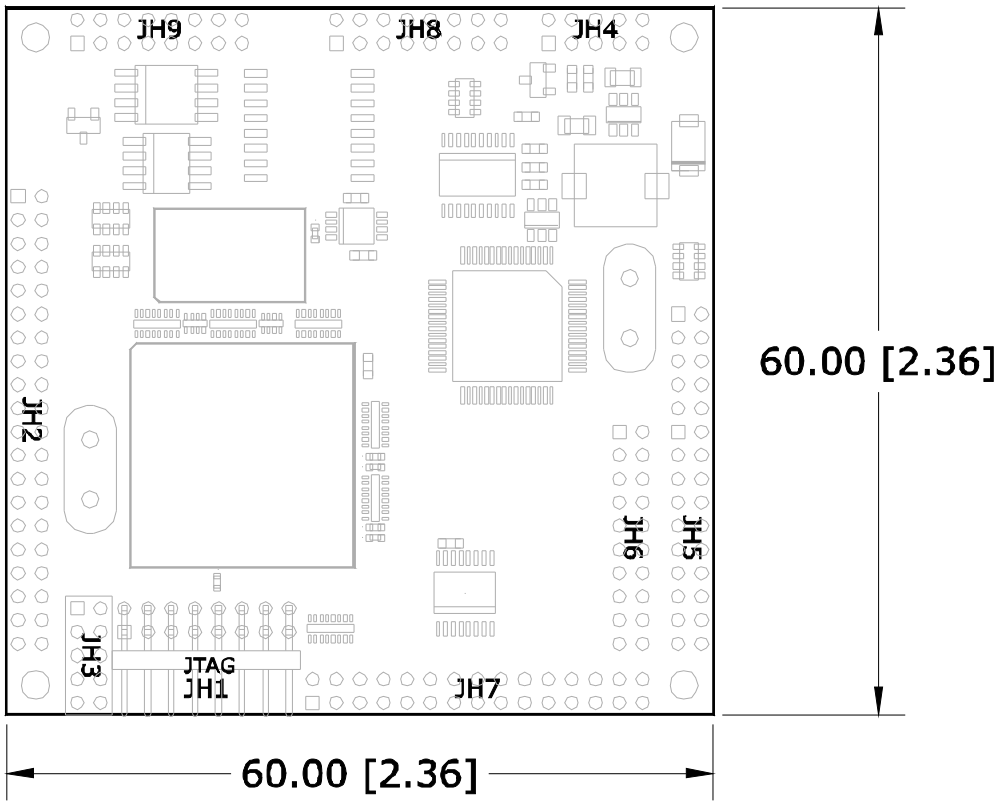
Danville Signal Processing, Inc.

dspblok™
Designing for Compatibility



Version 3.1

Danville Signal Processing, Inc.
dspblok™ Designing for Compatibility

Danville Signal Processing, Inc. strives to deliver the best product to our customers. As part of this goal, we are constantly trying to improve our products. Danville Signal Processing, Inc., therefore, reserves the right to make changes to product specification or documentation without prior notice. Updated operating manuals and product specification sheets are available at our website for downloading.

This manual may contain errors, omissions or "typo's". Please send your comments, suggestions and corrections to:

Danville Signal Processing, Inc.
29687 82nd Avenue Way
Cannon Falls, MN 55009

Trademark Notice

dspblok, dspstak and dspBootloader are trademarks of Danville Signal Processing, Inc. VisualDSP++, SHARC and Blackfin are trademarks of Analog Devices, Inc.

Contact Information

Danville Signal Processing, Inc.
29687 82nd Avenue Way
Cannon Falls, MN 55009

E-mail: dsp@danvillesignal.com
Web Site: http://www.danvillesignal.com

Voice: 507-263-5854

Table of Contents

Overview

Danville Signal's dspblok™ family of products delivers the power of digital signal processing in a small 60mm x 60mm form factor. Connections are brought out to standard 2mm dual row headers. The dspblok reduces development costs, risk and time.

Danville's dspblok DSP function modules are often incorporated directly into larger custom embedded systems. By taking advantage of pretested signal processing modules, pc board layouts become simpler and projects are completed quickly and cost effectively.

Danville's dspblok DSP Engines are largely pin compatible. This allows your application to take advantages of new processor technology and extended features in the future. The purpose of this document is to help you integrate dspblok DSP or FPGA modules into your design or adapt an earlier dspblok based design to use a newer dspblok module.

dspblok™ Development Boards

Many dspblok modules had a companion developer's version, which included an Analog Devices' EZ-KIT style debugger. These boards were physically larger (60mm x 115mm) to accommodate the debugger but have a matching footprint to our production modules (60mm x 60mm).

Since the latest ADI tools are substantially less expensive than Visual DSP++ and the earlier emulators, Danville no longer produces these boards. The ICE-1000 functions similarly to the "with ICE" dspblok debug agents and will work with any of our production targets. The ICE-2000 is the faster alternative.

dspblok™ History

Danville has manufactured dspblok DSP Modules since 2002. The first dspblok module was based on the ADSP-21161 2nd generation SHARC. The primary motivation for this product was to facilitate quick design of new products by reusing the standard DSP functions without the design burden of creating a DSP Core that tended to be the same from project to project. It's usually the I/O requirements that drive the diversity in most products.

In the early days of the dspblok, BGA assembly was often an obstacle to board manufacturing, especially for small production runs. Creating a function module that relied on 2mm headers greatly simplified pc board layout and assembly without sacrificing valuable pc board real estate. For many users, this decoupled complex manufacturing for less complicated hand or automated assembly.

There are other advantages to physically decoupling the DSP functions from a larger system. The DSP (or other similar core processor functions) generally require more pcb layers and higher precision pcb fabrication than typical I/O and power supply circuits. This rapidly contributes to pc board fabrication costs since the whole board must accommodate the demands of all the circuits. Low noise data converter and signal conditioning circuits are also much easier to isolate when high speed digital switching resides off board.

In many respects, the value of dspblok modules hasn't changed at all. Certainly BGA assembly is somewhat more routine today than in the early 2000s, but circuit board layout and design have become increasingly complex as devices are operating at much higher clocks than the earlier processors and memory. Careful attention to signal integrity, crosstalk, trace length matching and other design issues are absolute requirements for modern digital design. These issues were far less of a concern when the DSP only operated at 100MHz.

With the introduction of the 3rd generation SHARC, the dspblok became much more flexible. The first member of this generation was the ADSP-21262. This device featured a crossbar switch that abstracted 20 pins (DAI) from SPORTs and other peripherals. This meant that any of these pins could be routed to a peripheral in an arbitrary manner via software in a similar manner that CPLDs and FPGAs are configured.

This idea was further extended with the introduction of the ADSP-21369. In addition to the original 20 pin DAI, microcontroller peripherals such as UARTs, SPI, I2C and timers were added and abstracted with a similar crossbar interface called DPI. The ADSP-21369 also added support for SDRAM.

The dspblok 21469+USB, based on the ADSP-21469 is foundation dspblok that this document uses as the root of dspblok compatibility. The dspblok 21369zx is largely compatible whereas earlier generation dspblok DSP modules are loosely compatible. If you are migrating a design from an early dspblok (21161, 21261 or 21262), consult Danville for specific suggestions in this regard.

With the advent of the DAI & DPI, SHARC processors, FPGAs and CPLDs all have pins that can be reconfigured via software. This is key to building compatible targets.

dspblok Versions

The following sections outline the basic features of current dspblok DSP modules. It is certainly possible that some of the definitions and therefore some of the pin definitions could change in future dspblok products. We are providing this information on the assumption that planning from a probable roadmap is better than not knowing the plan in the first place.

In general, information about a specific dspblok model will be identical for both the production (60 x 60mm) version as well as the legacy companion developer (60 x 115mm, ICE) version. Earlier dspblok "with ice" versions were larger (60 x 140mm). All dspbloks are supported with external emulators. You should review specific product user manuals for detailed information about each module.

Early dspbloks including non SHARC based versions

Danville has produced dspbloks using early 1st and 2nd generation SHARCs as well as a Blackfin version. These are not very compatible and are not discussed in this document. They are no longer produced by Danville except by special order. If you have a need to migrate a design from one of these targets, please contact us and we can look at your specific case in detail.

3rd Generation

The 3rd generation dspbloks are all based on the ADSP-21369 SHARC. The most popular product in this group is the dspblok 21369zx. The dspblok 21369cy25 and dspblok 21369cy05 combine an Altera FPGA and a ADSP-21369. The 3rd generation dspbloks are now considered "legacy" at Danville. This means that we only make them via special order with typical minimum order quantities of 25.

3rd generation dspbloks are candidates for migration to newer 4th generation targets. The newer dspbloks offer either lower cost or higher performance and often both. There is a later section in this manual that discusses migration from 3rd generation to 4th generation dspbloks.

The largest distinction between 3rd generation and 4th generation dspbloks is that the ADSP-21369 supported a 32 bit wide parallel data bus whereas the 4th generation SHARCs support either 8 bit, 16 bit or no external data bus at all. Since the SDRAM was on-board in the 3rd generation dspblok, very few designs actually took advantage of the wider external data bus. All 4th generation dspbloks support either an 8 bit data bus or omit the external parallel data bus altogether.

4th Generation

The 4th generation dspbloks are all based on 4th generation SHARCs (ADSP-214xx). The 4th generation dspbloks support two new features that were not part of the original dspblok family: on-board USB and our third generation dspBootloader with on-board bootloader mode pins. This requires the addition of JH8, a 2mm dual row header located adjacent to JH4 on earlier dspbloks. Ethernet is also supported on a few

dspbloks. This adds header JH9 located next to JH8. If your design does not need Ethernet, then the only concern you may have would be for connector clearance of JH9 on a mating board.

4th Generation SHARC/XMOS (USB Audio Class 2)

This new generation of dspbloks combines a 4th generation SHARC with a companion XMOS processor. The XMOS processor is used to support USB Audio Class 2 applications. This means that the USB port can stream audio data in addition to typical control or bulk data transfers that are supported in the general dspblok family.

Since the USB is now transferring audio data, some of the SHARC DAI ports needed to be redeployed to the XMOS processor and are no longer available externally. Inner communication between the XMOS processor and the SHARC DSP also restricts some of the DPI assignments.

If a design, might support this subclass of dspbloks, then pin assignments will need to be much more restrictive than the general purpose family of dspbloks. Mechanically, all dspbloks are still largely the same. This means that it is possible to design a board that supports both the general purpose and audio class 2 dspbloks. In this case, the general purpose dspbloks will use the same USB connections but will not be able to stream USB audio.

4th Generation with Ethernet interfaces

Some dspbloks can support Ethernet. These dspbloks use an additional header (JH9). This connector is located adjacent to JH8. The Ethernet transceiver and PHY reside on the dspblok. You add magnetics and a connector externally (usually in a composite RJ-45 connector).

Other dspbloks in the future

In addition to the obvious new dspblok designs based on future SHARC DSPs , there are other candidates. These are often driven by customer needs and inputs. This document will be continually updated as the “future” products are released and additional products are added to our roadmap.

dspblok Specific Products

dspblok 21469 & dspblok 21469+USB

Type: 4th Generation
P/N A.08469D & P/N A.08469D-USB

The dspblok™ 21469 is a highly integrated DSP module that incorporates an Analog Devices' ADSP-21469 SHARC DSP operating at 450 MHz with flash, EEProm and 1Gb DDR2 DRAM memory. The on-board switching power supply is powered by 3.3 to 5V. The I/O power supply is 3.3V. The ADSP-21469 peripherals include SPORTS (8), SPI(2), TWI (I2C), UART, timers, PWMs, JTAG, an 8 bit wide data bus and LINK ports(2). JH6 is used to support the 21469 link ports. Most designs do not take advantage of link ports .

Earlier revisions of the dspblok 21469 did not include JH8. JH8 is now universally used in all 4th generation dspbloks to support the dspBootloader™

The dspblok™ 21469+USB is identical to the dspblok 21469 with the addition of an FTDI FT2232H high speed USB 2.0 transceiver. The USB transceiver and Boot Mode register is mapped to the upper memory of MS1#

dspblok 21489

Type: 4th Generation
P/N A.08489C

The dspblok™ 21489 is a highly integrated DSP module that incorporates an Analog Devices' ADSP-21489 SHARC DSP operating at 400 MHz with flash, EEProm and high speed USB 2.0. The on-board switching power supply is powered by 3.3 to 5V. The I/O power supply is 3.3V. Since the ADSP-21489 has 5Mbits of internal RAM, the dspblok 21489 does not include external SDRAM. From our experience, we have determined that most applications do not benefit from external DRAM. When applications do need lots of external memory, the dspblok 21469 is much better suited since it supports much faster DDR2 SDRAM. The ADSP-21489 peripherals include SPORTS (8), SPI(2), TWI (I2C), UART, timers, PWMs and JTAG.

dspblok 21489+Ethernet

Type: 4th Generation with Ethernet
P/N A.08489C-ETH

The dspblok™ 21489+Ethernet is identical to the dspblok 21489 with the addition of a 100BaseT ethernet transceiver. This requires the addition of header JH9.

dspblok 21479

Type: 4th Generation
P/N A.08479A

The dspblok™ 21479 is a highly integrated DSP module that incorporates an Analog Devices' ADSP-21479 SHARC DSP operating at 250 MHz with flash, EEProm and full speed USB 2.0. It does not support an external parallel data bus so USB & bootloader modes are transferred via SPI communications.
The dspblok 21479 benefits from the low power consumption of the ADSP-2147x family.

The low power consumption opens the door to applications that can be USB bus powered. This is achieved by an extension to the JH4 power connections to include a power enable pin. This allows the USB to be enumerated and then request 500mA. An external FET switch can then power the remaining circuits in the dspblok as well as other external circuits.

dspblok 21369zx

Type: 3rd Generation
P/N A.08369B

The dspblok 21369zx is a popular foundation dspblok. Existing designs can usually be accommodated for either lower cost or better performance. The biggest compatibility difference between ADSP-21369 based designs and later devices is that the ADSP-21369 has a 32 bit external data bus. This was primarily needed to support wide SDRAM. Very few external I/O devices ever required the full 32 bits. It fact, Danville has never created a companion board that took advantage of all 32 bits. A few designs such as our dspstak 21369zx2, required 16 bits for the PLX NET2272 USB transceiver. The dspstak 21369zx2 cannot be supported by the 4th generation dspbloks. There is a new dspstak 4th generation baseboard that supports all the basic 4th generation dspbloks including support for high speed USB.

dspblok 21369cy25

Type: 3rd Generation
P/N A.11369-25

The dspblok 21369+fpga combines an Altera Cyclone III FPGA with an ADSP-21369.
The FPGA connects to the DSP via the DSP's 32-bit wide data bus.

dspblok 21479u8a

Type: 4th Generation SHARC/XMOS USB Audio Class 2
P/N A.07479A

The dspblok™ 21479u8a is a highly integrated DSP module that incorporates an Analog Devices' ADSP-21479 SHARC DSP operating at 250 MHz and an XMOS XS1-U8A MCU with flash, EEProm and high speed

USB 2.0. The XMOS MCU supports USB Audio Class 2 primarily for stereo applications. S/PDIF or AES3 applications can operate as fast as 96k. This is a function of the DSP core clock.

It does not support an external parallel data bus so USB & bootloader modes are transferred via SPI communications. The dspblok 21479u8a benefits from the low power consumption of the ADSP-2147x family.

The low power consumption opens the door to applications that can be USB bus powered. This is achieved by an extension to the JH4 power connections to include a power enable pin. This allows the USB to be enumerated and then request 500mA. An external FET switch can then power the remaining circuits in the dspblok as well as other external circuits.

dspblok 21489uac2

Type: 4th Generation SHARC/XMOS USB Audio Class 2
P/N A.14489

The dspblok™ 21489uac2 is a highly integrated DSP module that incorporates an Analog Devices' ADSP-21489 SHARC DSP operating at 400 MHz and an XMOS XU208 MCU with flash and EEProm memory. The XMOS MCU supports USB Audio Class 2 to sample rates of up to 384k. S/PDIF or AES3 applications can operate as fast as 192k. This is a function of the DSP core clock.

The dspblok 21489uac2 has eliminated the Wolfson WM8805 S/PDIF/AES3 transceiver used in the dspblok 21479u8a since this capability is already supported internally in the SHARC. This frees up more DAI pins.

It does not support an external parallel data bus.

dspblok 21469uac2

Type: 4th Generation SHARC/XMOS USB Audio Class 2
P/N A.14469

The dspblok™ 21469uac2 is a highly integrated DSP module that incorporates an Analog Devices' ADSP-21469 SHARC DSP operating at 450 MHz and an XMOS XU208 MCU with flash, EEProm and 1Gb DDR2 DRAM memory. The XMOS MCU supports USB Audio Class 2 to sample rates of up to 384k. S/PDIF or AES3 applications can operate as fast as 192k. This is a function of the DSP core clock.

The dspblok 21469uac2 has eliminated the Wolfson WM8805 S/PDIF/AES3 transceiver used in the dspblok 21479u8a since this capability is already supported internally in the SHARC. This frees up more DAI pins.

The faster processing speed and large DDR2 memory make it suitable for larger designs and large FIR calculations.

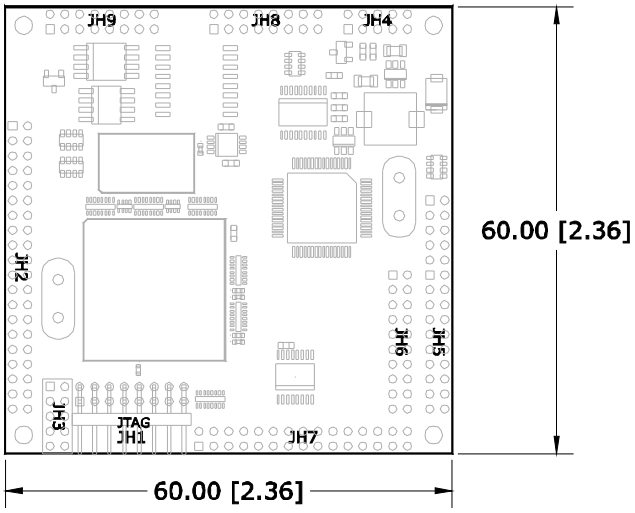
It does not support an external parallel data bus.

Universal dspblok features

Every production dspblok board is 60mm x 60mm. Male 2mm dual row headers are located on the bottom side of the pcb. The top side of the pcb includes a JTAG header and sometimes configuration headers.

Many production dspblok boards were supported by a companion development ("with ICE") dspblok version. These modules were larger (60mm x 115mm) to accommodate the Analog Devices' Debug Agent. Earlier versions of the dspblok "with ICE" were 60mm x 140mm. The debugger portion essentially replaced the JTAG connections and extended out from the JTAG connector side. Since lower cost tools such as the ICE-1000 are available, Danville no longer manufactures these products.

The following table shows the locations of each header. Most dspbloks do not include all of these headers, however you may want to insure that components on a mating board avoid the areas if you want to insure maximum compatibility. The origin (0,0) is the lower left corner of the board. All dimensions are in mm.



Header	Pin 1
JH2	1,44
JH4	46,57
JH5	57,24
JH6	52,24
JH7	26,1
JH8	28,57
JH9	6,57

Each header supports a group of common functions. These functions are often identical on all dspbloks but can differ in small detail as extended features have been added or modified for greater flexibility. From a compatibility point of view, generally the bottom headers are the only connections that need to be addressed. The next sections detail the pin connections as viewed from a header perspective.

JH1 - JTAG

- Location – Top Side
- Right Angle Dual Row – 16 Pin
- PCB Compatibility Issues - None

JTAG connections are available for connection to external emulators and programmers. JTAG Adapters are available to convert the small 2mm connection to the larger connectors use by the external device.
There are currently two different pin definitions for JH1. Two different pins are used as keys to prevent the wrong JTAG adapter from being inserted. Early dspbloks did not have the pin cut so this should be checked.

If JH1 is only required to support an Analog Devices' DSP, the pin outs are identical to the ADI JTAG 14 pin (100mil dual row header) footprint except for additional pins 15 & 16. These pins carry 3.3V that could be used to power buffers and are usually just used to power a LED on the JTAG adapter. The JTAG adapter is P/N A.08153.

If JH1 is connected to multiple devices, typically a SHARC DSP and FPGA or CPLD, then the pin assignments are reassigned to support two independent JTAG chains. This allows the board to be connected simultaneously to two independent emulators or programmers. For example, you could run VisualDSP++ or CCES with an emulator at the same time you were connected to Altera's Quartus via a USB Blaster. The JTAG adapter is P/N A.08154.

JH2 – DAI, DPI, I/O & SPI

- Location – Bottom Side
- Straight Dual Row - 40 Pin
- PCB Compatibility Issues - Minimal

JH2 is primarily a third (and fourth) generation SHARC expansion connection for DAI (Digital Applications Interface) and DPI (Digital Peripheral Interface). These connections are supplemented by RESET#, Flags and GND connections.

There are two fundamental dspblok types: Standard and USB Audio Class 2. The following table outlines the pin assignments.

On the standard dspblok, there are 12 of 14 DPI pins and all 20 DAI pins. FPGA connections can use any of the pins arbitrarily with the exception of GND, pins 1, 39 & 40 and RESET#, pin 17.

Since there are internal connections needed for control and streaming on the USB Audio Class 2 dspbloks, some of the DAI and DPI ports have been reclaimed. SS ports are specifically used for SPI expansion. OUT ports are output only and several DAI lines have specific uses that are more restrictive than the standard DAI.

Pins 13 & 18 are now RESERVED, but are usually connected to GND on most boards. This means that these connections are now limited to digital inputs, since driving an external GND would not be a good idea.

JH2 Pin	Default		21369zx	21369 fpga	21469 USB	21489	21479	21479 u8a	21469/21489 uac2
1	**GND**		**GND**	**GND**	**GND**	**GND**	**GND**	**GND**	**GND**
2	DPI 8		DPI 8	DPI 8	DPI 8	DPI 8	DPI 8	DPI 8	DPI 8
3	DPI 4		DPI 4	DPI 4	DPI 4	DPI 4	DPI 4	DPI 4	DPI 4
4	DPI 13		DPI 13	DPI 13	DPI 13	DPI 13	DPI 13	SS0#	SS0#
5	DPI 14		DPI 14	DPI 14	DPI 14	DPI 14	DPI 14	SS1#	SS1#
6	FLG 0		FLG 0	FLG 0	FLG 0	FLG 0	FLG 0	FLG 0	FLG 0
7	DPI 7		DPI 7	DPI 7	DPI 7	DPI 7	DPI 7	DPI 7	DPI 7
8	DPI 11		DPI 11	DPI 11	DPI 11	DPI 11	DPI 11	SS2#	SS2#
9	DPI 12		DPI 12	DPI 12	DPI 12	DPI 12	DPI 12	SS3#	SS3#
10	DPI 9		DPI 9	DPI 9	DPI 9	DPI 9	DPI 9	DPI 9	DPI 9
11	DPI 10		DPI 10	DPI 10	DPI 10	DPI 10	DPI 10	DPI 10	DPI 10
12	FLG 1		FLG 1	FLG 1	FLG 1	FLG 1	FLG 1	FLG 1	FLG 1
13	Reserved		Reserved	Reserved	FLG 3	FLG 3	FLG 3	Reserved	OUT2
14	**MOSI**		**MOSI**	**MOSI**	**MOSI**	**MOSI**	**MOSI**	**MOSI**	**MOSI**
15	**SCK**		**SCK**	**SCK**	**SCK**	**SCK**	**SCK**	**SCK**	**SCK**
16	**MISO**		**MISO**	**MISO**	**MISO**	**MISO**	**MISO**	**MISO**	**MISO**
17	**RESET#**		**RESET#**	**RESET#**	**RESET#**	**RESET#**	**RESET#**	**RESET#**	**RESET#**
18	Reserved		GND	GND	GND	Reserved	Reserved	Reserved	Reserved
19	DAI 1		DAI 1	DAI 1*	DAI 1	DAI 1	DAI 1	MCLK_USB	MCLK_OUT
20	DAI 2		DAI 2	DAI 2*	DAI 2	DAI 2	DAI 2	OUT0	OUT0
21	DAI 3		DAI 3	DAI 3*	DAI 3	DAI 3	DAI 3	RESET_EXT#	OUT1
22	DAI 4		DAI 4	DAI 4*	DAI 4	DAI 4	DAI 4	PS_SYNC	PS_SYNC
23	DAI 5		DAI 5	DAI 5*	DAI 5	DAI 5	DAI 5	DAI 5	DAI 5
24	DAI 6		DAI 6	DAI 6*	DAI 6	DAI 6	DAI 6	DAI 6	DAI 6
25	DAI 7		DAI 7	DAI 7*	DAI 7	DAI 7	DAI 7	DAI 7	DAI 7
26	DAI 8		DAI 8	DAI 8*	DAI 8	DAI 8	DAI 8	DAI 8	DAI 8
27	DAI 9		DAI 9	DAI 9*	DAI 9	DAI 9	DAI 9	DAI 9	DAI 9
28	DAI 10		DAI 10	DAI 10*	DAI 10	DAI 10	DAI 10	DAI 10	DAI 10
29	DAI 11		DAI 11	DAI 11*	DAI 11	DAI 11	DAI 11	DAI 11	DAI 11
30	DAI 12		DAI 12	DAI 12*	DAI 12	DAI 12	DAI 12	DAI 12	CLK_EXT DAI 12
31	DAI 13		DAI 13	DAI 13*	DAI 13	DAI 13	DAI 13	DAI 13	DAI 13
32	DAI 14		DAI 14	DAI 14*	DAI 14	DAI 14	DAI 14	DAI 14	DAI 14
33	DAI 15		DAI 15	DAI 15*	DAI 15	DAI 15	DAI 15	DAI 15	DAI 15
34	DAI 16		DAI 16	DAI 16*	DAI 16	DAI 16	DAI 16	DIR0	DAI 16
35	DAI 17		DAI 17	DAI 17*	DAI 17	DAI 17	DAI 17	DIR1	DAI 17
36	DAI 18		DAI 18	DAI 18*	DAI 18	DAI 18	DAI 18	DIR2	DAI 18
37	DAI 19		DAI 19	DAI 19*	DAI 19	DAI 19	DAI 19	DIR3	DAI 19
38	DAI 20		DAI 20	DAI 20*	DAI 20	DAI 20	DAI 20	OUT1	DAI 20
39	**GND**		**GND**	**GND**	**GND**	**GND**	**GND**	**GND**	**GND**
40	**GND**		**GND**	**GND**	**GND**	**GND**	**GND**	**GND**	**GND**

* Also connected to FPGA pin

The peripherals abstracted by the DAI & DPI are not necessarily identical. For example, SPORTs operate faster with higher DSP Core Clock. Some peripherals may not exist on all processors. Consult the individual dspblok user manuals and device data sheets and reference manuals to review this aspect of compatibility.

JH3 – DSP Mode Configuration

- Location – Bottom Side, early 21369, Top Side, all others
- Straight Dual Row - 10 Pin
- PCB Compatibility Issues – None

JH3 was originally used for JTAG expansion to a target board. With the advent of dspblok "with ICE" versions of the dspblok, this function was no longer needed. New designs may use JH3 as a configuration header to select Clock Configuration and Boot Modes for the target on board DSP. Since the header is on the top side of the pcb, the specific definitions can vary depending on the DSP requirement and do not have a specific issue for compatibility. Individual dspblok user manuals should be reviewed for the correct jumper settings.

The dspblok 21369zx and dspblok 21369+fpga no longer include JH3. This means that there is no reason to allocate clearance on mating boards.

JH4 – Power Supply & Clock

- Location – Bottom Side
- Straight Dual Row - 10 Pin
- PCB Compatibility Issues – Moderate

Universal power supply connections are fairly straightforward with the dspblok. All dspbloks require 3.3V for at least their I/O power requirements. The dspblok has a separate input for their core switching supplies. This can be a DC supply from 3.3V to 5V. It must be connected even if powered by 3.3V. For maximum capability connect 3.3V to both pins 5 and 6.

The most important consideration for compatibility is available power. Usually this is determined by the requirements of the core supply (Vdd). The ADSP-21369 and ADSP-21469 have similar power requirements when each is operating at full speed. This can be as high as 1.5W at maximum speed and operating temperature. Obviously, a DSP + FPGA module will require more power than a DSP only dspblok. FPGA power consumption can be difficult to predict. FPGA current draw is largely a function of its configuration. You also need a low impedance supply to account for the initial in-rush current on these devices. FPGA current will also vary depending on the density of the device. We usually assume that an FPGA will draw twice the current of a companion DSP. Low power designs can benefit from dspbloks based on the ADSP-2147x family of DSPs. These processors trade off speed for power consumption. This opens the door for USB bus powered applications. USB bus powered applications are described in a later section of this document.

In most cases, a 5V, 500mA linear supply for Vdd will be sufficient. Switching supplies are often higher impedance and therefore may need to have larger current capability. A large capacitor placed across Vdd, after an external switching supply may also be helpful for dspbloks with FPGAs.

Most designs will not require the remaining pins to function. These are used to take advantage of certain features but may restrict compatibility. In most cases, connections to these pins will not cause problems unless an extended feature is required in the target design. For example, PWR_EN# is used for USB bus powered applications.

JH4 Pin	Default		21369zx	21369 fpga	21469 USB	21489	21479	21479 u8a	21469/21489 uac2
1	**GND**		**GND**	**GND**	**GND**	**GND**	**GND**	**GND**	**GND**
2	**Ext Clk**		**Ext Clk**	**Ext Clk**	**Ext Clk**	**Ext Clk**			
3			Core Mon	Core Mon	Core Mon	PWR EN#	PWR EN#	PWR EN#	
4	ClkOut		DSP Clk	DSP Clk	DSP Clk				
5	**Vd+3.3**		**Vd+3.3**	**Vd+3.3**	**Vd+3.3**	**Vd+3.3**	**Vd+3.3**	**Vd+3.3**	
6	**Vd+3.3**		**Vd+3.3**	**Vd+3.3**	**Vd+3.3**	**Vd+3.3 USB**			**Vd+3.3**
7	**Vdd**		**Vdd**	**Vdd**	**Vdd**	**Vdd**	**Vdd**	**Vdd**	**Vdd**
8	**Vdd**		**Vdd**	**Vdd**	**Vdd**	**Vdd**	**Vdd**	**Vdd**	**Vdd**
9			GND	Sync	Sync	Sync	Sync		
10	**GND**		**GND**	**GND**	**GND**	**GND**	**GND**	**GND**	**GND**

JH5, JH6, JH7 – External Data Bus, Link Ports and other I/O

- Location – Bottom Side
- PCB Compatibility Issues – None to Unworkable

The combination of JH5, JH6 and JH7 comprise the external data bus in many dspblok modules. In others, they represent a combination of the external data bus and link ports or additional I/O. In FPGA designs, they can be assigned for whatever you want them to be.

The reason that the pin assignments have changed is that the computational devices are a little different. On the ADSP-21369, the SDRAM interface is shared with the external bus. This requires the data bus to be wide (32 bits). The ADSP-21469 supports DDR2. The DDR2 interface is independent from the external data bus. Once the DRAM interface requirements were removed from the external data bus, the bus width could be reduced to 8 bits. The ADSP-21489 has a 16 bit bus, but we have elected to bring out only the lower 8 bits. This keeps it compatible with the ADSP-21469.

This may seem like an insurmountable obstacle for compatibility but in most cases it's not a big issue. Since the dspblok modules already encompass the memory, computational functions and USB, the external data bus is generally used exclusively for a few select external I/O devices. The dspblok 21369zx supports a 32 bit wide data bus on JH5 & JH6. With the exception of the internal SDRAM, we have never used the upper 16 bits on any Danville design. We have used the lower half (D15-D0) of the dspblok 21369zx on our dspstak 21369zx2 to support a PLX NET2272 USB transceiver. This board is now obsolete since we provide onboard high speed USB with many dspbloks. We estimate that at least 90% of current dspblok 21369zx applications operate with either no external data bus requirements or use only the lowest 8 bit portion of the data bus (D7-D0). Any of our boards that use FTDI USB transceivers meet this criterion.

The conclusion to draw from this situation is that it's easy to port a dspblok 21369zx based design to any of the newer dspblok SHARC based designs as long as you don't actually need more than an 8 bit external data bus.

JH5 is generally used for the lower portion of the data bus. JH7 is the address bus. JH6 varies considerably depending on the module. The following table of pin assignments should make this clearer. Not all dspbloks have JH6 headers. To maintain compatibility, target boards should not have components placed below JH6.

JH5 Pin	Default	21369zx	21369 fpga	21469 USB	21489	21479	21479 u8a	21469/21489 uac2
1		D15	D15*	NC	NC			
2		D14	D14*	NC	NC			
3		D13	D13*	NC	NC			
4		D12	D12*	NC	NC			
5		D11	D11*	NC	NC			
6		D10	D10*	NC	NC			
7		D9	D9*	NC	NC			
8		D8	D8*	NC	NC			
9	D7	D7	D7*	D7	D7			
10	D6	D6	D6*	D6	D6			
11	D5	D5	D5*	D5	D5			
12	D4	D4	D4*	D4	D4			
13	D3	D3	D3*	D3	D3			
14	D2	D2	D2*	D2	D2			
15	D1	D1	D1*	D1	D1			
16	D0	D0	D0*	D0	D0			
17	RD#	RD#	RD#*	RD#	RD#			
18	WR#	WR#	WR#*	WR#	WR#			
19	ACK	ACK	ACK*	ACK	ACK			
20	Reserved	Reserved	Reserved	Reserved	Reserved			

* Can be freely assigned, data bus assignments are only used for compatibility

JH6 Pin	Default		21369zx	21369 fpga	21469 USB	21489	21479	21479 u8a	21469 uac2
1			D31	D31	L0_DAT0				
2			D30	D30	L0_DAT1				
3			D29	D29	L0_DAT2				
4			D28	D28	L0_DAT3				
5			D27	D27	L0_DAT4				
6			D26	D26	L0_DAT5				
7			D25	D25	L0_DAT6				
8			D24	D24	L0_DAT7				
9			D23	D23	LCLK0				
10			D22	D22	LACK0				
11			D21	D21	L1_DAT0				
12			D20	D20	L1_DAT1				
13			D19	D19	L1_DAT2				
14			D18	D18	L1_DAT3				
15			D17	D17	L1_DAT4				
16			D16	D16	L1_DAT5				
17			Reserved	Reserved	L1_DAT6				

18			Reserved	Reserved	L1_DAT7				
19			Reserved	Reserved	LCLK1				
20			Reserved	Reserved	LACK1				

JH7 Pin	Default		21369zx	21369 fpga	21469 USB	21489	21479	21479 u8a	21469/21489 uac2
1	Reserved		NC	NC	NC	NC			
2	Reserved		NC	NC	NC	NC			
3	Reserved		NC	NC	NC	NC			
4	A23		A23	A23*	A23	A23			
5	A22		A22	A22*	A22	A22			
6	A21		A21	A21*	A21	A21			
7	A20		A20	A20*	A20	A20			
8	A19		A19	A19*	A19	A19			
9	A18		A18	A18*	A18	A18			
10	A17		A17	A17*	A17	A17			
11	A16		A16	A16*	A16	A16			
12	A15		A15	A15*	A15	A15			
13	A14		A14	A14*	A14	A14			
14	A13		A13	A13*	A13	A13			
15	A12		A12	A12*	A12	A12			
16	A11		A11	A11*	A11	A11			
17	A10		A10	A10*	A10	A10			
18	A9		A9	A9*	A9	A9			
19	A8		A8	A8*	A8	A8			
20	A7		A7	A7*	A7	A7			
21	A6		A6	A6*	A6	A6	NC		
22	A5		A5	A5*	A5	A5	NC		
23	A4		A4	A4*	A4	A4	NC		
24	A3		A3	A3*	A3	A3	NC		
25	A2		A2	A2*	A2	A2	NC		
26	A1		A1	A1*	A1	A1	NC		
27	A0		A0	A0*	A0	A0	NC		
28	MS1#		MS1#	MS1#*	MS1#**	MS1#**	NC		
29	MS2#		MS2#	MS2#*	MS2#	MS2#	FLG 2		
30	MS3#		MS3#	MS3#*	MS3#	MS3#	FLG 3		

* Can be freely assigned, data bus assignments are only used for compatibility
** MS1# is used to support on board USB, Address assigned to upper memory A23-A21 = 111
*** MS2# & MS3# can also be assigned as FLG 2 & FLG 3, but both must be of the same type.

JH8 –USB & dspBootloader Mode

- Location – Top Side – 21369 unpopulated, Bottom Side – All others where populated
- Straight Dual Row – 16 Pins
- PCB Compatibility Issues – Minimal

JH8 is the USB expansion connection. It also is used to support the mode pins of the Danville dspBootloader (version 3). All fourth generation dspbloks have JH8 headers.

JH8 served a different purpose on the dspblok 21369zx and dspblok 21369+fpga. These connections are do not extend below the pcb and can be safely ignored from a target pcb compatibility perspective.

JH8 Pin	Default		21369zx	21369 fpga	21469 USB	21489	21479	21479 u8a	21469/21489 uac2
1					NC	NC	NC	NC	NC
2					NC	NC	NC	NC	NC
3					USB VBUS	USB VBUS	USB VBUS	USB VBUS	USB VBUS
4					NC	NC	NC	NC	NC
5					GND	GND	GND	GND	GND
6					GND	GND	GND	GND	GND
7					USB DP	USB DP	USB DP	USB DP	USB DP
8					USB_DP	USB_DP	USB_DP	USB_DP	USB_DP
9					GND	GND	GND	GND	GND
10					GND	GND	GND	GND	GND
11					Reserved	NC	NC	NC	Mode 2
12					GND	GND	GND	GND	GND
13					Mode 1	Mode 1	Mode 1	Mode 1	Mode 1
14					GND	GND	GND	GND	GND
15					Mode 0	Mode 0	Mode 0	Mode 0	Mode 0
16					GND	GND	GND	GND	GND

* The dspblok 21469 does not include the USB connections. This connectivity is provided by the dspblok 21469+USB.

JH9 –Ethernet

- Straight Dual Row – up to 40 Pins
- PCB Compatibility Issues – Minimal

JH9 is the Ethernet expansion connection. Most dspbloks do not have this expansion. There is also a JH9 on the dspblok 21479u8a. This is used for an entirely different purpose and can be safely ignored.

JH9 Pin	Default		21369zx	21369 fpga	21469 USB	21489 ETH	21479	21479 u8a	21469/21489 uac2
1						RD-			
2						RD+			
3						NC			
4						NC			
5						NC			
6						TCT			
7						TD+			
8						TD-			
9						Connector GND			
10						NC			
11						Link LED#			
12						NC			
13						Full Dplx LED#			
14						GND			
15						GND			
16						3.3V			

Migrating from existing dspblok 21369zx designs

Migrating an existing dspblok 21369zx design to use a 4th generation dspblok is often an easy process. This document concentrates primarily on hardware issues. Obviously, there will also software modifications necessary as well. If the migration is from an earlier SHARC to a later version, the software changes will be quite minimal.

The 4th generation dspbloks are based on Analog Devices' ADSP-21469, ADSP-21489 or ADSP-21479 DSPs. One of these newer processors are almost always a better choice than the ADSP-21369. In some cases, you may need better performance, in others, lower cost. Sometimes you can get both.

This section will help you determine if your design is already hardware compatible with a newer dspblok module. If it is not, you may find that the "Designing a Target for Universal Support" in conjunction with this section may be helpful. Danville engineers are also available to discuss specific migration questions and review schematics and boards.

There are really only a few issues that need to be addressed.

Migration Checklist

1. Does the new target have a wide enough data bus?
2. Is there a conflict with MS1#?
3. Is the power supply sufficient for the new module?
4. Is the new dspblok module fast enough for the application?
5. Does the new module have all the peripherals needed for the application?
6. Does the application need to operate over extended temperature?
7. Is the memory sufficient?
8. Is there a clearance problem with JH8?

Using the checklist as a guide:

1. Does the new target have a wide enough data bus?

   This is the big question. If the application needs a 16 bit or 32 bit external data bus, then migration will be difficult without the addition of some glue logic. If the application does not use an external data bus, any of the standard fourth generation dspbloks will probably be a drop in upgrade for the dspblok 21369zx. If the application requires just the lower 8 bits of the external data bus, the dspblok 21469 or dspblok 21489 are both candidates.

2. Is there a conflict with MS1#?

   The dspblok 21469+USB and the dspblok 21489 both map the USB, dspBootloader modes and optional Ethernet into the upper portion of MS1# where A[23..20] = 11xx. This is not an issue with the dspblok 21479 since USB and dspBootloader modes are mapped via SPI.

3. Is the power supply sufficient for the new module?

   The new dspbloks draw about the same current or less than the dspblok 21369zx even at their maximum core clocks. Power supply connections to JH4 do not need to change.

4. Is the new dspblok module fast enough for the application?

   The dspblok 21469 (450MHz) and dspblok 21489 (400MHz) are both faster than the dspblok 21369zx (333MHz). The dspblok 21479 is slower (250MHz) since it is optimized for low power consumption.

5. Does the new module have all the peripherals needed for the application?

   The fourth generation SHARC has one UART whereas the ADSP-21369 has two. There is also one less timer in the newer SHARCS.

6. Does the application need to operate over extended temperature?

   The dspblok 21469 and dspblok 21469+USB are available in extended temperature versions (–40 to +85 operation, 400MHz core clock). The dspblok 21489 family dspblok 21479 are specified for 0 to 70 operation.

7. Is the memory sufficient?

   The internal memory has been increased from 2Mbits to 5Mbits in the ADSP-21469 and ADSP-21489. For most applications, there is no need to ever use slow external DRAM. The ADSP-21469 supports DDR2 memory. This memory is substantially faster and larger than the plain vanilla SDRAM supported by the ADSP-21489.

   In our experience, we have found that applications will either need access to large amounts of fast memory (usually processing large FFTs or delay lines) or never need external memory at all. Given this situation, we could see no justification for adding external DRAM to the ADSP-21489 or ADSP-21479 DSPs. This helps allow the dspblok 21479 & dspblok 21489 to sell at lower prices than the dspblok 21369zx. If you actually need the SDRAM, the dspblok 21469 will have substantially better performance.

8. Is there a clearance problem with JH8?

   JH8 is needed for selecting the dspBootloader (version 3) mode pins. It is possible to add jumpers on the JH8 connector if there is sufficient clearance. The dspBootloader will also operate in its default mode with the jumpers open. If you overwrite the dspBootloader with your own loader file (*.ldr), then clearance will be the only concern.

Designing a Target for Universal Support

Clearly there is no absolute guarantee that a design will always be able to accommodate every new version of the dspblok since these products are a combination of existing features and new capabilities. In many cases, the extended features will be used for specific situations and will never have a functional equivalent.

Designing for universal support entails making sure that the least common denominator of requirements is supported, that clearance is provided for unused headers and that power supply is adequate.

We are going to start with the assumption that all targets will use either an 8 bit external data bus or require no external bus at all. This requirement allows either the dspblok 21489 or the dspblok 21479 to be the basic core.

The dspbloks with USB Audio Class 2 support place tighter restrictions on a universal design. The design of these targets will be discussed in the "Designing a Target for USB Audio Class 2 Support" section. In general, the standard dspbloks will also work (without the USB Audio) in the same footprint. USB bus power presents another compatibility issue. This is discussed in the "Designing for USB Bus Power" section, which affects some standard dspbloks as well as the USB Audio dspbloks. These issues are ignored in this section.

Universal Support Checklist

1. Is the power supply sufficient for all the supported targets?
2. Is there clearance for unused headers?
3. Is there a need for an external data bus?
4. Is there room for a dspblok "with ICE" or a JTAG adapter for an external programmer or ICE?

Using the checklist as a guide:

1. Is the power supply sufficient for all the supported targets?

   If we assume that the target is not operating in a USB bus power application, then this question comes down to how large should the power supply be to support the current dspbloks that the application might use as well as dspbloks in the future.

   The dspblok 21469 or dspblok 21469uac2 operating at full speed and executing floating point operations with substantial DMA operations from the DDR2 memory might be the worst case scenario for those applications that do not have additional coprocessors or FPGAs. Operating temperature becomes a very important consideration in this analysis. The Analog Devices' data sheets can help access this situation.

   So the simple answer is "it depends". If a design is going to be very performance driven, then it may be very desirable to assume that the dspblok 21469 or dspblok 21469uac2 is the target even if a less powerful and less expensive dspblok might be used.

A SHARC DSP can draw as much as 2.0W at it maximum temperature. Almost all of this will be supplied from the Vdd connections, which can range from 3.3 to 5.0V on a dspblok. The Vdd supply is the input to on-board switcher(s) used to power most of the core supplies on the DSP and optional coprocessors. The 3.3V I/O supply current will likely be much less.

The dspblok 21489 will be somewhat less due almost entirely to the lack of DDR2 memory fetches. The dominant current of the SHARC itself is mostly due to leakage current with is a function of temperature and not MIPs.

2. Is there clearance for unused headers?

With the possible exception of JH9, most designs will use JH2, JH4 and JH8. Design should allow clearance for JH5, JH6 and JH7. The clearance constraint usually means that there should not be components placed directly under the unused headers. There is no need to necessarily have mating connectors on the target if the connections are not needed. Often, the PCB will have tracks running in this area without the need to navigate through the unused mating connectors. If the design is very minimal, the dspblok 21479 is a good "least common denominator" footprint.

It is probably prudent to add 0.250 2-56 swage standoffs in the 4 corners of the board. These are located 2.5mm from the edge of the dspblok in all four corners. This is especially true for designs with no need for JH5 and JH7.

Headers are also placed on the topside of some dspbloks. Ideally, it is best to avoid clearance around the perimeter on the topside of the dspblok.

3. Is there a need for an external data bus?

Given the likelihood that communications peripheral support and external memory is already on the dspblok, there may not be a need to use an external bus. The remaining control interfaces may be well served by either I2C or SPI bus expansion. This opens the door to simpler routing and expansion on the target PCB and also may reduce the cost of the companion dspblok. If you can eliminate the need for an external bus, the dspblok 21479 is the basic footprint that is most universal. If you need an external data bus, the dspblok 21489 is a good foundation target.

4. Is there room for a JTAG adapter for an external programmer or ICE?

Connector Checklist

JH2 – DAI, DPI & FLAGS

The connections on JH2 have changed very little since the dspblok 21369zx. The main difference is that Pins 13 & 18 are no longer GND. Pin 13 is now FLAG3, which is current limited by a 1K00 resistor. MS3# which is also configurable as FLAG3, is located on JH7. It is essentially the same connection without the current limiting resistor. The main resistor for also locating FLAG3 on JH2 is that it may allow JH7 to be completely omitted and still take advantage of this resource.

- Most designs should leave Pins 13 & 18 open. In earlier boards, this connection was probably tied to GND.

- Add series terminators (22 to 33 ohms) on the driving side of each interface. These should be located close to JH2.

JH4 – Power Supply

Assuming that the target is not a USB Bus powered design:

- Connect 3.3V to Pins 5&6.
- Connect 3.3V to 5V to Pins 7&8. 5V is more efficient.
- Connect GND to Pins 1&10.

Optional:

- Connect a 3.3V logic level sync clock to Pin 9. The frequency should be in the range of 1.4M to 1.6M. Usually it will be the audio MCLK/16, for example 24.576M/16 = 1.536M. Some dspblok do not have sync inputs. On these dspbloks the pin is a NC. The only issue might be EMI.

- For a universal design, it is best to avoid the external clocks. These connections are not available on most fourth generation dspbloks. Order the dspblok 21469 family with a –25.000 suffix, to include an on-board crystal.

USB Powered targets are discussed in the "Designing for USB Bus Power" section.

JH5, JH6, JH7 – External Data Bus

Assuming that the target needs an external data bus:

- Remember that the upper portion of MS1# is used for on-board peripherals. The lower ¾ of the address space is available. MS2# and MS3# can also be used as FLAG2 and FLAG3, respectively.

They are configured together so MS1# might be used for the external data bus and FLAG2 & FLAG3 for other purposes.

- You may want to use a smaller 10 pin header for JH7, oriented so that Pin 1 on the mating connector is aligned to Pin 21. This is often sufficient for I/O expansion and may make PCB routing easier.

- Ignore JH6, but allocate clearance on the PCB.

- Remember that not all dspbloks include an external data bus.

Assuming that the target does not needs an external data bus:

- You may want to use a 10 pin header for JH7 oriented so that Pin 1 on the mating connector is aligned to Pin 21. This gives you access to MS2# & MS3#, which can be configured as FLAG2 & FLAG3.

- Place 0.250″ 2-56 threaded swage standoffs in the corners to support the dspblok.

JH8 – USB & dspBootloader Mode.

- Consider transient protection on USB circuits. A Texas Instruments' TPD2E001 is a good choice. Tracks need to be carefully laid out for high speed USB. Avoid stubs.

- Bring out the Mode pins to an external configuration header or dip switch.

Designing a Target for USB Audio Class 2 Support

USB Audio Class 2 is rapidly becoming a preferred audio interface to PCs and MACs. The asynchronous clocking allows digital audio to be transported with very low clock jitter at the data converters. Danville's dspbloks support USB Audio Class 2 with XMOS processors. The dspblok 21479u8a uses an XMOS XS1-U8A processor as its communications engine. This is ideal for high performance stereo applications with full bit depth and sample rates to 192k. The dspblok 21469uac2 and dspblok 21489uac2 use the XMOS XU-208. This device can operate up to 384k.

Since USB audio streaming is part of the new dspblok requirements, the available DAI and DPI is restricted for interprocessor control and data. The connections on JH2 are therefore a subset of the general purpose dspbloks. If you design a target to meet the requirements of the USB Audio dspbloks, you will probably be compatible with the standard dspbloks as well. Obviously, you will lose the audio streaming, but control via USB is still available.

If the target is going to be USB bus powered as well, there are further considerations. This will be addressed in the next section, "Designing for USB Bus Power". This feature only applies to the dspbloks based on the ADSP-21479. The other processors are optimized for speed at the cost of power consumption.

With the possible exception of USB Bus Power, the connections for JH4 (Power Supply) and JH8 (USB & dspBootloader Mode) are unchanged from the general purpose dspblok. Currently none of the USB Audio dspbloks support an external data bus so this completely eliminates JH5, JH6 & JH7 from consideration. If there is a possibility of substituting a standard dspblok, then you still will want to maintain the phantom clearance rules for JH5, JH6 & JH7.

This leaves the DAI, DPI & FLAGS header, JH2, as the focus of most of the attention.

DPI & FLAG Changes:

Standard dspbloks provide 12 of the 14 DPI lines externally. The two remaining are used for internal peripherals including flash, EEProm and sometimes USB support. Three of the DPI lines are dedicated to a SPI master controlled by the DSP.

On the USB Audio dspblok, a dedicated SPI interface has been added for XMOS/DSP control. This required the use of some of the DPI. In most applications, the DPI is used to support SPI or I2C communications and sometimes a UART function. Since there are now fewer DPI connections available, SPI slave selects which usually consumed DPI resources have been replaced with four new dedicated slave selects (SS0#-SS3#).

FLAG2 and FLAG3, which were often not used as flags anyway, are now used for internal operations. FLAG1 and FLAG0 are unchanged.

The new restrictions generally will not impose much of a design change except that the arbitrary DPI assignments are a little less flexible. There are still 5 DPIs that can be used for miscellaneous functions including UART, I2C, Interrupts or GPIO.

DAI Changes:

Standard dspbloks provide all 20 DAI lines externally. Clearly, this is no longer possible since the USB audio streams need access to the SPORTs. DAI1 through DAI4 and DAI16 through DAI20 are used internally on the dspblok 21479u8a. DAI13 through DAI15 are sometimes connected to both the XMOS processor and the DSP (called XDAI for our purposes). DAI12 is also a useful feed for a low phase jitter master clock.

The dspblok 21469uac2 and dspblok 2189uac2 have reclaimed a few of the DAI pins. DAI16 through DAI20 are now available and will continue to be available on future audio dspbloks.

Since the number of DAI connections has been reduced, the previously defined pins on the headers have new definitions. These can generally be duplicated by standard dspbloks.

The table on page 10 shows the new assignments. You can also review the specific dspblok manual for more details.

General Rules:

- Review the "Designing a Target for Universal Support" rules.
- Use the function specific connections such as SPI slave selects or the various DAI replacements before utilizing more general purpose DPI or DAI.
- Use the unassigned DPI for functions that are not specifically supported such as I2C or UART interfaces. You can always use a SPI expander to get more GPIO. The Microchip MCP23S08 or MCP23S17 are useful for these purposes. Simple outputs can be constructed with 74LV594 or 74LV595 devices.
- If the design uses an external low phase jitter clock, connect it to DAI12.
- Use the unassigned DAI for functions that are not specifically supported such data converter interfacing. Use DAI13 & DAI 15 last.
- Add series terminators (22 to 33 ohms) on the driving side of each interface. These should be located close to JH2.

Designing for USB Bus Power

Designing for USB Bus Power is often a difficult challenge. There are three basic rules to meet if you want the design to be USB compliant. Many designs break these rules (and will never be Logo compliant).

Rules:

1. Must not draw more than 100mA before enumeration.
2. Must not draw more than 500mA after enumeration (and permission)
3. Must not draw more than 2.5mA during USB suspend.

The USB transceivers used in some of the dspbloks including the FTDI based devices and the XMOS XS1-U8A have the capability of implementing these restrictions with a little help.

In general, the dspbloks based on ADSP-21479 DSPs are going to be the best candidates since power is at a premium. If the target requires high S/N performance this may be very difficult to achieve since high performance data converters are not low power. A practical, but perhaps not perfectly "kosher" solution is to use a USB 3.0/3.1 cable and connector. This allows the maximum current to be 900mA. Officially USB 3.0 high speed (480M/s) is still restricted to 500mA, but is likely to work to 900mA for the USB 3 superspeed (5G/s) requirement. If you are using an audio dspblok, you will also find that computers that support USB 3 are much more likely to work well with USB Audio Class 2.

On several of the dspbloks, the USB transceiver is always bus powered, even for an overall self powered design. Both the dspblok 21479 and dspblok 21479u8a are configured in this manner. The dspblok 21489 can also be USB bus powered but its wiring will need to be a little different. It will also create a somewhat larger challenge since this processor draws much more current than the boards based on ADSP-21479.

To comply with the 100mA and suspend 2.5mA requirements it is necessary to operate the USB subsection independently from the other components on the board. This means that the USB subsection will need to turn on the power to the remaining components after enumeration. This requires a few external components.

dspblok 21479 & dspblok 21479u8a:

JH4 pin 3 is the USB controlled PWR_EN#. It is active low. The on-board USB device gets its power from the USB VBUS line on JH8. After enumeration, PWR_EN# goes low. This signal should be fed into a FET load switch. A Fairchild FDC6330L is one of many choices. The load switch will connect the USB VBUS supply to the input of the power supply that feeds Vdd and 3.3 volts as well as the circuits on the mating PCB.

Clearly there are many specific power supply considerations that need to be addressed in this situation. Danville engineers can discuss these possibilities.

dspblok 21489:

The dspblok needs to a separate 3.3V supply for the USB subsection. This is an exception to the rule that JH4 Pin 5 and Pin 6 should be tied together. You need to add an external 3.3V LDO from USB VBUS to Pin 6 of JH4. An NCP699SN33 or TPS79333 regulator is adequate. The remaining load switch circuits are essentially the same. Remember that JH4 Pin 5 will need 3.3V (and Vdd on Pins 7&8) after enumeration. This is the 3.3V supply for the rest of the components.

Conclusion

The dspblok is a flexible and adaptable platform that minimizes product obsolescence as technology marches on. In most cases, simple steps can be taken to migrate existing products and provide support for the future.

Danville Engineers are available to discuss and review any of these considerations.

We follow the Analog Devices and XMOS roadmaps very carefully. Obviously, there is no absolute guarantee that all new devices will always be 100% compatible since each generation of new parts tend to bring new features and higher levels of performance or integration.

This document is periodically updated to reflect some of the changes as they occur.