOTCC.C4RE** is set. When host software tries to read 08h, the following sequence occurs:

- Host tries to read 08h - trapped by the CS4281
- CS4281 forces host to retry read cycle
- CS4281 acquires bus and reads legacy/real 8237 location 08h
- CS4281 stores legacy 08h data in DLSR register
  - Legacy bits in DLSR are read-only; therefore, writing this register combines host status with CS4821 DMA status.
- Host retries the read of 08h and the CS4281 redirect to a read of DLSR register

  - Host reading DLSR clears all TC bits in register.

The *All Mask* register is read-write, similar to the 82357 DMA controller. In the original 82C37A, the *All Mask* register is write-only. The CS4281 does not support write-trapping this register since the register is shared by all legacy DMA controllers. If **IOTCC.C4WSE** is set, the CS4281 snoops writes to DmaBase+0Fh and multiplexes the data into the CS4281's **DCRn.MSK** bit (bit 0), as illustrated in Figure 42.

If **IOTCC.C4RE** is set, the CS4281 traps reads from DmaBase+0Fh and causes the following sequence.

- Host tries to read 0Fh - trapped by the CS4281
- CS4281 forces host to retry read cycle
- CS4281 acquires bus and reads legacy/real 8237 location 0Fh
- CS4281 stores legacy 0Fh data in DLMR register
  - Legacy bit in DLMR is read-only; therefore, writing this register combines host status with CS4821 DMA status.
- Host retries the read of 0Fh and the CS4281 redirect to a read of DLMR register

The Table 9 lists the legacy upper register set and the affect of traps to I/O legacy addresses.

13.4.1 CCLS Unsupported Legacy DMA Registers

The *Command* register at location 08h is not supported. This write-only register enables/disables the **entire** DMA chip, sets priority, and DRQ/DACK polarity. No game should ever hit this register since it affects the operation of all the DMA engines. *Command* is setup by BIOS at boot time and not used thereafter.

The *Request* register at location 09h is not supported. This write-only register forces the request for a particular engine active. This register uses the lower two data bits to determine the DMA engine - similar to the *Mode* register. To our knowledge, games don't use this bit since the Sound Blaster engine is synchronized with the legacy DMA controller using DRQ/DACK signals. If a game would use this register, the counts in Sound Blaster verses the DMA counts would have to be tweaked since *Request* causes the legacy DMA to do a transfer that the SB hardware didn't request.

The *Master Clear* register at location 0Dh is not supported. This write-only register forces all four legacy DMA engines to respond as if the bus reset (power up) pin was active. No game should ever hit this register since it affects the operation of all the engines.

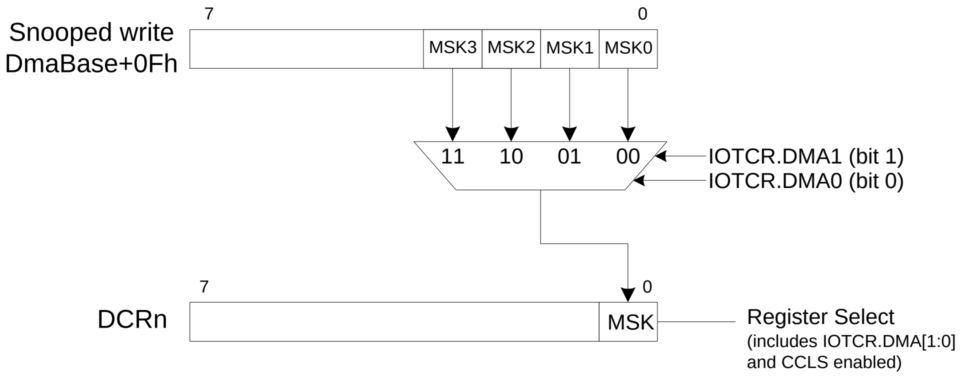


Figure 42. CCLS All-Mask Register Conceptual Logic

The *Temporary* register at location 0Dh is not supported. This read-only register contains data used in memory-to-memory transfers. The CS4281 and SB games don’t support memory-to-memory transfers.

13.5 Interface to PC/PCI

Configurations using Intel’s PC/PCI mechanism for handling Sound Blaster DMA do not configure one of the DMA engines to emulate the 8237. The Legacy DMAn controller, when configured for PC/PCI, will redirect DMA REQ# and GNT# to the PC/PCI engine. When the PCI bus agent implementing PC/PCI functionality (typically the south bridge) needs to access the CS4281 for the purpose of writing data previously read from system memory or for the purpose of reading data that will be written to system memory, it does so by generating I/O read and write transactions targeted to addresses 0x00 and 0x04. Address 0x00 is used for “normal” I/O accesses, while address 0x04 is used when the agent implementing PC/PCI generates the last I/O read or write transaction associated with a DMA transfer (use of address 0x04 is the “terminal count” signal for the overall DMA transfer). The I/O Trap logic redirects IOTPCP read accesses to the Legacy DMAn controller and write accesses to the Legacy DMAn controller. Data will only be transferred one byte (one channel) at a time.

See the *PC/PCI* section for functional details.

Dma Base+	Access	Name	[1]Host Reg.	Comment
08h	(write)	Command	no trap	CS4281 currently doesn’t support
08h		[3]Status	DLSR	Request Pending and TC status
09h	(write)	Request	no trap	CS4281 currently doesn’t support
09h	(read)		no trap	not used
0Ah	write snoop	[2]Single Mask	complicated DCRn.MSK	If PCI.data bits [1:0] = n, PCI.bit[2] redirected to DCRn.bit[0] which is the MSK bit.
0Ah	(read)		no trap	not used
0Bh	write snoop	[2]Mode	DMRn.bits[7:0]	If PCI.data bits [1:0] = n, PCI.bits[7:0] are sent to DMRn.bits[7:0]
0Bh	(read)		no trap	not used
0Ch	write snoop	Clear byte pointer f/f	IOTCC.BPFF = 0	Byte pointer used on accesses to 00h-07h
0Ch	(read)		no trap	not used
0Dh	(write)	Master Clear	no trap	CS4281 currently doesn’t support
0Dh	(read)	Temporary Register	no trap	CS4281 currently doesn’t support
0Eh	write snoop	Clear all Masks	DCRn.MSK = 0	
0Eh	(read)		no trap	not used
0Fh	write snoop	Multi-Engine Mask Register	complicated DCRn.MSK	The legacy DMAn engine MSK bit is mapped to DCRn.MSK (bit 0)
0Fh	read	[3]Multi-Engine Mask Register	DLMR	Reads DCRn.MSK bit in D0

1. In Host Reg., “n” is selected by **IOTCR.DMA[1:0]**
2. Lower two data bits determine DMA engine. If they match ‘n’, store a copy of the data and execute
3. These reads are for all 4 legacy engines. Requires trapping cycle, doing retry, mastering to real Legacy DMA controller, getting data, combining data with internal register data, and sending back to host to complete re-tried cycle. Their is a bit to disable this feature.

Table 9. CCLS Shared Register I/O Map

13.6 Interface to DDMA

Assuming I/O Trapping is enabled (**IOTCR.ITD**=0), and the CS4281 is setup for DDMA operation (**IOTCR.EDMA[1:0]** = 11), the IOTDMA traps all accesses to DmaBase+00 through DmaBase+0Fh for read and write operations. The CS4281 DMA engine registers are setup to operate like DDMA; therefore, the I/O Trap mechanism only redirects the I/O based on the DMA engine selected for DDMA (**IOTCR.DMA[1:0]**) and supports the unused DDMA registers properly (reads return zeros and writes go to bit bucket). The response for each register is listed in Table 10 in the *DDMA* section.

14. PC/PCI

The CS4281 supports Intel's PC/PCI mechanism for emulating ISA DMA on the PCI bus via **PCREQ#** and **PCGNT#** pins multiplexed with the EEPROM interface pins. (Note that these multiplexed pins take on PC/PCI request and grant functionality only when the EEPROM is not used. This chapter only discusses the PC/PCI request/grant signalling interface; a complete PC/PCI implementation also requires handling I/O read and write accesses targeted to addresses 0x00 and 0x04 — this aspect of PC/PCI is covered in the *PCI I/O Trapping* section using the IOTPCP register and setting the legacy DMAn engine to PC/PCI operation, **IOTCR.EDMA[1:0]** = 10.

The PC/PCI specification from Intel, *Mobile PC/PCI DMA Arbitration and Protocols* - Revision 2.2, contains several complicated provisions for situations where a device wants to request access to multiple DMA engines simultaneously. The PC/PCI implementation in the CS4281 only supports requesting one DMA engine at any one time.

PC/PCI uses a host PC/PCI-capable DMA controller that is programmed by the game software. The CS4281 DMAn engine programmed for PC/PCI legacy operation doesn't work like traditional DMA (doesn't inc/dec address pointers and doesn't dec. a counter). The Legacy DMAn engine configured for PC/PCI acts as if the stream was in polled I/O mode in that no address or current counters are changed and data is just transferred from the host to and from the FIFOs using PCI I/O writes and reads. All standard DMA functionality is handled by the host PC/PCI agent. The BIU bus mastering engine is also idled while the PC/PCI engine is enabled. The following bits are related to PC/PCI operation:

- BIOS should set **CFLR.CB1[1:0]** bits to indicate that PC/PCI is used and which channel
- IOTPCP register - setup to read and write trap addresses 00 and 04
- **SPMC.EESPD** set - to disable the EEPROM port
- **IOTCR.DMA[1:0]** = 'n' - selects the Legacy DMAn engine and **CHx** bit for **PCREQ#**
- **IOTCR.EDMA[1:0]** = 10 - Legacy DMAn engine configured for PC/PCI operation
- **DMRn.TBC** set - forces transfers by channel/byte. PC/PCI engine only supports one byte at a time.
- **PCPCR.PCIEN** set, to enable PC/PCI engine (and disable BIU Bus Master for Legacy DMAn controller)

When the FIFO makes a request, the request is sent to the PC/PCI engine, which serializes the request, as shown in Figure 43, and sends the serial data over the **PCREQ#** line. The PC/PCI engine sets the **PCREQ#** engine number based on the legacy DMA engine, **IOTCR.DMA[1:0]**. The DMA engines supported by Gershwin are 0 through 3. For example, if **IOTCR.DMA[1:0]** = 01, then the CH1 bit is set high by the PC/PCI engine indicating the DMA1 engine. In response to the **PCREQ#** signal, the host PC/PCI DMA controller sends a grant on **PCGNT#** with the same DMA engine, sent during the request, encoded on the bits. Since the CS4281 doesn't support DMA engines 4-7, bit2 is always 0. The encoding is:

- bit[1:0] = 00 is DMA0
- bit[1:0] = 01 is DMA1
- bit[1:0] = 10 is DMA2
- bit[1:0] = 11 is DMA3

The four CS4281 DMA engine combinations are illustrated in Figure 44. The PC-PCI engine sets the appropriate bit on the **PCGNT#** line, and host PC/PCI DMA engine responds with the appropriate **PCGNT#** signal. The DMA controller must be set in DMA (vs. Polled) mode by the Sound Blaster engine, before the PC/PCI engine will send out a request.

Since the CS4281 only supports one legacy engine, the bits returned on the **PCGNT#** line are of little use. After the PC/PCI engine receives the grant (after bit2), the **PCREQ#** signal is taken away for a minimum of 2 PCICLKs . The Host PC/PCI DMA controller transfers data to/from address 00h/04h which the CS4281 traps (only when **PCGNT#** is low) through the IOTPCP register. When the host PC/PCI agent reads 00/04, the Legacy DMAn engine transfers data from FIFOn, through the formatter, and out the BIU. When the PC/PCI agent writes 00/04, the BIU sends the data to the Legacy DMAn engine which transfers the data through the formatter to FIFOn. The Legacy DMAn controller does NOT update addresses or counters while in PC/PCI mode (similar to Polled FIFO mode).

The PC/PCI engine attached to the Legacy DMAn engine, works in a round-robin fashion with the other three DMAn controllers, which could still be operating in normal mode. If no other DMAn engines need access to the PCI bus when **PCGNT#** is de-asserted, the Legacy DMAn engine reasserts the **PCREQ#** with the encoded data, and waits for another grant on **PCGNT#**. The PC/PCI engine clocks the **PCGNT#** signal using PCICLK. The **PCGNT#** high-to-low detection (low sample is the start bit) indicates that the CS4281 is receiving a grant. Once the start bit is sampled, the transfer must be completed. If **PCREQ#** is sent out but no **PCGNT#** has been received, and the Sound Blaster engine gets a Halt or Reset command, the **PCREQ#** is de-asserted and the transfer cancelled.

The BIU only traps IOTPCP data when the **PCGNT#** signal is low, indicating that the CS4281 is being accessed for reading or writing. The

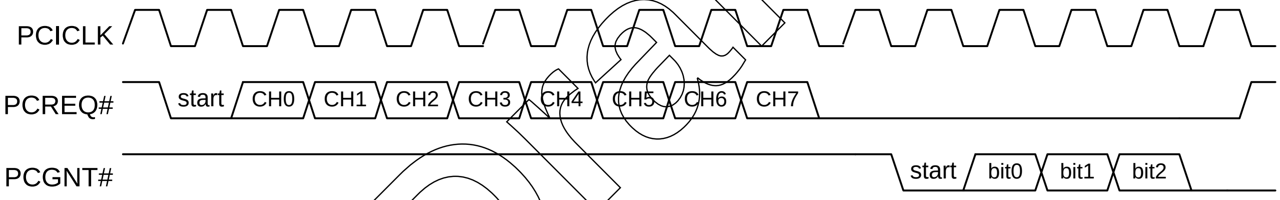


Figure 43. PC-PCI Serial Engine-Passing Protocol

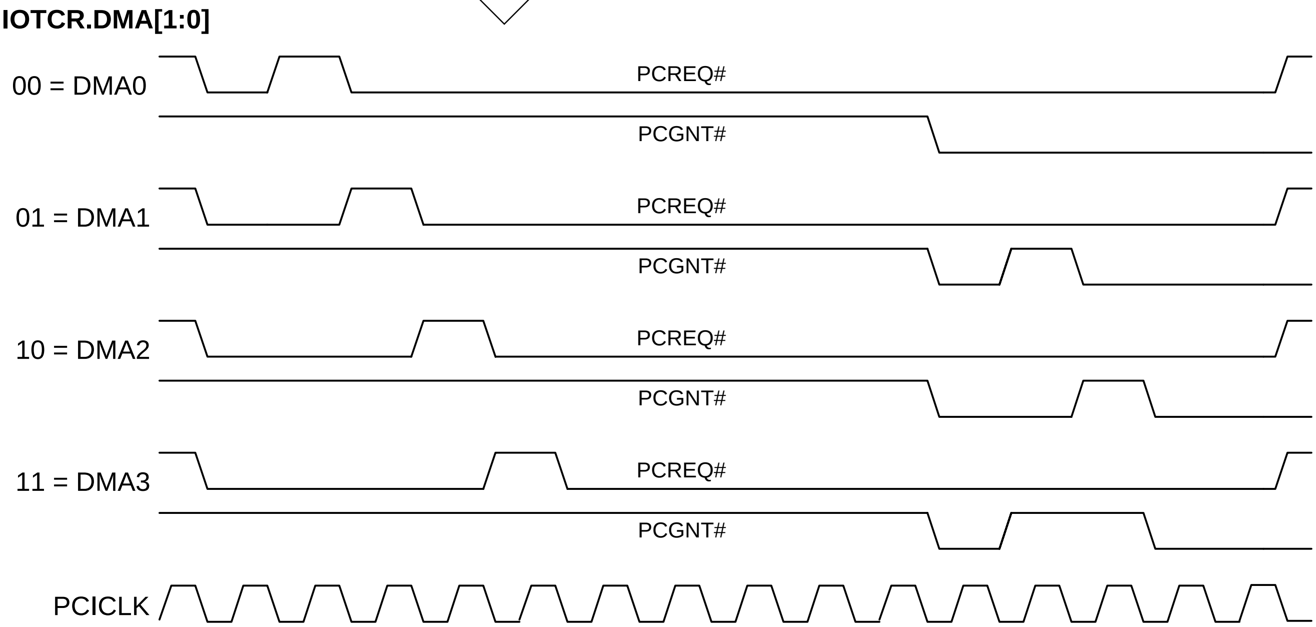


Figure 44. PC-PCI CS4281-Supported Serial Engines

DMAn Legacy controller makes no distinction between address 00 and 04. Address 04 is used as a terminal count indication when supporting demand mode (which the CS4281 doesn't support). PC/PCI always transfers data dword aligned. Sound Blaster always transfers bytes; hence PC/PCI transfers will be bytes to/from 00 or 04.

A capture sequence is illustrated in Figure 45, in which the CS4281 is requesting service for Legacy DMA1 on **PCREQ#**. The host PC/PCI DMA engine responds with **PCGNT#** having DMA1 encoded in the serial bit stream. The host DMA engine then reads a byte from the CS4281 at I/O location 00/04. The host, at some time later, writes the byte to memory. The two transactions may not be back to back. Note that the **PCGNT#** line is removed before the memory write is completed. If the CS4281 doesn't have any other DMA engines that need service, the PC/PCI engine can reassert **PCREQ#** (with serial data) after **PCGNT#** goes high. The memory write cycle length can vary. The I/O read cycle length is based on the CS4281.

A playback sequence is illustrate in Figure 46, in which the CS4281 is requesting service for Legacy DMA3 on **PCREQ#**. The host PC/PCI DMA engine responds with **PCGNT#** having DMA3 encoded in the serial bit stream. The host DMA engine then reads a byte from memory. The host, at some time later, writes the byte to the CS4281 I/O Trap location 00/04. The two transactions may not be back to back. If the CS4281 doesn't have any other DMA engines that need service, the PC/PCI engine can reassert **PCREQ#** (with serial data) after **PCGNT#** goes high. The memory read cycle length can vary. The I/O write cycle length is based on the CS4281. The capture and playback sequences are depicted with different Legacy DMA engines for example purposes only. In real operation, Sound Blaster capture and playback will use the same Legacy DMAn engine.

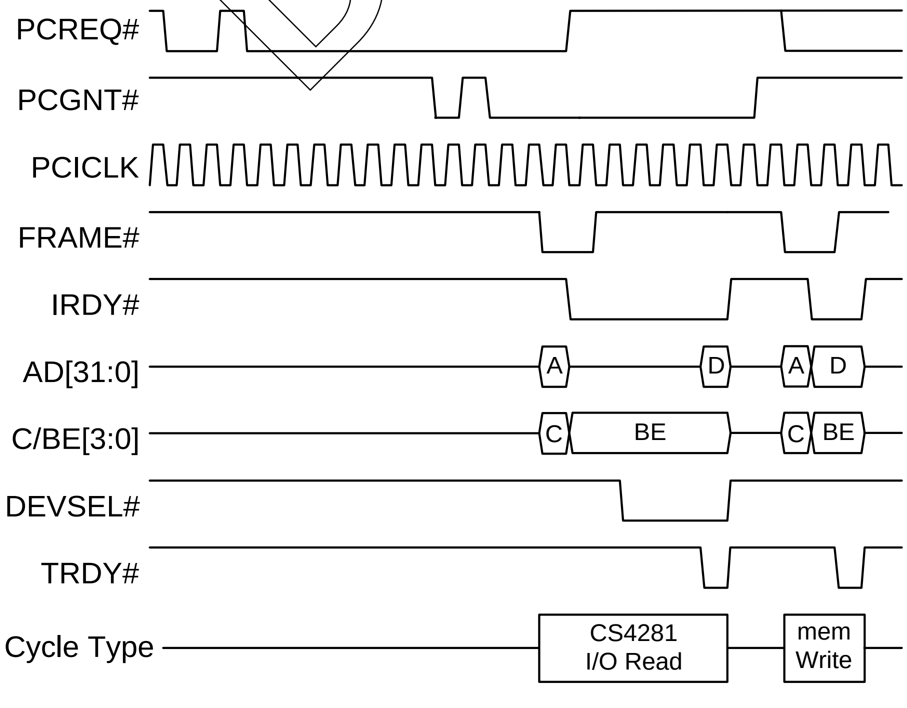


Figure 45. PC/PCI Capture Sequence

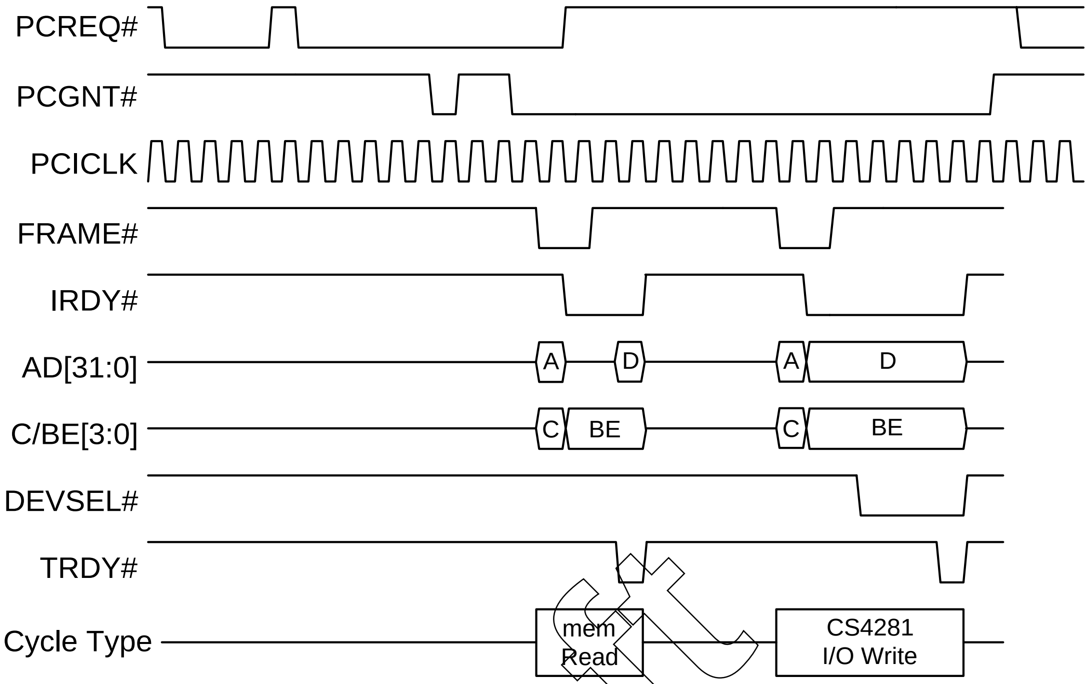


Figure 46. PC/PCI Playback Sequence

14.1 Register Interface

14.1.1 PC/PCI Request Register (PCPRR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ										res	res	res	RDC2	RDC1	RDC0

Address: BA0: 600h, Read-Only

Default: 0000h

Definition: Provided for backwards compatibility and debug purposes. The **RDC[2:0]** bits are sent out the **PCREQ#** line when requesting service and contain the **IOTCR.DMA[1:0]** bits

Bit Descriptions:

RDC2 Requested DMA Engine bit 2: This bit is always 0 since the CS4281 does not support any DMA channel above 3.

RDC1 Requested DMA Engine bit 1: This bit is a copy of **IOTCR.DMA1**.

RDC0 Requested DMA Engine bit 0: This bit is a copy of **IOTCR.DMA0**.

REQ DMA Request Pending: This read-only bit reflects the DMAn engine’s request to the PC/PCI engine to send a request across the **PCREQ#** pin/signal.
0 = No DMAn engine request
1 = DMAn has a PC/PCI request pending. PC/PCI logic should send out a request.

14.1.2 PC/PCI Grant Register (PCPGR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VL													GDC2	GDC1	GDC0

Address: BA0: 604h, Read-Only

Default: 0000h

Definition: Provided for debug purposes only and is the grant status of the PC/PCI interface.

Bit Descriptions:

GDC[2:0] Granted DMA Engine: These bits are the encoded DMA engine number taken directly from the PC/PCI grant signal. No value over 3 should ever be received from the host PC/PCI DMA engine.
0 0 0 = DMA engine 0 (reset default)
0 0 1 = DMA engine 1
0 1 0 = DMA engine 2
0 1 1 = DMA engine 3
1 0 1 = DMA engine 5
1 1 0 = DMA engine 6
1 1 1 = DMA engine 7

VL Valid: This bit indicates whether or not the **GDC[2:0]** bits contain valid information. The PC/PCI engine automatically sets this bit to 1 when it receives a grant from the host PC/PCI agent, and the PC/PCI engine automatically clears **VL** when a DMA transfer completes or is terminated.

14.1.3 PC/PCI Control Register (PCPCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															PCPEN

Address: BA0: 608h, Read-Write

Default: 0000h

Definition: Master enable for the PC/PCI logic. BIOS sets the PC/PCI flag in the config space, CFLR register at boot time. Host configuration software, seeing the PC/PCI flag set, should enable the PC/PCI logic via this register and set the IOTPCP PCI trapping register and set the legacy mode to PC/PCI, **IOTCR.EDMA[1:0]** = 10.

Bit Descriptions:

PCPEN PC/PCI Enable: Setting this bit to 1 enables the PC/PCI logic and enables the **PCREQ#** and **PCGNT#** pins for PC/PCI functions (as opposed to EEPROM or GPIO functions).
0 = PC/PCI disabled (reset default)
1 = PC/PCI enabled

15. DDMA

Information in this section is based on the *Distributed DMA Support for PCI Systems,* revision 6.0 document. DDMA is enabled by setting **IOTCR.EDMA[1:0]** = 11. DDMA takes a legacy DMA controller and tweaks it to support only 1 DMA engine. Reads from registers not supported return 0, and writes to un-supported registers do not have any adverse affects.

Dma Base+	Access	Name	[1]Host Reg.	Comment
00h	write	Legacy Base Address (+Current) low	DBAn.BAL[7:0], DCAn.CAL[7:0]	
00h	read	Legacy Current Address low	DCAn.CAL[7:0]	
01h	write	Legacy Base Address (+Current) high	DBAn.BAH[7:0], DCAn.CAH[7:0]	
01h	read	Legacy Current Address high	DCAn.CAH[7:0]	
02h	write	Legacy Page Register	DBAn.BAU[7:0], DCAn.CAU[7:0]	
02h	read	Legacy Page Register	DBAn.BAU[7:0]	
03h	write	Base Address bits 24 - 31	bit bucket	optional in DDMA
03h	read	Current Address bits 24 - 31	00h	
04h	write	Legacy Base Count (+Current) low	DBCn.BCL[7:0], DCCn.BCL[7:0]	
04h	read	Legacy Current Count low	DCCn.BCL[7:0]	
05h	write	Legacy Base Count (+Current) high	DBCn.BCH[7:0], DCCn.BCH[7:0]	
05h	read	Legacy Current Count high	DCCn.BCH[7:0]	
06h	write	Base Count bits 16-23	bit bucket	optional in DDMA
06h	read	Current Count bits 16-23	00h	
07h	write	DDMA Reserved	bit bucket	
07h	read	DDMA Reserved	00h	
08h	write	Command	bit bucket	CS4281 currently doesn’t support
08h	read	Status	HDSRn.bits[7:0]	Request Pending and TC status
09h	write	Request	bit bucket	CS4281 currently doesn’t support
09h	read		00h	not used
0Ah	write	Single Mask	bit bucket	Single engine mask not used by DDMA. DDMA redirects to 0Fh.
0Ah	read		00h	not used
0Bh	write	Mode	DMRn.bits[7:0]	Auto-init, transfer mode, and direction
0Bh	read		00h	not used
0Ch	write	Clear byte pointer f/f	IOTCC.BPFF = 0	Byte pointer not used by DDMA
0Ch	read		00h	not used
0Dh	write	Master Clear	bit bucket	CS4281 currently doesn’t support
0Dh	read		00h	Temporary register not supported in DDMA
0Eh	write	Clear all Masks	bit bucket	DDMA host redirects to 0Fh
0Eh	read		00h	not used
0Fh	write	Multi-Engine Mask Register	DCRn.bits[7:0]	Sets/Clears DCRn.MSK bit in D0
0Fh	read	Multi-Engine Mask Register	DCRn.bits[7:0]	Reads DCRn.MSK bit in D0

[1] In Host Reg., “n” is selected by IOTCR.DMA[1:0]

Table 10. DDMA I/O Map

Legacy 8237 controllers support 4 DMA engines and access each engine through register pairs at locations DmaBase+0 to DmaBase+7. DDMA redefines registers 0-6 by expanding these registers to support only one DMA engine. DDMA also redefines the rest of the trapped registers to support status/setup for only a single DMA engine (as opposed to combined setup/status on legacy 8237s). Custom start-up software will write the CS4281 IOTDMA register base and let the host DDMA engine know where the CS4281 DDMA engine is mapped. The Host DDMA engine traps Legacy DMA requests and redirects the request to as many as 4 DDMA engines. The CS4281 DDMA engine directs this register I/O to one of the four CS4281 DMA engines, base on the **IOTCR.DMA[1:0]** bits. Therefore, in Table 10, for DmaBase+0, 1, 2, 4, 5, 8, 0Bh, 0Fh:

- DMA[1:0] = 0, map n to DMA engine 0 registers
- DMA[1:0] = 1, map n to DMA engine 1 registers
- DMA[1:0] = 2, map n to DMA engine 2 registers
- DMA[1:0] = 3, map n to DMA engine 3 registers

Unused registers trapped in the DmaBase+0 to DmaBase+0Fh need reads to respond with zeros and writes to go into the bit bucket.

16. SOUND BLASTER

The Sound Blaster Pro emulation is done in hardware and enabled through the *Sound System Control Register*, **SSCR.SB**. When **SSCR.SB** goes from a 0 to a 1, the Sound Blaster engine forces a Sound Blaster reset, which initializes the state machine and sets the default mixer values.

The Sound Blaster emulation controller recognizes accesses to the Sound Blaster trap range. Typically this range is I/O address 0220h to 022Fh. The address and read/write function are interpreted and mapped to a read/write of the appropriate physical register. All Sound Blaster I/O registers are mapped into PCI memory.

Sound Blaster FM functions map directly to the FM hardware registers. This is a simple address translation from the I/O trap address to the memory mapped FM address.

The Sound Blaster DSP is emulated in hardware by responding to the I/O traps through the IOTSB register. The I/O trapping logic maps the I/O locations to PCI memory mapped space. The controller recognizes access to the DSP registers and performs the appropriate function.

Sound Blaster ADPCM is supported by a ADPCM engine in the DSP controller. Full ADPCM 2:1 (ADPCM4) decompression is supported. The ADPCM 3:1 (ADPCM2.6), and 4:1 (ADPCM2) are decompressed as zeros (no data). Sound Blaster ADPCM compression is not supported.

When SB playback is enabled, the SRC slot assignments in SRCSA for playback are copied to the slot assignment bits in the FCRn associated with the legacy DMAn engine. This forces the playback SRC to be connected to the proper FIFO. Host software must set the Playback SRC mapping to the desired slots, typically 3 and 4. Similarly, when SB record is enabled, the SRC slot assignments in SRCSA for record are copied to the same slot assignment bits in the FCRn associated with the legacy DMAn. Host software must also set the record SRC mapping to the desired slots; typically 3 and 4 on the primary AC link. Note that AC-link slot numbers are encoded with the encoding listed in Table 37 on page 207. When the SB engine changes from playback to record and vice-versa, **FCRn.FEN** is toggled to flush the legacy FIFOn.

16.1 Sound Blaster Register Addressing

The following table defines the legacy Sound Blaster I/O address mapping into PCI memory address space. The SbBase address is set in I/O Trap for Sound Blaster (IOTSB).

I/O Address	Description	Type	Address (BA0 Offset)
SBbase+0	Left FM Status Port	Read	730h
SBbase+0	Left FM Register Address Port	Write	730h
SBbase+1	Left FM Data Port	Read/Write	734h
SBbase+2	Right FM Status Port	Read	738h
SBbase+2	Right FM Register Address Port	Write	738h
SBbase+3	Right FM Data Port	Read/Write	73Ch
SBbase+4	Mixer Register Address	Read/Write	700h
SBbase+5	Mixer Data Port	Read/Write	704h
SBbase+6	Reset	Write	708h
SBbase+8	FM Status Port	Read	730h
SBbase+8	FM Register Port	Write	730h
SBbase+9	FM Data Port	Read/Write	734h
SBbase+A/B	Read Data Port	Read	70Ch
SBbase+C/D	Command/Write Data	Write	710h
SBbase+C/D	Write Buffer Status	Read	710h
SBbase+E/F	Data Available Status	Read	714h

Table 11. Sound Blaster Direct Registers

16.2 Sound Blaster Register Description

16.2.1 Sound Blaster Mixer Address Register (SBMAR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	MAD0

Address: BA0: 700h, Read Only
I/O: SBbase+4

Default: 000000FFh

Definition: The address port for selecting the Sound Blaster Pro Indirect Mixer register to be accessed through the Mixer Data Register. Although not the norm, the Mixer address and data register can be written simultaneously using PC word accesses.

Bit Descriptions:

MAD[7:0] Mixer Address.

16.2.2 Sound Blaster Mixer Data Register (SBMDR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0

Address: BA0: 704h, Read/Write
I/O: SBbase+5

Default: 00000011h

Definition: The data port for reading and writing the selected (SBMAR register) Sound Blaster Pro Mixer register. See Table 22. Although not the norm, the Mixer address and data register can be written simultaneously using PC word accesses.

Bit Descriptions:

MD[7:0] Mixer Data

16.2.3 Sound Blaster Reset Register (SBRR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															SBR

Address: BA0: 708h, Write-Only
I/O: SBbase+6

Default: 000000FFh

Definition: Controls Sound Blaster Reset functionality. The Sound Blaster interface will reset on a logical 1 to 0 transition of the **SBR** bit.

Bit Descriptions:

SBR Reset control bit

16.2.4 Sound Blaster Read Data Port (SBRDP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

Address: BA0: 70Ch, Read-Only
I/O: SBbase+A/B

Default: 000000AAh

Definition: Read data port for Sound Blaster Command and data access. When **SBDAS.RBF** = 1, new data can be read from this register

Bit Descriptions:

RD[5:0] Read Data

16.2.5 Sound Blaster Write Data Port (SBWDP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								WD7	WD6	WD5	WD4	WD3	WD2	WD1	WD0

Address: BA0: 710h, Write-Only
I/O: SBbase+C/D

Default: 00000000h

Definition: Write data port for Sound Blaster Command and data access. When **SBWBS.WBE** = 1, data can be written to this register

Bit Descriptions:

WD[7:0] Write Data

16.2.6 Sound Blaster Write Buffer Status (SBWBS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								WBB	RD6	RD5	RD4	RD3	RD2	RD1	RD0

Address: BA0: 710h, Read-Only
I/O: SBbase+C/D

SB Reset Default: 000000AAh (will change to 2Ah after delay)
POR Default: 0000002Ah (**SSCR.SB** = 0)

Definition: Write status port for Sound Blaster Command and Data write access.

Bit Descriptions:

WBB Write Buffer Empty:
0 - Next data item can be written to SBWDP
1 - Write Buffer Busy. Software must wait before writing data to SBWDP

RD[6:0] Lower 7 bits from data in SBRDP.

16.2.7 Sound Blaster Read Buffer Status (SBRBS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RBF	RD6	RD5	RD4	RD3	RD2	RD1	RD0

Address: BA0: 714h, Read-Only
I/O: SBbase+E/F

SB Reset Default: 0000002Ah (will change to AAh after delay)
POR Default: 00000000h (**SSCR.SB** = 0)

Definition: Read status port for Sound Blaster Command and Data read access. When bit 7 is a 1, new read data is available in the data read port. Reading this register also clears all Sound Blaster generated interrupts.

Bit Descriptions:

RBF Read Buffer Full:
0 - Stale Data
1 - New Data available in SBRDP

RD[6:0] Lower 7 bits from data in SBRDP.

16.3 Sound Blaster Hardware Overview

The Sound Blaster block accepts commands and data trapped by the PCI interface and passed along by the Control Block. Mixer data and DSP commands and data are translated to appropriate reads and writes of the CS4281's target registers which set up the digital data path in the CS4281 and the analog path in the AC '97 part to effectively emulate a Sound Blaster chip set.

The Sound Blaster Block supports both playback and capture for DMA and PIO. Support is also included for ADPCM through DMA-only playback ADPCM 2:1, 3:1, and 4:1 decoding (although 4:1 and 3:1 will play silence). Capture ADPCM will not be supported. Other accesses that set up the data paths through both the DSP Cmd/Data and Mixer ports are also supported. The SB hardware supports the same number of capture and playback sample frequencies (mono and stereo are considered one sample) from 6 kHz to 48 kHz.

16.4 Sound Blaster Mixer

This block communicates with the SB DSP block as well as the PCI interface and AC-link. Some of its data is required by the DSP and also a few DSP commands such as Speaker On & Off configure the AC '97 codec.

16.4.1 Sound Blaster Mixer Mapping onto the AC '97/CS4281 Mixer

The Sound Blaster Mixer is mapped onto the AC '97 generic mixer and the digital portion on board the CS4281 chip as described in the following table. Since the AC '97 link supports two codecs, the particular codec to update is controlled through the **SSCR.MVCS** bit. When **SSCR.MVCS** is clear (default) the primary codec is

Register	D7	D6	D5	D4	D3	D2	D1	D0
00h	DATA RESET							
02h	RESERVED							
04h	VOICE VOLUME LEFT			1	VOICE VOLUME RIGHT			1
06h	RESERVED							
08h	RESERVED							
0Ah		X	X	X	X	MIC MIXING		1
0Ch	X	X	X	X	X	INPUT SELECT		X
0Eh	X	X	X	X	X	X	VSTC	X
20h	RESERVED							
22h	MASTER VOLUME LEFT			1	MASTER VOLUME RIGHT			1
24h	RESERVED							
26h	FM VOLUME LEFT			1	FM VOLUME RIGHT			1
28h	CD VOLUME LEFT			1	CD VOLUME RIGHT			1
2Ah	RESERVED							
2Ch	RESERVED							
2Eh	LINE VOLUME LEFT			1	LINE VOLUME RIGHT			1
40h - FFh	1	1	1	1	1	1	1	1

Table 12. Sound Blaster Mixer (indirect) registers

selected. When **SSCR.MVCS** is set, the secondary codec is selected.

The SB Master Volume data can to be written to all three AC ‘97 Master Volume controls since the Sound Blaster user might be using any or all of these outputs to listen to his game audio. Master volume translation is controlled through the Sound System Control Register (SSCR). This register contains bits that determine which AC ‘97 codec to update (primary or secondary) and which volumes on that codec to update. The bits in SSCR control master volume updates for SB Master and Hardware Volumes and are:

- **MVCS** - Selects Primary (0) or Secondary (1) AC ‘97 codec.
- **MVLD** - Master Volume Line-Out Disable. When set, register 0x02 is NOT updated.
- **MVAD** - Master Volume Alternate/Headphone Disable. When set, register 0x04 is NOT updated.
- **MVMD** - Master Volume Mono Out Disable. When set, register 0x06 is NOT updated.

The Sound Blaster mixer supports independent volume controls for the FM and the PCM (PSRC) path. Since the CS4281 mixes FM digitally before sending the data to the AC Link, independent controls for PCM (voice) and FM are provided prior to mixing.

The CS4281 will write the appropriate volume to the Master Input Volume depending on which input is selected by the Input Setting Register. The Input Select to AC Link translations are in Table 25.

The mapping of the Sound Blaster Mixer into the CS4281/AC ‘97 mixer is drawn out schematically

SB Pro Mixer Read Table

Mixer Index	Data Returned
0	*last data read
2	[22h]
4	[04h]
6	[26h]
8	[28h]
0Ah	[0Ah]
0Ch	[0Ch]
0Eh	[0Eh]
20h	11h
22h	[22h]
24h	*last data read
26h	[26h]
28h	[28h]
2Ah	*last data read
2Ch	*last data read
2Eh	[2Eh]
40h - FFh	FFh

* return the same thing as the previous SB mixer data register read
[n] - contents of mixer register n
Mixer Index bits 0 and 4 are don’t cares and alias to the above locations

Table 13. SB Pro Mixer - All Read Values

Sound Blaster Register	SB Index	CS4281/AC ‘97 Register	AC ‘97 Index
Voice Volume	x04[7:5] x04[3:1]	CS4281 PPLVC CS4281 PPRVC	--
Mic Mix	x0A[2:1]	Mic Volume	x0E[7:0]
Input Setting	x0C	L Record Select R Record Select	x1A[15:8] x1A[7:0]
Master Volume*	x22[7:5] x22[3:1]	L Master Volume* R Master Volume* L Headphone/Alternate Volume* R Headphone/Alternate Volume* Master Volume Mono*	x02[15:8] x02[7:0] x04[15:8] x04[7:0] x06[7:0]
FM Volume	x26[7:5] x26[3:1]	CS4281 FMLVC CS4281 FMRVC	--
CD Volume	x28[7:5] x28[3:1]	L CD Volume R CD Volume (Master Input Volume)	x12[15:8] x12[7:0]
Line Volume	x2E[7:5] x2E[3:1]	L Line In Volume R Line In Volume (Master Input Volume)	x10[15:8] x10[7:0]

* Master Volume mapping is software-controllable through the SSCR register, bits **MVCS**, **MVLD**, **MVAD**, **MVMD**. See text for more information.

Table 14. Sound Blaster Mixer Mapping to AC ‘97 Registers

SB Volume Value	Master Volume		Line/CD Volume		Voice/FM Volume	Mic (Mix) Volume	
	AC ‘97 Value (Hex)	Atten. (dB)	AC‘97 Value (Hex)	Atten. (dB)	Atten. (dB)	AC’97 Value (Hex)	Gain (dB)
0	1F	-94.5/mute*	1F	-34.5/mute*	mute	45	+24.5
1	12	-27.5	1A	-27	-27	49	+18.5
2	0E	-21	16	-21	-21	01	+10.5
3	0B	-16.5	13	-16.5	-16.5	04	+6
4	08	-12	10	-10	-12		
5	05	-7	0D	-7.5	-7.5		
6	02	-3	0A	-3	-3		
7	00	0	08	0	0		

* If left and right channels are both 0, AC ‘97 mute bit is set. If one channel is not 0, the other channel is set to maximum attenuation. Maximum value dependent on AC ‘97 codec used. SB supports mute/channel and AC ‘97 codecs do not.

Table 15. Sound Blaster Volume Translations

SB Input Select (0Ch)	SB Value (Hex)	AC ‘97 1Ah Register (Hex)
Mic Source (default)	0	0000h
CD Source	1	0101h
Mic Source	2	0000h
Line-In Source	3	0404h

Table 16. Sound Blaster Input Select

in Figure 47. Valid SB audio paths are highlighted.

16.5 SB Configuration Setup Requirements

- Use Environment BLASTER Variable to determine which DMAn (0-3) to use as Legacy
- Setup legacy DMAn (CCLS, PC/PCI, DDMA, and Polled operation); **IOTCR.DMA[1:0]** = n
  - Set FIFOn size to minimum (2-4 samples); **FCRn.SZ[7:0]**
  - Turn on Legacy DMA engine (will set Mask); **DMRn.DMA**
  - Set SB format: only DMRn format bits on: **CBC, SWAPC, USIGN, SIZE8**

    - **MONO** function is controlled internally by SB engine, based on game-programmed format

    - **DMA/POLL** and **DMRn.TR[1:0]** functions will also be controlled directly by SB engine
  - Set **FCRn.DACZ**
  - Clear **FCRn.PSH** (required for SB silence command 80h)
  - Enable FIFOm; **FCRn.FEN**
  - Initialize Upper Base Count and Upper Address Count to 0
- Interrupt Setup - check environment BLASTER variable
  - Mask unused interrupts (usually all but SBINT); HIMR
  - Optionally switch from PCI to ISA interrupts; HIMR, IIER
- Turn on Sound Blaster and FM; **SSCR.SB** = 1, **SSPM.FMEN** = 1.
- Disable Host Interrupts from legacy DMA; DCRn interrupt enable bits = 0
- Setup and turn on legacy I/O trapping for SB, FM, and Game Port; IOTSB, IOTFM, and IOTGP
- If PC/PCI, set up PC/PCI registers and I/O Trap for PC/PCI; IOTPCP and **PCPCR.PCPEN** set
- Select, setup, and turn on legacy I/O trapping for DMA (plus global trapping); IOTCR
- Setup CSRC and PSRC slot mapping (between AC link and FIF0m); SRCSA
  - NOTE: SRCSA mapping will be loaded into the FIFO slot mapping in FCRm when SB playback or record is enabled.
- Enable SRCs; **SSPM.PSRCEN** and **SSPM.CSRCEN** set.
- Do SB Reset to setup mixer registers
- Set AC Link Output Slot Valid bits in ACOSV
- Make sure Codec is powered up and ADC/DAC running in proper mode (sets ACISV or ACISV2 valid bits).
- Write Codec DAC volume to 0 dB (Index 18h = 0808h). SB uses CS4281 registers for Voice volume

When exiting Sound Blaster mode (tear-down):

- Sound Blaster Voice Volume unmuted, 0 dB; PPLVC/PPRVC = 0
- Turn off Sound Blaster; **SSCR.SB** = 0
- Disable legacy I/O trapping; **IOTCR.ITD** = 1
- Disable legacy DMA

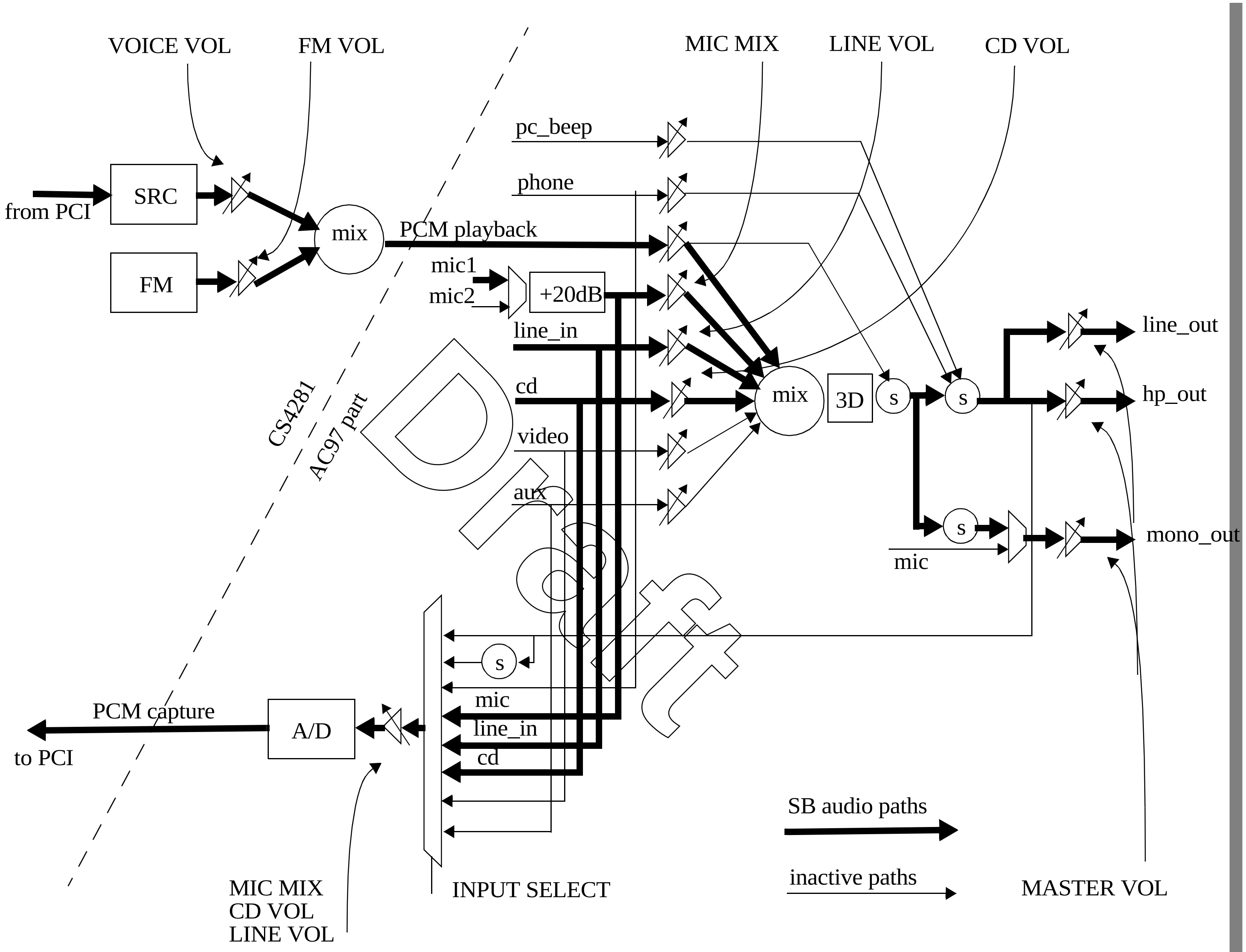


Figure 47. Sound Blaster to AC '97 Mixer Mapping

16.6 Sound Blaster DSP

The Sound Blaster DSP consists of the main SB state machine and slave logic to convert SB written values to values appropriate for the CS4281 target registers. These target registers include some of the Mixer registers described in the previous section to facilitate execution of such commands as the Speaker On and Off commands, and also include the registers that control the DMA, FIFO, SRC, and AC-link slot assignment. The DSP also manages the SB Data Available and Busy handshake mechanism between the CS4281 and the host software.

In Sound Blaster, both PIO and DMA consist only of 8-bit data transfers. There is no 16 or 20/24 bit Sound Blaster data path. The Sound Blaster standard does not support full duplex data transfers, so that one DMA/FIFO/AC ‘97 stereo stream is all that is required. Also, it should be noted that *Sound Blaster stereo reverses the normal Sound System byte order* -- for Sound Blaster, first the right, then the left byte of stereo data arrive in a DMA transfer.

16.7 Sound Blaster Hardware Handshake

The Sound Blaster interface uses a hardware handshake mechanism for processing commands. The mixer does not use a handshake mechanism and is always assumed to be available for ISA bus accesses. Two handshake bits are used: Command Busy and Data Available. The Command Busy (otherwise referred to as SB Busy) is located in the Write Buffer Status (SBWBS) Register, bit 7. The Data Available bit is located in the Data Available Status (SBDAS) Register, bit 7. The Command Busy bit indicates when the CS4281 is still busy executing a Sound Blaster command. The Data Available bit is used to indicate when the CS4281 has responded to a SB command with the required data. Reading this register also clears any Sound Blaster generated interrupts.

The handshake works as follows:

Command Busy: a write to the SB Reset Register, SBbase + 6, with bit 0 equal to one or any write to the SB Command Register, SBbase + C/D, will set this bit. It is cleared by the CS4281 when the command or reset has been fully executed from the point of view of the SB state machine. In other words, if a write is pending across the AC-link, but the appropriate Mixer Register has already been updated, the command is considered to have been executed and Command Busy would be released.

Data Available: when the SB state machine loads the Read Data Port with requested data, this bit is set. It is cleared by a read of the Read Data Port, SBbase + A/B, via the PCI bus.

16.8 Sound Blaster Reset

A Sound Blaster Reset Command is generated by writing a one and then a zero to the Sound Blaster Reset Register (SBRR), SBbase+6, successively. This one-to-zero transition is detected in the SB state machine and initiates a reset of the Sound Blaster hardware. This reset should completely reconfigure the SB hardware, including the CS4281 digital data path and AC ‘97 analog mixer so that SB is in its initial state and ready for commands.

16.9 Sound Blaster Interrupts

Sound Blaster commands can cause the hardware in the CS4281 to generate Sound Blaster interrupts (HISR.SBINT) for some SB commands. Sound Blaster interrupts are cleared by the SB state machine upon reading the *Data Available Status* (SBDAS) register (SBbase+E). Commands like Silence, Generate Interrupt, and all the ADPCM commands cause the SB interrupt which comes directly from the SB state (DSP) machine. These interrupts appear on the bus (either PCI INTA or ISA IRQ[A:C]).

SB Volume Control	SB Value (Hex)	SB Register (Hex)	AC '97 Registers (Hex)	AC '97 Value (Hex)	Volume (dB)
Master Volume	99	22	02, 04, 06	0808	-12
Voice Volume	99	04	CS4281 PPxVC	07, 07	-10.5
FM Volume	99	26	CS4281 FMxVC	07, 07	-10.5
CD Volume	11	28	12	9F1F	muted
Mic Mix	11	0A	0E	0045	+24.5
Input Setting	11	0C	1A	0000	mic selected
Output Setting	11		N/A		mono
Line Volume	11	2E	10	9F1F	muted

Table 17. Sound Blaster Reset Volume Values

TC	SB Mono Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
0-89	≤ 5988	6024	lots	255
90	6024	6024	-0.01	255
91	6060	6071	0.18	253
92	6097	6095	-0.03	252
93	6134	6144	0.16	250
94	6172	6194	0.35	248
95	6211	6244	0.53	246
96	6250	6269	0.31	245
97	6289	6321	0.51	243
98	6329	6347	0.29	242
99	6369	6400	0.49	240
100	6410	6427	0.26	239
101	6451	6481	0.47	237
102	6493	6508	0.24	236
103	6535	6564	0.45	234
104	6578	6592	0.22	233
105	6622	6649	0.41	231
106	6666	6678	0.18	230
107	6711	6737	0.39	228
108	6756	6767	0.16	227
109	6802	6827	0.36	225
110	6849	6857	0.12	224
111	6896	6919	0.33	222
112	6944	6950	0.09	221
113	6993	7014	0.30	219
114	7042	7046	0.05	218
115	7092	7111	0.27	216
116	7142	7144	0.03	215
117	7194	7211	0.24	213
118	7246	7245	-0.01	212
119	7299	7314	0.21	210

Table 18. Mono SB - TC to Sample Frequency Translation

TC	SB Mono Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
120	7352	7349	-0.04	209
121	7407	7420	0.18	207
122	7462	7456	-0.08	206
123	7518	7529	0.15	204
124	7575	7567	-0.11	203
125	7633	7642	0.12	201
126	7692	7719	0.35	199
127	7751	7797	0.59	197
128	7812	7837	0.32	196
129	7874	7918	0.55	194
130	7936	8000	0.81	5
131	8000	8000	0.00	5
132	8064	8000	-0.79	5
133	8130	8170	0.49	188
134	8196	8214	0.22	187
135	8264	8303	0.47	185
136	8333	8348	0.18	184
137	8403	8440	0.44	182
138	8474	8486	0.14	181
139	8547	8581	0.40	179
140	8620	8629	0.11	178
141	8695	8727	0.37	176
142	8771	8777	0.07	175
143	8849	8879	0.33	173
144	8928	8930	0.03	172
145	9009	9035	0.29	170
146	9090	9089	-0.01	169
147	9174	9198	0.26	167
148	9259	9253	-0.06	166
149	9345	9366	0.22	164
150	9433	9423	-0.10	163
151	9523	9540	0.18	161
152	9615	9600	-0.16	160
153	9708	9722	0.14	158

Table 18. Mono SB - TC to Sample Frequency Translation (Continued)

TC	SB Mono Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
154	9803	9783	-0.20	157
155	9900	9910	0.10	155
156	10000	9974	-0.26	154
157	10101	10105	0.04	152
158	10204	10240	0.35	150
159	10309	10378	0.67	148
160	10416	10449	0.32	147
161	10526	10593	0.64	145
162	10638	10667	0.27	144
163	10752	10817	0.60	142
164	10869	10894	0.23	141
165	10989	11025	0.32	4
166	11111	11025	-0.77	4
167	11235	11294	0.53	136
168	11363	11378	0.13	135
169	11494	11549	0.48	133
170	11627	11636	0.08	132
171	11764	11815	0.44	130
172	11904	11907	0.03	129
173	12048	12094	0.39	127
174	12195	12190	-0.04	126
175	12345	12387	0.34	124
176	12500	12488	-0.10	123
177	12658	12694	0.29	121
178	12820	12800	-0.16	120
179	12987	13017	0.23	118
180	13157	13128	-0.22	117
181	13333	13357	0.18	115
182	13513	13474	-0.29	114
183	13698	13714	0.12	112
184	13888	13838	-0.36	111
185	14084	14092	0.05	109
186	14285	14222	-0.44	108
187	14492	14491	-0.01	106
188	14705	14629	-0.52	105
189	14925	14913	-0.08	103
190	15151	15208	0.38	101
191	15384	15515	0.85	99
192	15625	15673	0.31	98
193	15873	16000	0.80	3
194	16129	16000	-0.80	3
195	16393	16516	0.75	93
196	16666	16696	0.18	92
197	16949	17067	0.69	90
198	17241	17258	0.10	89
199	17543	17655	0.64	87
200	17857	17860	0.02	86
201	18181	18286	0.58	84
202	18518	18506	-0.06	83

Table 18. Mono SB - TC to Sample Frequency Translation (Continued)

TC	SB Mono Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
203	18867	18963	0.51	81
204	19230	19200	-0.16	80
205	19607	19692	0.44	78
206	20000	19948	-0.26	77
207	20408	20480	0.35	75
208	20833	20757	-0.37	74
209	21276	21333	0.27	72
210	21739	22061	1.48	2
211	22222	22061	-0.72	2
212	22727	22588	-0.61	68
213	23255	23273	0.08	66
214	23809	23631	-0.75	65
215	24390	24381	-0.04	63
216	25000	24774	-0.90	62
217	25641	25600	-0.16	60
218	26315	26034	-1.07	59
219	27027	26947	-0.29	57
220	27777	27429	-1.25	56
221	28571	28444	-0.44	54
222	29411	29538	0.43	52
223	30303	30720	1.38	50
224	31250	31347	0.31	49
225	32258	32681	1.31	47
226	33333	33391	0.17	46
227	34482	34909	1.24	44
228	35714	35721	0.02	43
229	37037	37463	1.15	41
230	38461	38400	-0.16	40
231	40000	40421	1.05	38
232	41666	41514	-0.37	37
233	43478	44122	1.48	1
234	45454	44122	-2.93	1
235	47619	48000	0.80	0
236-255	≥ 50000	48000	lots	0

Table 18. Mono SB - TC to Sample Frequency Translation (Continued)

TC	SB Stereo Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
0-172	≤ 5952	6024	lots	255
173	6024	6024	-0.01	255
174	6097	6095	-0.03	252
175	6172	6169	-0.05	249
176	6250	6244	-0.10	246
177	6329	6321	-0.13	243
178	6410	6427	0.26	239
179	6493	6508	0.24	236
180	6578	6592	0.22	233
181	6666	6678	0.18	230
182	6756	6767	0.16	227
183	6849	6857	0.12	224
184	6944	6950	0.09	221
185	7042	7046	0.05	218
186	7142	7144	0.03	215
187	7246	7245	-0.01	212
188	7352	7349	-0.04	209
189	7462	7456	-0.08	206
190	7575	7567	-0.11	203
191	7692	7680	-0.16	200
192	7812	7797	-0.19	197
193	7936	8000	0.81	5
194	8064	8000	-0.79	5
195	8196	8214	0.22	187
196	8333	8348	0.18	184
197	8474	8486	0.14	181
198	8620	8629	0.11	178
199	8771	8777	0.07	175
200	8928	8930	0.03	172
201	9090	9089	-0.01	169
202	9259	9253	-0.06	166
203	9433	9423	-0.10	163
204	9615	9600	-0.16	160
205	9803	9783	-0.20	157
206	10000	9974	-0.26	154
207	10204	10172	-0.31	151
208	10416	10378	-0.36	148

Table 19. Stereo SB - TC to Sample Frequency Translation

TC	SB Stereo Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
209	10638	10593	-0.42	145
210	10869	11025	1.44	4
211	11111	11025	-0.77	4
212	11363	11378	0.13	135
213	11627	11636	0.08	132
214	11904	11907	0.03	129
215	12195	12190	-0.04	126
216	12500	12488	-0.10	123
217	12820	12800	-0.16	120
218	13157	13128	-0.22	117
219	13513	13474	-0.29	114
220	13888	13838	-0.36	111
221	14285	14222	-0.44	108
222	14705	14629	-0.52	105
223	15151	15059	-0.61	102
224	15625	16000	2.40	3
225	16129	16000	-0.80	3
226	16666	16696	0.18	92
227	17241	17258	0.10	89
228	17857	17860	0.02	86
229	18518	18506	-0.06	83
230	19230	19200	-0.16	80
231	20000	19948	-0.26	77
232	20833	20757	-0.37	74
233	21739	22061	1.48	2
234	22727	22061	-2.93	2
235	23809	23631	-0.75	65
236	25000	24774	-0.90	62
237	26315	26034	-1.07	59
238	27777	27429	-1.25	56
239	29411	28981	-1.46	53
240	31250	30720	-1.70	50
241	33333	32681	-1.96	47
242	35714	35721	0.02	43
243	38461	38400	-0.16	40
244	41666	44122	5.89	1
245	45454	44122	-2.93	1
246-255	≥ 50000	48000	lots	0

Table 19. Stereo SB - TC to Sample Frequency Translation (Continued)

17. SOUND BLASTER

The Sound Blaster Pro emulation is done in hardware and enabled through the *Sound System Control Register*, **SSCR.SB**. When **SSCR.SB** goes from a 0 to a 1, the Sound Blaster engine forces a Sound Blaster reset, which initializes the state machine and sets the default mixer values.

The Sound Blaster emulation controller recognizes accesses to the Sound Blaster trap range. Typically this range is I/O address 0220h to 022Fh. The address and read/write function are interpreted and mapped to a read/write of the appropriate physical register. All Sound Blaster I/O registers are mapped into PCI memory.

Sound Blaster FM functions map directly to the FM hardware registers. This is a simple address translation from the I/O trap address to the memory mapped FM address.

The Sound Blaster DSP is emulated in hardware by responding to the I/O traps through the IOTSB register. The I/O trapping logic maps the I/O locations to PCI memory mapped space. The controller recognizes access to the DSP registers and performs the appropriate function.

Sound Blaster ADPCM is supported by a ADPCM engine in the DSP controller. Full ADPCM 2:1 (ADPCM4) decompression is supported. The ADPCM 3:1 (ADPCM2.6), and 4:1 (ADPCM2) are decompressed as zeros (no data). Sound Blaster ADPCM compression is not supported.

When SB playback is enabled, the SRC slot assignments in SRCSA for playback are copied to the slot assignment bits in the FCRn associated with the legacy DMAn engine. This forces the playback SRC to be connected to the proper FIFO. Host software must set the Playback SRC mapping to the desired slots, typically 3 and 4. Similarly, when SB record is enabled, the SRC slot assignments in SRCSA for record are copied to the same slot assignment bits in the FCRn associated with the legacy DMAn. Host software must also set the record SRC mapping to the desired slots; typically 3 and 4 on the primary AC link. Note that AC-link slot numbers are encoded with the encoding listed in Table 37 on page 207. When the SB engine changes from playback to record and vice-versa, **FCRn.FEN** is toggled to flush the legacy FIFOn.

17.1 Sound Blaster Register Addressing

The following table defines the legacy Sound Blaster I/O address mapping into PCI memory address space. The SbBase address is set in I/O Trap for Sound Blaster (IOTSB).

I/O Address	Description	Type	Address (BA0 Offset)
SBbase+0	Left FM Status Port	Read	730h
SBbase+0	Left FM Register Address Port	Write	730h
SBbase+1	Left FM Data Port	Read/Write	734h
SBbase+2	Right FM Status Port	Read	738h
SBbase+2	Right FM Register Address Port	Write	738h
SBbase+3	Right FM Data Port	Read/Write	73Ch
SBbase+4	Mixer Register Address	Read/Write	700h
SBbase+5	Mixer Data Port	Read/Write	704h
SBbase+6	Reset	Write	708h
SBbase+8	FM Status Port	Read	730h
SBbase+8	FM Register Port	Write	730h
SBbase+9	FM Data Port	Read/Write	734h
SBbase+A/B	Read Data Port	Read	70Ch
SBbase+C/D	Command/Write Data	Write	710h
SBbase+C/D	Write Buffer Status	Read	710h
SBbase+E/F	Data Available Status	Read	714h

Table 20. Sound Blaster Direct Registers

17.2 Sound Blaster Register Description

17.2.1 Sound Blaster Mixer Address Register (SBMAR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	MAD0

Address: BA0: 700h, Read Only
I/O: SBbase+4

Default: 000000FFh

Definition: The address port for selecting the Sound Blaster Pro Indirect Mixer register to be accessed through the Mixer Data Register. Although not the norm, the Mixer address and data register can be written simultaneously using PC word accesses.

Bit Descriptions:

MAD[7:0] Mixer Address.

17.2.2 Sound Blaster Mixer Data Register (SBMDR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0

Address: BA0: 704h, Read/Write
I/O: SBbase+5

Default: 00000011h

Definition: The data port for reading and writing the selected (SBMAR register) Sound Blaster Pro Mixer register. See Table 22. Although not the norm, the Mixer address and data register can be written simultaneously using PC word accesses.

Bit Descriptions:

MD[7:0] Mixer Data

17.2.3 Sound Blaster Reset Register (SBRR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															SBR

Address: BA0: 708h, Write-Only
I/O: SBbase+6

Default: 000000FFh

Definition: Controls Sound Blaster Reset functionality. The Sound Blaster interface will reset on a logical 1 to 0 transition of the **SBR** bit.

Bit Descriptions:

SBR Reset control bit

17.2.4 Sound Blaster Read Data Port (SBRDP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0

Address: BA0: 70Ch, Read-Only
I/O: SBbase+A/B

Default: 000000AAh

Definition: Read data port for Sound Blaster Command and data access. When **SBDAS.RBF** = 1, new data can be read from this register

Bit Descriptions:

RD[5:0] Read Data

17.2.5 Sound Blaster Write Data Port (SBWDP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								WD7	WD6	WD5	WD4	WD3	WD2	WD1	WD0

Address: BA0: 710h, Write-Only
I/O: SBbase+C/D

Default: 00000000h

Definition: Write data port for Sound Blaster Command and data access. When **SBWBS.WBE** = 1, data can be written to this register

Bit Descriptions:

WD[7:0] Write Data

17.2.6 Sound Blaster Write Buffer Status (SBWBS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								WBB	RD6	RD5	RD4	RD3	RD2	RD1	RD0

Address: BA0: 710h, Read-Only
I/O: SBbase+C/D

SB Reset Default: 000000AAh (will change to 2Ah after delay) POR Default: 0000002Ah (**SSCR.SB** = 0)

Definition: Write status port for Sound Blaster Command and Data write access.

Bit Descriptions:

WBB Write Buffer Empty:
0 - Next data item can be written to SBWDP
1 - Write Buffer Busy. Software must wait before writing data to SBWDP

RD[6:0] Lower 7 bits from data in SBRDP.

17.2.7 Sound Blaster Read Buffer Status (SBRBS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								RBF	RD6	RD5	RD4	RD3	RD2	RD1	RD0

Address: BA0: 714h, Read-Only
I/O: SBbase+E/F

SB Reset Default:0000002Ah (will change to AAh after delay)
POR Default:00000000h (**SSCR.SB** = 0)

Definition: Read status port for Sound Blaster Command and Data read access. When bit 7 is a 1, new read data is available in the data read port. Reading this register also clears all Sound Blaster generated interrupts.

Bit Descriptions:

RBF Read Buffer Full:
0 - Stale Data
1 - New Data available in SBRDP

RD[6:0] Lower 7 bits from data in SBRDP.

17.3 Sound Blaster Hardware Overview

The Sound Blaster block accepts commands and data trapped by the PCI interface and passed along by the Control Block. Mixer data and DSP commands and data are translated to appropriate reads and writes of the CS4281's target registers which set up the digital data path in the CS4281 and the analog path in the AC '97 part to effectively emulate a Sound Blaster chip set.

The Sound Blaster Block supports both playback and capture for DMA and PIO. Support is also included for ADPCM through DMA-only playback ADPCM 2:1, 3:1, and 4:1 decoding (although 4:1 and 3:1 will play silence). Capture ADPCM will not be supported. Other accesses that set up the data paths through both the DSP Cmd/Data and Mixer ports are also supported. The SB hardware supports the same number of capture and playback sample frequencies (mono and stereo are considered one sample) from 6 kHz to 48 kHz.

17.4 Sound Blaster Mixer

This block communicates with the SB DSP block as well as the PCI interface and AC-link. Some of its data is required by the DSP and also a few DSP commands such as Speaker On & Off configure the AC '97 codec.

17.4.1 Sound Blaster Mixer Mapping onto the AC '97/CS4281 Mixer

The Sound Blaster Mixer is mapped onto the AC '97 generic mixer and the digital portion on board the CS4281 chip as described in the following table. Since the AC '97 link supports two codecs, the particular codec to update is controlled through the **SSCR.MVCS** bit. When **SSCR.MVCS** is clear (default) the primary codec is selected. When **SSCR.MVCS** is set, the secondary codec is selected.

The SB Master Volume data can to be written to all three AC '97 Master Volume controls since the Sound Blaster user might be using any or all of these outputs to listen to his game audio. Master volume translation is controlled through the Sound System Control Register (SSCR). This register contains bits that determine which AC '97 codec to update (primary or secondary) and which volumes on that codec to update. The bits in SSCR control master volume updates for SB Master and Hardware Volumes and are:

Register	D7	D6	D5	D4	D3	D2	D1	D0
00h	DATA RESET							
02h	RESERVED							
04h	VOICE VOLUME LEFT			1	VOICE VOLUME RIGHT			1
06h	RESERVED							
08h	RESERVED							
0Ah		X	X	X	X	MIC MIXING		1
0Ch	X	X	X	X	X	INPUT SELECT		X
0Eh	X	X	X	X	X	X	VSTC	X
20h	RESERVED							
22h	MASTER VOLUME LEFT			1	MASTER VOLUME RIGHT			1
24h	RESERVED							
26h	FM VOLUME LEFT			1	FM VOLUME RIGHT			1
28h	CD VOLUME LEFT			1	CD VOLUME RIGHT			1
2Ah	RESERVED							
2Ch	RESERVED							
2Eh	LINE VOLUME LEFT			1	LINE VOLUME RIGHT			1
40h - FFh	1	1	1	1	1	1	1	1

Table 21. Sound Blaster Mixer (indirect) registers

- **MVCS** - Selects Primary (0) or Secondary (1) AC ‘97 codec.
- **MVLD** - Master Volume Line-Out Disable. When set, register 0x02 is NOT updated.
- **MVAD** - Master Volume Alternate/Headphone Disable. When set, register 0x04 is NOT updated.
- **MVMD** - Master Volume Mono Out Disable. When set, register 0x06 is NOT updated.

The Sound Blaster mixer supports independent volume controls for the FM and the PCM (PSRC) path. Since the CS4281 mixes FM digitally before sending the data to the AC Link, independent controls for PCM (voice) and FM are provided prior to mixing.

The CS4281 will write the appropriate volume to the Master Input Volume depending on which input is selected by the Input Setting Register. The Input Select to AC Link translations are in Table 25.

The mapping of the Sound Blaster Mixer into the CS4281/AC ‘97 mixer is drawn out schematically in Figure 47. Valid SB audio paths are highlighted.

SB Pro Mixer Read Table	
Mixer Index	**Data Returned**
0	*last data read
2	[22h]
4	[04h]
6	[26h]
8	[28h]
0Ah	[0Ah]
0Ch	[0Ch]
0Eh	[0Eh]
20h	11h
22h	[22h]
24h	*last data read
26h	[26h]
28h	[28h]
2Ah	*last data read
2Ch	*last data read
2Eh	[2Eh]
40h - FFh	FFh

1. * return the same thing as the previous SB mixer data register read
2. [n] - contents of mixer register n
3. Mixer Index bits 0 and 4 are don’t cares and alias to the above locations

Table 22. SB Pro Mixer - All Read Values

Sound Blaster Register	SB Index	CS4281/AC ‘97 Register	AC ‘97 Index
Voice Volume	x04[7:5] x04[3:1]	CS4281 PPLVC CS4281 PPRVC	--
Mic Mix	x0A[2:1]	Mic Volume	x0E[7:0]
Input Setting	x0C	L Record Select R Record Select	x1A[15:8] x1A[7:0]
Master Volume*	x22[7:5] x22[3:1]	L Master Volume* R Master Volume* L Headphone/Alternate Volume* R Headphone/Alternate Volume* Master Volume Mono*	x02[15:8] x02[7:0] x04[15:8] x04[7:0] x06[7:0]
FM Volume	x26[7:5] x26[3:1]	CS4281 FMLVC CS4281 FMRVC	--
CD Volume	x28[7:5] x28[3:1]	L CD Volume R CD Volume (Master Input Volume)	x12[15:8] x12[7:0]
Line Volume	x2E[7:5] x2E[3:1]	L Line In Volume R Line In Volume (Master Input Volume)	x10[15:8] x10[7:0]

4. * Master Volume mapping is software-controllable through the SSCR register, bits **MVCS**, **MVLD**, **MVAD**, **MVMD**. See text for more information.

Table 23. Sound Blaster Mixer Mapping to AC ‘97 Registers

SB Volume Value	Master Volume		Line/CD Volume		Voice/FM Volume	Mic (Mix) Volume	
	AC ‘97 Value (Hex)	Atten. (dB)	AC‘97 Value (Hex)	Atten. (dB)	Atten. (dB)	AC’97 Value (Hex)	Gain (dB)
0	1F	-94.5/mute*	1F	-34.5/mute*	mute	45	+24.5
1	12	-27.5	1A	-27	-27	49	+18.5
2	0E	-21	16	-21	-21	01	+10.5
3	0B	-16.5	13	-16.5	-16.5	04	+6
4	08	-12	10	-10	-12		
5	05	-7	0D	-7.5	-7.5		
6	02	-3	0A	-3	-3		
7	00	0	08	0	0		

* If left and right channels are both 0, AC ‘97 mute bit is set. If one channel is not 0, the other channel is set to maximum attenuation. Maximum value dependent on AC ‘97 codec used. SB supports mute/channel and AC ‘97 codecs do not.

Table 24. Sound Blaster Volume Translations

17.5 SB Configuration Setup Requirements

- Use Environment BLASTER Variable to determine which DMAn (0-3) to use as Legacy
- Setup legacy DMAn (CCLS, PC/PCI, DDMA, and Polled operation); **IOTCR.DMA[1:0]** = n
  - Set FIFOn size to minimum (2-4 samples); **FCRn.SZ[7:0]**
  - Turn on Legacy DMA engine (will set Mask); **DMRn.DMA**
  - Set SB format: only DMRn format bits on: **CBC, SWAPC, USIGN, SIZE8**
    - **MONO** function is controlled internally by SB engine, based on game-programmed format
    - **DMA/POLL** and **DMRn.TR[1:0]** functions will also be controlled directly by SB engine
  - Set **FCRn.DACZ**
  - Clear **FCRn.PSH** (required for SB silence command 80h)
  - Enable FIFOm; **FCRn.FEN**
  - Initialize Upper Base Count and Upper Address Count to 0
- Interrupt Setup - check environment BLASTER variable
  - Mask unused interrupts (usually all but SBINT); HIMR
  - Optionally switch from PCI to ISA interrupts; HIMR, IIER
- Turn on Sound Blaster and FM; **SSCR.SB** = 1, **SSPM.FMEN** = 1.
- Disable Host Interrupts from legacy DMA; DCRn interrupt enable bits = 0
- Setup and turn on legacy I/O trapping for SB, FM, and Game Port; IOTSB, IOTFM, and IOTGP
- If PC/PCI, set up PC/PCI registers and I/O Trap for PC/PCI; IOTPCP and **PCPCR.PCPEN** set
- Select, setup, and turn on legacy I/O trapping for DMA (plus global trapping); IOTCR
- Setup CSRC and PSRC slot mapping (between AC link and FIF0m); SRCSA
  - NOTE: SRCSA mapping will be loaded into the FIFO slot mapping in FCRm when SB playback or record is enabled.
- Enable SRCs; **SSPM.PSRCEN** and **SSPM.CSRCEN** set.
- Do SB Reset to setup mixer registers
- Set AC Link Output Slot Valid bits in ACOSV
- Make sure Codec is powered up and ADC/DAC running in proper mode (sets ACISV or ACISV2 valid bits).
- Write Codec DAC volume to 0 dB (Index 18h = 0808h). SB uses CS4281 registers for Voice volume

When exiting Sound Blaster mode (tear-down):

- Sound Blaster Voice Volume unmuted, 0 dB; PPLVC/PPRVC = 0
- Turn off Sound Blaster; **SSCR.SB** = 0
- Disable legacy I/O trapping; **IOTCR.ITD** = 1
- Disable legacy DMA

SB Input Select (0Ch)	SB Value (Hex)	AC ‘97 1Ah Register (Hex)
Mic Source (default)	0	0000h
CD Source	1	0101h
Mic Source	2	0000h
Line-In Source	3	0404h

Table 25. Sound Blaster Input Select

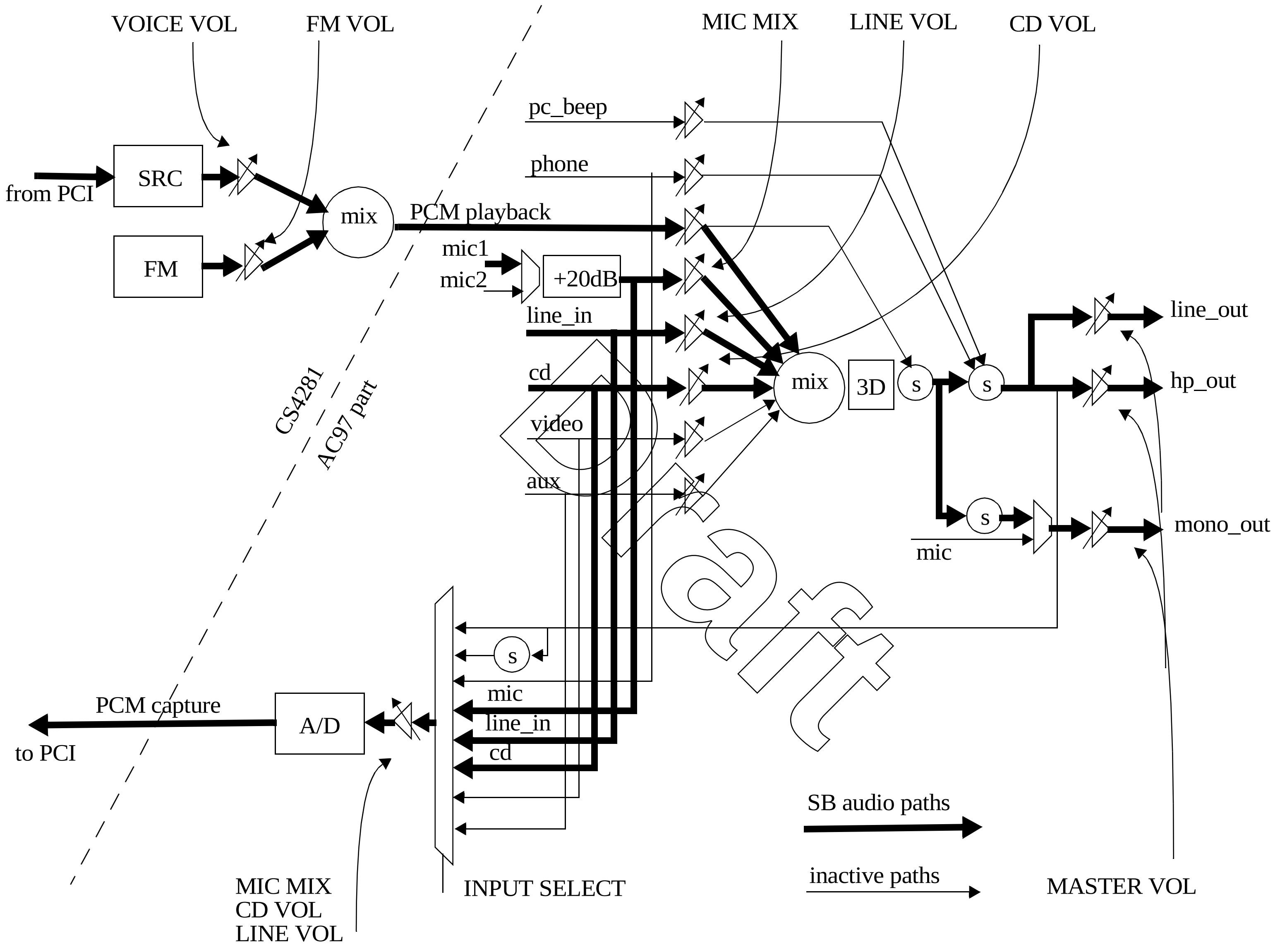


Figure 48. Sound Blaster to AC ‘97 Mixer Mapping

17.6 Sound Blaster DSP

The Sound Blaster DSP consists of the main SB state machine and slave logic to convert SB written values to values appropriate for the CS4281 target registers. These target registers include some of the Mixer registers described in the previous section to facilitate execution of such commands as the Speaker On and Off commands, and also include the registers that control the DMA, FIFO, SRC, and AC-link slot assignment. The DSP also manages the SB Data Available and Busy handshake mechanism between the CS4281 and the host software.

In Sound Blaster, both PIO and DMA consist only of 8-bit data transfers. There is no 16 or 20/24 bit Sound Blaster data path. The Sound Blaster standard does not support full duplex data transfers, so that one DMA/FIFO/AC ‘97 stereo stream is all that is required. Also, it should be noted that *Sound Blaster stereo reverses the normal Sound System byte order* -- for Sound Blaster, first the right, then the left byte of stereo data arrive in a DMA transfer.

17.7 Sound Blaster Hardware Handshake

The Sound Blaster interface uses a hardware handshake mechanism for processing commands. The mixer does not use a handshake mechanism and is always assumed to be available for ISA bus accesses. Two handshake bits are used: Command Busy and Data Available. The Command Busy (otherwise referred to as SB Busy) is located in the Write Buffer Status (SBWBS) Register, bit 7. The Data Available bit is located in the Data Available Status (SBDAS) Register, bit 7. The Command Busy bit indicates when the CS4281 is still busy executing a Sound Blaster command. The Data Available bit is used to indicate when the CS4281 has responded to a SB command with the required data. Reading this register also clears any Sound Blaster generated interrupts.

The handshake works as follows:

Command Busy: a write to the SB Reset Register, SBbase + 6, with bit 0 equal to one or any write to the SB Command Register, SBbase + C/D, will set this bit. It is cleared by the CS4281 when the command or reset has been fully executed from the point of view of the SB state machine. In other words, if a write is pending across the AC-link, but the appropriate Mixer Register has already been updated, the command is considered to have been executed and Command Busy would be released.

Data Available: when the SB state machine loads the Read Data Port with requested data, this bit is set. It is cleared by a read of the Read Data Port, SBbase + A/B, via the PCI bus.

17.8 Sound Blaster Reset

A Sound Blaster Reset Command is generated by writing a one and then a zero to the Sound Blaster Reset Register (SBRR), SBbase+6, successively. This one-to-zero transition is detected in the SB state machine and initiates a reset of the Sound Blaster hardware. This reset should completely reconfigure the SB hardware, including the CS4281 digital data path and AC ‘97 analog mixer so that SB is in its initial state and ready for commands.

SB Volume Control	SB Value (Hex)	SB Register (Hex)	AC ‘97 Registers (Hex)	AC ‘97 Value (Hex)	Volume (dB)
Master Volume	99	22	02, 04, 06	0808	-12
Voice Volume	99	04	CS4281 PPxVC	07, 07	-10.5
FM Volume	99	26	CS4281 FMxVC	07, 07	-10.5
CD Volume	11	28	12	9F1F	muted
Mic Mix	11	0A	0E	0045	+24.5
Input Setting	11	0C	1A	0000	mic selected
Output Setting	11		N/A		mono
Line Volume	11	2E	10	9F1F	muted

Table 26. Sound Blaster Reset Volume Values

17.9 Sound Blaster Interrupts

Sound Blaster commands can cause the hardware in the CS4281 to generate Sound Blaster interrupts (HISR.SBINT) for some SB commands. Sound Blaster interrupts are cleared by the SB state machine upon reading the *Data Available Status* (SBDAS) register (SBbase+E). Commands like Silence, Generate Interrupt, and all the ADPCM commands cause the SB interrupt which comes directly from the SB state (DSP) machine. These interrupts appear on the bus (either PCI INTA or ISA IRQ[A:C]).

TC	SB Mono Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
0-89	≤ 5988	6024	lots	255
90	6024	6024	-0.01	255
91	6060	6071	0.18	253
92	6097	6095	-0.03	252
93	6134	6144	0.16	250
94	6172	6194	0.35	248
95	6211	6244	0.53	246
96	6250	6269	0.31	245
97	6289	6321	0.51	243
98	6329	6347	0.29	242
99	6369	6400	0.49	240
100	6410	6427	0.26	239
101	6451	6481	0.47	237
102	6493	6508	0.24	236
103	6535	6564	0.45	234
104	6578	6592	0.22	233
105	6622	6649	0.41	231
106	6666	6678	0.18	230
107	6711	6737	0.39	228
108	6756	6767	0.16	227
109	6802	6827	0.36	225
110	6849	6857	0.12	224
111	6896	6919	0.33	222
112	6944	6950	0.09	221
113	6993	7014	0.30	219
114	7042	7046	0.05	218
115	7092	7111	0.27	216
116	7142	7144	0.03	215
117	7194	7211	0.24	213
118	7246	7245	-0.01	212
119	7299	7314	0.21	210
120	7352	7349	-0.04	209
121	7407	7420	0.18	207
122	7462	7456	-0.08	206
123	7518	7529	0.15	204
124	7575	7567	-0.11	203
125	7633	7642	0.12	201
126	7692	7719	0.35	199
127	7751	7797	0.59	197
128	7812	7837	0.32	196
129	7874	7918	0.55	194
130	7936	8000	0.81	5
131	8000	8000	0.00	5
132	8064	8000	-0.79	5
133	8130	8170	0.49	188
134	8196	8214	0.22	187
135	8264	8303	0.47	185
136	8333	8348	0.18	184
137	8403	8440	0.44	182
138	8474	8486	0.14	181
139	8547	8581	0.40	179
140	8620	8629	0.11	178
141	8695	8727	0.37	176
142	8771	8777	0.07	175
143	8849	8879	0.33	173
144	8928	8930	0.03	172
145	9009	9035	0.29	170
146	9090	9089	-0.01	169
147	9174	9198	0.26	167
148	9259	9253	-0.06	166
149	9345	9366	0.22	164
150	9433	9423	-0.10	163
151	9523	9540	0.18	161
152	9615	9600	-0.16	160
153	9708	9722	0.14	158
154	9803	9783	-0.20	157
155	9900	9910	0.10	155
156	10000	9974	-0.26	154
157	10101	10105	0.04	152
158	10204	10240	0.35	150
159	10309	10378	0.67	148
160	10416	10449	0.32	147
161	10526	10593	0.64	145
162	10638	10667	0.27	144
163	10752	10817	0.60	142
164	10869	10894	0.23	141
165	10989	11025	0.32	4
166	11111	11025	-0.77	4
167	11235	11294	0.53	136
168	11363	11378	0.13	135
169	11494	11549	0.48	133
170	11627	11636	0.08	132
171	11764	11815	0.44	130
172	11904	11907	0.03	129
173	12048	12094	0.39	127
174	12195	12190	-0.04	126
175	12345	12387	0.34	124
176	12500	12488	-0.10	123

Table 27. Mono SB - TC to Sample Frequency Translation

Table 27. Mono SB - TC to Sample Frequency Translation (Continued)

TC	SB Mono Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
177	12658	12694	0.29	121
178	12820	12800	-0.16	120
179	12987	13017	0.23	118
180	13157	13128	-0.22	117
181	13333	13357	0.18	115
182	13513	13474	-0.29	114
183	13698	13714	0.12	112
184	13888	13838	-0.36	111
185	14084	14092	0.05	109
186	14285	14222	-0.44	108
187	14492	14491	-0.01	106
188	14705	14629	-0.52	105
189	14925	14913	-0.08	103
190	15151	15208	0.38	101
191	15384	15515	0.85	99
192	15625	15673	0.31	98
193	15873	16000	0.80	3
194	16129	16000	-0.80	3
195	16393	16516	0.75	93
196	16666	16696	0.18	92
197	16949	17067	0.69	90
198	17241	17258	0.10	89
199	17543	17655	0.64	87
200	17857	17860	0.02	86
201	18181	18286	0.58	84
202	18518	18506	-0.06	83
203	18867	18963	0.51	81
204	19230	19200	-0.16	80
205	19607	19692	0.44	78
206	20000	19948	-0.26	77

Table 27. Mono SB - TC to Sample Frequency Translation (Continued)

TC	SB Mono Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
207	20408	20480	0.35	75
208	20833	20757	-0.37	74
209	21276	21333	0.27	72
210	21739	22061	1.48	2
211	22222	22061	-0.72	2
212	22727	22588	-0.61	68
213	23255	23273	0.08	66
214	23809	23631	-0.75	65
215	24390	24381	-0.04	63
216	25000	24774	-0.90	62
217	25641	25600	-0.16	60
218	26315	26034	-1.07	59
219	27027	26947	-0.29	57
220	27777	27429	-1.25	56
221	28571	28444	-0.44	54
222	29411	29538	0.43	52
223	30303	30720	1.38	50
224	31250	31347	0.31	49
225	32258	32681	1.31	47
226	33333	33391	0.17	46
227	34482	34909	1.24	44
228	35714	35721	0.02	43
229	37037	37463	1.15	41
230	38461	38400	-0.16	40
231	40000	40421	1.05	38
232	41666	41514	-0.37	37
233	43478	44122	1.48	1
234	45454	44122	-2.93	1
235	47619	48000	0.80	0
236-255	≥ 50000	48000	lots	0

Table 27. Mono SB - TC to Sample Frequency Translation (Continued)

TC	SB Stereo Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
0-172	≤ 5952	6024	lots	255
173	6024	6024	-0.01	255
174	6097	6095	-0.03	252
175	6172	6169	-0.05	249
176	6250	6244	-0.10	246
177	6329	6321	-0.13	243
178	6410	6427	0.26	239
179	6493	6508	0.24	236
180	6578	6592	0.22	233
181	6666	6678	0.18	230
182	6756	6767	0.16	227
183	6849	6857	0.12	224
184	6944	6950	0.09	221
185	7042	7046	0.05	218
186	7142	7144	0.03	215
187	7246	7245	-0.01	212
188	7352	7349	-0.04	209
189	7462	7456	-0.08	206
190	7575	7567	-0.11	203
191	7692	7680	-0.16	200
192	7812	7797	-0.19	197
193	7936	8000	0.81	5
194	8064	8000	-0.79	5
195	8196	8214	0.22	187
196	8333	8348	0.18	184
197	8474	8486	0.14	181
198	8620	8629	0.11	178
199	8771	8777	0.07	175
200	8928	8930	0.03	172
201	9090	9089	-0.01	169
202	9259	9253	-0.06	166
203	9433	9423	-0.10	163
204	9615	9600	-0.16	160
205	9803	9783	-0.20	157
206	10000	9974	-0.26	154
207	10204	10172	-0.31	151
208	10416	10378	-0.36	148

Table 28. Stereo SB - TC to Sample Frequency Translation

TC	SB Stereo Fs	Gershwin Fs	Percent Error	DACSR/ ADCSR
209	10638	10593	-0.42	145
210	10869	11025	1.44	4
211	11111	11025	-0.77	4
212	11363	11378	0.13	135
213	11627	11636	0.08	132
214	11904	11907	0.03	129
215	12195	12190	-0.04	126
216	12500	12488	-0.10	123
217	12820	12800	-0.16	120
218	13157	13128	-0.22	117
219	13513	13474	-0.29	114
220	13888	13838	-0.36	111
221	14285	14222	-0.44	108
222	14705	14629	-0.52	105
223	15151	15059	-0.61	102
224	15625	16000	2.40	3
225	16129	16000	-0.80	3
226	16666	16696	0.18	92
227	17241	17258	0.10	89
228	17857	17860	0.02	86
229	18518	18506	-0.06	83
230	19230	19200	-0.16	80
231	20000	19948	-0.26	77
232	20833	20757	-0.37	74
233	21739	22061	1.48	2
234	22727	22061	-2.93	2
235	23809	23631	-0.75	65
236	25000	24774	-0.90	62
237	26315	26034	-1.07	59
238	27777	27429	-1.25	56
239	29411	28981	-1.46	53
240	31250	30720	-1.70	50
241	33333	32681	-1.96	47
242	35714	35721	0.02	43
243	38461	38400	-0.16	40
244	41666	44122	5.89	1
245	45454	44122	-2.93	1
246-255	≥ 50000	48000	lots	0

Table 28. Stereo SB - TC to Sample Frequency Translation (Continued)

18. SOUND SYSTEM INTERFACE

The Sound System Interface provides all the registers and controls to operate the entire sound system. The Sound System is the hardware that processes data from the sample FIFOs to the AC Link. This includes the AC Link side FIFO controls, sample rate conversion, hardware volume control, and FIFO exception handling functions. The Sound System registers are logically located in this block. In the CS4281, the codec/mixer analog registers are accessed through the AC Link. All Sound System registers are PCI memory mapped.

18.1 Sound System Register Interface

18.1.1 Sound System Power Management (SSPM)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MIXEN	CSRCEN	PSRCEN	JSEN	ACLEN	FMEN	

Address: BA0: 740h

Default: 00000000h

Definition: Enables for FM, Joystick, AC Link, and SRCs. When disabled, each device consumes minimal power.

Bit Descriptions:

MIXEN When set, the digital mixer is enabled and data coming out of the Playback SRC is mixed with FM or Wavetable data before going to the AC Link. When clear, the Digital Mixer and PSRC are bypassed.

CSRCEN When set, the capture sample rate converter is enabled. When clear, the CSRC is powered down; however, the output latch is still used if the SRC is assigned a slot number.

PSRCEN When set, the playback sample rate converter is enabled. When clear, the PSRC is powered down; however, the output latch is still used if the SRC is assigned a slot number. **MIXEN** must be set when **PSRCEN** is set.

JSEN When set, joystick logic is enabled. When clear, joystick logic powered down and coordinate input circuits are disabled, input clamp transistors are disabled, and the joystick registers are read-only with a value of 0. Therefore, if a joystick is attached, the pin will float up to Vcc reducing power. If no joystick is attached, the coordinate pins will float but the input circuit is disabled to stop excess current from flowing into the pad.

ACLEN When set, serial port engine and AC-Link are enabled. When clear, both are powered down

FMEN When set, FM synthesis block is enabled and mixed with the PCM output stream connected to the playback SRC. When clear, FM synthesis block is powered down and the digital stream is available for Host wavetable.

18.1.2 DAC Sample Rate (DACSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SRDA7	SRDA6	SRDA5	SRDA4	SRDA3	SRDA2	SRDA1	SRDA0

Address: BA0: 744h

Default: 00000000h

Definition: Selects the sample rate for the Playback SRC going to the AC Link and Codec DAC.

Bit Descriptions:

SRDA[7:0] DAC sample rate select code. See the *Sample Rate* section for the supported sample rates.

18.1.3 ADC Sample Rate (ADCSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SRAD7	SRAD6	SRAD5	SRAD4	SRAD3	SRAD2	SRAD1	SRAD0

Address: BA0: 748h

Default: 00000000h

Definition: Selects the sample rate for the Capture SRC which contains audio data from the AC Link, which is from an ADC.

Bit Descriptions:

SRAD[7:0] ADC sample rate select code. See the *Sample Rate* section for the supported sample rates.

18.1.4 *Sound System Control Register (SSCR)*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								HVS1				MVCS	MVLD	MVAD	MVMD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							XLPSRC	LPSRC		CDTX		HVC			

Address: BA0: 74Ch

Default: 00000000h

Definition: Core powered. This register is used to manage the Sound System operating configuration.

Bit Descriptions:

HVS1 Hardware Volume Step by 1. When set, each volume step (up or down) is 1 (1.5 dB). When clear, each volume step is 2 (3.0 dB).

MVCS Master Volume Codec Select. Chooses the codec that hardware volume is directed to.
0 - Primary AC Link codec (Slot 1/2 tag bits valid)
1 - Secondary AC Link codec (Slot 1/2 tag bits invalid, slot 0 Codec select bits set to **SERMC.TCID[1:0]**)

MVLD Master Volume Line Out Disable. When set, disables Line Out Master Volume (02h) from being updated by hardware volume pins.

MVAD Master Volume Alternate Out Disable. When set, disables the Headphone/Alternate Volume (04h) from being updated by hardware volume pins.

MVMD Master Volume Mono Out Disable. When set, disables the Mono Out Volume (06h) from being updated by hardware volume pins.

CDTX CD Transfer data. When set, and **SSPM.FMEN** clear, audio data received from ASDIN slots 3 and 4 (Slot IDs 10 and 11 resp.) on the AC Link are transferred by the SSC to the Wavetable interface in the digital mixer. It is host software's responsibility to make sure that no FIFOn is programmed to use the wavetable interface (Slot IDs 29/30). This bit enabled CDROM data in a notebook, to be digitized on the primary Codec ADC and played out a docking station (secondary) Codec's DACs without host software support.

HVC Hardware Volume Control Enable:
0 - no hardware volume control (disabled)
1 - hardware volume control pins active and modify AC-Link registers based on **MVCS**, **MVLD**, **MVAD**, and **MVMD** disable bits. **HVC** also enables HW volume interrupt capability.

LPSRC SRC loopback mode. When set, the internal data from a FIFO goes through the Playback SRC and is looped back through the Capture SRC which then goes to a different FIFO. The AC Link is taken out of the path.

XLPSRC External SRC loopback mode. When set, the external data coming from the AC Link goes through the Capture SRC and is looped back to the Playback SRC, which then goes back out the AC Link. The FIFOs are taken out of the path.

18.1.5 FM Left Volume Control (FMLVC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								LFM		LFA5	LFA4	LFA3	LFA2	LFA1	LFA0

Address: BA0: 754h

Default: 00000000h

Definition: Controls the left channel FM mute and volume that is mixed into the PCM stream after the playback SRC.

Bit Descriptions:

LFM Left FM Mute: When 1 mute left FM audio.

LFA[5:0] Left FM volume control. The least significant bit represents 1.5 dB, with 000000 = 0 dB. The total range is 0 to -93 dB, with lowest setting (1Fh) equal to mute.

18.1.6 FM Right Volume Control (FMRVC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RFM		RFA5	RFA4	RFA3	RFA2	RFA1	RFA0

Address: BA0: 758h

Default: 00000000h

Definition: Controls the right channel FM mute and volume that is mixed into the PCM stream after the playback SRC.

Bit Descriptions:

RFM Right FM Mute: When 1 mute right FM audio.

RFA[5:0] Right FM volume control. The least significant bit represents 1.5 dB, with 000000 = 0 dB. The total range is 0 to -93 dB, with lowest setting (1Fh) equal to mute.

18.1.7 PCM Playback Left Volume Control (PPLVC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								LPM		LPA5	LPA4	LPA3	LPA2	LPA1	LPA0

Address: BA0: 760h

Default: 00000000h

Definition: Controls the volume for left channel PCM playback and is primarily used to control Voice (PCM) and FM volumes independently. After the playback SRC, the PCM stream is volume modified, mixed with the volume-modified FM stream and output to the AC link slot assigned.

Bit Descriptions:

LPM Left PCM Mute: When 1 mute left PCM audio.

LPA[5:0] Left PCM volume control. The least significant bit represents 1.5 dB, with 000000 = 0 dB. The total range is 0 to -93 dB, with lowest setting (1Fh) equal to mute.

18.1.8 PCM Playback Right Volume Control (PPRVC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RPM		RPA5	RPA4	RPA3	RPA2	RPA1	RPA0

Address: BA0: 764h

Default: 00000000h

Definition: Controls the volume for right PCM playback and is primarily used to control Voice (PCM) and FM volumes independently. After the playback SRC, the PCM stream is volume modified, mixed with the volume-modified FM stream and output to the AC link slot assigned.

Bit Descriptions:

RPM Right PCM Mute: When 1 mute right PCM audio.

RPA[5:0] Right PCM volume control. The least significant bit represents 1.5 dB, with 000000 = 0 dB. The total range is 0 to -93 dB, with lowest setting (1Fh) equal to mute.

18.1.9 SRC Slot Assignment (SRCSA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			CRSS4	CRSS3	CRSS2	CRSS1	CRSS0				CLSS4	CLSS3	CLSS2	CLSS1	CLSS0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			PRSS4	PRSS3	PRSS2	PRSS1	PRSS0				PLSS4	PLSS3	PLSS2	PLSS1	PLSS0

Address: BA0: 75Ch

Default: 1F1F1F1Fh

Definition: Controls the slot assignments for the playback and capture SRCs. For playback SRC architecture, see Figure 48. Although SRC channels can be programmed to any slot, the SRC Left/Right channels cannot be split across different FIFOs. In addition, the left and right channels must match the left/right pair of the FIFO attached (cannot swap channels).

Bit Descriptions:

PLSS[4:0] Playback Left SRC Slot assignment. When a FIFO is programmed for this slot, the data is sent to the playback left SRC, then volume control through **PPVC.LPA[5:0]**, then mixed with left FM stream (slot 29) and output to the AC Link slot indicated by **PLSS[4:0]**. If no FIFO channel is programmed for this slot, 0 is sent to the AC Link slot. See Table 29.

PRSS[4:0] Playback Right SRC Slot assignment. When a FIFO is programmed for this slot, the data is sent to the playback right SRC, then volume control through **PPVC.RPA[5:0]**, then mixed with right FM stream (slot 30) and output to the AC Link slot indicated by **PRSS[4:0]**. If no FIFO channel is programmed for this slot, 0 is sent to the AC Link slot. See Table 29.

CLSS[4:0] Capture Left SRC Slot assignment. When this AC Link slot has valid data, the data is sent to the left capture SRC first. Once through the SRC, the data is sent to a FIFO channel programmed for the same slot, **CLSS[4:0]**. If no FIFO channel is programmed for this slot, the data is discarded. See Table 30.

CRSS[4:0] Capture Right SRC Slot assignment. When this AC Link slot has valid data, the data is sent to the right capture SRC first. Once through the SRC, the data is sent to a FIFO channel programmed for the same slot, **CRSS[4:0]**. If no FIFO channel is programmed for this slot, the data is discarded. See Table 30.

PLSS[4:0]/ PRSS[4:0]*	AC '97 Output Slot	AC '97 Slot Function (primary & secondary codec)
0	3	Left PCM Playback
1	4	Right PCM Playback
2	5	Phone Line 1 DAC
3	6	Center PCM Playback
4	7	Left Surround PCM Playback
5	8	Right Surround PCM Playback
6	9	LFE PCM Playback
7	10	Phone Line 2 DAC
8	11	HeadSet DAC
31	Not used	SRC channel is not used - power down

25. * Left and Right channels cannot be split across different FIFOs. Only one FIFOn can be attached to PSRC.

Table 29. SRC Playback Slot Assignments

CLSS[4:0]/ CRSS[4:0]*	AC ‘97 Input Slot	AC ‘97 Codec	AC ‘97 Slot Function
10	3	Primary	Left PCM Record
11	4	Primary	Right PCM Record
12	5	Primary	Phone Line 1 ADC
13	6	Primary	Mic ADC
14	7	Primary	reserved
15	8	Primary	reserved
16	9	Primary	reserved
17	10	Primary	Phone Line 2 ADC
18	11	Primary	HeadSet ADC
20	3	Secondary	Left PCM Record
21	4	Secondary	Right PCM Record
22	5	Secondary	Phone Line 1 ADC
23	6	Secondary	Mic ADC
24	7	Secondary	reserved
25	8	Secondary	reserved
26	9	Secondary	reserved
27	10	Secondary	Phone Line 2 ADC
28	11	Secondary	HeadSet ADC
31	Not used		SRC channel is not used - power down

26. * Left and Right channels cannot be split across different FIFOs. Only one FIFOn can be attached to CSRC.

Table 30. Record SRC Slot Assignments

18.2 Sound System Functional Description

The Sound System Controller (SSC) moves data from the AC Link to and from the FIFO’s and SRCs. Figure 47 illustrates the flow of data between the AC-Link engine and the FIFO memory. The Sound System Controller gets requests from the AC-Link engine when it needs data. For playback, the AC-Link engine requests data (**ac97_dacmux[3:0]**) one slot prior to sending the slot on the AC Link. The AC-Link engine will only ask for **ASDOUT** slots that have their Output Slot Valid bits set (ACOSV). For record, the AC-Link engine will request data be transferred (**ac97_adcen[3:0]**) for incoming slots that have their Input Slot Valid bits set (ACISV for **ASDIN** and ACISV2 for **ASDIN2**). When the AC-Link engine has data, the SSC moves the data (one channel at a time) into the FIFO. For FIFOs that have one channel inactive (slot ID 31), the SSC sends both valid channel signals to the FIFO to make the write pointer increment. The same is true when reading from a FIFO with one channel inactive.

If the AC-Link engine gives the SSC AC-Link slots that don’t have an attached FIFO, the data is discarded. A typical example is when the CS4281 is attached to a CS4297. Unless the ADCs are powered down, the slot 3 and slot 4 data will always be valid. The Serial port engine will pass the data to the SSC. Most of the time host software won’t be recording. Therefore, slot 3 and 4 data will always be valid, but no FIFO will be programmed for **ASDIN** slots 3 and 4 (Slot IDs 10, 11). The SSC takes the data from the AC link and discard it.

If the AC Link has output slot valid bits set (ACOSV), and no FIFO is attached to those slots, the SSC sends 0’s when the AC-Link engine requests data for those slots.

When the Sample Rate Converters are enabled, the sample frequency clocks provide the “requests for data movement” to the SSC. The PSRC, enabled by setting **SSPM.PSRCEN**, uses the DACSR register for the playback sample frequency. The CSRC, enabled by setting **SSPM.CSRCEN**, uses the ADCSR register for the capture sample frequency.

Host-wavetable mixing is enabled by having FM disabled (**SSPM.FMEN** clear) and setting a FIFO's slot IDs in the FCRn register to 29 and 30. Given these criteria are met, the SSC will use the 48 kHz frame clock (**48k_clk**) to transfer data from the FIFOn to the mixer in the PSRC, where the host-wavetable data is mixed PSRC before being transferred to the AC-Link engine.

18.2.1 FM/Host Wavetable Stream Architecture

The FM Synthesis block generates a stereo 48 kHz audio stream. This stream (after volume control through FMVC) is digitally mixed with PCM data (which also has volume control in PPVC) in the SRC. Then the composite PCM data is sent out the assigned AC-Link slot, typically 3 and 4. In many systems FM is not used. In these cases (where **SSPM.FMEN** is clear), the FM audio stream is avail-

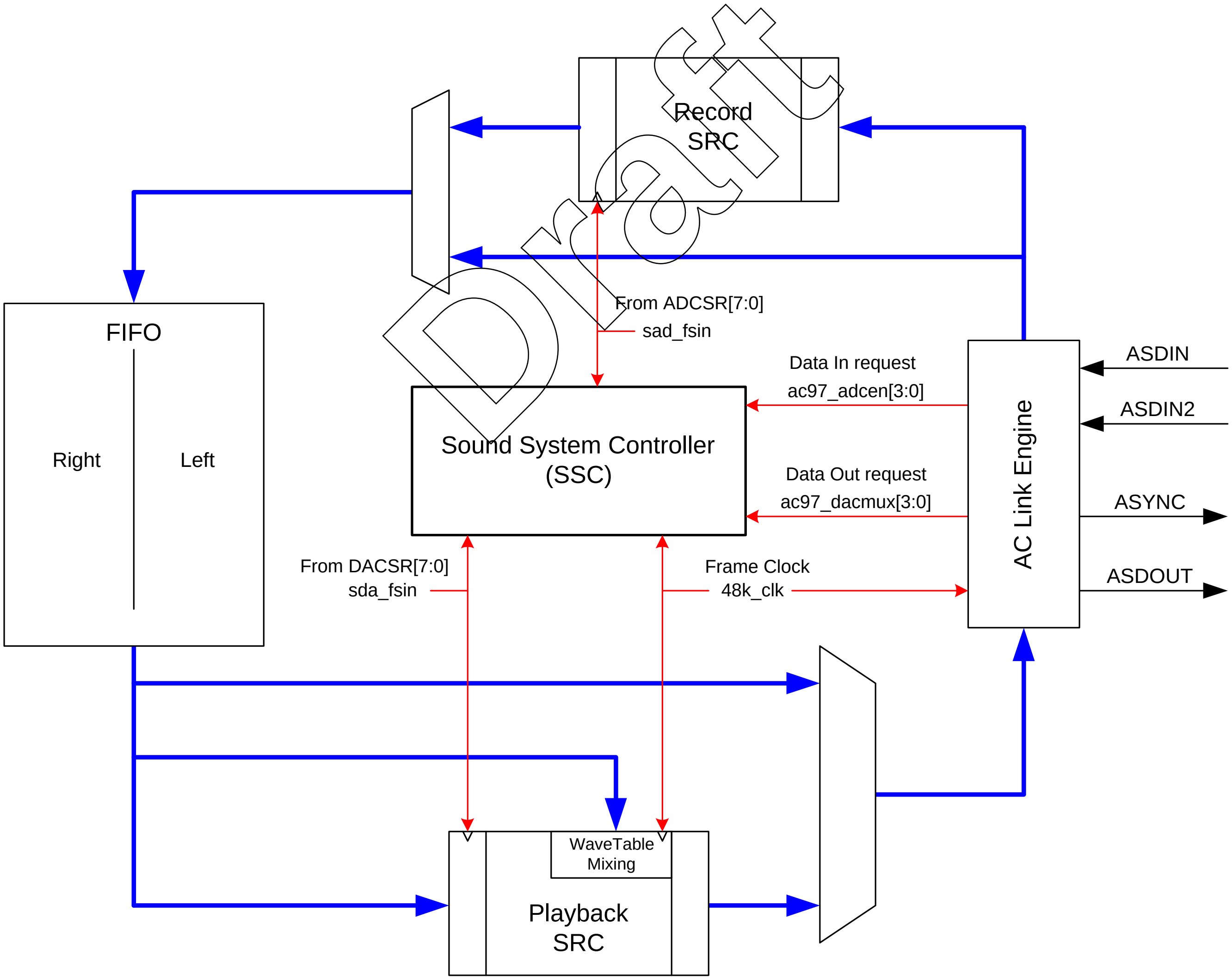


Figure 47. Sound System Controller Data Flow

able to the host with the CS4281 providing digital mixing (acceleration) before being sent to the AC Link. The conceptual architecture for the Host path and playback SRC is illustrated in Figure 48. Host software can set one of the 4 FIFO's for playback and connect it to slot IDs 29 and 30. These "special" slot IDs send the data through the Playback SRC which adds FM volume control, mixes FM with PCM data, and sends the data to the assigned AC-Link slots, typically slots 2 and 3. The FM mixing and volume control function is done after the playback SRC. When the playback SRC is powered down, data from the FIFOs is transferred from the input of the SRC directly to the output to support the mixing function. If the Playback SRC is connected to slots other than Slots 3 and 4 (PCM playback), the FM mixing will follow the new slot assignments. FM audio data is output at a fixed 48 kHz rate and is mixed with the PSRC data after sample rate conversion.

18.3 Sample Rate Converters

Their are two stereo sample rate converters, one for playback and one for capture. Each is enabled by setting the **SSPM.PSRCEN** bit and **SSPM.CSRCEN**, respectively. If the enable bit is cleared, the PSRC/CSRC is powered down and flushed. Once enabled they are attached to a particular AC Link slot. The capture SRC (CSRC) is assigned to a particular AC Link slot using **SRCSA.CLSS[4:0]** for the left channel and **SRCSA.CRSS[4:0]** for the right. If the CSRC is given valid data from the AC Link for the slots programmed for the CSRC, the data is sent through the CSRC and then given to the FIFOn channel programmed to the same slot (through **FCRn.L/RS[4:0]**).

The playback SRC (PSRC) is assigned to a particular AC-Link slot using **SRCSA.PLSS[3:0]** for the left channel and **SRCSA.PRSS[3:0]** for the right. If the PSRC finds a FIFO attached to the same slot the

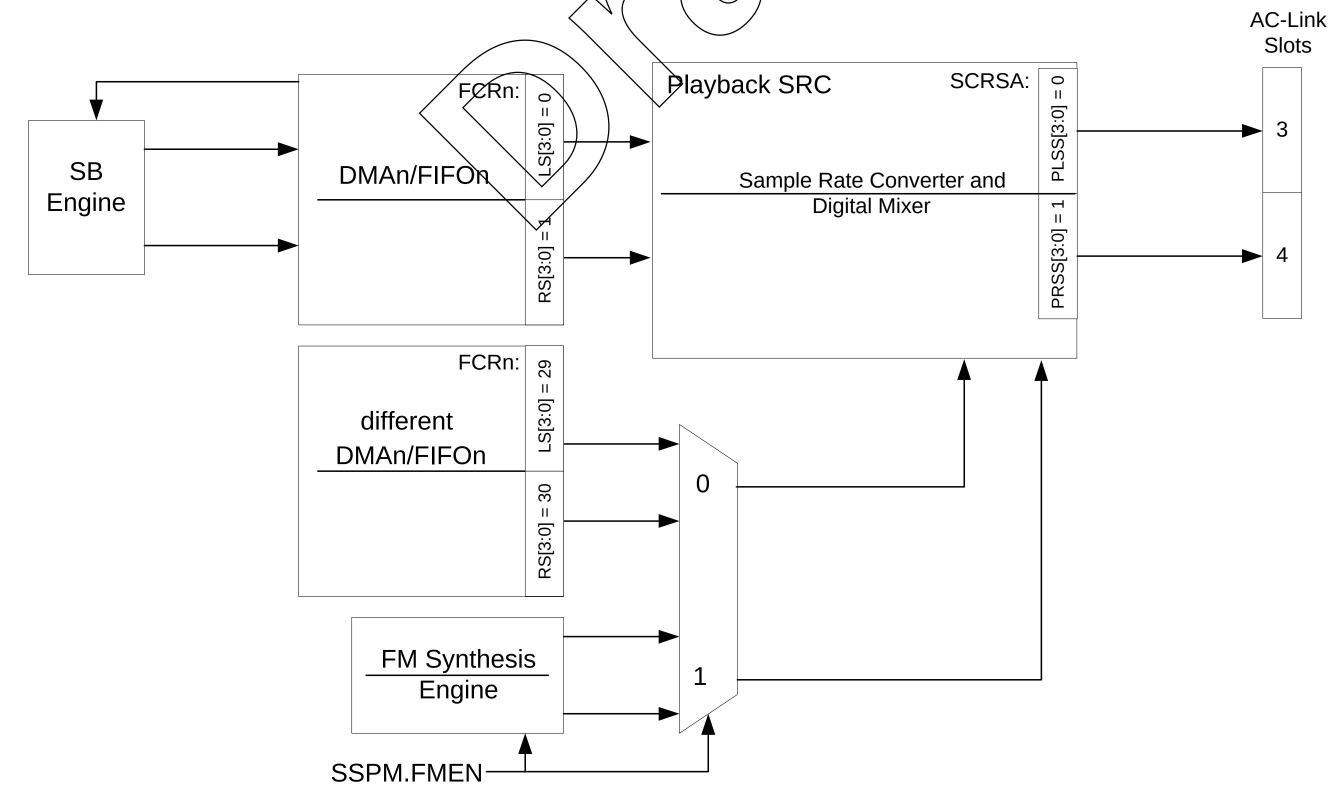


Figure 48. FM/Host-Wavetable and Playback SRC

PSRC is assigned to, the FIFO data is first sent through the PSRC and then sent to the AC Slot appropriate AC slot. The Playback SRC also adds volume control to the playback data controlled through the PPVC register. The PSRC also takes in data from the FM/Host Wavetable channels (slots 30/31) add volume control (FMVC register) and mixes that data with the playback data before sending the data to the AC Link as illustrated in Figure 48.

18.3.1 Sample Rates

The CS4281 Sample Rate Converters (SRCs) support independent sample rates for the record and playback paths. The sample frequencies are set in the DACSR for playback and ADCSR register for recording in an encoded fashion. The value needed for a particular sample frequency is listed in Table 31 for standard rates and Table 32 for special rates. For the standard rates, the register value needed is:

$$\text{ADCSR or DACSR} = \frac{2 \times \mathbf{ABITCLK}}{\text{Fs} \times 16}$$

or for the standard **ABITCLK** frequency of 12.288 MHz:

$$\text{ADCSR or DACSR} = \frac{1,536,000}{\text{Fs}}$$

DACSR/ ADCSR	Internal Divider	Samples Frequency	DACSR/ ADCSR	Internal Divider	Samples Frequency	DACSR/ ADCSR	Internal Divider	Samples Frequency
255	4080	6023.5	180	2880	8533.3	105	1680	14628.6
254	4064	6047.2	179	2864	8581.0	104	1664	14769.2
253	4048	6071.1	178	2848	8629.2	103	1648	14912.6
252	4032	6095.2	177	2832	8678.0	102	1632	15058.8
251	4016	6119.5	176	2816	8727.3	101	1616	15207.9
250	4000	6144.0	175	2800	8777.1	100	1600	15360.0
249	3984	6168.7	174	2784	8827.6	99	1584	15515.2
248	3968	6193.5	173	2768	8878.6	98	1568	15673.5
247	3952	6218.6	172	2752	8930.2	97	1552	15835.1
246	3936	6243.9	171	2736	8982.5	96	1536	16000.0
245	3920	6269.4	170	2720	9035.3	95	1520	16168.4
244	3904	6295.1	169	2704	9088.8	94	1504	16340.4
243	3888	6321.0	168	2688	9142.9	93	1488	16516.1
242	3872	6347.1	167	2672	9197.6	92	1472	16695.7
241	3856	6373.4	166	2656	9253.0	91	1456	16879.1
240	3840	6400.0	165	2640	9309.1	90	1440	17066.7
239	3824	6426.8	164	2624	9365.9	89	1424	17258.4
238	3808	6453.8	163	2608	9423.3	88	1408	17454.5
237	3792	6481.0	162	2592	9481.5	87	1392	17655.2
236	3776	6508.5	161	2576	9540.4	86	1376	17860.5
235	3760	6536.2	160	2560	9600.0	85	1360	18070.6
234	3744	6564.1	159	2544	9660.4	84	1344	18285.7
233	3728	6592.3	158	2528	9721.5	83	1328	18506.0
232	3712	6620.7	157	2512	9783.4	82	1312	18731.7
231	3696	6649.4	156	2496	9846.2	81	1296	18963.0
230	3680	6678.3	155	2480	9909.7	80	1280	19200.0
229	3664	6707.4	154	2464	9974.0	79	1264	19443.0
228	3648	6736.8	153	2448	10039.2	78	1248	19692.3
227	3632	6766.5	152	2432	10105.3	77	1232	19948.1
226	3616	6796.5	151	2416	10172.2	76	1216	20210.5
225	3600	6826.7	150	2400	10240.0	75	1200	20480.0
224	3584	6857.1	149	2384	10308.7	74	1184	20756.8

Table 31. Sound System Sample Frequencies

DACSR/ ADCSR	Internal Divider	Samples Frequency	DACSR/ ADCSR	Internal Divider	Samples Frequency	DACSR/ ADCSR	Internal Divider	Samples Frequency
223	3568	6887.9	148	2368	10378.4	73	1168	21041.1
222	3552	6918.9	147	2352	10449.0	72	1152	21333.3
221	3536	6950.2	146	2336	10520.5	71	1136	21633.8
220	3520	6981.8	145	2320	10593.1	70	1120	21942.9
219	3504	7013.7	144	2304	10666.7	69	1104	22260.9
218	3488	7045.9	143	2288	10741.3	68	1088	22588.2
217	3472	7078.3	142	2272	10816.9	67	1072	22925.4
216	3456	7111.1	141	2256	10893.6	66	1056	23272.7
215	3440	7144.2	140	2240	10971.4	65	1040	23630.8
214	3424	7177.6	139	2224	11050.4	64	1024	24000.0
213	3408	7211.3	138	2208	11130.4	63	1008	24381.0
212	3392	7245.3	137	2192	11211.7	62	992	24774.2
211	3376	7279.6	136	2176	11294.1	61	976	25180.3
210	3360	7314.3	135	2160	11377.8	60	960	25600.0
209	3344	7349.3	134	2144	11462.7	59	944	26033.9
208	3328	7384.6	133	2128	11548.9	58	928	26482.8
207	3312	7420.3	132	2112	11636.4	57	912	26947.4
206	3296	7456.3	131	2096	11725.2	56	896	27428.6
205	3280	7492.7	130	2080	11815.4	55	880	27927.3
204	3264	7529.4	129	2064	11907.0	54	864	28444.4
203	3248	7566.5	128	2048	12000.0	53	848	28981.1
202	3232	7604.0	127	2032	12094.5	52	832	29538.5
201	3216	7641.8	126	2016	12190.5	51	816	30117.6
200	3200	7680.0	125	2000	12288.0	50	800	30720.0
199	3184	7718.6	124	1984	12387.1	49	784	31346.9
198	3168	7757.6	123	1968	12487.8	48	768	32000.0
197	3152	7797.0	122	1952	12590.2	47	752	32680.9
196	3136	7836.7	121	1936	12694.2	46	736	33391.3
195	3120	7876.9	120	1920	12800.0	45	720	34133.3
194	3104	7917.5	119	1904	12907.6	44	704	34909.1
193	3088	7958.5	118	1888	13016.9	43	688	35720.9
192	3072	8000.0	117	1872	13128.2	42	672	36571.4
191	3056	8041.9	116	1856	13241.4	41	656	37463.4
190	3040	8084.2	115	1840	13356.5	40	640	38400.0
189	3024	8127.0	114	1824	13473.7	39	624	39384.6
188	3008	8170.2	113	1808	13592.9	38	608	40421.1
187	2992	8213.9	112	1792	13714.3	37	592	41513.5
186	2976	8258.1	111	1776	13837.8	36	576	42666.7
185	2960	8302.7	110	1760	13963.6	35	560	43885.7
184	2944	8347.8	109	1744	14091.7	34	544	45176.5
183	2928	8393.4	108	1728	14222.2	33	528	46545.5
182	2912	8439.6	107	1712	14355.1	32	512	48000.0
181	2896	8486.2	106	1696	14490.6	31-6	512	48000.0

Table 31. Sound System Sample Frequencies (Continued)

19. FM SYNTHESIS

The FM synthesis engine, enabled by **SSPM.FMEN** = 1, will generate a 48 kHz sample rate stereo data stream. This stream will be volume adjusted (FMLVC/FMRVC) and digitally mixed with the sample rate converted WAV audio data. If FM is disabled, the digital mixing stream can be used by the host (wavetable) by setting the FIFO-to-AC Link Slot to special slot IDs (29 and 30) in the FCRn register.

The CS4281 internal FM block provides full compatibility with market standard FM-based music synthesis as used in DOS games and educational software. FM mixing will occur in the CS4281 with the data stream going to the DACs in the AC '97 Codec. The digital mixer, after the playback SRC, will combine PCM and FM data, after applying volume controls to each, and then send the data to the programmed AC-Link slots, typically 3 and 4. This configuration is designed to support Sound Blaster FM but may be used in host mode, no restrictions. See the *FM/Host Wavetable Stream Architecture* section for more information on the FM mixing architecture.

DACSR/ ADCSR	Internal Divider	Desired Fs	Actual Fs	Percent Error
5	3072	8000	8000.0	0.000
4	2229	11025	11025.6	0.005
3	1536	16000	16000.0	0.000
2	1114	22050	22061.0	0.050
1	557	44100	44122.1	0.050
0	512	48000	48000.0	0.000

Table 32. Sound System Special Sample Frequencies

19.1 FM Direct Register Description

19.1.1 FM Status Register (FMSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								IRQ	FT1	FT2					BUSY

Address: BA0: 730h, Read-Only

Default: 00000000h

Definition: Contains the FM status for interrupts and counters.

Bit Descriptions:

IRQ Interrupt Request Flag. **IRQ** is set to one when either **FT1** or **FT2** is set to one. **IRQ** is reset to zero when the **RST** bit in array 0, address 0x4, is set to one.

FT1 Timer Flag 1. **FT1** is set to one when timer 1 has reached its terminal count. **FT1** is reset to zero when the **RST** bit in array 0, address 0x4, is set to one.

FT2 Timer Flag 2. **FT2** is set to one when timer 2 has reached its terminal count. **FT1** is reset to zero when the **RST** bit in array 0, address 0x4, is set to one.

BUSY The FM Synthesis core requires a wait time between when an address is written to the address register and when data is written to the data register. The **BUSY** bit when = 1 indicates when the FM interface is “not ready” to receive data. When the **BUSY** bit is a zero then the FM interface is ready to receive data. **BUSY** is enabled only when **NEW3** (Bank 1, Index 05h) is a 1.

19.1.2 Bank 0 Address Port (B0AP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FMAD7	FMAD6	FMAD5	FMAD4	FMAD3	FMAD2	FMAD1	FMAD0

Address: BA0: 730h, Write-Only

Default: 00000000h

Definition: Indirect address port for selecting FM registers in array 0, accessed through the FM Data Port register, FMDP.

Bit Descriptions:

FMAD[7:0]FM Register indirect address.

19.1.3 FM Data Port (FMDP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FMDP7	FMDP6	FMDP5	FMDP4	FMDP3	FMDP2	FMDP1	FMDP0

Address: BA0: 734h, Read-Write

Default: 00000000h

Definition: Data port for accessing the FM register last addressed by an address write operation, either B0AP or B1AP.

Bit Descriptions:

FMDP[7:0] FM Register data.

19.1.4 Bank 1 Address Port (B1AP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FMAD7	FMAD6	FMAD5	FMAD4	FMAD3	FMAD2	FMAD1	FMAD0

Address: BA0: 738h, Read-Write

Default: 00000000h

Definition: Indirect address port for selecting FM registers in array/bank 1, accessed through the FM Data Port, FMDP, or Bank 1 Data Port, B1DP, registers.

Bit Descriptions:

FMAD[7:0]FM Register Indirect Address.

19.1.5 Bank 1 Data Port (B1DP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FMDP7	FMDP6	FMDP5	FMDP4	FMDP3	FMDP2	FMDP1	FMDP0

Address: BA0: 73Ch, Read-Write

Default: 00000000h

Definition: Data port for accessing the FM register last addressed by an address write operation to register array/bank 1 using the B1AP register.

Bit Descriptions:

FMDP[7:0] FM Register data.

20. PERIPHERAL DEVICES

The Peripheral Devices include the Hardware Volume Control, Clock Generation, General Purpose I/O (GPIO), MIDI UART Port, Game (Joystick) Port, and AC Link.

20.1 Hardware Volume Control

The CS4281 supports Hardware Volume Control using a 2 button method. The **VOLUP** and **VOLDN** pins perform the volume up and volume down functions respectively. When both button are pushed the volume mutes. The CS4281 maintains a local copy of the master volume data to simplify updating transactions. The **VOLUP** and **VOLDN** input pins are digitally debounced with a 50 ms timer. The volume control is incremented or decremented one or two (based on **SSCR.HVS1**) step each time the pin is sensed low and then released. Holding the pin low for 0.5 second activates an auto-repeat function, where the volume control is stepped one/two counts each 0.25 second. Pressing both buttons for 0.1 second activates the mute. When muted, pressing either **VOLUP** or **VOLDN** exits the mute state and returns to the last volume setting before mute was entered, without doing a up or down step. Thus the user can press both buttons to mute, then press either button to unmute, and repeating this process will toggle the volume control between the mute state and the volume control state last established.

Hardware volume is enabled by setting the **HVC** bit in the *Sound System Control Register* (SSCR). Hardware volume can modify one of the two possible codecs on the serial link, selected by the **SSCR.MVCS** bit. Volume control updates are software-configurable to control:

- the AC '97 Master Volume Register at address 02h (**SSCR.MVLD**=0)
- the Alternate Master Volume Register at address 04h (**SSCR.MVAD**=0)
- the Mono Out Volume register at 06h (**SSCR.MVMD**=0)

The volume up/down step size is programmable with the default being 2 counts per step (**SSCR.HVS1** = 0). Two counts on a standard AC '97 Codec translates to 3 dB/step. Setting **SSCR.HVS1** causes the step size to change to 1 count per step. The Hardware Volume range is 0 down to 3Fh (6 bits). When the most significant bit is set (bit 5), the volume jumps to the last setting of 3Fh. For example, if the volume setting is 1Eh and the step size is 2, then pressing the Volume Down button causes 1Eh to decrement to 20h which sets the MSB and forces the volume to 3Fh. If host software programs the master volume to a number less than 3Fh but with the MSB set (20h through 3Eh), then a decrement will jump to 3Fh.

If the Volume Up button is pressed when the MSB is set (range of 20h through 3Fh), the hardware volume will jump to 1Fh if **SSCR.HVS1** is set and 1Eh if **SSCR.HVS1** is clear (basically 20h - step value). If the Volume Up is pressed when the hardware volume is 0, nothing will happen since 0 is the highest value.

If software wants to be informed of hardware volume changes, the extended GPIO logic on the **VOLUP/VOLDN** pins can be configured to generate interrupts on hardware volume changes (hardware volume logic generates a pulse to the GPIO logic which can be configured to generate interrupts). If software wants total control of master volume updating, the above disable bits can all be set causing the hardware volume control logic to generate interrupts only - no actual volume updates. Every volume change generates an internal pulse signal. This pulse signal is generated at the 100 ms point for each debounced volume signal and at the repetition rate for signals held down for longer than 0.5 seconds. The pulse generated for each signal is sent to the GPIOR logic, which can create interrupts which are cleared through GPIOR. The individual interrupt pulses go to the Extended GPIO control for the **VOLUP/VOLDN** pins. See the *General Purpose Input/output Pins* section and Figure 51 for a conceptual view of how the signal interfaces with the extended GPIO logic.

A shadow register on the CS4281 snoops writes to the AC Link registers and stores writes to the Master Volume register on the Codec. Hardware volume changes then read the shadow register, increments/decrements one step and, write the new value to both the shadow register and the Codec (through the AC Link), thereby keeping the hardware volume relative to any software updates.

Host writes to the master volume control registers take precedence over volume control button operation. The Host write will have the same effect as if the user released the button just before the host write, and then pressed the button again just after the write. Host software that does read-modify-writes of the volume registers will maintain any changes from the last hardware volume update. In addition, an interrupt can be generated on a hardware volume change thereby keeping hardware volume changes synchronized with the host software sliders/values.

If the AC Link is down (no **ABITCLK**) the hardware volume values are frozen. **VOLUP**/**VOLDN** button changes are ignored and registers are NOT updated. Hardware volume registers are powered from the core (unlike the AC ‘97 codec volume registers). If the core power is removed, the hardware volume is lost and host software must write the master volume to re-initialize the hardware volume registers when the power and link come back up.

Host software can avoid this loss of synchronization by either doing a read-modify-write on the volume (which will take into account any hardware volume changes) or enable the hardware volume interrupt, through the GPIOR logic, which will let the software know when the hardware volume changes the master volume registers.

The volume control pins have an internal 20 kΩ pullup to the positive supply. It is expected that the pins would use SPST momentary switches between the pin and ground to activate the function. No external debounce capacitors are required, although EMI filter capacitors may be needed.

20.2 Clock Control

The CS4281 clocking model includes a delay locked loop for high frequency core clock generation, and power control via selective clock gating.

20.2.1 Clock Control Register 1 (CLKCR1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						CLKON	DLLRDY							CKRN	CKRA

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									DLLOS	SWCE	DLLP	DLLSS1	DLLSS0	res	res

Address: BA0: 400h, Read-Write

Default: 00000000h

Definition: Core powered. Clock Control Register 1 provides a host port for general device configuration, control, and clock status. For normal operation, the lower word would contain 0030h. This register only affects the part of the chip controlled by **ABITCLK** and has no affect on **PCICLK**-based sections. For normal operation **DLLP** and **SWCE** should be set.

Bit Descriptions:

res Reserved

DLLSS[1:0] DLL Source Select: This field selects the clock source for the pre-divider and the DLL.
0 0 = Serial Port Bit Clock: For AC ‘97 configurations (reset default).
0 1 = Reserved.
1 0 = Reserved.
1 1 = Reserved.

DLLP DLL Power Up: This bit is the master enable/disable for the DLL. Note that “powering up” the DLL requires a certain stabilization delay before operating the device.
0 = Stop DLL (reset default).
1 = DLL running.

SWCE Software Clock Enable: This bit is the master enable/disable for the core clocks. Clearing this bit without clearing **DLLP** allows a reduced power state without the start-up latency of stopping the DLL.
0 = Device clocks stopped (reset default).
1 = Device clocks enabled.

DLLOS DLL Output Select: This bit selects the clock fed to the device core.
0 = Normal DLL output (reset default).
1 = Reserved.

CKRA CLKRUN# Always. When set, forces **CLKRUN#** to be asserted (with the normal **CLKRUN#** protocol), whenever the Central Resource indicates that it wants to stop **CLKRUN#**. **CKRA** is a debug/backup plan that puts **CLKRUN#** assertion under host software control. This bit is similar to any internal section that indicates to the CLKRUN engine that it needs the PCI clock to keep running to finish its tasks.

CKRN CLKRUN Never. When set, blocks **CLKRUN#** output from ever being asserted. Mainly provided as a debug/backup bit. This bit overrides **CKRA**.

DLLRDY Read-Only bit indicating status of the DLL. When set, indicates that the DLL is ready and locked to the clock source (generally **ABITCLK**). This bit also resets logic in the serial port engine and allows PME support from the **ASDIN/ASDIN2** lines. See Figure 18 in *PCM# Assertion* section for **PME#** conceptual logic. **DLLRDY** locks (goes high) to the clock source in approx. 30 clocks. **DLLRDY** goes low when the **CLKON** signal is detected low (quite a long time).

CLKON Read-only bit indicating the current status of the loss-of-clock detector. Provided for debug/test purposes.

20.3 General Purpose Input/Output Pins

Some CS4281 device pins are available for general purpose input or output use. Some of these pins are dual function and will not be available in certain system configurations. When available, they can be used for system-defined functionality driven by host software.

The following table shows the pins available for GPIO use and the system configurations in which they are available.

The definition and control for the **EECLK/GPOUT** and **EEDAT/GPIO2** pins can be found in the *EEPROM Configuration Interface* section. The other GPIO pins contain extended capabilities and are described below. If the Extended GPIO pins are setup to generate a modem wake-up event (**PME#** assertion), the pins must be connected to a *Ring Discriminator* to be compliant with Microsoft. Otherwise, false triggering will occur.

20.3.1 Enabling Extended GPIO pins

The **VOLUP** and **VOLDN** extended GPIO logic is always enabled since these pins are always inputs. However, the input to the GPIO logic can be controlled by the hardware volume control, enabled by setting the **HVC** bit in the *Sound System Control Register*, BA0:74C. With **HVC** = 1, the GPIO logic can be controlled directly from the pin as a general purpose input, or can also be controlled through the hardware volume control logic giving software hardware-volume status (through interrupts) every time hardware volume is changed. The **VOLUP/VOLDN** pins are powered from the **VAUX** supply, thereby supporting the wake-up capability (**PME#** assertion) from the D3$_{\text{cold}}$ power-off state. Conceptual logic for the **VOLUP/VOLDN** pins is illustrated in Figure 51.

GPIO1 is set to a general purpose I/O by clearing the **ASDIN2** enable bit (**ASDI2E**) in the *Serial Port Power Management Control* (SPMC) register. The **ASDIN2/GPIO1** pin is powered from the **VAUX** supply, thereby supporting the wake-up capability from the D3$_{\text{cold}}$ power-off state. Conceptual logic for the **ASDIN2/GPIO1** pin is illustrated in Figure 50.

The **GPIO3** pin has no alternate function so it is always available as a GPIO. The **GPIO3** pin is powered from the PCI power supply; therefore, wake-up events are supported in every power state except D3$_{\text{cold}}$. Conceptual logic for the **GPIO3** pin is illustrated in Figure 49. In designs/sockets that are backward compatible with the CS4614 and CS4280, the **GPIO3** pin must remain high-impedance since this pin was formally a PCI power supply pin.

Pin Name	Type	Usage	Access
ASDIN2/GPIO1	I/O	Limited availability with dual codec. Powered from **VAUX**.	GPIOR Register, 3E8h and SPMC, 3ECh
GPIO3	I/O	Powered from PCI bus	GPIOR Register, 3E8h
VOLUP	Input	Can be used while HW volume active	GPIOR Register, 3E8h
VOLDN	Input	Can be used while HW volume active	GPIOR Register, 3E8h
EECLK/GPOUT	Output	Available if no EE required, may be used if EE present and protocol observed.	CFGI Register, 4B0h
EEDAT/GPIO2	I/O	Available if no EE required, may be used if EE present and protocol observed. Powered from **VAUX**.	CFGI Register, 4B0h

Table 33. GPIO Pin Usage and Register Control

20.3.2 General Purpose I/O Register (GPIOR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GP3W	GP3ST	GP3PT	GP3OE	GP1W	GP1ST	GP1PT	GP1OE	VUPW	VUPST	VUPPO	VUPLT	VDNW	VDNST	VDNPO	VDNLT
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				GP3D	GP1D					GPPS	GPSS	GP3S	GP1S	VUPS	VDNS

Address: Host BA0: 3E8h, Read-Write
PCI CFG: 0E8h, Read-Write if CWPR configured, otherwise Read-Only

Default: 00000000h

Definition: Vaux powered. The General Purpose I/O register provides a host port for accessing extended general-purpose I/O pins. This register is unaffected by the PCI **RST#** signal. The default value is set by a Vaux POR signal. In addition to functionality in this register, these pins can generate a host interrupt through the *Host Interrupt Mask* (HIMR) and *Host Interrupt Status* (HISR) Registers. See the *Interrupt Subsystem* section for more details, and the figures below. They can also generate PME# events. See Figure 18 in *PCM# Assertion* section for PME# conceptual logic.

Bit Descriptions:

VDNS **VOLDN** input Status: This bit reflects the status of the **VOLDN** input pin. If configured as sticky (**VDNST**=1), **VDNS** reads one when the **VOLDN** pin goes active (edge sensitive - edge defined by polarity bit **VDNPO**), and is cleared by writing a 0 to **VDNS**.
If configured as level sensitive (**VDNST**=0), this bit reflects the current state of the **VOLDN** pin qualified by the polarity bit **VDNPO**.

VUPS **VOLUP** input Status: This bit reflects the status of the **VOLUP** input pin. If configured as sticky (**VUPST**=1), **VUPS** reads one when the **VOLUP** pin goes active (edge sensitive - edge defined by polarity bit **VUPPO**), and is cleared by writing a 0 to **VUPS**.
If configured as level sensitive (**VUPST**=0), this bit reflects the current state of the **VOLUP** pin qualified by the polarity bit **VUPPO**.

GP1S **ASDIN2/GPIO1** input Status: Assuming this pin is not configured for **ASDIN2**, this bit reflects the status of the **ASDIN2/GPIO1** pin. If **ASDIN2/GPIO1** is an output, this bit reflects the actual state of the pin. If **ASDIN2/GPIO1** is an input:
If configured as sticky (**GP1ST**=1), this bit reads one when the **ASDIN2/GPIO1** pin goes active (edge sensitive - edge defined by polarity bit **GP1PT**), and is cleared by writing a 0 to **GP1S**.
If configured as level sensitive (**GP1ST**=0), this bit reflects the current state of the **ASDIN2/GPIO1** pin qualified by the polarity bit **GP1PT**.
See the *Serial Port Power Management Control* (SPMC) register description of ASDI2E bit.

GP3S **GPIO3** input Status: This bit reflects the status of the **GPIO3** pin. If **GPIO3** is an output, this bit reflects the actual state of the pin. If **GPIO3** is an input:
If configured as sticky (**GP3ST**=1), this bit reads one when the **GPIO3** pin goes active (edge sensitive - edge defined by polarity bit **GP3PT**), and is cleared by writing a 0 to **GP3S**.
If configured as level sensitive (**GP3ST**=0), this bit reflects the current state of the **GPIO3** pin qualified by the polarity bit **GP3PT**.

GPSS GP_INT input Secondary Status. A general purpose input pin on the Secondary Codec (**ASDIN2**) caused slot 12, GP_INT to set (**SLT12M2.GP_INT** = 1). Writing **GPSS** = 0 clears the locally stored copy; however, since the interrupt condition occurred in the Secondary Codec, the condition must be removed through the Secondary Codec (**ACCTL.TC** = 1) GPIO Pin Sticky register, Index 54h. See Figure 52.

GPPS GP_INT input Primary Status. A general purpose input pin on the Primary Codec caused slot 12, GP_INT to set (**SLT12M.GP_INT** = 1). Writing **GPPS** = 0 clears the locally stored copy; however, since the interrupt condition occurred in the Primary codec, the condition must be removed through the Primary Codec (**ACCTL.TC** = 0) GPIO Pin Sticky register, Index 54h. See Figure 52.

GP1D **GPIO1** output data. When **ASDIN2/GPIO1** is not ASDIN2 and is configured as an output (**GP1OE** = 1), writes to this bit are presented on the **ASDIN2/GPIO1** pin.

GP3D **GPIO3** output data. When configured as an output (**GP3OE** = 1), writes to this bit are presented on the **GPIO3** pin. Note that in backward-compatible sockets, this pin is a PCI power supply pin.

VDNLT Volume Down Load/Type. Function dependent on hardware volume control enable.
Hardware Volume Control Enabled (**SSCR.HVC** = 1)
0 - GPIO logic input reflects the pin status directly
1 - GPIO logic input is pulse from Down hardware volume control logic. When a hardware volume change is generated from **VOLDN**, a pulse is sent to this GPIO input.
Hardware Volume Control Disabled (**SSCR.HVC** = 0)
0 - Enable **VOLDN** pin pullup
1 - Disable **VOLDN** pin pullup

VDNPO Volume Down input Polarity.
0 - active low
1 - active high

VDNST Volume Down input Sticky.
1 - **VOLDN** input pin is latched, for edge sensitive inputs, and presented on the **VNDS** bit. The **VDNS** bit is cleared by writing a 0 to **VDNS**.
0 - **VOLDN** input pin (after **VNDPO**) is presented on **VDNS** bit for level sensitive inputs.

VDNW Volume Down Wake. When set, **VOLDN** can cause a wake-up event (asserts **PME#**). **VDNST** must be set sticky for this bit to be effective.

VUPLT Volume Up Load/Type. Function dependent on hardware volume control enable.
Hardware Volume Control Enabled (**SSCR.HVC** = 1)
0 - GPIO logic input reflects the pin status directly
1 - GPIO logic input is pulse from Up hardware volume control logic. When a hardware volume change is generated from **VOLUP**, a pulse is sent to this GPIO input.
Hardware Volume Control Disabled (**SSCR.HVC** = 0)
0 - Enable **VOLUP** pin pullup
1 - Disable **VOLUP** pin pullup

VUPPO Volume Up input Polarity.
0 - active low
1 - active high

VUPST Volume Up input Sticky.
1 - **VOLUP** input pin is latched, for edge sensitive inputs, and presented on the **VUPS** bit. The **VUPS** bit is cleared by writing a 0 to **VUPS**.
0 - **VOLUP** input pin (after **VUPPO**) is presented on **VUPS** bit for level sensitive inputs.

VUPW Volume Up Wake-up. When set, **VOLUP** can cause a wake-up event (asserts **PME#**). **VUPST** must be set sticky for this bit to be effective.

GP1OE Output Enable **ASDIN2/GPIO1**. When this pin is not configured as **ASDIN2**, setting this bit enables the output buffer allowing writes to the **GP1D** bit to be presented on the pin.
0 = Output disabled, pin configured as input (reset default)
1 = Output enabled

GP1PT **GPIO1** input Polarity/output Type. When **ASDIN2/GPIO1** is not configured as **ASDIN2**:
When **ASDIN2/GPIO1** pin is configured as an input (**GP1OE** = 0), this bit sets the polarity.
0 - active low input
1 - active high input
When **ASDIN2/GPIO1** pin is configured as an output (**GP1OE** = 1), this bit sets the type
0 - CMOS output
1 - open drain output

GP1ST **GPIO1** input Sticky. Assumes **GP1OE** = 0 and pin not configured for **ASDIN2**.
1 - **GPIO1** input pin is latched, for edge sensitive inputs, and presented on the **GP1S** bit. The **GP1S** bit is cleared by writing a 0 to **GP1S**.
0 - **GPIO1** input pin (after **GP1PT**) is presented on **GP1S** bit for level sensitive inputs.

GP1W **GPIO1** Wake. When set, **GPIO1** can cause a wake-up event (asserts **PME#**). **GP1ST** must be set sticky for this bit to be effective and the pin must not be configured for **ASDIN2**.

GP3OE Output Enable **GPIO3**: Setting this bit enables the output buffer allowing writes to the **GP3D** bit to be presented on the **GPIO3** pin. Note that in backwards-compatible sockets, this pin is a PCI power supply pin.
0 = Output disabled, pin configured as input (reset default)
1 = Output enabled

GP3PT **GPIO3** input Polarity/output Type.
When the **GPIO3** pin is configured as an input (**GP3OE** = 0), this bit sets the polarity.
0 - active low input
1 - active high input
When the **GPIO3** pin is configured as an output (**GP3OE** = 1), this bit sets the type
0 - CMOS output
1 - open drain output

GP3ST **GPIO3** input Sticky. Assumes **GP3OE** = 0.
1 - **GPIO3** input pin is latched, for edge sensitive inputs, and presented on the **GP3S** bit. The **GP3S** bit is cleared by writing a 0 to **GP3S**.
0 - **GPIO3** input pin (after **GP3PT**) is presented on **GP3S** bit for level sensitive inputs.

GP3W **GPIO3** Wake. When set, **GPIO3** can cause a wake-up event (asserts **PME#**). **GP3ST** must be set sticky for this bit to be effective.

The **VOLUP** and **VOLDN** logic support software-control/notification of hardware volume. When Hardware Volume is enabled (**SSCR.HVC** = 1), these pins can indicate the status of the hardware volume pins, after debounce and enough time to allow for the other volume control to arrive (both being pressed) indicated a mute state. To enable hardware volume interrupts, the GPIO bits must be configured

- **VUPLT/VDNLT** = 1, GPIO input pulse from HW volume logic
- **VUPPO/VDNPO** = 1, active high (pulse)
- **VUPST/VDNST** = 1, sticky (capture pulse)
- **HIMR.VUPIM/HIRM.VDNIM** = 0, **VOLUP/VOLDN** interrupts enabled

Now host software will get an interrupt whenever hardware master volume does an update. To clear the interrupt **VxxS** must be written zero for the particular pin causing the interrupt.

If software wants total control over the master volume, then the update disable bits should all be set (SSCR bits **MVLD** = **MVAD** = **MVMD** = 1) which keeps the hardware volume control logic from updating any master volume. Then when an interrupt occurs:

- Read HISR to see if **VUPI/VDNI** set
- If so, write GPIOR, bits **VUPS** and/or **VDNS** to 0 which clears the interrupt (read-modify-write: don't mess with other GPIO bits)
- update master volumes using host software

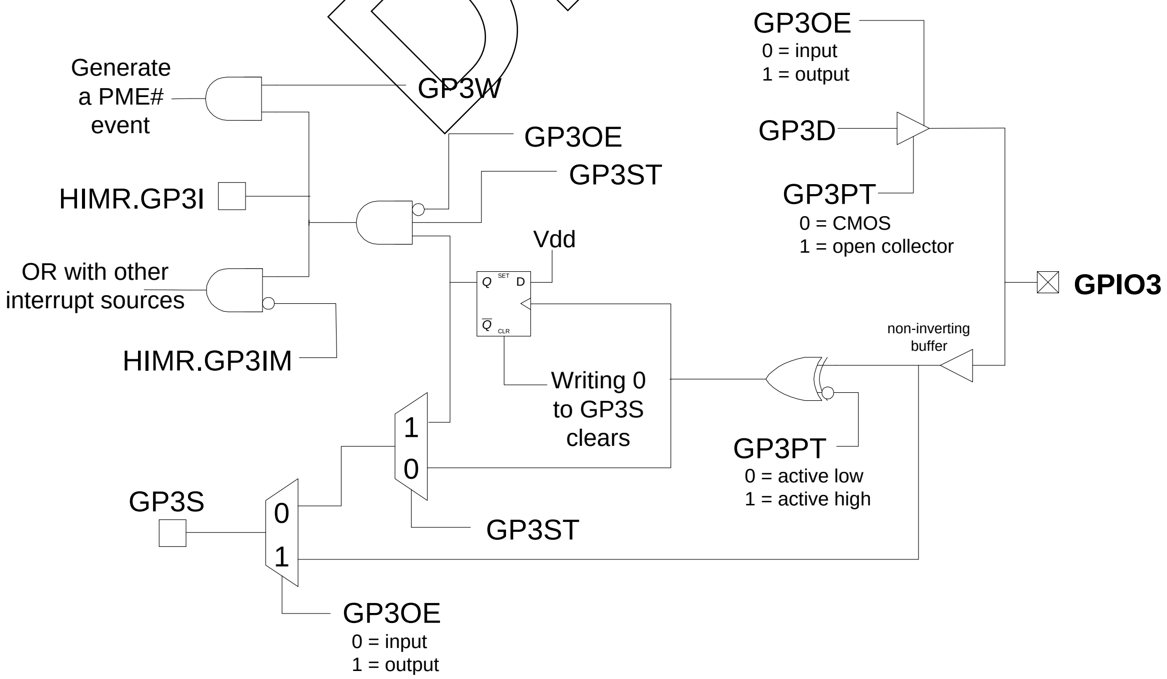


Figure 49. GPIO3 Conceptual Logic

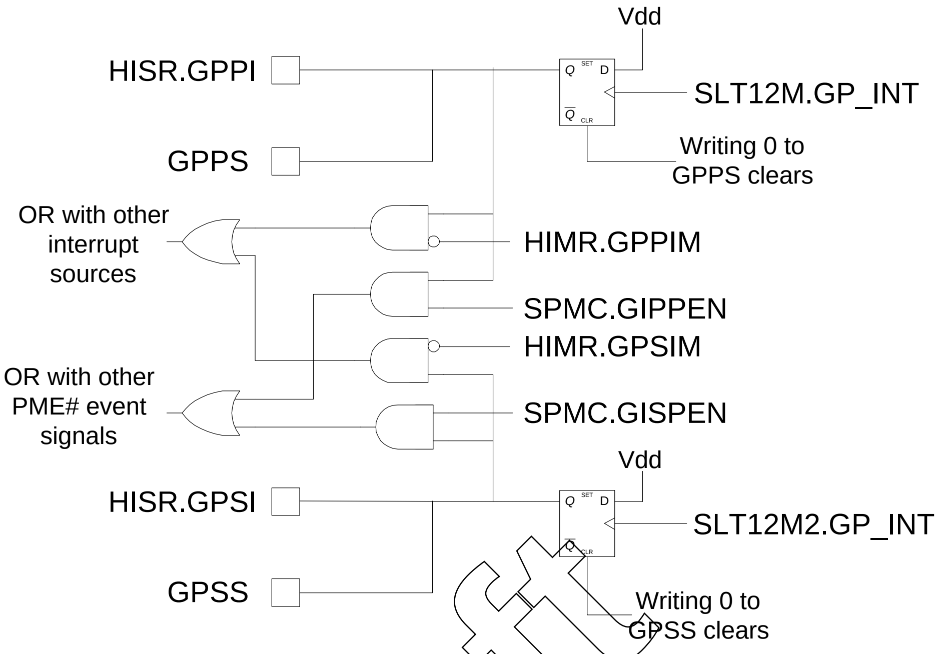


Figure 52. AC-Link Slot-12 GP_INT Conceptual Logic

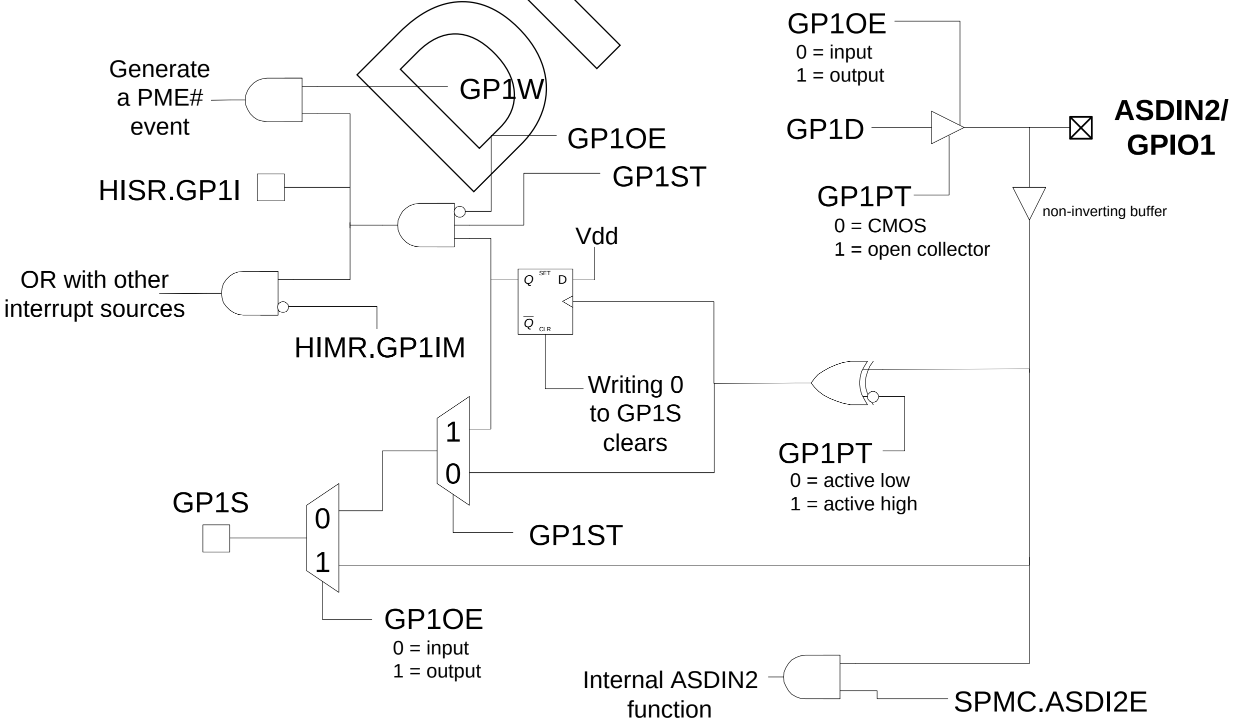


Figure 50. ASDIN2/GPIO1 Conceptual Logic

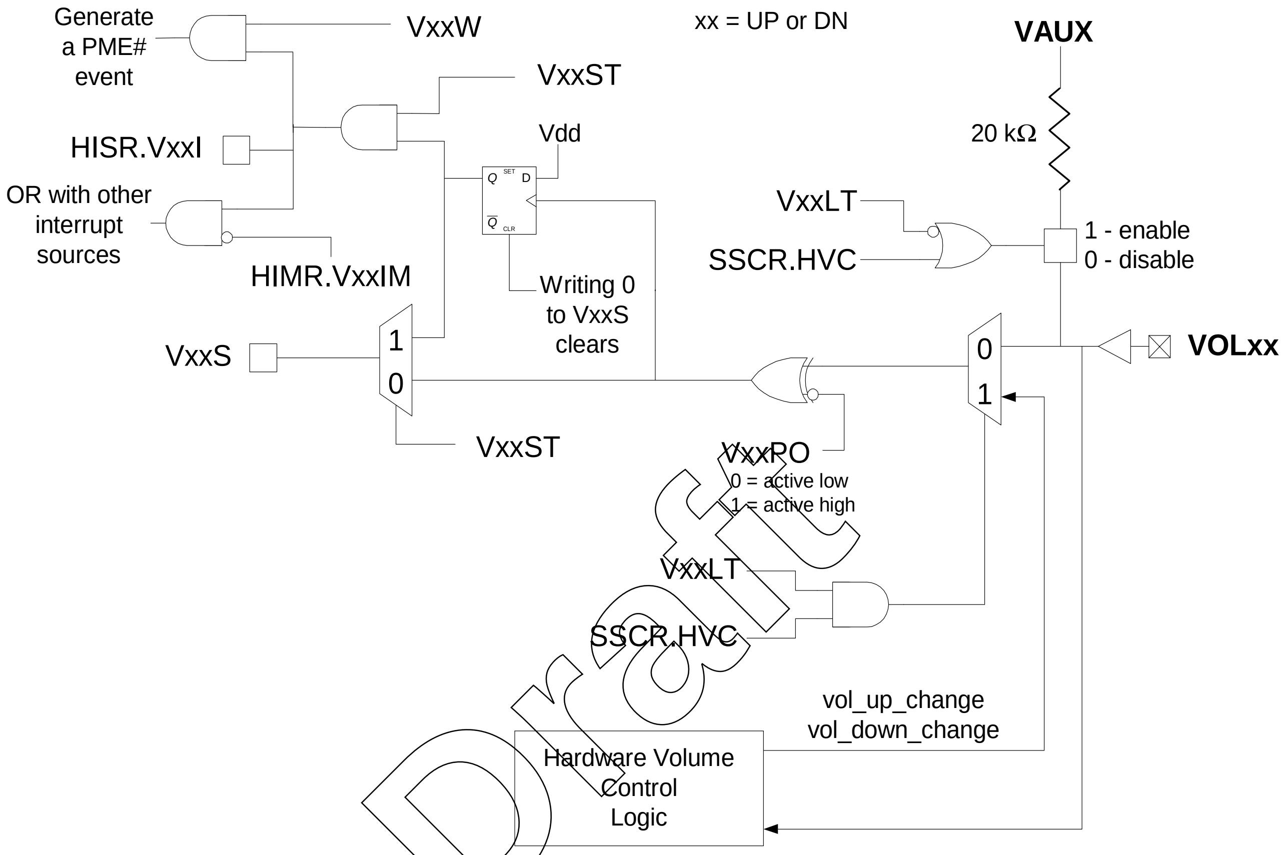


Figure 51. VOLUP/VOLDN GPIO Conceptual Logic

20.4 MIDI Port

In general, the MIDI port has two major roles. The MIDI port provides a basic bi-directional external MIDI interface for connection to “outside of the box” musical instruments, and a general MIDI port has non-UART mode wherein it responds to commands with appropriate responses (although no real work happens.) Most software just sends the MPU-401 the commands to go into “UART” mode and then sends MIDI data.

A hardware MPU-401 port (non-UART) mode does not exist on the CS4281. Therefore, the MIDI port is assumed to always be in “UART” mode and does not accept commands to enter or exit “UART” mode. Eliminating MPU-401 hardware implies that real-mode DOS software expecting an MPU-401 will not find one, ergo, the CS4281 does not support real-mode DOS MPU-401. In Windows, the MPU-401 can be coded into the driver to provide full MPU-401 interface to other software applications expecting an MPU-401 interface. Therefore, in Windows and a DOS box, the MPU-401 support can be provided by software.

20.4.1 MIDI Port Functional Description

The MPU-401 is an intelligent MIDI interface that was introduced by Roland in 1984. Voyetra Technologies subsequently introduced an IBM-PC plug in card that incorporated the MPU-401 functionality. The MPU-401 has become the defacto standard for controlling MIDI devices via IBM-PC compatible personal computers.

Although the standard MPU-401 does have some intelligence, the CS4281 implements the non-

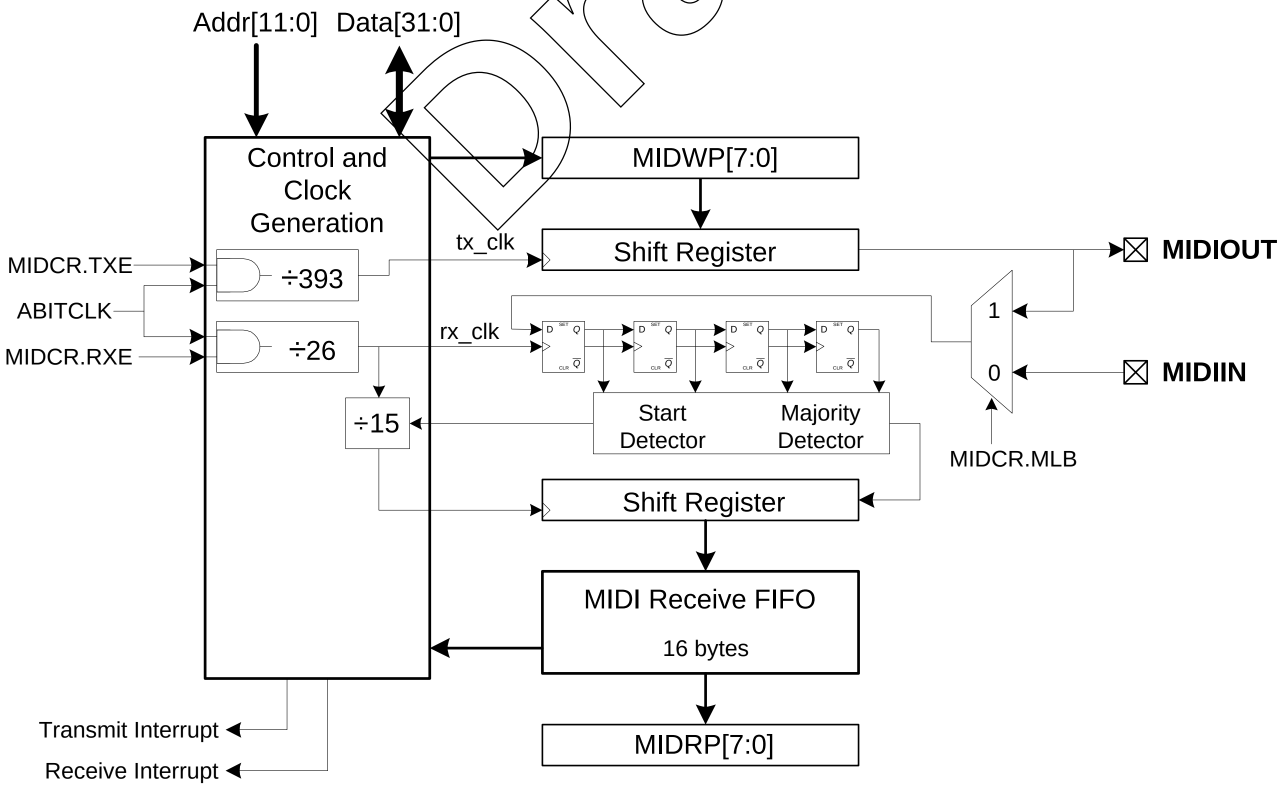


Figure 53. MIDI UART Architecture

intelligent mode in which the MPU-401 operates as a basic UART.

The MIDI interface is enabled through the MIDI Control register MIDCR, where **MIDCR.TXE** enables the transmitter and **MIDCR.RXE** enables the receiver. Interrupts, enabled through MIDCR, are also supported for both transmit and receive directions. Since the MIDI interface is considered to always be in the UART Mode, the MPU-401 Command register, MIDCMD, doesn't affect the MIDI port but is provided for backwards compatibility with older products. The MIDI receiver has a 16-byte FIFO allowing software more time to respond to interrupts. When the receive FIFO changes from empty to having a byte available, an interrupt can be generated. When the transmit buffer moves data from the MIDWP to the Transmit shift register, an interrupt can be generated. The interrupt conceptual logic is shown in Figure 54.

Since the interface is always in "UART" mode:

- All writes to the Transmit Port, MIDWP, are placed in the transmit buffer FIFO. Whenever the transmit buffer FIFO is not empty, the next byte is read from the buffer and sent out the **MIDIOUT** pin. The Status Register **MIDSR.TBF** is updated to reflect the transmit buffer FIFO status.
- All reads of the Receive Port, MIDRP, return the next byte in the receive buffer FIFO. When serial data is received from the **MIDIIN** pin, it is placed in the next receive buffer FIFO location. If the buffer is full, the last location is overwritten with the new data. The **MIDSR.RBE** is updated to reflect the new receive buffer FIFO state.
- All writes to the Command Register, MIDCMD, are ignored.

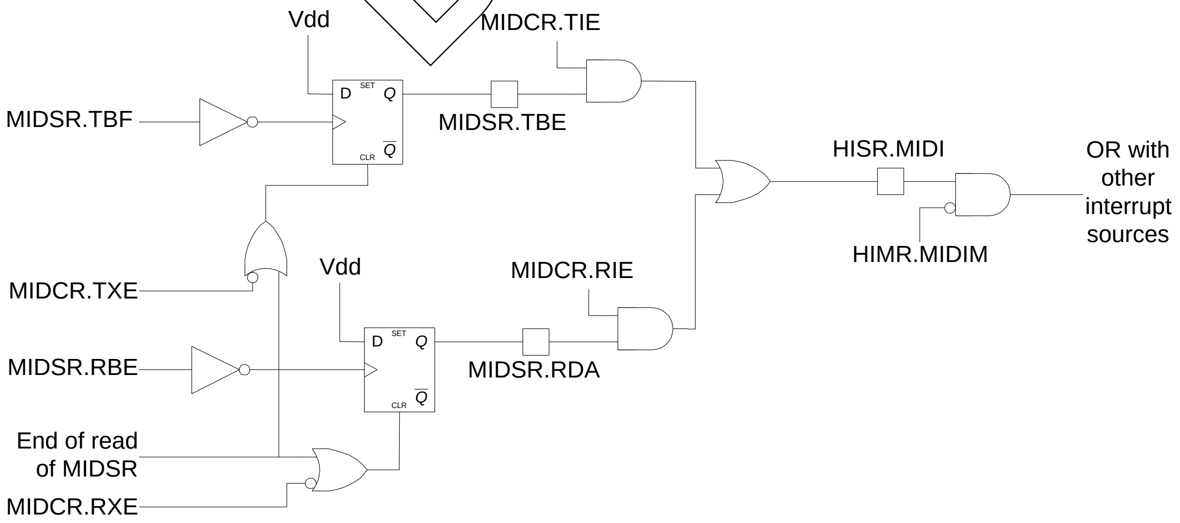


Figure 54. MIDI Interrupt Conceptual Logic

20.4.2 MIDI Control Register (MIDCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										MRST	MLB	TIE	RIE	RXE	TXE

Address: BA0: 490h, Read-Write

Default: 00000000h

Definition: The control port for the MIDI input and output ports.

Bit Descriptions:

TXE MIDI Transmit Enable: This bit controls the MIDI output port.
0 = Disable MIDI transmit (reset default)
1 = Enable MIDI transmit (output port enabled)

RXE MIDI Receive Enable: This bit controls the MIDI input port.
0 = Disable MIDI receive (reset default)
1 = Enable MIDI receive (input port enabled)

RIE MIDI Receive Interrupt Enable: This bit controls generation of host interrupts by the MIDI input port.
0 = Disable receive interrupts (reset default)
1 = Enable MIDI receive interrupts

TIE MIDI Transmit Interrupt Enable: This bit controls generation of host interrupts by the MIDI output port.
0 = Disable transmit interrupts (reset default)
1 = Enable MIDI transmit interrupts

MLB MIDI Loop Back Enable: This bit controls the loop back feature of the MIDI port.
0 = Disable loop back/normal operation (reset default)
1 = Enable loop back of MIDIOUT (internal) to MIDIIN (internal). When set, the external MIDIIN pin is disconnected from the path.

MRST MIDI Reset: This bit resets the MIDI interface, including the receive and transmit FIFOs.
0 = Normal operation (reset default)
1 = Reset MIDI interface (remains reset until this bit is set back to 0)

20.4.3 Host MIDI Command Register (MIDCMD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CS7	CS6	CS5	CS4	CS3	CS2	CS1	CS0

Address: BA0: 494h, Write-Only

Default: 00000000h

Definition: A host command port for the MPU-401 functions. This register has no effect and is provided for backwards compatibility.

Bit Descriptions:

CS[7:0] The MPU-401 Command Register.

20.4.4 Host MIDI Status Register (MIDSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDA	TBE							RBE	TBF	CS5	CS4	CS3	CS2	CS1	CS0

Address: BA0: 494h, Read-Only

Default: 00000080h

Definition: Host status port for the MIDI input and output ports.

Bit Descriptions:

CS[5:0] The 6 LSBs of the last command written to the Command Register.

TBF Transmit Buffer Full: This bit returns the full/not full status of the MIDI transmit FIFO.
0 = FIFO not full (reset default)
1 = FIFO full

RBE Receive Buffer Empty: This bit returns the empty/not empty status of the MIDI receive FIFO.
0 = FIFO not empty
1 = FIFO empty (reset default)

TBE This bit is sticky and is set when the **TBF** flag transitions from a 1 to a 0 and is cleared by reading this register. **TBE** is forced to 0 by **MIDCR.TIE** = 0. This bit is OR'd with **RDA** and reflected in the Interrupt Status register, **HISR.MIDI** bit. See Figure 54.

RDA This bit is sticky and is set when the **RBE** flag goes from a 1 to a 0 (data available) and cleared by reading this register. **RDA** is forced to 0 by **MIDCR.RIE** = 0. This bit is OR'd with **TBE** and reflected in the Interrupt Status register, **HISR.MIDI**. See Figure 54.

20.4.5 MIDI Write Port (MIDWP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MWD7	MWD6	MWD5	MWD4	MWD3	MWD2	MWD1	MWD0

Address: BA0: 498h, Read-Write

Default: 00000000h

Definition: MIDI transmit port.

Bit Descriptions:

MWD[7:0] MIDI Write Data: This byte is placed into the MIDI transmit port when written. A read of this port from the BA0 space will return the last byte written.

20.4.6 MIDI Read Port (MIDRP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								MRD7	MRD6	MRD5	MRD4	MRD3	MRD2	MRD1	MRD0

Address: BA0: 49Ch, Read-Only

Default: 00000000h

Definition: MIDI receive FIFO Port.

Bit Descriptions:

MRD[7:0] MIDI Read Data: This byte is removed from the MIDI receive FIFO when read.

20.5 Joystick Port

The Joystick (or Game) Port provides an interface to a standard personal computer type joystick. The joystick interface supports two joystick coordinate pairs and four push buttons. The PCI interface to the Game Port includes an address decoder and read/write strobe generator as well as a trapping register, IOTGP, to trap the legacy ISA I/O location of 200h/201h. The Game port operates in traditional polled mode only (no digital assist).

The Game Port hardware interface consists of four 555-like timers, read/write strobe generator, and data buffer. The joystick itself consists of two 100 kΩ potentiometers; one for the x-axis and one for the y-axis. As the joystick position is varied the resistance of the x and y axis potentiometers will also vary in direct proportion to the joystick movement. In addition one-to-four push buttons may be included. One timer is connected to each potentiometer. Two joysticks therefore require four timers. Once triggered the timer output pulse width is determined by the output current supplied by the timer, the joystick potentiometer resistance, and an external capacitor. Host software on the personal computer continually reads the timer pulse outputs and determines joystick position depending on the width of the pulses. The state of the push buttons are also monitored by the host software.

The joystick logic is enabled when **SSPM.JSEN** is set. When **SSPM.JSEN** is clear, the joystick logic is powered down, the coordinate input circuits are disabled, the registers are read-only with a value of 0h, and input clamp transistors are disabled. Therefore, if a joystick is attached, the pin will float up to Vcc reducing power. If no joystick is attached, the coordinate pins will float but the input circuit is disabled to stop excess current from flowing into the pad.

Traditional polled mode operates all 4 coordinates in parallel. The trigger event (caused by a write to the joystick register) releases the RC clamps, allowing the capacitor to charge through the joystick potentiometer.

The joystick default read upon power up (assuming no buttons are pressed) is 000000F0 hex. This is also the idle value once all RC time constants are finished. When the joystick is triggered, the data value read (with no button pressed) will be 000000FF hex. This scenario is illustrated in Figure 55.

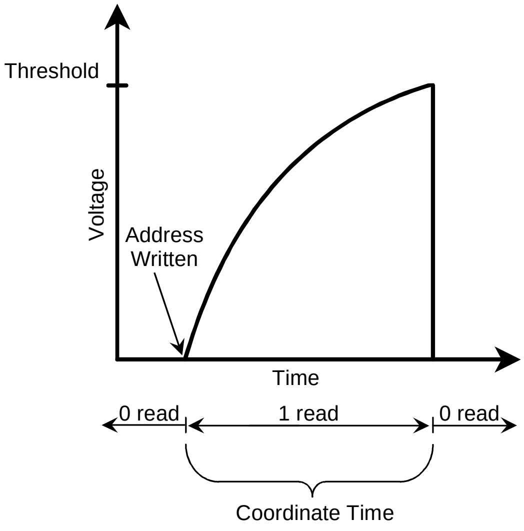


Figure 55. Joystick Coordinate Timing

20.5.1 Joystick Poll/Trigger Register (JSPT)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								BB2	BB1	BA2	BA1	CBY	CBX	CAY	CAX

Address: BA0: 480h, Read-Write

Default: 000000F0h

Definition: A port for polling the button inputs and coordinate comparator outputs. A write (data irrelevant) to this port triggers the coordinate charge/discharge cycle for polled measurement of coordinate values (the data is ignored on a write).

Bit Descriptions:

CAX AX Coordinate Comparator Output: This status bit returns the comparator output value for the AX coordinate input associated with the **JACX** pin.
0 = Charge point reached, now discharged (reset default)
1 = Charging in progress, comparator trip point not reached yet

CAY AY Coordinate Comparator Output: This status bit returns the comparator output value for the AY coordinate input associated with the **JACY** pin.
0 = Charge point reached, now discharged (reset default)
1 = Charging in progress, comparator trip point not reached yet

CBX BX Coordinate Comparator Output: This status bit returns the comparator output value for the BX coordinate input associated with the **JBCX** pin
0 = Charge point reached, now discharged (reset default)
1 = Charging in progress, comparator trip point not reached yet.

CBY BY Coordinate Comparator Output: This status bit returns the comparator output value for the BY coordinate input associated with the **JBCY** pin.
0 = Charge point reached, now discharged (reset default)
1 = Charging in progress, comparator trip point not reached yet

BA1 Button A1 Input: This status bit returns the input value associated with the **JAB1** pin.
0 = Button pressed (input pin is low)
1 = Button released (input pin is high)

BA2 Button A2 Input: This status bit returns the input value associated with the **JAB2** pin.
0 = Button pressed (input pin is low)
1 = Button released (input pin is high)

BB1 Button B1 Input: This status bit returns the input value associated with the **JBB1** pin.
0 = Button pressed (input pin is low)
1 = Button released (input pin is high)

BB2 Button B2 Input: This status bit returns the input value associated with the **JBB2** pin.
0 = Button pressed (input pin is low)
1 = Button released (input pin is high)

20.5.2 Joystick Control Register (JSCTL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
														SP1	SP0

Address: BA0: 484h, Read-Write

Default: 00000000h

Definition: Core powered. This register controls joystick port options.

Bit Descriptions:

SP[1:0] Comparator Set Point: This field controls the comparator trigger/threshold point, and therefore the relative speed of the joystick coordinate charging process. These times assume that the coordinate capacitors on the board are 12 nF and the joystick is powered from +5 V.
0 0 = Slowest speed, trip point 3.00 V (1.00 × original IBM joystick logic)
0 1 = Medium slow speed, trip point 2.52 V (1.31 × original IBM joystick logic)
1 0 = Medium fast speed, trip point 1.93 V (1.88 × original IBM joystick logic)
1 1 = Fastest speed, trip point 1.20 V, (3.34 × original IBM joystick logic)

The standard board connections for the CS4281 are illustrated in Figure 56.

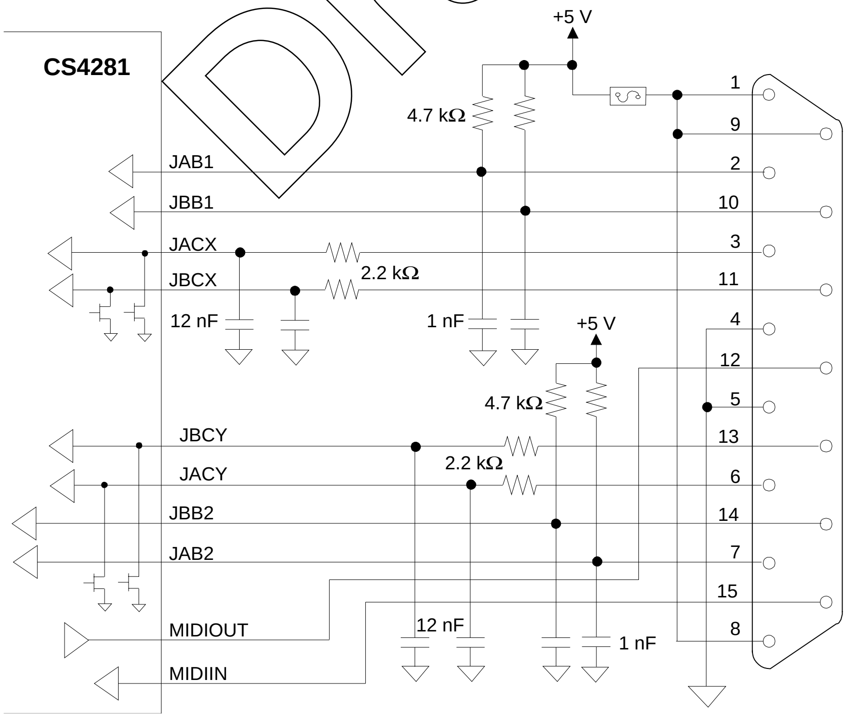


Figure 56. Joystick External Logic

21. EEPROM CONFIGURATION INTERFACE

The CS4281 configuration interface allows connection of an external serial EEPROM to the device in order to provide power-up configuration information. The chip configuration information must be loaded by a driver.

The external serial EEPROM is not required for proper operation of the device. However, it may be required to support specific operating system compatibility requirements; specifically, Microsoft requires PCI devices to support the configuration space subsystem vendor ID and subsystem ID (these are optional in the PCI 2.1 specification), and if the CS4281 is used on an expansion card, then an external serial EEPROM must be used to load these IDs. Four bytes of configuration data are also supported. If an external serial EEPROM is not present, then the device is configured by default as follows:

CS4281 Default Configuration

- Subsystem ID fields (ID and Vendor ID) = 00000000h
- Configuration Load Register (CFLR) = 00000000h

After a hardware reset, if the **SPMC.EESPD** bit is clear, a state machine in the CS4281 attempts to load the configuration data, if present (otherwise the two EEPROM interface pins are configured for use as PC/PCI request and grant signals). The EEPROM device is accessible to the host processor for reading/writing via a control register (described below). If **SPMC.EESPD** is set, the CS4281 state machine (defined as EEPROM engine) is held in reset and disconnected from the associated pins.

21.1 External Connection

Presence of the external configuration EEPROM is detected by attempting to load data from it and checking for a valid header. Connection of the device is via 2 pins on the device, EECLK and EEDAT. EECLK is used to provide the serial clock, and EEDAT is used to provide serial data I/O capability.

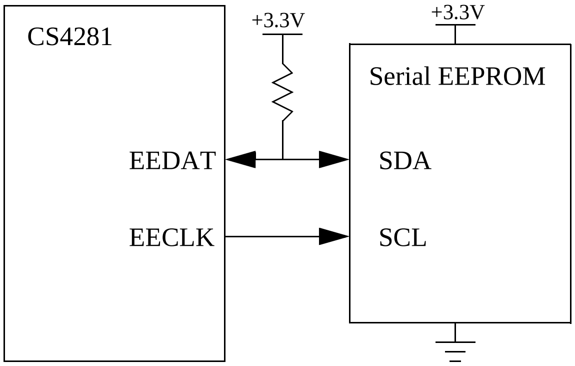


Figure 57. EEPROM interface block diagram

21.2 Initialization

On a hardware reset, if **SPMC.EESPD** is clear, a hardware-based EEPROM controller steps through the following sequence to initialize the CS4281:

CS4281 Initialization

- Enable EEPROM interface (switches mode of EECLK pin).
- Send a dummy write to set the byte address to 0.
- Start sequential read of bytes from EEPROM.
- Check signature header as loaded; abort if an invalid signature is detected.
- Load fixed number of bytes, transferring data into destination configuration bits as loaded.

The only time when the CS4281 accesses the EEPROM is after a hardware reset; the CS4281 can only read EEPROM devices — it cannot write them unassisted. Writing a EEPROM can be accomplished through a configuration interface register accessible from the host. The timing of the data and clock signals for the initialization load are generated by a hardware state machine. The minimum timing relationship between the clock and data is shown in the figure below. The state of the data line can change only when the clock line is low. A state change of the data line during the time that the clock line is high is used to indicate start and stop conditions.

The EEPROM device read access sequence is shown in the figure below. The timing follows that of a random read sequence. The CS4281 first performs a dummy write operation, generating a start condition followed by the slave device address and an byte address of zero. The slave address is made up of a device identifier (0xA) and a bank select (bits A2-A0). The bank select bits select among eight 256 byte blocks. The bank select bits may be used to select among multiple 256 byte blocks within a single device, i.e. a 1 kbyte memory may be comprised of a single 1 kbyte EEPROM with four 256 byte banks. The CS4281 always begins access at byte address zero and continues accessing one byte at a time. In the EEPROM, the byte address automatically increments by one until a stop condition is detected.

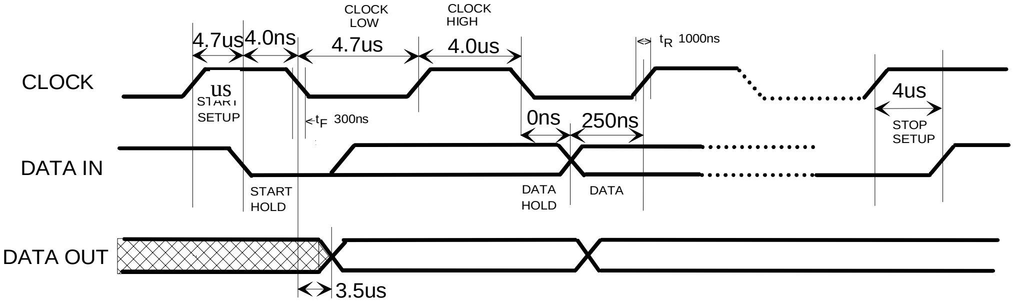


Figure 58. EEPROM timing requirements

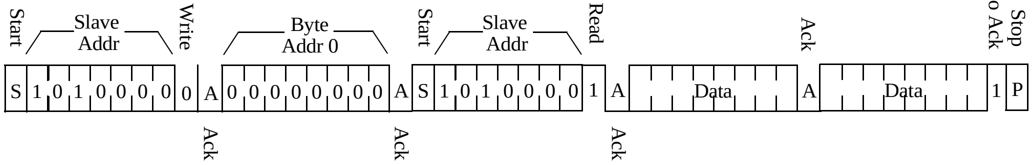

Start
Slave Addr
Write
Byte Addr 0
Start
Slave Addr
Read
Ack
No Ack
Stop
S 1 0 1 0 0 0 0 0 A 0 0 0 0 0 0 0 0 A S 1 0 1 0 0 0 0 1 A Data A Data 1 P
Ack
Ack
Ack

Figure 59. EEPROM read sequence

21.3 Control Register

The configuration interface register (CFGI) contains the control bits for host software-based manipulation of the external EEPROM (if attached).

21.3.1 Configuration Interface Register (CFGI)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											PC/PCI	EELD	DIN/EEN	DOUT	CLK

Address: BA0: 4B0h, Read-Write

Default: 00000000h

Definition: A host port for reading and writing an external serial EEPROM. This register operates independently of the **SPMC.EESPD** bit. Although **SPMC.EESPD** disabled the EEPROM engine from reading the EEPROM, host software may still bit-bang the interface through this register.

Bit Descriptions

PC/PCI Read-Only status bit indicating configuration condition of the PC/PCI support:
0 = PC/PCI is not configured, the EEPROM may be present.
1 = PC/PCI is configured, there is not EEPROM device attached.

EELD Indicates a successful load of initialization data happened after hardware reset.
0 = EEPROM load aborted (EE not present or header mismatch).
1 = EEPROM load successful.

DIN/EEN On reads, **DIN** is used to read back data from the EEPROM, on the **EEDAT** pin. On writes, **EEN** enables the **DOUT** and **CLK** bits onto the CS4281 **EEDAT** and **EECLK** pins.

DOUT When **DIN/EEN** = 1, the **EEDAT** pin is driven low when this bit is a zero, and when this bit is a one, the **EEDAT** pin floats.

CLK When **DIN/EEN** = 1, the **EECLK** pin follows the state of this register bit.

Usage: The CS4281 provides a two wire serial interface that is directly controlled by three register bits. The timing is controlled by the host software. One pin is used as a clock (**EECLK**). One pin is used as a bi-directional data wire (**EEDAT**) to the EEPROM device, where the output is controlled by the **DOUT** bit and the input is read from the **DIN/EEN** bit (when the **DOUT** bit is a one).

PCI bus access to the EEPROM is enabled via the **DIN/EEN** bit in the CFGI register. When the **DIN/EEN** bit is written to a one then the **CLK** and **DOUT** bits are enabled on the **EECLK** and **EEDAT** pins, respectively. The timing of the clock and data signals is completely determined by host-based software and should meet the timing requirements as shown previously. In order to read back data from the EEPROM device, the **DOUT** bit must be set to a one.

21.4 EEPROM Memory Format

Nine bytes are needed in the EEPROM with the following table showing the EEPROM contents. The Subsystem ID's are written to the PCI Config space FCh base address. This address is read-writable. The 2Ch base address in config space is a read-only version of the data at FCh. The Configuration bytes are located in the *Feature and Configuration Reporting* section, CFLR register.

Byte Offset	Field Description	Configuration Location	Notes
0	Header / Version: Constant 55h	N/A	Abort if read byte not 55h
1	Subsystem vendor ID low byte	Offsets 2Ch, FCh	Write FCh, visible at 2Ch
2	Subsystem vendor ID high byte	Offsets 2Dh, FDh	Write FDh, visible at 2Dh
3	Subsystem ID low byte	Offsets 2Eh, FEh	Write FEh, visible at 2Eh
4	Subsystem ID high byte	Offsets 2Fh, FFh	Write FFh, visible at 2Fh
5	Configuration byte 1 for CFLR register	Offset F0h	Contents defined by driver
6	Configuration byte 2 for CFLR register	Offset F1h	Contents defined by driver
7	Configuration byte 3 for CFLR register	Offset F2h	Contents defined by driver
8	Configuration byte 4 for CFLR register	Offset F3h	Contents defined by driver

Table 34. EEPROM Data Format

22. AC ‘97 INTERFACE

The dual codec architecture of AC ‘97 revision 2.1 is supported, wherein the second codec uses the same pins from the primary, with the exception of a separate serial data in line. The **ASDIN/ASDIN2** AC-Link pins is powered from **VAUX** to insure wakeup capability when Core and PCI power are removed. The serial port engine is enabled and running when **SSPM.ACLEN** is set. When **SSPM.ACLEN** is clear, the serial port engine is reset, powered down, and all serial port registers reset to their default states. All registers are also reset by the PCI **RST#** signal except as noted in the register description (as in SPMC).

22.1 Hardware Architecture

The following pins are used by the AC ‘97/98 serial port.

- **ABITCLK**
- **ASYNC**
- **ASDOUT**
- **ASDIN**
- **ASDIN2/GPIO1**
- **ARST#**

If the system supports only one AC ‘97 codec, the **ASDIN2/GPIO1** pin has extended GPIO functionality similarity to the GPIO functionality in the *Audio Codec ‘97* spec, revision 2.1. Other pins with extended functionality include **GPIO3**, **VOLUP**, and **VOLDN**. **VOLUP** and **VOLDN** can only be programmed as inputs. More information can be found in the *General Purpose Input/Output Pins* section.

22.1.1 AC ‘97 Register Management

The AC ‘97 codec registers are accessed through slots 0, 1, and 2. Slot 0 contains tag bits for each slot. Since the interface operates at a constant 48 kHz rate, register updates can occur faster than they can be sent down the AC-Link. Host software accesses the Codec registers by setting up the address (ACCAD) and data (ACCDA) and selecting the particular codec (**ACCTL.TC**). Then host software sets the **ACCTL.DCV** bit. Once **DCV** is set, state machine waits for the next available frame, sets the tag bits in slot 0 for slots 1 and 2 valid, copies the address from ACCAD to slot 1, copies the data from ACCDA to slot 2, and then automatically clears the **DCV** bit (once the frame is on its way). For the host to send data, it must actually check that **ACCTL.DCV** is clear otherwise it would overwrite the last data. Therefore, the host has a feedback mechanism to delay updating more registers.

Table 35 lists the registers needed to configure the AC-Link and the registers in common when two codecs are connected. The **SSPM.ACLEN** bit enables the AC-Link engine, **SPMC.RSTN** releases the **ARST#** signal, and the **ACCTL.ESYN** enables **ASYNC** generation. These three steps should enable the codec to start AC-Link operation. Since the two codecs share the same **ASDOUT** line, the **ASDOUT**-specific registers are common to both codecs. Codec Register access is supported through the ACCDA and ACCAD registers with the ACCTL register enabling the actual transfer. The ACOSV sets the audio data slot valid bits, sent during slot 0, and cause the AC-Link engine to start requesting slot IDs from the Sound System Controller. The final common register supports the EGPIO bits in slot 12 that are programmed (in the modem codec) for outputs.

Although the ASDOUT line is shared across the two codecs, a separate data-in line exists for each codec. Table 36 lists the registers that are unique to each codec. Before any operations or data can be sent to a codec, the codec ready bit, **ACSTSx.CRDY** must be active. Another bit exists in ACSTSx that indicates if the address and data slots/registers from the codec contain valid data. Valid data will remain in these registers until read. No new data is accepted until the old data is read by host software. The data received is from register read requests made across the ASDOUT line on the previous frame. The ACISV register indicates which audio slots have valid data in them (bits set from codec), and cause the AC-Link engine to tell the Sound System Controller that the AC-Link has valid data to transfer to a FIFO or the CSRC. The last input register is SLT12Mx which stores EGPIO bits (programmed as inputs on the modem codec) from slot 12.

22.1.2 FIFO

The CS4281 contains a single ported FIFO RAM with 128 locations. Each location holds a left/right stereo audio sample pair. The locations are arranged into 4 controllers called FIFO0 - FIFO3, each of which defaults to 32 entries deep. The size of each FIFO is configurable. There is one pointer for the entire FIFO RAM. The FIFO Control Registers (FCRn) define the mapping from PCI bus to FIFO RAM and from the FIFO to AC Link slots. The Sample Rate Converter hardware determines the transfer rate through two of the audio data streams. The other two streams are fixed at the same rate as the AC Link, 48 kHz. The FIFO Mapping section details which FIFO audio channels are mapped to which AC-Link slots.

Reg Name	BA0 Address	Description
General Configuration registers:		
SSPM	740h	AC-Link engine enable
SERMC	420h	Loopbacks, Target Codec ID, Variable Sample Rate Support
SPMC	3ECh	Controls ARST# pin, and asynchronous generation of ASDOUT and ASYNC, and ASDIN2 enable for secondary codec support
ACCTL	460h	Enables ASYNC generation
Common AC-Link (ASDOUT) registers:		
ACCTL	460h	Controls sending of register address/data writes to either codec on ASD-OUT
ACCAD	46Ch	Command/Register address field (slot 1) on ASDOUT
ACCDA	470h	Command/Register data field (slot 2) on ASDOUT
ACOSV	468h	Audio data Slot Valid Tag bits send during slot 0 of ASDOUT
SLT12O	41Ch	EGPIO bits for slot 12 of ASDOUT

Table 35. AC-Link Common Register Table

Primary (ASDIN)		Secondary (ASDIN2)		
Name	BA0 Addr.	Name	BA0 Addr.	Description
ACSTS	464h	ACSTS2	4E4h	Codec Ready bit, Valid register data bit
ACSAD	478h	ACSAD2	4F8h	Register Address for register reads from codec - slot 1
ACSDA	47Ch	ACSDA2	4FCh	Register Data for register reads from codec - slot 2
ACISV	474h	ACISV2	4F4h	Audio data Input Slot Valid Tag bits received from slot 0
SLT12M	45Ch	SLT12M2	4DCh	EGPIO bits from slot 12

Table 36. AC-Link Codec-Specific Registers

AC-Link Slots	Slot Identification	
	Decimal	Hex
AC 97 primary and secondary links output slot 3	0	00
AC 97 primary and secondary links output slot 4	1	01
AC 97 primary and secondary links output slot 5	2	02
AC 97 primary and secondary links output slot 6	3	03
AC 97 primary and secondary links output slot 7	4	04
AC 97 primary and secondary links output slot 8	5	05
AC 97 primary and secondary links output slot 9	6	06
AC 97 primary and secondary links output slot 10	7	07
AC 97 primary and secondary links output slot 11	8	08
Not used	9	09
AC 97 primary link input slot 3	10	0A
AC 97 primary link input slot 4	11	0B
AC 97 primary link input slot 5	12	0C
AC 97 primary link input slot 6	13	0D
AC 97 primary link input slot 7	14	0E
AC 97 primary link input slot 8	15	0F
AC 97 primary link input slot 9	16	10
AC 97 primary link input slot 10	17	11
AC 97 primary link input slot 11	18	12
AC 97 primary link input slot 12	na	
AC 97 secondary link input slot 3	20	14
AC 97 secondary link input slot 4	21	15
AC 97 secondary link input slot 5	22	16
AC 97 secondary link input slot 6	23	17
AC 97 secondary link input slot 7	24	18
AC 97 secondary link input slot 8	25	19
AC 97 secondary link input slot 9	26	1A
AC 97 secondary link input slot 10	27	1B
AC 97 secondary link input slot 11	28	1C
Left Wavetable Data (FM Digital mixing stream)	29	1D
Right Wavetable Data (FM Digital mixing stream)	30	1E
Throw-away slot (FIFO channel turned off)	**31**	**1F**

Table 37. FIFO-to-AC ‘97 Link Slot Identification

22.1.3 AC-Link Variable Sample-Rate Support

The CS4281 supports the *AC '97 Codec Spec.*, revision 2.1 extension (Appendix A) for Codecs that have built-in sample rate converters (SRC). This feature provides variable sample frequency support on the CS4281 streams not attached to the internal SRCs. The AC Link always operates at a frame rate of 48 kHz. For data coming from the Codec (record) data, the Slot Valid tag bits in slot 0 will be dynamically controlled to indicate when a valid sample is available. Only the AC-Link slots with valid slot bits will be transferred into the FIFO. Therefore, dynamic changing of input slot bits controls the sample frequency of data coming from the AC Link.

For data sent to the Codec (playback), the Codec indicates when it wants data for a particular slot. Generally host software sets the output slot valid bits (ACOSV) to indicate active slots. Under fixed sample rate support (Codecs designed for the original 1.03 spec), the serial port engine will transfer samples from the FIFO to the valid slot at a 48 kHz rate. The AC '97 specification was expanded to use reserved bits in slot 1 on **ASDIN** (or **ASDIN2**) to indicate that the Codec needs data in the **next** frame. The spec. supports slots 3 through 12, but the CS4281 uses slot 12 for GPIO indication so the CS4281 only supports *on-demand* samples for slots 3 through 11. On demand support is enabled by setting **SERMC.ODSEN1** for the primary **ASDIN** input, and **SERMC.ODSEN2** for the secondary **ASDIN2** input. When the next frame comes along (**ASYNC** going high), the latched data is transferred to register AODSD1/2. The AODSD1/2 registers contain read-only bits **NDS[11:3]** which, when high, indicate data should not be sent in the next frame. Since the CS4281 supports two Codecs, if either of the bits from the two Codecs is set, the output data is blocked from being sent down the AC Link and the tag bit in **ASDOUT** slot 0 for the blocked slot is cleared.

22.2 AC-Link Power Management

The majority of work required to power down the AC link is left to Host Software (HSW). The CS4281 must already be configured properly before the host sends data to the Codec to power down the AC Link (PR4). The only function handled by the serial port engine is when the serial port engine detects loss of **ABITCLK**, it will reset the frame generation logic forcing **ASYNC** and **ASDOUT** low, since a frame hasn't started. This step could also be done by host software using **ACCTL.ESYN**. If the AC Link is configured for both codecs and the AC Link is powered back up, the serial port engine will start transmitting data on **ASDOUT** when **either** *Codec Ready* bit comes active. Host software has the responsibility to not start sending data until the proper codec is active. The *Audio Codec '97 Specification*, revision 2.1, contains more details regarding AC-Link power-down (sections 6.3.11, 7.0, 9.2.5, and Appendix D).

The following is the sequence to power down the AC link:

1) HSW makes sure all PCM output channels are off
   - all FIFO and DMA engines idled/off (SB must be idled/off)
   - all slot valid bits clear; ACOSV
2) HSW powers down ADC/DACs in Codecs, register 26h (PR0 = PR1 = 1)
   - all PCM slot valid input bits should be clear; ACISV and ACISV2
3) HSW sets primary codec register 26h to power down AC link (PR4 set).

- The CS4281 will detect the loss of **ABITCLK** and drive **ASYNC** low and **ASDOUT** low (and reset serial port state machine/frame gen. engine)
  Hardware Volume control registers will be frozen (will not respond to **VOLUP/DN** changes).

Once powered down, host software must not restart the AC Link for a minimum of 4 frames. To restart the AC Link:

1) HSW does one of the following:
   - Warm power-up: drive **ASYNC** high for 1 µs, then drive **ASYNC** low; **ACCTL.ASYN**, or
   - Cold power-up: drive **ARST#** low for 1 µs, then drive **ARST#** high; **SERMC.RSTN**.
2) This will cause the Codec to start **ABITCLK**.
3) The CS4281 will detect start of **ABITCLK** and release **ASYNC** and **ASDOUT** to start a new frame.
   - Hardware Volume Controls will remain frozen until the Codec Ready bit from the Codec hardware volume is connected to is set.
4) HSW monitors for Codec Ready bit; **ACSTS.CRDY** and **ACSTS2.CRDY2**
5) If a Warm power-up, HSW powers up ADC/DACs in Codecs, register 26h (PR0 = PR1 = 1)
6) If a Cold power-up, HSW reinitializes the entire Codec, which is in the power-up default configuration.
7) HSW then re-initialize streams (DMA/Polled) for operation.

AC '97 Codecs containing modem functionality will treat a cold power-up as a warm power-up to maintain internal registers.

22.3 Loopback Modes

The CS4281 has numerous loopback modes that support and enhance the loopback modes available on the AC '97 Codecs. The loopbacks are centered on the FIFO-to-AC Link-to-Codec. To verify the PC-to-FIFO link, the FIFO memory can be read and written directly. The loopbacks can be divided into two types based on direction: Analog In/Analog Out, and Bus-Centric or Digital In/Digital Out.

The Bus-Centric loopbacks are illustrated in Figure 60. These loopback modes start at a CS4281 Playback FIFO and end at a Record FIFO. In general, they start and end at the PCI bus. All Codec loopbacks are part of the Modem extension to the AC '97 Specification and the *AC-Link Data Loopback* (Loopback 101 at index 56h), is only available on the CS4298 Codec.

The Analog-Centric loopbacks are illustrated in Figure 61 and start and end in the analog domain. These loopbacks generally require external analog test equipment such as an Audio Precision System Two. They allow different amounts of circuitry to be testing in the analog domain to see what affect the added circuitry has on the analog performance. All Codec loopbacks, with the exception of the **LPBK** bit at index 20h, are part of the Modem extension to the AC '97 Specification.

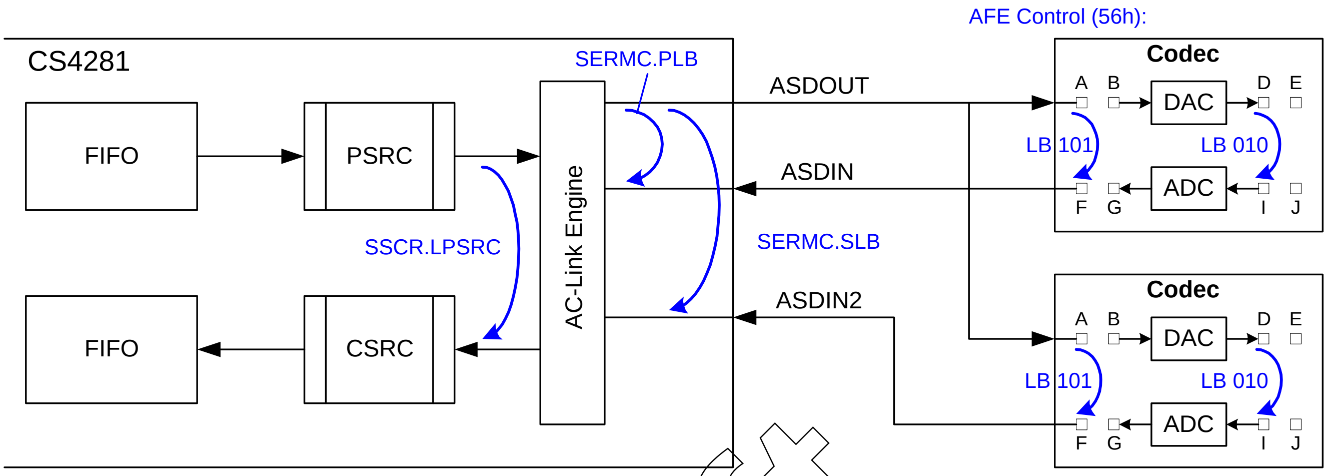


Figure 60. Bus-Centric Loopback Modes

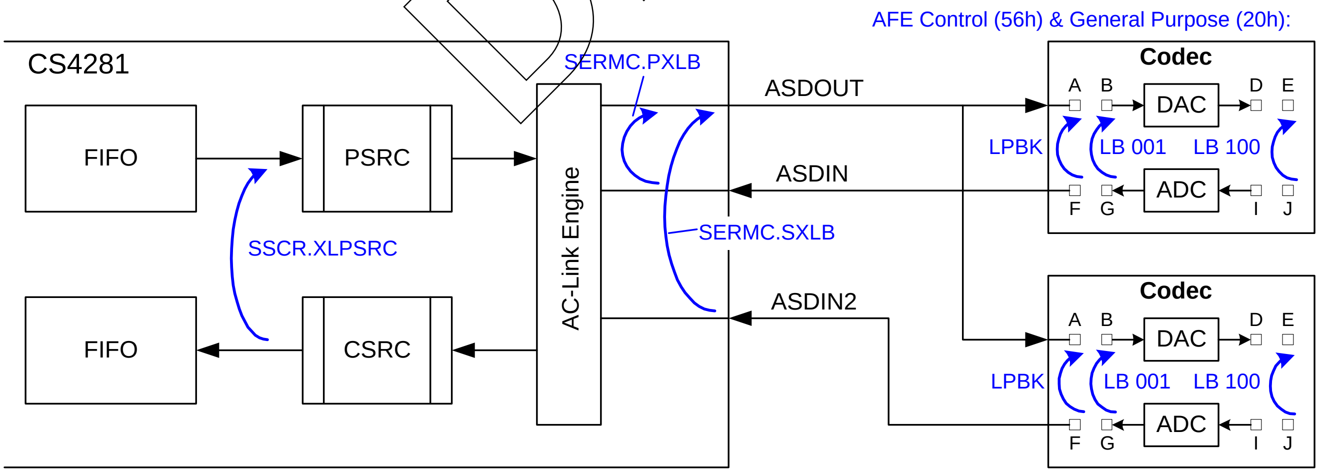


Figure 61. Analog-Centric Loopback Modes

22.4 General Configuration Registers

22.4.1 Serial Port Master Control Register (SERMC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				FCRN		ODSEN2	ODSEN1			SXLB	SLB	LOFV		TCID1	TCID0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										PXLB	PLB	PTC2	PTC1	PTC0	MSPE

Address: BA0: 420h, Read-Write, some bits Read-Only

Default: 00010003h

Definition: Control of the CS4281 AC Link loopback modes.

Bit Descriptions:

MSPE Master Serial Port Enable: This bit is read-only as the AC Link is always enabled.
1 = AC Link operational

PTC[2:0] Port Timing Configuration: This 3 bit field is read-only and identifies the timing configuration of the CS4281 serial ports.
0 0 1 = AC ‘97 link mode, **ABITCLK** in, **ASYNC** out

PLB Primary Port Internal Loop Back: This bit controls the internal loopback of **ASDOUT** to **ASDIN**.
0 = Loopback disabled (reset default)
1 = **ASDOUT** to **ASDIN** loopback enabled

PXLB Primary Port External Loop Back: This bit controls the pin level loopback of **ASDIN** to **ASDOUT**.
0 = Loopback disabled (reset default)
1 = **ASDIN** to **ASDOUT** loopback enabled

TCID[1:0] Target Codec ID. These bits are output on **ASDOUT**, slot 0, Codec ID bits 1 and 0 (respectively) when **ACCTL.TC** is set, and specify the particular secondary codec attached. Since these bits support the secondary codecs, they should never be 00.

LOVF Loopback Output Valid Frame bit. This bit is OR’d with the internal Valid Frame bit for **ASDOUT**. Allows host software to force valid frames. Typically, only used when **SLB/PLB** set Normally the Valid Frame bit is controlled by hardware based on the codec input valid bit and when output data needs to be sent; however, in loopback mode, the normal logic would be incestuous so this bit provides a way to set the Frame Valid bit when **ASDOUT** is looped back to **ASDIN** or **ASDIN2**.

SLB **ASDOUT** to **ASDIN2** Loopback: This bit controls the internal loopback of **ASDOUT** to **ASDIN2**.
0 = Loopback disabled (reset default)
1 = **ASDOUT** to **ASDIN2** internal loopback enabled

SXLB **ASDIN2** to **ASDOUT** Loopback: This bit controls the pin level loopback of **ASDIN2** to **ASDOUT**. This bit and the **PXLB** bit cannot be active simultaneously. **PXLB** has priority over **SXLB**.
0 = Loopback disabled (reset default)
1 = **ASDIN2** to **ASDOUT** pin level loopback enabled

ODSEN1 On-Demand Support Enable. When set, the AODSD1 register stores the On-Demand Slot bits from **ASDIN**, slot 1. These bits support the variable-rate extension to the 2.1 AC '97 spec. When clear, the AODSD1 register is forced clear, and all output slots with their valid bits set (**ACOSV.SLT[11:3]**, transfer data every frame.

ODSEN2 On-Demand Support Enable. When set, the AODSD2 register stores the On-Demand Slot bits from **ASDIN2**, slot 1. These bits support the variable-rate extension to the 2.1 AC '97 spec. When clear, the AODSD2 register is forced clear, and all output slots with their valid bits set (**ACOSV.SLT[11:3]**, transfer data every frame.

FCRN Force Codec Ready Not. When clear, the serial port engine sends **ASDOUT** valid slot data regardless of the input codec ready bits (solely based on the output slot valid bits). When set, at least one codec ready bit from **ASDIN** or **ASDIN2** must be set before data is sent out **ASDOUT**.

22.4.2 Serial Port Power Management Control (SPMC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIPPEN	GISPEN					EESPD	ASDI2E	ASDO				WUP2	WUP1	ASYN	RSTN

Address: BA0: 3ECh, Read-Write
PCI CFG: 0ECh, Read-Write if CWPR configured, otherwise Read-Only

Default: 0000h

Definition: Vaux powered. Supports power management of the AC Link and the enable for **ASDIN2**. This register resides in the PCI config space and is only reset by a Vaux POR circuit. This register is unaffected by the PCI **RST#** signal. The default value is set by a Vaux POR signal. Host software should initialize this register before use.

Bit Descriptions:

RSTN Reset NOT!: This bit controls the **ARST#** pin. Note the negative sense of the bit, which matches the active low output pin definition. The **ARST#** pin is a logical OR of **RSTN** with the PCI reset pin **RST#**.
0 = **ARST#** active, AC-Link and Codec reset (reset default)
1 = **ARST#** inactive, AC-Link and Codec not reset (normal operation).

ASYN Asynchronous **ASYNC** Assertion: This bit allows the unclocked assertion of the **ASYNC** pin for AC-Link management protocol requirements.
0 = Normal **ASYNC** generation (reset default)
1 = Force **ASYNC** valid (with no clocking dependencies other than PCI clock)

WUP1 Wakeup for primary input: This bit indicates that a CS4298 codec attached to the **ASDIN** pin signaled a wake-up event by forcing a low-to-high transition on **ASDIN** while the AC-Link is down. This bit remains set until host driver software issues a warm reset of the AC-Link by setting the **ASYN** bit; specifically, the falling edge of the **ASYNC** warm reset pulse clears this bit. See Figure 18 in *PCM# Assertion* section for conceptual logic. If the AC Link is powered up, the host driver should check the **GP_INT** bit in SLT12M for **PME#** event generation.
0 = No wake-up event signaled by **ASDIN**
1 = Wake-up event signaled by **ASDIN**

WUP2 Wakeup for secondary input: This bit indicates that a AC ‘97 2.0 Codec is attached to the **ASDIN2** pin signaled a wake-up event by forcing a low-to-high transition on **ASDIN2** while the AC-Link is down. This bit remains set until host driver software issues a warm reset of the AC-Link by setting the **ASYN** bit; specifically, the falling edge of the **ASYNC** warm reset pulse clears this bit. See Figure 18 in *PCM# Assertion* section for conceptual logic. If the AC Link is powered up, the host driver should check the **GP_INT** bit in SLT12M2 for **PME#** event generation.
0 = No wake-up event signaled **ASDIN2**
1 = Wake-up event signaled by **ASDIN2**

ASDO Asynchronous **ASDOUT** Assertion: This bit allows the unclocked assertion of the **ASDOUT** pin to support Codec debug test modes. Test modes for the AC ‘97 codec require some combination of **ASDOUT** and **ASYNC** high when **ARST#** is driven high (de-asserted). Once test modes are enabled, a test register inside the Codec enables the particular test of interest. **ASYNC** is controlled by the **ASYN** bit and **ARST#** by the **RSTN** bit.
0 = Normal **ASDOUT** generation (reset default)
1 = Force **ASDOUT** valid (with no clocking dependencies other than PCI clock)

GIPPEN **GP_INT** Primary **PME#** Enable for primary **ASDIN2** Slot 12 data. When set, allows **SLT12M.GP_INT** to generate a PME event when **GP_INT** goes from 0 to a 1. See Figure 52 in the *General Purpose Input/Output Pins* section for conceptual logic.

GISPEN **GP_INT** Secondary **PME#** Enable for secondary **ASDIN2** Slot 12 data. When set, allows **SLT12M2.GP_INT** to generate a PME event when **GP_INT** goes from 0 to a 1. See Figure 52 in the *General Purpose Input/Output Pins* section for conceptual logic.

EESPD EEPROM Serial Port Disable. When set, the EEPROM engine is disabled and does NOT try and read the EERPOM on a power-on reset. The two EEPROM pins are also disconnected from the EEPROM engine. When clear, the EEPROM engine is enabled and goes out on the EEPROM port and tries to read the EEPROM after a power-on reset.

ASDI2E **ASDIN2** Enable.
0 = **ASDIN2** function disabled (reset default) (converts to extended **GPIO1**).
1 = **ASDIN2** function enabled (implies a Secondary Codec is attached).

22.4.3 Serial Port Configuration Register 1 (SERC1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SO1F2	SO1F1	SO1F0	SO1EN

Address: BA0: 428h, Read-Only

Default: 00000003h

Definition: Lists the configuration of the primary output port for backwards compatibility only.

Bit Descriptions:

SO1EN Primary Output Port Enable. This bit is read only.
1 = Port enabled, **ASDOUT** pin always driven.

SO1F[2:0] Primary Output Port Format: This 3 bit field is read only and identifies the output data format for this port.
0 0 1 = AC ‘97 format

22.4.4 Serial Port Configuration Register 2 (SERC2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SI1F2	SI1F1	SI1F0	SI1EN

Address: BA0: 42Ch, Read-Only

Default: 00000003h

Definition: Lists the configuration of the primary input port for backwards compatibility only.

Bit Descriptions:

SI1EN Primary Input Port Enable.
1 = Port enabled, **ASDIN** is always the primary input.

SI1F[2:0] Primary Input Port Format: This 3 bit field is read only and identifies the input data format for this port.
0 0 1 = AC ‘97 format

22.5 AC ‘97 Registers

22.5.1 AC ‘97 Control Register (ACCTL)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
									TC	res	CRW	DCV	VFRM	ESYN	res

Address: BA0: 460h, Read-Write

Default: 00000000h

Definition: Control port for the AC Codec register interface. This register is reset by the PCI **RST#** signal, **SSPM.ACLEN** being clear, and the internal **dll_rdy** being clear (**CLKCR1.DLLRDY** clear).

Bit Descriptions:

ESYN Enable Sync: This bit controls **ASYNC** generation for the AC ‘97 link (the **ASYNC** pin).
0 = **ASYNC** generation disabled. **ASYNC** & **ASDOUT** outputs held low (reset default). Note **SPMC.ASYN** can independently drive the **ASYNC** output. **ESYN** just holds the serial port frame generation engine in reset. This should have the effect of automatically holding **ASYNC** and **ASDOUT** low since a frame hasn’t started.
1 = **ASYNC** generation enabled, AC ‘97 framing will start (FIFO accesses and data shifting will not start until host bypass)

VFRM Valid Frame: This bit controls the “valid frame” value in the AC ‘97 output stream (**ASDOUT** pin).
0 = Valid frame disabled (reset default)
1 = Valid frame enabled, AC ‘97 codec can interpret frame time slots

DCV Dynamic Command Valid: This bit controls dynamic command address and data generation on the first AC 97 link. To generate a valid command data slot:
1) Write the ACCAD and ACCDA registers
2) Set **TC** bit if the 2nd codec in a dual CS4298 system is to be addressed (otherwise clear it)
3) Set this bit **- DCV**
4) The serial port will dynamically set the slot valid bits for command address and data slots
5) The serial port will shift out command address and data at the appropriate time slots
6) The serial port will clear **DCV** and **TC** bits automatically, to indicate completion of the process

CRW Control Read/Write: This bit indicates type of transaction requested to the AC ‘97 codec control registers.
0 = Write command
1 = Read command

TC Target Codec: This bit specifies the target codec for dynamic command address and data generation on the AC '97 link. This is only useful in a system where two CS4298 codecs are in use. Setting this bit to 0 (the reset default) sets the codec ID bits (positions 0 and 1) in output slot 0 to 0x0, thereby addressing the primary codec; setting this bit to 1 sets the codec ID bits in output slot 0 to bits **SERMC.TCID[1:0]**, thereby addressing one of the secondary codecs. Finally, this bit is automatically reset upon completion of a dynamic command operation.
0 = Target primary CS4297/98 codec - set slot 0, bits 1:0 to 00
1 = Target a secondary CS4298 codec - set slot 0, bits 1:0 to **SERMC.TCID[1:0]**

22.5.2 AC ‘97 Status Register (ACSTS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														VSTS	CRDY

Address: BA0: 464h, Read-Only

Default: 00000000h

Definition: Status port for the first AC ’97 link (the **ASDIN** pin). This register is cleared by PCI **RST#**, the Serial Port enable bit **SSPM.ACLEN**, or when **ABITCLK** stops (**CLKCR1.DLLRDY** low).

Bit Descriptions:

CRDY Codec Ready: This bit returns the last frame’s “codec ready” indicator in the primary AC 97 input data stream on **ASDIN**.
0 = Codec not ready
1 = Codec ready

VSTS Valid Status: This set/reset bit support dynamic status capture from the primary AC ‘97 input data stream, according to the following rules:
1) A valid status data slot (slot 2 tag) in a given input frame will set this bit
2) The status address and data will be stored in the ACSAD and ACSDA registers
3) Subsequent valid status address and data slots will be ignored until the current set is read
4) Read ACSAD and ACSDA registers upon seeing this bit set
5) Read of the ACSDA register will automatically clear/reset this bit

22.5.3 AC ‘97 Output Slot Valid Register (ACOSV)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						SLV12	SLV11	SLV10	SLV9	SLV8	SLV7	SLV6	SLV5	SLV4	SLV3

Address: BA0: 468h, Read-Write

Default: 00000000h

Definition: Slot valid signals for the AC ‘97 tag phase (slot 0) on the AC ‘97 link (the **ASDOUT** pin). Except for **SLV12**, these bits are qualified with the On-Demand bits in registers AODSD1 and AODSD2, which support the variable sample rate extensions to the AC ‘97 specification, before data is sent out **ASDOUT**. If **SLV12** is set, the SLT12O register, bits 19-0, is output during Slot 12.

Bit Descriptions:

SLV[12:3] Slot Valid bits: These bits set the static slot valid bits in the AC ‘97 output data stream.
0 = Slot not valid. Corresponding output data slot forced to 0.
1 = Slot valid.

22.5.4 AC ‘97 On-Demand Slot Disable for primary link (AODSD1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							NDS11	NDS10	NDS9	NDS8	NDS7	NDS6	NDS5	NDS4	NDS3

Address: BA0: 4A8h, Read Only

Default: 00000000h

Definition: Supports the variable rate extensions to the 2.1 revision of the *AC ‘97 Codec Specification* for the primary Codec. The Codec will set these bits from **ASDIN**, slot 1, bits 3-11, to indicate that the Codec *DOES NOT* want data in the next frame. Since CS4281 uses slot 12 for GPIO control, On-Demand for slot 12 is not supported. This register’s ability to block data transmission is only enabled when **SERMC.ODSEN1** is set. The bits in this register are latched in the frame they are received and control data in the following frame.

Bit Descriptions:

NDS[11:3] No Data, Slots 11 - 3: These bits block the sending of data for output slots where the output slot valid bits are set, **ACOSV.SLV[11:3]**. AC ‘97 Codecs that support the 2.1 variable rate extension, will set these bits to block the CS4281 from sending data in the next frame (frame after these bits were received.) These bits are copied from **ASDIN**, Slot 1, where: **NDS11** is bit 3, **NDS10** is bit 4, and so on up to **NDS3** is bit 11.

0 = Send data in the next frame if corresponding **ACISV.SLV** bit is set.
1 = Do not send data in the next frame.

22.5.5 AC ‘97 On-Demand Slot Disable for secondary link (AODSD2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							NDS11	NDS10	NDS9	NDS8	NDS7	NDS6	NDS5	NDS4	NDS3

Address: BA0: 4ACh, Read Only

Default: 00000000h

Definition: Supports the variable rate extensions to the 2.1 revision of the *AC ‘97 Codec Specification* for the secondary Codec. The Codec will set these bits from **ASDIN2**, slot 1, bits 3-11, to indicate that the Codec *DOES NOT* want data in the next frame. Since CS4281 uses slot 12 for GPIO control, On-Demand for slot 12 is not supported. This register’s ability to block data transmission is only enabled when **SERMC.ODSEN2** is set. The bits in this register are latched in the frame they are received and control data in the following frame.

Bit Descriptions:

NDS[11:3] No Data, Slots 11 - 3: These bits block the sending of data for output slots where the output slot valid bits are set, **ACOSV.SLV[11:3]**. AC ‘97 Codecs that support the 2.1 variable rate extension, will set these bits to block the CS4281 from sending data in the next frame (frame after these bits were received.) These bits are copied from **ASDIN2**, Slot 1, where: **NDS11** is bit 3, **NDS10** is bit 4, and so on up to **NDS3** is bit 11.

0 = Send data in the next frame if corresponding **ACOSV.SLV** bit is set.
1 = Do not send data in the next frame.

22.5.6 AC ‘97 Command Address Register (ACCAD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									CI6	CI5	CI4	CI3	CI2	CI1	CI0

Address: BA0: 46Ch, Read-Write

Default: 00000000h

Definition: The command address field for an AC ‘97 frame on the AC ‘97 link (the **ASDOUT** pin). Note that the contents of this register will not be sent out in an output frame unless the dynamic command valid bit is set in ACCTL. The **TC** bit in ACCTL selects the target codec in dual codec configurations.

Bit Descriptions:

CI[6:0] Control Register Index: This 7 bit field addresses the 64 16-bit registers in the AC ‘97 control register address space. Normally bit 0 should always be set to 0 (even addresses only).

22.5.7 AC ‘97 Command Data Register (ACCDA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CD15	CD14	CD13	CD12	CD11	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0

Address: BA0: 470h, Read-Write

Default: 00000000h

Definition: The command data field for an AC ‘97 frame on the primary codec on the AC ‘97 link (the **ASDOUT** pin). Note that the contents of this register will not be sent out in an output frame unless the dynamic command data valid (**DCV**) bit is set in ACCTL. The **TC** bit in ACCTL selects the target codec).

Bit Descriptions:

CD[15:0] Control Register Data: This 16 bit field provides data during writes to the AC ‘97 control register address space. This field should be set to 0000h during control register reads.

22.5.8 AC '97 Input Slot Valid Register (ACISV)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						ISV12	ISV11	ISV10	ISV9	ISV8	ISV7	ISV6	ISV5	ISV4	ISV3

Address: BA0: 474h, Read-Only

Default: 00000000h

Definition: Status bits that sense the input slot valid signals for the AC '97 tag phase (slot 0) on the first AC '97 link (the **ASDIN** pin). The contents of this register are dynamic updated with each AC '97 input frame.

Bit Descriptions:

ISV[12:3] Slot Valid bits: These bits sense the slot valid bits in the AC 97 input data stream.
0 = Slot not valid
1 = Slot valid

22.5.9 AC '97 Status Address Register (ACSAD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SI6	SI5	SI4	SI3	SI2	SI1	SI0

Address: BA0: 478h, Read-Only

Default: 00000000h

Definition: The address register field for an AC '97 input frame on the first AC '97 link (the **ASDIN** pin). The contents of this register will not be overwritten by another input frame's valid status address unless the valid status bit is cleared in ACSTS by a read of ACSDA.

Bit Descriptions:

SI[6:0] Status Register Index: This 7 bit field returns the captured status address returned in slot 1 of the AC '97 input frame.

22.5.10 AC '97 Status Data Register (ACSDA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SD15	SD14	SD13	SD12	SD11	SD10	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

Address: BA0: 47Ch, Read-Only

Default: 00000000h

Definition: The register data field for an AC '97 input frame on the first AC '97 link (the **ASDIN** pin). The contents of this register will not be overwritten by another input frame's valid data unless the valid status bit is cleared in ACSTS by a read of this register.

Side Effect: When this register is read, the valid status bit in ACSTS is cleared.

Bit Descriptions:

SD[15:0] Status Data: This 16 bit field returns data of a read from the AC '97 control register.

22.5.11 Slot 12 Output Register for AC Link Codec (SLT12O)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												AGPO15	AGPO14	AGPO13	AGPO12

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AGPO11	AGPO10	AGPO9	AGPO8	AGPO7	AGPO6	AGPO5	AGPO4	AGPO3	AGPO2	AGPO1	AGPO0	VSPO2	VSPO1	VSPO0	

Address: BA0: 41Ch, Read-Write

Default: 00000000h

Definition: The AC-Link Slot 12 GPIO outputs bits. These bit positions match the slot 12 recommendation in the AC '97 2.1 Audio Codec spec. For these bits to be sent out **ASDOUT**, the slot 12 tag bit must be set, **ACOSV.SLV12**.

Bit Descriptions:

AGPO[15:0]AC Link GPIO Output bits. Bits sent to slot 12 of the AC link (**ASDOUT**).

VSPO[2:0] Vendor Specific bits. Sent to same position in Slot 12 of **ASDOUT**.

22.5.12 Slot 12 Monitor Register for Primary Codec (SLT12M)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												AGPI15	AGPI14	AGPI13	AGPI12

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AGPI11	AGPI10	AGPI9	AGPI8	AGPI7	AGPI6	AGPI5	AGPI4	AGPI3	AGPI2	AGPI1	AGPI0	VSPI2	VSPI1	VSPI0	GP_INT

Address: BA0: 45Ch, Read-Only

Default: 00000000h

Definition: Monitor port for the Primary AC Link (**ASDIN**), Slot 12 data. These bit positions match the slot 12 recommendation in the AC '97 2.1 Audio Codec spec. **ASDIN** slot 12, bits 19 through 0, are stored in this register in the same bit position (slot 12 has 19 bits and this register uses 19 bits). For slot 12 on **ASDIN** to be stored in this register, the slot 12 tag bit from the codec must have been set, **ACISV.ISV12**.

Bit Descriptions:

AGPI[15:0] AC Link GPIO bits. Bits from slot 12 of the Primary AC link (**ASDIN**).

VSPI[2:0] Vendor Specific Input bits. From same position in Slot 12.

GP_INT GPIO Interrupt - Slot 12, bit 0. Indicates that at least one of the Primary codec's GPIO pins has generated an interrupt. The rising edge of **GP_INT** is stored in **GPIOR.GPPS** and can generate a host interrupt or PME event. The interrupt is maskable through **HIMR.GPPIM** and clearable by writing a zero **GPIOR.GPPS** which clears the CS4281 interrupt. The ability to cause a **PME#** event is maskable through **SPMC.GIPPEN** and must be cleared by writing a zero to **GPIOR.GPPS**. **GP_INT** is cleared by writing to the Primary Codec's GPIO Sticky register (Index 50h), and clearing the bit causing the interrupt. See Figure 52 in the *General Purpose Input/Output Pins* section for conceptual logic.

22.5.13 *Slot 12 Monitor Register 2 for Secondary Codec (SLT12M2)*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												AGPI15	AGPI14	AGPI13	AGPI12
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AGPI11	AGPI10	AGPI9	AGPI8	AGPI7	AGPI6	AGPI5	AGPI4	AGPI3	AGPI2	AGPI1	AGPI0	VSPI2	VSPI1	VSPI0	GP_INT

Address: BA0: 4DCh, Read-Only

Default: 00000000h

Definition: Monitor port for the Secondary AC Link (**ASDIN2**), Slot 12 data. These bit positions match the slot 12 recommendation in the AC '97 2.1 Audio Codec spec. **ASDIN2** slot 12, bits 19 through 0, are stored in this register in the same bit position (slot 12 has 19 bits and this register uses 19 bits). For slot 12 on **ASDIN2** to be stored in this register, the slot 12 tag bit from the codec must have been set, **ACISV2.ISV12**.

Bit Descriptions:

AGPI[15:0] AC Link GPIO bits. Bits from slot 12 of the Secondary AC link (**ASDIN2**).

VSPI[2:0] Vendor Specific Input bits. From same position in Slot 12.

GP_INT GPIO Interrupt - Slot 12, bit 0. Indicates that at least one of the Secondary codec's GPIO pins has generated an interrupt. The rising edge of **GP_INT** is stored in **GPIOR.GPSS** and can generate a host interrupt or a PME event. The interrupt is maskable through **HIMR.GPSIM** and clearable by writing a zero **GPIOR.GPSS** which clears the CS4281 interrupt. The ability to cause a **PME#** event is maskable through **SPMC.GISPEN** and must be cleared by writing a zero to **GPIOR.GPSS**. **GP_INT** is cleared by writing to the Secondary Codec's GPIO Sticky register (Index 50h), and clearing the bit causing the interrupt. See Figure 52 in the *General Purpose Input/Output Pins* section for conceptual logic.

22.5.14 *AC '97 Status Register 2 (ACSTS2)*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														VSTS2	CRDY2

Address: BA0: 4E4h, Read-Only

Default: 00000000h

Definition: Status port for the second AC '97 link in a dual codec system (the **ASDIN2** pin). This register is cleared by PCI **RST#**, the Serial Port enable bit **SSPM.ACLEN**, or when **ABITCLK** stops (**CLKCR1.DLLRDY** low).

Bit Descriptions:

CRDY2 Codec Ready: This bit returns the last frame's "codec ready" indicator in the second AC '97 input data stream.
0 = Codec not ready
1 = Codec ready

VSTS2 Valid Status: This set/reset bit support dynamic status capture from the second AC ‘97 input data stream, according to the following rules:
1) A valid status data slot (slot 2 tag) in a given input frame will set this bit
2) The status address and data will be stored in the ACSAD2 and ACSDA2 registers
3) Subsequent valid status address and data slots will be ignored until the current set is read
4) Read ACSAD2 and ACSDA2 registers upon seeing this bit set
5) Read of the ACSDA2 register will automatically clear/reset this bit

22.5.15 AC ‘97 Input Slot Valid Register 2 (ACISV2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						ISV12	ISV11	ISV10	ISV9	ISV8	ISV7	ISV6	ISV5	ISV4	ISV3

Address: BA0: 4F4h, Read-Only

Default: 00000000h

Definition: Status register for the input slot valid signals for the AC ‘97 tag phase (slot 0) on the second AC ‘97 link (the **ASDIN2** pin) in a dual codec system. The contents of this register are dynamically updated with each AC ‘97 input frame.

Bit Descriptions:

ISV[12:3] Slot Valid bits: These bits sense the slot valid bits in the AC ‘97 input data stream.
0 = Slot not valid
1 = Slot valid

22.5.16 AC ‘97 Status Address Register 2 (ACSAD2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SI6	SI5	SI4	SI3	SI2	SI1	SI0

Address: BA0: 4F8h, Read-Only

Default: 00000000h

Definition: The address register field for an AC 97 input frame on the second AC 97 link (the **ASDIN2** pin) in a dual codec system. The contents of this register will not be overwritten by another input frame’s valid address unless the valid status bit is cleared in ACSTS2 by a read of ACSDA2.

Bit Descriptions:

SI[6:0] Status Register Index: This 7 bit field returns the captured status address returned in slot 1 of the AC 97 input frame.

22.5.17 AC ‘97 Status Data Register 2 (ACSDA2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SD15	SD14	SD13	SD12	SD11	SD10	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

Address: BA0: 4FCh, Read-Only

Default: 00000000h

Definition: Data register field for an AC ‘97 input frame on the second AC ‘97 link (the **ASDIN2** pin) in a dual codec system.The contents of this register will not be overwritten by another input frame’s valid data unless the valid status bit is cleared in ACSTS2 by a read of this register.

Side Effect: When this register is read, the valid status bit in ACSTS2 is cleared.

Bit Descriptions:

SD[15:0] Status Data: This 16 bit field returns data of a read from the AC ‘97 control register.

23. TEST

The Testability Block contains all the normal test functions and the special purpose test features. The test functions are separated into two categories - those that can be run in the system are the Normal Test Functions, and those that must be in a special test fixture or evaluation board are Special Test Functions. In addition to these functions there are several observation registers in the PCI space that are used to observe the behavior of internal state machines.

With the elimination of the internal 8052, the pin remapping that was used is not needed. The only pin remapping needed is to support the scan testing and clock multiplier bypass/monitor testing. The joystick button and coordinate pins will be used for the scan test. The **VOLUP** and **VOLDN** pins will be used for the clock multiplier testing. The joystick button and **MIDIIN** pins select the test mode when using hardware test mode entry.

23.1 Normal Test Modes

- RAM/ROM test
- MIDI loopback - MIDI out - MIDI in
- SRC loopback - PCI - FIFO - SRC - FIFO - PCI
- Primary AC Link Loopback - PCI - FIFO - ASDOUT - ASDIN - FIFO - PCI
- Secondary AC Link Loopback - PCI - FIFO - ASDOUT - ASDIN2 - FIFO - PCI
- Joystick comparator test
- Fast Count

Special Test Modes

- Force all outputs low
- Force all outputs high
- Force all outputs float
- Input XOR tree
- Bypass clock multiplier, input 512× fs clock
- DLL monitor
- Scan Test

23.2 Boundary Test Mode

The Boundary Test Mode is a the same as the Special Test Mode 4, Input XOR Tree Test. A simplified entry procedure is provided. The Boundary Test Mode is entered by asserting the **TEST** pin high with the **TESTSEL** pin high. The part will remain in the Boundary Test Mode until the **TEST** pin is deasserted. See the Input XOR Tree description for the specific functional behavior.

23.3 Normal Test mode Entry

Normal Test Modes are entered by software or hardware mechanisms. Software test selection is setup by writing the desired mode selection to the Test Select Register, PCI Configuration F8h and setting the **IISR.TMSE** bit. Hardware test selection is setup by the code on the Joystick Button pins and the **MIDIIN** pin, with the Test Select Register zero. The test is entered by either asserting the **TEST** pin or writing the Test Enable Bit Pattern. The test mode is exited by writing zeros to the Test Select Register, deasserting the **TEST** pin, or by changing the Test Enable Bit Pattern. The Test Enable Bit Pattern is implemented by a set of reserved bits spread among two different registers. The intent is that the test condition cannot be entered accidently, and be very difficult to hack. The Test Select Register will be reserved also. Test Select value of zero is a no test condition, normal operation.

23.4 Special Test Mode Entry

Special Test Modes should only be entered through hardware, since they will operate only in a special fixture, or on an isolated bus. The desired test function is selected by the code applied to the Joystick Button pins, the **TESTSEL** pin is held low, and then the **TEST** pin is asserted high. The test mode is exited by deasserting the **TEST** pin. A value of all zeros on the test select pins is a no test condition, normal operation. The table below defines the pin states for hardware test mode entry.

23.5 Normal Test Mode Description

0) No test

Null test mode, all functions operate normally.

2) RAM/ROM test

All internal RAM and ROM block are mapped into the PCI address space with their normal word widths. Base address register BA1 specifies the starting location to avoid existing system allocations. The SRC sample RAM is not included in this function as it operates on a different clock domain, and is easily tested using the SRC loopback test. When this test is enabled, normal functionality for the RAM/ROM under test is disabled.

3) MIDI loopback

The MIDI serial output is internally looped back to the MIDI input buffer. **MIDIOUT** is still active in its normal operation. The **MIDIIN** input signal is ignored.

5) SRC loopback

PCI - FIFO - SRC - FIFO - PCI. The playback SRC output is looped back to the capture SRC input. This test mode internally sets the **SSCR.LPSRC** bit. All other registers associated with the stream must be setup properly for this test mode to work.

7) Primary AC-Link Loopback

PCI - FIFO - ASDOUT - ASDIN - FIFO - PCI. The **ASDOUT** signal keeps its normal activity and is internally muxed to replace the **ASDIN** input. The **ASDIN** signal is ignored. This test mode internally sets the **SERMC.PLB** bit. All other registers associated with the stream must be setup properly for this test mode to work.

MIDIN	JBB2	JBB1	JAB2	JAB1	Test Mode
0	0	0	0	0	No test mode
0	0	0	0	1	No test mode
0	0	0	1	0	RAM/ROM test
0	0	0	1	1	MIDI Loopback test
0	0	1	0	0	No test mode
0	0	1	0	1	SRC Loopback
0	0	1	1	0	No test mode
0	0	1	1	1	Primary AC-Link Loopback
0	1	0	0	0	Secondary AC-Link Loopback
0	1	0	0	1	Joystick Comparator Test
0	1	0	1	0	Fast Count
0	1	0	1	1	No test mode
1	0	0	0	0	No test mode
1	0	0	0	1	Force All Outputs Low
1	0	0	1	0	Force All Outputs High
1	0	0	1	1	Force All outputs Float
1	0	1	0	0	Input Level XOR Tree
1	0	1	0	1	Bypass Clock Multiplier
1	0	1	1	0	DLL Monitor Test
1	0	1	1	1	Scan Test

Table 38. Special Test Modes

8) Secondary AC-Link Loopback

PCI - FIFO - ASDOUT - ASDIN2 - FIFO - PCI. The **ASDOUT** signal keeps its normal activity and is internally muxed to replace the **ASDIN2** input. The **ASDIN2** signal is ignored. This test mode internally sets the **SERMC.SLB** bit. All other registers associated with the stream must be setup properly for this test mode to work.

9) Joystick Comparator test

Map the joystick comparator outputs directly to the Joystick Poll/Trigger Register (JSPT) so that the comparator trip levels can be directly measured. The discharge transistors are released and the latches bypassed.

10) Fast Count

Changes the 10 ms Time Base from using the Fs clock to using **ABITCLK** to generate the clock. This change decreases the 10 ms clock 256 times changing the 10 ms rate to approx. 39 µs.

23.6 Special Test Mode Description

1) Force all outputs low

All pins that have a pad driver attached will be driven low. This test is used for measuring Vol.

2) Force all outputs high

All pins that have a pad driver attached will be driven high. This test is used for measuring Voh.

3) Force all outputs float

All pins that have a pad driver attached will float. Any pins with an internal pullup or pulldown will have that pullup or pulldown disconnected so that the real input leakage can be measured.

4) Input level XOR tree

All pins with an input buffer have their input buffer outputs connected in a XOR tree. Most device pins are used except **TEST**, input pins and bidirectional pins are used as inputs to the XOR tree and output pins are used to observe the XOR result. This test mode allows independent measurement of all input buffer thresholds. The pin ordering is not important since there is only one tree. The XOR tree is output on five pins that are exclusively outputs: EECLK/GPOUT/PCREQ#, MIDIOUT, ARST#, ASYNC, and ASDOUT. The TEST and AD[0] pins start the XOR tree and then get XOR'd with AD[1] and so on as listed in Table 39. The final XOR pin is JAB1 and the XOR output appears on the five pins listed above.

Pin r	Pin r+1	Pin r+2	Pin r+3	Pin r+4
*TEST	AD[0]	AD[1]	AD[2]	AD[3]
AD[4]	AD[5]	AD[6]	AD[7]	C/BE[0]#
AD[8]	AD[9]	AD[10]	AD[11]	AD[12]
AD[13]	AD[14]	AD[15]	C/BE[1]#	PAR
PERR#	STOP#	DEVSEL#	TRDY#	IRDY#
FRAME#	C/BE[2]#	AD[16]	AD[17]	AD[18]
AD[19]	AD[20]	AD[21]	AD[22]	AD[23]
IDSEL	C/BE[3]#	AD[24]	AD[25]	AD[26]
AD[27]	AD[28]	AD[29]	AD[30]	AD[31]
GNT#	PCICLK	RST#	GPIO3	EEDAT/GPIO2/PCGNT#
ASDIN	ABITCLK	VOLDN	VOLUP	CLKRUN#
ASDIN2/ GPIO1	TESTSEL	MIDIIN	JBB2	JBB1
JAB2	JAB1			

* Even though TEST is an input, setting TEST low removes the CS4281 from this test mode.

Table 39. Special Test Mode 4: XOR Tree Inputs

A number of outputs were not included due to excess control-logic needed. The INTA-C output pins are not included due to high-impedence controls. The REQ#, INTA#, and PME# aren't included due to being open-drain. The JACX, JACY, JBCX, and JBCY pins were not included due to the analog nature of the inputs and the clamp transistors.

5) Bypass clock multiplier, input 512×fs clock

The internal 2× clock multiplier will be bypassed. The clock on the **VOLUP** pin will be used as a 512×Fs clock. All other function operate normally. This is used to synchronously drive the core clock when running full speed vectors.

6) DLL monitor

Drive the 4x master clock from the clock multiplier out to the **VOLDN** pin. All other functions operate normally. This is used to test the DLL lock range and duty cycle.

7) Scan Test

Enable scan testing. If in Scan Test mode, bringing JAB1 high sets the Scan Active, and JACX goes high as an indication that Scan is active. Three independent scan chains in the CS4281 support each of the three main internal clocks: **pci_clk**, **clk256** (**abitclk**), and **clk512**. The other joystick button pins are the scan input signals and the other joystick coordinate pins are the scan output signals. The **PCICLK** pin is the scan clock for the **pci_clk** scan chain which contains all the PCI blocks. The **ABITCLK** pin is the scan clock for the **clk256** scan chain which contains the FM and Peripheral Interface blocks. The **VOLUP** pin is the scan clock for the **clk512** scan chain which contains the SRC and Sound System blocks. While in scan test mode the DLL is bypassed and external clock pins are connected to the appropriate internal clocks as mentioned above. The table below defines the pin usage for the Scan Test Mode.

23.7 Test Observation Registers

Two of the three Test Observation Registers are planned: one for observing the Sound Blaster state machine, and another for observing the AC Link state machine.

Name	Description	Direction
JACX	Scan Active Indication	OUT
JACY	**pci_clk** Scan Data Output	OUT
JBCX	**clk256** Scan Data Output	OUT
JBCY	**clk512** Scan Data Output	OUT
JAB1	Scan Enable	IN - PU
JAB2	**pci_clk** Scan Data Input	IN - PU
JBB1	**clk256** Scan Data Input	IN - PU
JBB2	**clk512** Scan Data Input	IN - PU
ABITCLK	**clk256** External Scan Clock	IN
PCICLK	**pci_clk** External Scan Clock	IN
VOLUP	**clk512** External Scan Clock	IN

Table 40. Scan Test Mode Pin Usage

23.7.1 SB Observation Register 0 (OR0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LS6	LS5	LS4	LS3	LS2	LS1	LS0	ARG	AS4	AS3	AS2	AS1	AS0	DEAD	HISPD	AUTO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLAY	CAPT	DMA	POLL	DRU	FEN	TMR	PIOR	UARTM	DC2	DC1	DC0	RQF	RQE	AMR	AMG

Address: BA0: 7F0h, For Observation Register 0

Default: 00000000h

Definition: Core powered. A read-only port for observing internal state machine operation and critical signal operation for the Sound Blaster compatible engine.

LS[6:0] Last State bits of state machine.

ARG Argument from state machine.

AS[4:0] Argument State from state machine.

DEAD Unresponsive commands.

HISPD High Speed unresponsive commands.

AUTO Auto-initialize

PLAY Playback direction.

CAPT Capture direction.

DMA DMA mode.

POLL Polled FIFO mode.

DRU DMA Request Ungated. Gate signal (from FIFO) to stop/start DMA engine.

FEN FIFO Enable.

TMR Table Munge DMA Request.

PIOR PIO DMA Request.

UARTM UART Mode.

DC[2:0] DMA Count LSBs.

RQF Read Queue Full.

RQE Read Queue Empty.

AMR AC-Link Mixer Request. Request is pending to transfer data from Sound Blaster to Serial port.

AMG AC-Link Mixer Grant. Grant is pending from AC-Link serial port.

23.7.2 Observation Register 1/2 (OR1/2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Address: BA0: 7F4h, For Observation Register 1
BA0: 7F8h, For Observation Register 2

Default: 00000000h

Definition: Core powered. These registers provide a read-only port for observing internal state machine operation and critical signal operation. Currently undefined/unused.

Bit Descriptions:

A[31:0] Observation state signals. See separate table.

23.7.3 Test Mode Select Register (TMS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSM7	TSM6	TSM5	TSM4	TSM3	TSM2	TSM1	TSM0				TS4	TS3	TS2	TS1	TS0

Address: BA0: 3F8h, Read-Write
PCI CFG: 0F8h, Read-Write if CWPR configured, otherwise Read-Only

Default: 00000000h

Definition: Vaux powered. This register selects test mode entered from software test entry. This register must be zero for hardware test mode entry. The **TSM** field must match 81h to enable software test entry. The proper procedure is to write the **TS** bits to the desired test mode, then write the **TSM** field, then set the **IISR.TSME** bit to activate the software test mode. If written through configuration space, writing config. space requires setting the CWPR register correctly. **The Special Test Modes (TS[4] = 1) require a special fixture and won't operate on a standard PCI audio card.** Setting a Special Test Mode in a card that doesn't support that test mdoe could damage the device. The TMS register is reset by the PCI **RST#** signal and the Vaux POR signal.

Bit Descriptions:

TS[4:0] Test Mode Select code.

00000	No test mode
00001	No test mode
00010	RAM/ROM Enable
00011	MIDI Loopback test
00100	No test mode
00101	SRC Loopback
00110	No test mode
00111	Primary AC-Link Loopback
01000	Secondary AC-Link Loopback
01001	Joystick Comparator Test
01010	Fast Count Test
01011	No test mode
10000	No test mode
10001	Force All Outputs Low
10010	Force All Outputs High
10011	Force All outputs Float
10100	Input Level XOR Tree
10101	Bypass Clock Multiplier
10110	DLL Monitor Test
10111	Scan Test

TSM[7:0] Test Mode Match. The **TMS** bits must match 81h to enable the **TS** test mode bits. In addition, the **IISR.TSME** bit must be set.