



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



### FEATURES

ADSP-2199x, 16-bit, fixed-point DSP core with up to 160 MIPS sustained performance

48K words of on-chip RAM, as 32K words on-chip 24-bit program RAM, and 16K words on-chip, 16-bit data RAM

External memory interface

Dedicated memory DMA controller for data/instruction transfer between internal/external memory

Programmable PLL and flexible clock generation circuitry enables full-speed operation from low speed input clocks

IEEE JTAG Standard 1149.1 test access port supports on-chip emulation and system debugging

8-channel, 14-bit analog-to-digital converter system, with up to 20 MSPS sampling rate (at 160 MHz core clock rate)

3-phase 16-bit center based PWM generation unit with 12.5 ns resolution at 160 MHz core clock (CCLK) rate

Dedicated 32-bit encoder interface unit with companion encoder event timer

Dual 16-bit auxiliary PWM outputs

16 general-purpose flag I/O pins

3 programmable 32-bit interval timers

SPI communications port with master or slave operation

Synchronous serial communications port (SPORT) capable of software UART emulation

Controller area network (CAN) module, fully compliant with V2.0B standard

Integrated watchdog timer

Dedicated peripheral interrupt controller with software priority control

Multiple boot modes

Precision 1.0 V voltage reference

Integrated power-on-reset (POR) generator

Flexible power management with selectable power-down and idle modes

2.5 V internal operation with 3.3 V I/O

Operating temperature ranges of  $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$  and  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$

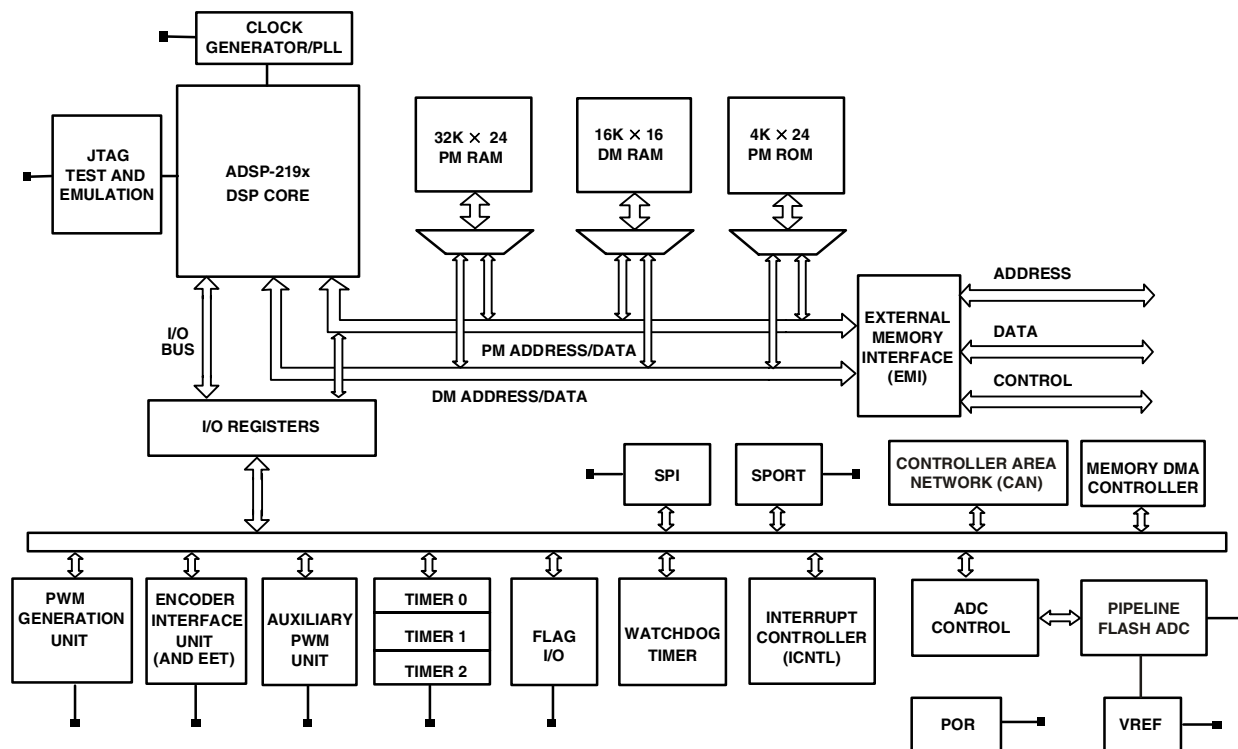


Figure 1. Functional Block Diagram

### Rev. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective companies.

# ADSP-21992\* PRODUCT PAGE QUICK LINKS

Last Content Update: 02/23/2017

---

## COMPARABLE PARTS

View a parametric search of comparable parts.

## EVALUATION KITS

- USB-Based Emulator and High Performance USB-Based Emulator

## DOCUMENTATION

### Application Notes

- AN-227: Digital Control System Design with the ADSP-2100 Family
  - AN-348: Avoiding Passive-Component Pitfalls
  - EE-06: ADSP-21xx Serial Port Startup Issues
  - EE-100: ADSP-218x External Overlay Memory
  - EE-102: Mode D and ADSP-218x Pin Compatibility - the FAQs
  - EE-104: Setting Up Streams with the VisualDSP Debugger
  - EE-11: ADSP-2181 Priority Chain & IDMA Holdoffs
  - EE-115: ADSP-2189 IDMA Interface to Motorola MC68300 Family of Microprocessors
  - EE-12: Interrupts and Programmable Flags on the ADSP-2185/2186
  - EE-121: Porting Code from ADSP-21xx to ADSP-219x
  - EE-122: Coding for Performance on the ADSP-219x
  - EE-123: An Overview of the ADSP-219x Pipeline
  - EE-124: Booting up the ADSP-2192
  - EE-125: ADSP-218x Embedded System Software Management and In-System-Programming (ISP)
  - EE-128: DSP in C++: Calling Assembly Class Member Functions From C++
  - EE-129: ADSP-2192 Interprocessor Communication
  - EE-130: Making Fast Transition from ADSP-21xx to ADSP-219x
  - EE-131: Booting the ADSP-2191/95/96 DSPs
  - EE-133: Converting From Legacy Architecture Files To Linker Description Files for the ADSP-218x
  - EE-139: Interfacing the ADSP-2191 to an AD7476 via the SPI Port
  - EE-142: Autobuffering, C and FFTs on the ADSP-218x
  - EE-144: Creating a Master-Slave SPI Interface Between Two ADSP-2191 DSPs
  - EE-145: SPI Booting of the ADSP-2191 using the Atmel AD25020N on an EZ-KIT Lite Evaluation Board
  - EE-146: Implementing a Boot Manager for ADSP-218x Family DSPs
  - EE-152: Using Software Overlays with the ADSP-219x and VisualDSP 2.0++
  - EE-153: ADSP-2191 Programmable PLL
  - EE-154: ADSP-2191 Host Port Interface
-

- 
- EE-156: Support for the H.100 protocol on the ADSP-2191
  - EE-158: ADSP-2181 EZ-Kit Lite IDMA to PC Printer Port Interface
  - EE-159: Initializing DSP System & Control Registers From C and C++
  - EE-164: Advanced EPROM Boot and No-boot Scenarios with ADSP-219x DSPs
  - EE-168: Using Third Overtone Crystals with the ADSP-218x DSP
  - EE-17: ADSP-2187L Memory Organization
  - EE-18: Choosing and Using FFTs for ADSP-21xx
  - EE-188: Using C To Implement Interrupt-Driven Systems On ADSP-219x DSPs
  - EE-2: Using ADSP-218x I/O Space
  - EE-226: ADSP-2191 DSP Host Port Booting
  - EE-227: CAN Configuration Procedure for ADSP-2192 DSPs
  - EE-249: Implementing Software Overlays on ADSP-218x DSPs with VisualDSP++®
  - EE-32: Language Extensions: Memory Storage Types, ASM & Inline Constructs
  - EE-35: Troubleshooting your ADSP-218x EZ-ICE
  - EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users
  - EE-38: ADSP-2181 IDMA Port - Cycle Steal Timing
  - EE-39: Interfacing 5V Flash Memory to an ADSP-218x (Byte Programming Algorithm)
  - EE-5: ADSP-218x Full Memory Mode vs. Host Memory Mode
  - EE-60: Simulating an RS-232 UART Using the Synchronous Serial Ports on the ADSP-21xx Family DSPs
  - EE-64: Setting Mode Pins on Reset
  - EE-68: Analog Devices JTAG Emulation Technical Reference
  - EE-71: Minimum Rise Time Specs for Critical Interrupt and Clock Signals on the ADSP-21x1/21x5
  - EE-74: Analog Devices Serial Port Development and Troubleshooting Guide
  - EE-78: BDMA Usage on 100 pin ADSP-218x DSPs Configured for IDMA Use
  - EE-79: EPROM Booting In Host Mode with 100 Pin 218x Processors
  - EE-82: Using an ADSP-2181 DSP's IO Space to IDMA Boot Another ADSP-2181
  - EE-89: Implementing A Software UART on the ADSP-2181 EZ-Kit-Lite
  - EE-96: Interfacing Two AD73311 Codecs to the ADSP-218x

## Data Sheet

- ADSP-21992: Mixed Signal DSP Controller With CAN Data Sheet

## Evaluation Kit Manuals

- ADSP-21992 EZ-KIT Lite® Evaluation System Manual

## Integrated Circuit Anomalies

- ADSP-2199x Anomaly List for Revisions up to 1.1

## Processor Manuals

- ADSP 21xx Processors: Manuals
- ADSP-2199x Mixed Signal DSP Controller Hardware Reference
- ADSP-219x DSP Instruction Set Reference
- Using the ADSP-2100 Family Volume 1

## Product Highlight

- The ADSP-2199x Family: Mixed-Signal DSPs For Embedded Control & Signal Processing Applications Product Highlight

## Software Manuals

- VisualDSP++ 3.5 Assembler and Preprocessor Manual for ADSP-218x and ADSP-219x DSPs
- VisualDSP++ 3.5 C Compiler and Library Manual for ADSP-218x DSPs
- VisualDSP++ 3.5 C/C++ Compiler and Library Manual for ADSP-219x Processors
- VisualDSP++ 3.5 Component Software Engineering User's Guide for 16-Bit Processors
- VisualDSP++ 3.5 Getting Started Guide for 16-Bit Processors
- VisualDSP++ 3.5 Kernel VDK User's Guide for 16-Bit Processors
- VisualDSP++ 3.5 Linker and Utilities Manual for 16-Bit Processors
- VisualDSP++ 3.5 Loader Manual for 16-Bit Processors
- VisualDSP++ 3.5 Quick Installation Reference Card
- VisualDSP++ 3.5 User's Guide for 16-Bit Processors

## SOFTWARE AND SYSTEMS REQUIREMENTS

- Software and Tools Anomalies Search

## TOOLS AND SIMULATIONS

- ADSP-21xx Processors: Software and Tools

## REFERENCE MATERIALS

### Technical Articles

- DSP Motor Control in Domestic Appliance Applications
-

---

## DESIGN RESOURCES

- [ADSP-21992 Material Declaration](#)
- [PCN-PDN Information](#)
- [Quality And Reliability](#)
- [Symbols and Footprints](#)

## DISCUSSIONS

[View all ADSP-21992 EngineerZone Discussions.](#)

## SAMPLE AND BUY

[Visit the product page to see pricing options.](#)

## TECHNICAL SUPPORT

[Submit a technical question or find your regional support number.](#)

## DOCUMENT FEEDBACK

[Submit feedback for this data sheet.](#)

---

## TABLE OF CONTENTS

General Description .....	3	Power Supplies .....	14
DSP Core Architecture .....	3	Bootling Modes .....	14
Memory Architecture .....	5	Instruction Set Description .....	14
Bus Request and Bus Grant .....	6	Development Tools .....	15
DMA Controller .....	7	Designing an Emulator-Compatible DSP Board .....	16
DSP Peripherals Architecture .....	7	Additional Information .....	16
Serial Peripheral Interface (SPI) Port .....	7	Pin Function Descriptions .....	17
DSP Serial Port (SPORT) .....	8	Specifications .....	20
Controller Area Network (CAN) Module .....	9	Operating Conditions .....	20
Analog-to-Digital Conversion System .....	9	Electrical Characteristics .....	23
Voltage Reference .....	9	Absolute Maximum Ratings .....	30
PWM Generation Unit .....	10	ESD Caution .....	30
Auxiliary PWM Generation Unit .....	10	Timing Specifications .....	30
Encoder Interface Unit .....	10	Power Dissipation .....	50
Flag I/O (FIO) Peripheral Unit .....	11	Test Conditions .....	51
Watchdog Timer .....	11	Output Disable Time .....	51
General-Purpose Timers .....	11	Output Enable Time .....	51
Interrupts .....	11	Example System Hold Time Calculation .....	51
Peripheral Interrupt Controller .....	12	Pin Configurations .....	52
Low Power Operation .....	12	Outline Dimensions .....	57
Clock Signals .....	13	Ordering Guide .....	59
Reset and Power-On Reset (POR) .....	13		

## REVISION HISTORY

8/07—Rev. 0 to Rev. A

Added RoHS part number to [Ordering Guide](#) ..... 59

## GENERAL DESCRIPTION

The ADSP-21992 is a mixed-signal DSP controller based on the ADSP-2199x DSP core, suitable for a variety of high performance industrial motor control and signal processing applications that require the combination of a high performance DSP and the mixed-signal integration of embedded control peripherals, such as analog-to-digital conversion with communications interfaces such as CAN. Target applications include industrial motor drives, uninterruptible power supplies, optical networking control, data acquisition systems, test and measurement Systems, and portable instrumentation.

The ADSP-21992 integrates the fixed-point ADSP-2199x family-based architecture with a serial port, an SPI-compatible port, a DMA controller, three programmable timers, general-purpose programmable flag pins, extensive interrupt capabilities, on-chip program and data memory spaces, and a complete set of embedded control peripherals that permits fast motor control and signal processing in a highly integrated environment.

The ADSP-21992 architecture is code compatible with previous ADSP-217x-based ADMCxxx products. Although the architectures are compatible, the ADSP-21992, with ADSP-2199x architecture, has a number of enhancements over earlier architectures. The enhancements to computational units, data address generators, and program sequencer make the ADSP-21992 more flexible and easier to program than the previous ADSP-21xx embedded DSPs.

Indirect addressing options provide addressing flexibility—pre-modify with no update, pre- and post-modify by an immediate 8-bit, twos complement value and base address registers for easier implementation of circular buffering.

The ADSP-21992 integrates 48K words of on-chip memory configured as 32K words (24-bit) of program RAM, and 16K words (16-bit) of data RAM.

Fabricated in a high speed, low power, CMOS process, the ADSP-21992 operates with a 6.25 ns instruction cycle time for a 160 MHz CCLK, with a 6.67 ns instruction cycle time for a 150 MHz CCLK, and with a 10.0 ns instruction cycle time for a 100 MHz CCLK. All instructions, except two multiword instructions, execute in a single DSP cycle.

The flexible architecture and comprehensive instruction set of the ADSP-21992 support multiple operations in parallel. For example, in one processor cycle, the ADSP-21992 can:

- Generate an address for the next instruction fetch.
- Fetch the next instruction.
- Perform one or two data moves.
- Update one or two data address pointers.
- Perform a computational operation.

These operations take place while the processor continues to:

- Receive and transmit data through the serial port.
- Receive or transmit data over the SPI port.
- Access external memory through the external memory interface.

- Decrement the timers.
- Operate the embedded control peripherals (ADC, PWM, EIU, etc.).

## DSP CORE ARCHITECTURE

- 6.25 ns instruction cycle time (internal), for up to 160 MIPS sustained performance (6.67 ns instruction cycle time for 150 MIPS sustained performance and 10.0 ns instruction cycle time for 100 MIPS sustained performance).
- ADSP-218x family code compatible with the same easy to use algebraic syntax.
- Single cycle instruction execution.
- Up to 1M words of addressable memory space with 24 bits of addressing width.
- Dual-purpose program memory for both instruction and data storage.
- Fully transparent instruction cache allows dual operand fetches in every instruction cycle.
- Unified memory space permits flexible address generation, using two independent DAG units.
- Independent ALU, multiplier/accumulator, and barrel shifter computational units with dual 40-bit accumulators.
- Single cycle context switch between two sets of computational and DAG registers.
- Parallel execution of computation and memory instructions.
- Pipelined architecture supports efficient code execution at speeds up to 160 MIPS.
- Register file computations with all nonconditional, non-parallel computational instructions.
- Powerful program sequencer provides zero overhead looping and conditional instruction execution.
- Architectural enhancements for compiled C code efficiency.
- Architecture enhancements beyond ADSP-218x family are supported with instruction set extensions for added registers, ports, and peripherals.

The clock generator module of the ADSP-21992 includes clock control logic that allows the user to select and change the main clock frequency. The module generates two output clocks: the DSP core clock, CCLK; and the peripheral clock, HCLK. CCLK can sustain clock values of up to 160 MHz, while HCLK can be equal to CCLK or CCLK/2 for values up to a maximum 80 MHz peripheral clock at the 160 MHz CCLK rate.

The ADSP-21992 instruction set provides flexible data moves and multifunction (one or two data moves with a computation) instructions. Every single word instruction can be executed in a single processor cycle. The ADSP-21992 assembly language uses

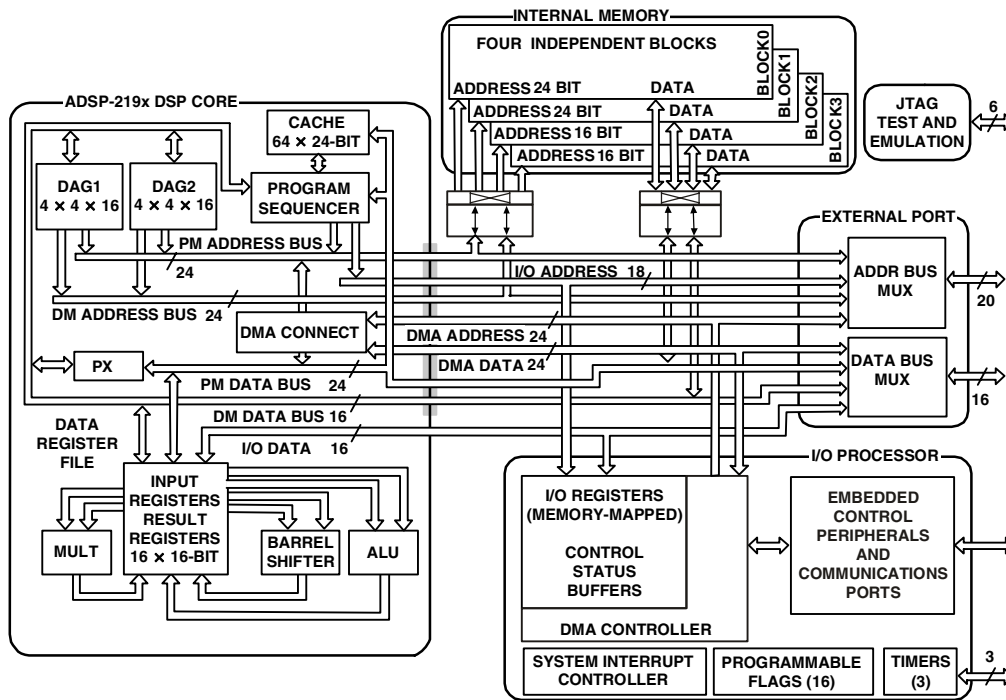


Figure 2. Block Diagram

an algebraic syntax for ease of coding and readability. A comprehensive set of development tools supports program development.

The block diagram (Figure 2) shows the architecture of the embedded SHARC core. It contains three independent computational units: the ALU, the multiplier/accumulator (MAC), and the shifter. The computational units process 16-bit data from the register file and have provisions to support multiprecision computations. The ALU performs a standard set of arithmetic and logic operations; division primitives are also supported. The MAC performs single cycle multiply, multiply/add, and multiply/subtract operations. The MAC has two 40-bit accumulators, which help with overflow. The shifter performs logical and arithmetic shifts, normalization, denormalization, and derive exponent operations. The shifter can be used to efficiently implement numeric format control, including multiword and block floating-point representations.

Register usage rules influence placement of input and results within the computational units. For most operations, the data registers of the computational units act as a data register file, permitting any input or result register to provide input to any unit for a computation. For feedback operations, the computational units let the output (result) of any unit be input to any unit on the next cycle. For conditional or multifunction instructions, there are restrictions on which data registers may provide inputs or receive results from each computational unit. For more information, see the *ADSP-2199x DSP Instruction Set Reference*.

A powerful program sequencer controls the flow of instruction execution. The sequencer supports conditional jumps, subroutine calls, and low interrupt overhead. With internal loop counters and loop stacks, the ADSP-21992 executes looped code with zero overhead; no explicit jump instructions are required to maintain loops.

Two data address generators (DAGs) provide addresses for simultaneous dual operand fetches (from data memory and program memory). Each DAG maintains and updates four 16-bit address pointers. Whenever the pointer is used to access data (indirect addressing), it is pre- or post-modified by the value of one of four possible modify registers. A length value and base address may be associated with each pointer to implement automatic modulo addressing for circular buffers. Page registers in the DAGs allow circular addressing within 64K word boundaries of each of the 256 memory pages, but these buffers may not cross page boundaries. Secondary registers duplicate all the primary registers in the DAGs; switching between primary and secondary registers provides a fast context switch.

Efficient data transfer in the core is achieved with the use of internal buses:

- Program memory address (PMA) bus
- Program memory data (PMD) bus
- Data memory address (DMA) bus
- Data memory data (DMD) bus
- Direct memory access address bus
- Direct memory access data bus



The two address buses (PMA and DMA) share a single external address bus, allowing memory to be expanded off-chip, and the two data buses (PMD and DMD) share a single external data bus. Boot memory space and I/O memory space also share the external buses.

Program memory can store both instructions and data, permitting the ADSP-21992 to fetch two operands in a single cycle, one from program memory and one from data memory. The DSP dual memory buses also let the embedded SHARC core fetch an operand from data memory and the next instruction from program memory in a single cycle.

## MEMORY ARCHITECTURE

The ADSP-21992 provides 48K words of on-chip SRAM memory. This memory is divided into three blocks: two 16K × 24-bit blocks (Blocks 0 and 1) and one 16K × 16-bit block (Block 2). In addition, the ADSP-21992 provides a 4K × 24-bit block of program memory boot ROM (that is reserved by ADI for boot load routines). The memory map of the ADSP-21992 is illustrated in Figure 2.

As shown in Figure 2, the three internal memory RAM blocks reside in memory page 0. The entire DSP memory map consists of 256 pages (Pages 0 to 255), and each page is 64K words long. External memory space consists of four memory banks (Banks 3–0) and supports a wide variety of memory devices. Each bank is selectable using unique memory select lines ( $\overline{MS3-0}$ ) and has configurable page boundaries, wait states, and wait state modes. The 4K words of on-chip boot ROM populates the top of Page 255, while the remaining 254 pages are addressable off-chip. I/O memory pages differ from external memory in that they are 1K word long, and the external I/O pages have their own select pin ( $\overline{IOMS}$ ). Pages 31–0 of I/O memory space reside on-chip and contain the configuration registers for the peripherals. Both the ADSP-2199x core and DMA capable peripherals can access the entire memory map of the DSP.

NOTE: The physical external memory addresses are limited by 20 address lines, and are determined by the external data width and packing of the external memory space. The Strobe signals ( $\overline{MS3-0}$ ) can be programmed to allow the user to change starting page addresses at runtime.

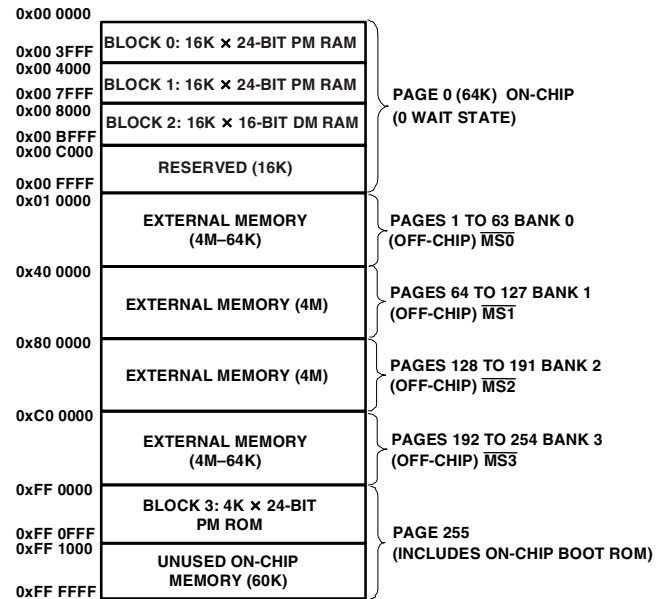


Figure 3. Core Memory Map at Reset

## Internal (On-Chip) Memory

The unified program and data memory space of the ADSP-21992 consists of 16M locations that are accessible through two 24-bit address buses, the PMA, and DMA buses. The DSP uses slightly different mechanisms to generate a 24-bit address for each bus. The DSP has three functions that support access to the full memory map.

- The DAGs generate 24-bit addresses for data fetches from the entire DSP memory address range. Because DAG index (address) registers are 16 bits wide and hold the lower 16 bits of the address, each of the DAGs has its own 8-bit page register (DMPG<sub>x</sub>) to hold the most significant eight address bits. Before a DAG generates an address, the program must set the DAG DMPG<sub>x</sub> register to the appropriate memory page. The DMPG<sub>1</sub> register is also used as a page register when accessing external memory. The program must set DMPG<sub>1</sub> accordingly, when accessing data variables in external memory. A “C” program macro is provided for setting this register.
- The program sequencer generates the addresses for instruction fetches. For relative addressing instructions, the program sequencer bases addresses for relative jumps, calls, and loops on the 24-bit program counter (PC). In direct addressing instructions (two word instructions), the instruction provides an immediate 24-bit address value. The PC allows linear addressing of the full 24-bit address range.
- For indirect jumps and calls that use a 16-bit DAG address register for part of the branch address, the program sequencer relies on an 8-bit indirect jump page (IJPg)

# ADSP-21992

register to supply the most significant eight address bits. Before a cross page jump or call, the program must set the program sequencer IJPG register to the appropriate memory page.

The ADSP-21992 has 4K words of on-chip ROM that holds boot routines. The DSP starts executing instructions from the on-chip boot ROM, which starts the boot process. For more information, see [Booting Modes on Page 14](#). The on-chip boot ROM is located on Page 255 in the DSP memory space map, starting at address 0xFF0000.

## External (Off-Chip) Memory

Each of the off-chip memory spaces of the ADSP-21992 has a separate control register, so applications can configure unique access parameters for each space. The access parameters include read and write wait counts, wait state completion mode, I/O clock divide ratio, write hold time extension, strobe polarity, and data bus width. The core clock and peripheral clock ratios influence the external memory access strobe widths. For more information, see [Clock Signals on Page 13](#). The off-chip memory spaces are:

- External memory space ( $\overline{MS3-0}$  pins)
- I/O memory space ( $\overline{IOMS}$  pin)
- Boot memory space ( $\overline{BMS}$  pin)

All of these off-chip memory spaces are accessible through the external port, which can be configured for 8-bit or 16-bit data widths.

## External Memory Space

External memory space consists of four memory banks. These banks can contain a configurable number of 64K word pages. At reset, the page boundaries for external memory have Bank0 containing pages 1 to 63, Bank1 containing pages 64 to 127, Bank2 containing pages 128 to 191, and Bank3 containing pages 192 to 254. The  $\overline{MS3-0}$  memory bank pins select Banks 3-0, respectively. Both the ADSP-2199x core and DMA capable peripherals can access the DSP external memory space.

All accesses to external memory are managed by the external memory interface unit (EMI).

## I/O Memory Space

The ADSP-21992 supports an additional external memory called I/O memory space. The I/O space consists of 256 pages, each containing 1024 addresses. This space is designed to support simple connections to peripherals (such as data converters and external registers) or to bus interface ASIC data registers. The first 32K addresses (I/O pages 0 to 31) are reserved for on-chip peripherals. The upper 224K addresses (I/O pages 32 to

255) are available for external peripheral devices. External I/O pages have their own select pin ( $\overline{IOMS}$ ). The DSP instruction set provides instructions for accessing I/O space.

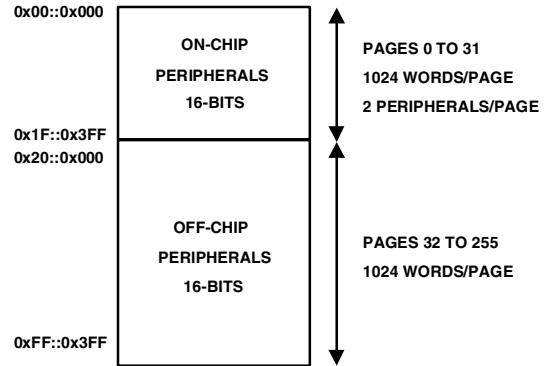


Figure 4. I/O Memory Map

## Boot Memory Space

Boot memory space consists of one off-chip bank with 254 pages. The  $\overline{BMS}$  memory bank pin selects boot memory space. Both the ADSP-2199x core and DMA capable peripherals can access the DSP off-chip boot memory space. After reset, the DSP always starts executing instructions from the on-chip boot ROM.

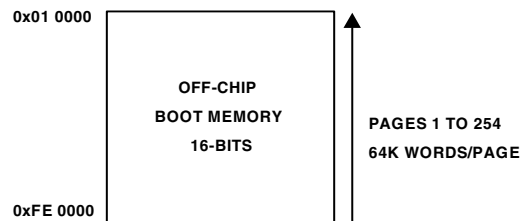


Figure 5. Boot Memory Map

## BUS REQUEST AND BUS GRANT

The ADSP-21992 can relinquish control of the data and address buses to an external device. When the external device requires access to the bus, it asserts the bus request ( $\overline{BR}$ ) signal. The ( $\overline{BR}$ ) signal is arbitrated with core and peripheral requests. External bus requests have the lowest priority. If no other internal request is pending, the external bus request will be granted. Due to synchronizer and arbitration delays, bus grants will be provided with a minimum of three peripheral clock delays. The ADSP-21992 will respond to the bus grant by:

- Three-stating the data and address buses and the  $\overline{MS3-0}$ ,  $\overline{BMS}$ ,  $\overline{IOMS}$ ,  $\overline{RD}$ , and  $\overline{WR}$  output drivers.
- Asserting the bus grant ( $\overline{BG}$ ) signal.

The ADSP-21992 will halt program execution if the bus is granted to an external device and an instruction fetch or data read/write request is made to external general-purpose or peripheral memory spaces. If an instruction requires two external memory read accesses, the bus will not be granted between the two accesses. If an instruction requires an external memory read and an external memory write access, the bus may be granted between the two accesses. The external memory interface can be configured so that the core will have exclusive use of the interface. DMA and bus requests will be granted. When the external device releases  $\overline{BR}$ , the DSP releases  $\overline{BG}$  and continues program execution from the point at which it stopped.

The bus request  $\overline{BREQ}$  operates at all times, even while the DSP is booting and  $\overline{RESET}$  is active.

The ADSP-21992 asserts the  $\overline{BGH}$  pin when it is ready to start another external port access, but is held off because the bus was previously granted. This mechanism can be extended to define more complex arbitration protocols for implementing more elaborate multimaster systems.

## DMA CONTROLLER

The ADSP-21992 has a DMA controller that supports automated data transfers with minimal overhead for the DSP core. Cycle stealing DMA transfers can occur between the ADSP-21992 internal memory and any of its DMA capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA capable peripherals and external devices connected to the external memory interface. DMA capable peripherals include the SPORT and SPI ports, and ADC control module. Each individual DMA capable peripheral has a dedicated DMA channel. To describe each DMA sequence, the DMA controller uses a set of parameters—called a DMA descriptor. When successive DMA sequences are needed, these DMA descriptors can be linked or chained together, so the completion of one DMA sequence autoinitiates and starts the next sequence. DMA sequences do not contend for bus access with the DSP core, instead DMAs “steal” cycles to access memory.

All DMA transfers use the DMA bus shown in [Figure 2 on Page 4](#). Because all of the peripherals use the same bus, arbitration for DMA bus access is needed. The arbitration for DMA bus access appears in [Table 1](#).

**Table 1. I/O Bus Arbitration Priority**

DMA Bus Master	Arbitration Priority
SPORT Receive DMA	0—Highest
SPORT Transmit DMA	1
ADC Control DMA	2
SPI Receive/Transmit DMA	3
Memory DMA	4—Lowest

## DSP PERIPHERALS ARCHITECTURE

The ADSP-21992 contains a number of special purpose, embedded control peripherals, which can be seen in the functional block diagram on Page 1. The ADSP-21992 contains a high performance, 8-channel, 14-bit ADC system with dual-channel simultaneous sampling ability across four pairs of inputs. An internal precision voltage reference is also available as part of the ADC system. In addition, a 3-phase, 16-bit, center-based PWM generation unit can be used to produce high accuracy PWM signals with minimal processor overhead. The ADSP-21992 also contains a flexible incremental encoder interface unit for position sensor feedback; two adjustable frequency auxiliary PWM outputs, 16 lines of digital I/O; a 16-bit watchdog timer; three general-purpose timers, and an interrupt controller that manages all peripheral interrupts. Finally, the ADSP-21992 contains an integrated power-on-reset (POR) circuit that can be used to generate the required reset signal for device power-on.

The ADSP-21992 has an external memory interface that is shared by the DSP core, the DMA controller, and DMA capable peripherals, which include the ADC, SPORT, and SPI communication ports. The external port consists of a 16-bit data bus, a 20-bit address bus, and control signals. The data bus is configurable to provide an 8- or 16-bit interface to external memory. Support for word packing lets the DSP access 16- or 24-bit words from external memory regardless of the external data bus width.

The memory DMA controller lets the ADSP-21992 move data and instructions from between memory spaces: internal-to-external, internal-to-internal, and external-to-external. On-chip peripherals can also use this controller for DMA transfers.

The embedded SHARC core can respond to up to 17 interrupts at any given time: three internal (stack, emulator kernel, and power-down), two external (emulator and reset), and 12 user-defined (peripherals) interrupts. Programmers assign each of the 32 peripheral interrupt requests to one of the 12 user-defined interrupts. These assignments determine the priority of each peripheral for interrupt service.

The following sections provide a functional overview of the ADSP-21992 peripherals.

## SERIAL PERIPHERAL INTERFACE (SPI) PORT

The serial peripheral interface (SPI) port provides functionality for a generic configurable serial port interface based on the SPI standard, which enables the DSP to communicate with multiple SPI-compatible devices. Key features of the SPI port are:

- Interface to host microcontroller or serial EEPROM.
- Master or slave operation (3-wire interface MISO, MOSI, SCK).
- Data rates to  $HCLK \div 4$  (16-bit baud rate selector).
- 8- or 16-bit transfer.
- Programmable clock phase and polarity.
- Broadcast Mode-1 master, multiple slaves.
- DMA capability and dedicated interrupts.

- PF0 can be used as slave select input line.
- PF1–PF7 can be used as external slave select output.

SPI is a 3-wire interface consisting of 2 data pins (MOSI and MISO), one clock pin (SCK), and a single slave select input (SPISS) that is multiplexed with the PF0 Flag I/O line and seven slave select outputs (SPISEL1 to SPISEL7) that are multiplexed with the PF1 to PF7 flag I/O lines. The  $\overline{\text{SPISS}}$  input is used to select the ADSP-21992 as a slave to an external master. The SPISEL1 to SPISEL7 outputs can be used by the ADSP-21992 (acting as a master) to select/enable up to seven external slaves in a multidevice SPI configuration. In a multimaster or a multidevice configuration, all MOSI pins are tied together, all MISO pins are tied together, and all SCK pins are tied together.

During transfers, the SPI port simultaneously transmits and receives by serially shifting data in and out on the serial data line. The serial clock line synchronizes the shifting and sampling of data on the serial data line.

In master mode, the DSP core performs the following sequence to set up and initiate SPI transfers:

- Enables and configures the SPI port operation (data size and transfer format).
- Selects the target SPI slave with the SPISELx output pin (reconfigured programmable flag pin).
- Defines one or more DMA descriptors in Page 0 of I/O memory space (optional in DMA mode only).
- Enables the SPI DMA engine and specifies transfer direction (optional in DMA mode only).
- In nonDMA mode only, reads or writes the SPI port receive or transmit data buffer.

The SCK line generates the programmed clock pulses for simultaneously shifting data out on MOSI and shifting data in on MISO. In DMA mode only, transfers continue until the SPI DMA word count transitions from 1 to 0.

In slave mode, the DSP core performs the following sequence to set up the SPI port to receive data from a master transmitter:

- Enables and configures the SPI slave port to match the operation parameters set up on the master (data size and transfer format) SPI transmitter.
- Defines and generates a receive DMA descriptor in Page 0 of memory space to interrupt at the end of the data transfer (optional in DMA mode only).
- Enables the SPI DMA engine for a receive access (optional in DMA mode only).
- Starts receiving the data on the appropriate SCK edges after receiving an SPI chip select on the SPISS input pin (reconfigured programmable flag pin) from a master.

In DMA mode only, reception continues until the SPI DMA word count transitions from 1 to 0. The DSP core could continue, by queuing up the next DMA descriptor.

The slave mode transmit operation is similar, except the DSP core specifies the data buffer in memory space, generates and relinquishes control of the transmit DMA descriptor, and begins filling the SPI port data buffer. If the SPI controller is not ready on time to transmit, it can transmit a “zero” word.

## DSP SERIAL PORT (SPORT)

The ADSP-21992 incorporates a complete synchronous serial port (SPORT) for serial and multiprocessor communications. The SPORT supports the following features:

- Bidirectional: The SPORT has independent transmit and receive sections.
- Double buffered: The SPORT section (both receive and transmit) has a data register for transferring data words to and from other parts of the processor and a register for shifting data in or out. The double buffering provides additional time to service the SPORT.
- Clocking: The SPORT can use an external serial clock or generate its own in a wide range of frequencies down to 0 Hz.
- Word length: Each SPORT section supports serial data word lengths from three to 16 bits that can be transferred either MSB first or LSB first.
- Framing: Each SPORT section (receive and transmit) can operate with or without frame synchronization signals for each data-word; with internally generated or externally generated frame signals; with active high or active low frame signals; with either of two pulse widths and frame signal timing.
- Companding in hardware: Each SPORT section can perform A law and  $\mu$  law companding according to CCITT recommendation G.711.
- Direct memory access with single cycle overhead: Using the built-in DMA master, the SPORT can automatically receive and/or transmit multiple memory buffers of data with an overhead of only one DSP cycle per data-word. The on-chip DSP, via a linked list of memory space resident DMA descriptor blocks, can configure transfers between the SPORT and memory space. This chained list can be dynamically allocated and updated.
- Interrupts: Each SPORT section (receive and transmit) generates an interrupt upon completing a data-word transfer, or after transferring an entire buffer or buffers if DMA is used.
- Multichannel capability: The SPORT can receive and transmit data selectively from channels of a serial bit stream that is time division multiplexed into up to 128 channels. This is especially useful for T1 interfaces or as a network communication scheme for multiple processors. The SPORTs also support T1 and E1 carrier systems.
- DMA Buffer: Each SPORT channel (Tx and Rx) supports a DMA buffer of up to eight 16-bit transfers.

- SPORT operates at a frequency of up to one-half the clock frequency of the HCLK.
- SPORT: Capable of UART software emulation.

### CONTROLLER AREA NETWORK (CAN) MODULE

The ADSP-21992 contains a controller area network (CAN) module. Key features of the CAN module are:

- Conforms to the CAN V2.0B standard.
- Supports both standard (11-bit) and extended (29-bit) identifiers.  
Supports data rates of up to 1 Mbps (and higher).
- 16 configurable mailboxes (all receive or transmit).
- Dedicated acceptance mask for each mailbox.
- Data filtering (first 2 bytes) which can be used for acceptance filtering.
- Error status and warning registers.
- Transmit priority by identifier.
- Universal counter module.
- Readable receive and transmit counters.

The CAN module is a low baud rate serial interface intended for use in applications where baud rates are typically under 1 Mbps. The CAN protocol incorporates a data CRC check, message error tracking and fault node confinement as means to improve network reliability to the level required for control applications.

The CAN module architecture is based around a 16-entry mailbox RAM. The mailbox is accessed sequentially by the CAN serial interface or the host CPU. Each mailbox consists of eight 16-bit data words. The data is divided into fields, which includes a message identifier, a time stamp, a byte count, up to 8 bytes of data, and several control bits. Each node monitors the messages being passed on the network. If the identifier in the transmitted message matches an identifier in one of its mailboxes, then the module knows that the message was meant for it, passes the data into its appropriate mailbox, and signals the host of its arrival with an interrupt.

The CAN network itself is a single, differential pair line. All nodes continuously monitor this line. There is no clock wire. Messages are passed in one of four standard message types or frames. Synchronization is achieved by an elaborate sync scheme performed in each CAN receiver. Message arbitration is accomplished one bit at a time. A dominant polarity is established for the network. All nodes are allowed to start transmitting at the same time following a frame sync pulse.

As each node transmits a bit, it checks to see if the bus is the same state that it transmitted. If it is, it continues to transmit. If not, then another node has transmitted a dominant bit so the first node knows it has lost the arbitration and it stops transmitting. The arbitration continues, bit by bit until only one node is left transmitting.

The electrical characteristics of each network connection are very stringent so the CAN interface is typically divided into two parts: a controller and a transceiver. This allows a single controller to support different drivers and CAN networks. The

ADSP-21992 CAN module represents only the controller part of the interface. The network I/O of this module is a single transmit line and a single receive line, which communicate to a line transceiver.

### ANALOG-TO-DIGITAL CONVERSION SYSTEM

The ADSP-21992 contains a fast, high accuracy, multiple input analog-to-digital conversion system with simultaneous sampling capabilities. This analog-to-digital conversion system permits the fast, accurate conversion of analog signals needed in high performance embedded systems. Key features of the ADC system are:

- 14-bit pipeline (6-stage pipeline) flash analog-to-digital converter.
- 8 dedicated analog inputs.
- Dual-channel simultaneous sampling capability.
- Programmable ADC clock rate to maximum of  $HCLK \div 4$ .
- First channel ADC data valid approximately 375 ns after CONVST (at 20 MSPS).
- All 8 inputs converted in approximately 725 ns (at 20 MSPS).
- 2.0 V peak-to-peak input voltage range.
- Multiple convert start sources.
- Internal or external voltage reference.
- Out of range detection.
- DMA capable transfers from ADC to memory.

The ADC system is based on a pipeline flash converter core, and contains dual input sample-and-hold amplifiers so that simultaneous sampling of two input signals is supported. The ADC system provides an analog input voltage range of 2.0 V p-p and provides 14-bit performance with a clock rate of up to  $HCLK \div 4$ . The ADC system can be programmed to operate at a clock rate from  $HCLK/4$  to  $HCLK/30$ , to a maximum clock rate of 20 MHz (at 160 MHz CCLK rate).

The ADC input structure supports eight independent analog inputs; four of which are multiplexed into one sample-and-hold amplifier (A\_SHA) and four of which are multiplexed into the other sample-and-hold amplifier (B\_SHA).

At the 20 MHz sampling rate, the first data value is valid approximately 375 ns after the convert start command. All eight channels are converted in approximately 725 ns.

The core of the ADSP-21992 provides 14-bit data such that the stored data values in the ADC data registers are 14 bits wide.

### VOLTAGE REFERENCE

The ADSP-21992 contains an on-board band gap reference that can be used to provide a precise 1.0 V output for use by the analog-to-digital system and externally on the VREF pin for biasing and level shifting functions. Additionally, the ADSP-21992 may be configured to operate with an external reference applied to the VREF pin, if required.

## PWM GENERATION UNIT

Key features of the 3-phase PWM generation unit are:

- 16-bit, center-based PWM generation unit.
- Programmable PWM pulse width, with resolutions to 12.5 ns (at 80 MHz HCLK Rate).
- Single/double update modes
- Programmable dead time and switching frequency.
- Twos complement implementation which permits smooth transition into full ON and full OFF states.
- Possibility to synchronize the PWM generation to an external synchronization.
- Special provisions for BDCM operation (crossover and output enable functions).
- Wide variety of special switched reluctance (SR) operating modes.
- Output polarity and clock gating control.
- Dedicated asynchronous PWM shutdown signal.
- Multiple shutdown sources, independently for each unit.

The ADSP-21992 integrates a flexible and programmable, 3-phase PWM waveform generator that can be programmed to generate the required switching patterns to drive a 3-phase voltage source inverter for ac induction (ACIM) or permanent magnet synchronous (PMSM) motor control. In addition, the PWM block contains special functions that considerably simplify the generation of the required PWM switching patterns for control of the electronically commutated motor (ECM) or brushless dc motor (BDCM). Tying a dedicated pin,  $\overline{\text{PWMSR}}$ , to GND, enables a special mode, for switched reluctance motors (SRM).

The six PWM output signals consist of three high side drive pins (AH, BH, and CH) and three low side drive signals pins (AL, BL, and CL). The polarity of the generated PWM signals may be set via hardware by the  $\text{PWMPOL}$  input pin, so that either active HI or active LO PWM patterns can be produced.

The switching frequency of the generated PWM patterns is programmable using the 16-bit  $\text{PWMTM}$  register. The PWM generator is capable of operating in two distinct modes, single update mode or double update mode. In single update mode the duty cycle values are programmable only once per PWM period, so that the resultant PWM patterns are symmetrical about the midpoint of the PWM period. In the double update mode, a second updating of the PWM registers is implemented at the midpoint of the PWM period. In this mode, it is possible to produce asymmetrical PWM patterns that produce lower harmonic distortion in 3-phase PWM inverters.

## AUXILIARY PWM GENERATION UNIT

Key features of the auxiliary PWM generation unit are:

- 16-bit, programmable frequency, programmable duty cycle PWM outputs.
- Independent or offset operating modes.
- Double buffered control of duty cycle and period registers.

- Separate auxiliary PWM synchronization signal and associated interrupt (can be used to trigger ADC convert start).
- Separate auxiliary PWM shutdown signal ( $\overline{\text{AUXTRIP}}$ ).

The ADSP-21992 integrates a 2-channel, 16-bit, auxiliary PWM output unit that can be programmed with variable frequency, variable duty cycle values and may operate in two different modes, independent mode or offset mode. In independent mode, the two auxiliary PWM generators are completely independent and separate switching frequencies and duty cycles may be programmed for each auxiliary PWM output. In offset mode the switching frequency of the two signals on the AUX0 and AUX1 pins is identical. Bit 4 of the  $\text{AUXCTRL}$  register places the auxiliary PWM channel pair in independent or offset mode.

The auxiliary PWM generation unit provides two chip output pins, AUX0 and AUX1 (on which the switching signals appear), and one chip input pin,  $\overline{\text{AUXTRIP}}$ , which can be used to shut down the switching signals—for example, in a fault condition.

## ENCODER INTERFACE UNIT

The ADSP-21992 incorporates a powerful encoder interface block to incremental shaft encoders that are often used for position feedback in high performance motion control systems.

- Quadrature rates to 53 MHz (at 80 MHz HCLK rate).
- Programmable filtering of all encoder input signals.
- 32-bit encoder counter.
- Variety of hardware and software reset modes.
- Two registration inputs to latch EIU count value with corresponding registration interrupt.
- Status of A/B signals latched with reading of EIU count value.
- Alternative frequency and direction mode.
- Single north marker mode.
- Count error monitor function with dedicated error interrupt.
- Dedicated 16-bit loop timer with dedicated interrupt.
- Companion encoder event (1/T) timer unit.

The encoder interface unit (EIU) includes a 32-bit quadrature up-/downconverter, programmable input noise filtering of the encoder input signals and the zero markers, and has four dedicated chip pins. The quadrature encoder signals are applied at the EIA and EIB pins. Alternatively, a frequency and direction set of inputs may be applied to the EIA and EIB pins. In addition, two north marker/strobe inputs are provided on pins EIZ and EIS. These inputs may be used to latch the contents of the encoder quadrature counter into dedicated registers,  $\text{EIZLATCH}$  and  $\text{EISLATCH}$ , on the occurrence of external events at the EIZ and EIS pins. These events may be programmed to be either rising edge only (latch event) or rising edge if the encoder is moving in the forward direction and falling edge if the encoder is moving in the reverse direction (software latched north marker functionality).

The encoder interface unit incorporates programmable noise filtering on the four encoder inputs to prevent spurious noise pulses from adversely affecting the operation of the quadrature counter. The encoder interface unit operates at a clock frequency equal to the HCLK rate. The encoder interface unit operates correctly with encoder signals at frequencies of up to 13.25 MHz, at the 80 MHz HCLK rate, corresponding to a maximum quadrature frequency of 53 MHz (assuming an ideal quadrature relationship between the input EIA and EIB signals).

The EIU may be programmed to use the north marker on EIZ to reset the quadrature encoder in hardware, if required.

Alternatively, the north marker can be ignored, and the encoder quadrature counter is reset according to the contents of a maximum count register, EIUMAXCNT. There is also a “single north marker” mode available in which the encoder quadrature counter is reset only on the first north marker pulse.

The encoder interface unit can also be made to implement some error checking functions. If an encoder count error is detected (due to a disconnected encoder line, for example), a status bit in the EIUSTAT register is set, and an EIU count error interrupt is generated.

The encoder interface unit of the ADSP-21992 contains a 16-bit loop timer that consists of a timer register, period register, and scale register so that it can be programmed to time out and reload at appropriate intervals. When this loop timer times out, an EIU loop timer timeout interrupt is generated. This interrupt could be used to control the timing of speed and position control loops in high performance drives.

The encoder interface unit also includes a high performance encoder event timer (EET) block that permits the accurate timing of successive events of the encoder inputs. The EET can be programmed to time the duration between up to 255 encoder pulses and can be used to enhance velocity estimation, particularly at low speeds of rotation.

## FLAG I/O (FIO) PERIPHERAL UNIT

The FIO module is a generic parallel I/O interface that supports 16 bidirectional multifunction flags or general-purpose digital I/O signals (PF15–PF0).

All 16 FLAG bits can be individually configured as an input or output based on the content of the direction (DIR) register, and can also be used as an interrupt source for one of two FIO interrupts. When configured as input, the input signal can be programmed to set the FLAG on either a level (level sensitive input/interrupt) or an edge (edge sensitive input/interrupt).

The FIO module can also be used to generate an asynchronous unregistered wake-up signal FIO\_WAKEUP for DSP core wake up after power-down.

The FIO lines, PF7–PF1 can also be configured as external slave select outputs for the SPI communications port, while PF0 can be configured to act as a slave select input.

The FIO lines can be configured to act as a PWM shutdown source for the 3-phase PWM generation unit of the ADSP-21992.

## WATCHDOG TIMER

The ADSP-21992 integrates a watchdog timer that can be used as a protection mechanism against unintentional software events. It can be used to cause a complete DSP and peripheral reset in such an event. The watchdog timer consists of a 16-bit timer that is clocked at the external clock rate (CLKIN or crystal input frequency).

In order to prevent an unwanted timeout or reset, it is necessary to periodically write to the watchdog timer register. During abnormal system operation, the watchdog count will eventually decrement to 0 and a watchdog timeout will occur. In the system, the watchdog timeout will cause a full reset of the DSP core and peripherals.

## GENERAL-PURPOSE TIMERS

The ADSP-21992 contains a general-purpose timer unit that contains three identical 32-bit timers. The three programmable interval timers (Timer0, Timer1, and Timer2) generate periodic interrupts. Each timer can be independently set to operate in one of three modes:

- Pulse waveform generation (PWM\_OUT) mode.
- Pulse width count/capture (WDTH\_CAP) mode.
- External event watchdog (EXT\_CLK) mode.

Each Timer has one bidirectional chip pin, TMR2-TMR0. For each timer, the associated pin is configured as an output pin in PWM\_OUT mode and as an input pin in WDTH\_CAP and EXT\_CLK modes.

## INTERRUPTS

The interrupt controller lets the DSP respond to 17 interrupts with minimum overhead. The DSP core implements an interrupt priority scheme as shown in Table 2. Applications can use the unassigned slots for software and peripheral interrupts. The peripheral interrupt controller is used to assign the various peripheral interrupts to the 12 user assignable interrupts of the DSP core.

**Table 2. Interrupt Priorities/Addresses**

Interrupt	IMASK/ IRPTL	Vector Address
Emulator (NMI) —Highest Priority	NA	NA
Reset (NMI)	0	0x00 0000
Power-Down (NMI)	1	0x00 0020
Loop and PC Stack	2	0x00 0040
Emulation Kernel	3	0x00 0060
User Assigned Interrupt (USR0)	4	0x00 0080

**Table 2. Interrupt Priorities/Addresses (Continued)**

Interrupt	IMASK/ IRPTL	Vector Address
User Assigned Interrupt (USR1)	5	0x00 00A0
User Assigned Interrupt (USR2)	6	0x00 00C0
User Assigned Interrupt (USR3)	7	0x00 00E0
User Assigned Interrupt (USR4)	8	0x00 0100
User Assigned Interrupt (USR5)	9	0x00 0120
User Assigned Interrupt (USR6)	10	0x00 0140
User Assigned Interrupt (USR7)	11	0x00 0160
User Assigned Interrupt (USR8)	12	0x00 0180
User Assigned Interrupt (USR9)	13	0x00 01A0
User Assigned Interrupt (USR10)	14	0x00 01C0
User Assigned Interrupt (USR11) —Lowest Priority	15	0x00 01E0

There is no assigned priority for the peripheral interrupts after reset. To assign the peripheral interrupts a different priority, applications write the new priority to their corresponding control bits (determined by their ID) in the interrupt priority control register.

Interrupt routines can either be nested with higher priority interrupts taking precedence or processed sequentially. Interrupts can be masked or unmasked with the IMASK register. Individual interrupt requests are logically ANDed with the bits in IMASK; the highest priority unmasked interrupt is then selected. The emulation, power-down, and reset interrupts are nonmaskable with the IMASK register, but software can use the DIS INT instruction to mask the power-down interrupt.

The interrupt control (ICNTL) register controls interrupt nesting and enables or disables interrupts globally.

The IRPTL register is used to force and clear interrupts. On-chip stacks preserve the processor status and are automatically maintained during interrupt handling. To support interrupt, loop, and subroutine nesting, the PC stack is 33 levels deep, the

loop stack is eight levels deep, and the status stack is 16 levels deep. To prevent stack overflow, the PC stack can generate a stack level interrupt if the PC stack falls below three locations full or rises above 28 locations full.

The following instructions globally enable or disable interrupt servicing, regardless of the state of IMASK.

- Ena Int
- Dis Int

At reset, interrupt servicing is disabled.

For quick servicing of interrupts, a secondary set of DAG and computational registers exist. Switching between the primary and secondary registers lets programs quickly service interrupts, while preserving the state of the DSP.

## PERIPHERAL INTERRUPT CONTROLLER

The peripheral interrupt controller is a dedicated peripheral unit of the ADSP-21992 (accessed via I/O mapped registers). The peripheral interrupt controller manages the connection of up to 32 peripheral interrupt requests to the DSP core.

For each peripheral interrupt source, there is a unique 4-bit code that allows the user to assign the particular peripheral interrupt to any one of the 12 user assignable interrupts of the embedded ADSP-2199x core. Therefore, the peripheral interrupt controller of the ADSP-21992 contains eight 16-bit interrupt priority registers (Interrupt Priority Register 0 (IPR0) to Interrupt Priority Register 7 (IPR7)).

Each interrupt priority register contains four 4-bit codes; one specifically assigned to each peripheral interrupt. The user may write a value between 0x0 and 0xB to each 4-bit location in order to effectively connect the particular interrupt source to the corresponding user assignable interrupt of the ADSP-2199x core.

Writing a value of 0x0 connects the peripheral interrupt to the USR0 user assignable interrupt of the ADSP-2199x core while writing a value of 0xB connects the peripheral interrupt to the USR11 user assignable interrupt. The core interrupt USR0 is the highest priority user interrupt, while USR11 is the lowest priority. Writing a value between 0xC and 0xF effectively disables the peripheral interrupt by not connecting it to any ADSP-2199x core interrupt input. The user may assign more than one peripheral interrupt to any given ADSP-2199x core interrupt. In that case, the burden is on the user software in the interrupt vector table to determine the exact interrupt source through reading status bits.

This scheme permits the user to assign the number of specific interrupts that are unique to their application to the interrupt scheme of the ADSP-2199x core. The user can then use the existing interrupt priority control scheme to dynamically control the priorities of the 12 core interrupts.

## LOW POWER OPERATION

The ADSP-21992 has four low power options that significantly reduce the power dissipation when the device operates under standby conditions. To enter any of these modes, the DSP executes an IDLE instruction. The ADSP-21992 uses the



configuration of the PD, STCK, and STALL bits in the PLLCTL register to select between the low power modes as the DSP executes the IDLE instruction. Depending on the mode, an IDLE shuts off clocks to different parts of the DSP in the different modes. The low power modes are:

- Idle
- Power-down core
- Power-down core/peripherals
- Power-down all

### Idle Mode

When the ADSP-21992 is in idle mode, the DSP core stops executing instructions, retains the contents of the instruction pipeline, and waits for an interrupt. The core clock and peripheral clock continue running.

To enter idle mode, the DSP can execute the IDLE instruction anywhere in code. To exit idle mode, the DSP responds to an interrupt and (after two cycles of latency) resumes executing instructions.

### Power-Down Core Mode

When the ADSP-21992 is in power-down core mode, the DSP core clock is off, but the DSP retains the contents of the pipeline and keeps the PLL running. The peripheral bus keeps running, letting the peripherals receive data.

To exit power-down core mode, the DSP responds to an interrupt and (after two cycles of latency) resumes executing instructions.

### Power-Down Core/Peripherals Mode

When the ADSP-21992 is in power-down core/peripherals mode, the DSP core clock and peripheral bus clock are off, but the DSP keeps the PLL running. The DSP does not retain the contents of the instruction pipeline. The peripheral bus is stopped, so the peripherals cannot receive data.

To exit power-down core/peripherals mode, the DSP responds to an interrupt and (after five to six cycles of latency) resumes executing instructions.

### Power-Down All Mode

When the ADSP-21992 is in power-down all mode, the DSP core clock, the peripheral clock, and the PLL are all stopped. The DSP does not retain the contents of the instruction pipeline. The peripheral bus is stopped, so the peripherals cannot receive data.

To exit power-down core/peripherals mode, the DSP responds to an interrupt and (after 500 cycles to restabilize the PLL) resumes executing instructions.

## CLOCK SIGNALS

The ADSP-21992 can be clocked by a crystal oscillator or a buffered, shaped clock derived from an external clock oscillator. If a crystal oscillator is used, the crystal should be connected across the CLKIN and XTAL pins, with two capacitors connected as shown in Figure 6. Capacitor values are dependent on crystal

type and should be specified by the crystal manufacturer. A parallel resonant, fundamental frequency, microprocessor grade crystal should be used for this configuration.

If a buffered, shaped clock is used, this external clock connects to the DSP CLKIN pin. CLKIN input cannot be halted, changed, or operated below the specified frequency during normal operation. This clock signal should be a TTL-compatible signal. When an external clock is used, the XTAL input must be left unconnected.

The DSP provides a user-programmable  $1\times$  to  $32\times$  multiplication of the input clock, including some fractional values, to support 128 external to internal (DSP core) clock ratios. The BYPASS pin, and MSEL6–0 and DF bits, in the PLL configuration register, decide the PLL multiplication factor at reset. At runtime, the multiplication factor can be controlled in software. To support input clocks greater than 100 MHz, the PLL uses an additional bit (DF). If the input clock is greater than 100 MHz, DF must be set. If the input clock is less than 100 MHz, DF must be cleared. For clock multiplier settings, see the *ADSP-2199x DSP Hardware Reference Manual*.

The peripheral clock is supplied to the CLKOUT pin.

All on-chip peripherals for the ADSP-21992 operate at the rate set by the peripheral clock. The peripheral clock (HCLK) is either equal to the core clock rate or one half the DSP core clock rate (CCLK). This selection is controlled by the IOSEL bit in the PLLCTL register. The maximum core clock is 160 MHz for the ADSP-21992BST, 150 MHz for both the ADSP-21992BBC and ADSP-21992YBC, and 100 MHz for the ADSP-21992YST. The maximum peripheral clock is 80 MHz for the ADSP-21992BST, 75 MHz for both the ADSP-21992BBC and ADSP-21992YBC, and 50 MHz for the ADSP-21992YST—the combination of the input clock and core/peripheral clock ratios may not exceed these limits.

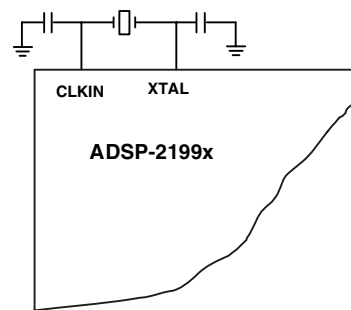


Figure 6. External Crystal Connections

## RESET AND POWER-ON RESET (POR)

The  $\overline{\text{RESET}}$  pin initiates a complete hardware reset of the ADSP-21992 when pulled low. The  $\overline{\text{RESET}}$  signal must be asserted when the device is powered up to assure proper initialization. The ADSP-21992 contains an integrated power-on reset (POR) circuit that provides an output reset signal,  $\overline{\text{POR}}$ , from the ADSP-21992 on power-up and if the power supply voltage falls below the threshold level. The ADSP-21992 may be reset

# ADSP-21992

from an external source using the  $\overline{\text{RESET}}$  signal, or alternatively, the internal power-on reset circuit may be used by connecting the POR pin to the RESET pin. During power-up the  $\overline{\text{RESET}}$  line must be activated for long enough to allow the DSP core's internal clock to stabilize. The power-up sequence is defined as the total time required for the crystal oscillator to stabilize after a valid VDD is applied to the processor and for the internal phase-locked loop (PLL) to lock onto the specific crystal frequency. A minimum of 512 cycles will ensure that the PLL has locked (this does not include the crystal oscillator start-up time).

The  $\overline{\text{RESET}}$  input contains some hysteresis. If an RC circuit is used to generate the  $\overline{\text{RESET}}$  signal, the circuit should use an external Schmitt trigger.

The master reset sets all internal stack pointers to the empty stack condition, masks all interrupts, and resets all registers to their default values (where applicable). When RESET is released, if there is no pending bus request, program control jumps to the location of the on-chip boot ROM (0xFF0000) and the booting sequence is performed.

## POWER SUPPLIES

The ADSP-21992 has separate power supply connections for the internal ( $V_{DDINT}$ ) and external ( $V_{DDEXT}$ ) power supplies. The internal supply must meet the 2.5 V requirement. The external supply must be connected to a 3.3 V supply. All external supply

pins must be connected to the same supply. The ideal power-on sequence for the DSP is to provide power-up of all supplies simultaneously. If there is going to be some delay in power-up between the supplies, provide  $V_{DD}$  first, then  $V_{DDIO}$ .

## BOOTING MODES

The ADSP-21992 supports a number of different boot modes that are controlled by the three dedicated hardware boot mode control pins (BMODE2, BMODE1, and BMODE0). The use of three boot mode control pins means that up to eight different boot modes are possible. Of these only five modes are valid on the ADSP-21992. The ADSP-21992 exposes the boot mechanism to software control by providing a nonmaskable boot interrupt that vectors to the start of the on-chip ROM memory block (at address 0xFF0000). A boot interrupt is automatically initiated following either a hardware initiated reset, via the  $\overline{\text{RESET}}$  pin, or a software initiated reset, via writing to the software reset register. Following either a hardware or a software reset, execution always starts from the boot ROM at address 0xFF0000, irrespective of the settings of the BMODE2, BMODE1, and BMODE0 pins. The dedicated BMODE2, BMODE1, and BMODE0 pins are sampled at hardware reset.

The particular boot mode for the ADSP-21992 associated with the settings of the BMODE2, BMODE1, BMODE0 pins is defined in [Table 3](#).

Table 3. Summary of Boot Modes

Boot Mode	BMODE2	BMODE1	BMODE0	Function
0	0	0	0	Illegal-Reserved
1	0	0	1	Boot from External 8-Bit Memory over EMI
2	0	1	0	Execute from External 8-Bit Memory
3	0	1	1	Execute from External 16-Bit Memory
4	1	0	0	Boot from SPI $\leq$ 4K Bits
5	1	0	1	Boot from SPI $>$ 4K Bits
6	1	1	0	Illegal-Reserved
7	1	1	1	Illegal-Reserved

## INSTRUCTION SET DESCRIPTION

The ADSP-21992 assembly language instruction set has an algebraic syntax that was designed for ease of coding and readability. The assembly language, which takes full advantage of the unique architecture of the processor, offers the following benefits:

- SHARC assembly language syntax is a superset of and source code compatible (except for two data registers and DAG base address registers) with ADSP-21xx family syntax. It may be necessary to restructure ADSP-21xx programs to accommodate the unified memory space of the ADSP-21992 and to conform to its interrupt vector map.
- The algebraic syntax eliminates the need to remember cryptic assembler mnemonics. For example, a typical arithmetic add instruction, such as  $AR = AX0 + AY0$ , resembles a simple equation.
- Every instruction, but two, assembles into a single, 24-bit word that can execute in a single instruction cycle. The exceptions are two dual word instructions. One writes 16- or 24-bit immediate data to memory, and the other is an absolute jump/call with the 24-bit address specified in the instruction.
- Multifunction instructions allow parallel execution of an arithmetic, MAC, or shift instruction with up to two fetches or one write to processor memory space during a single instruction cycle.
- Program flow instructions support a wider variety of conditional and unconditional jumps/calls and a larger set of conditions on which to base execution of conditional instructions.

## DEVELOPMENT TOOLS

The ADSP-21992 is supported with a complete set of CROSSCORE™ software and hardware development tools, including Analog Devices emulators and VisualDSP++™ development environment. The emulator hardware that supports other SHARC DSPs also fully emulates the ADSP-21992.

The VisualDSP++ project management environment lets programmers develop and debug an application. This environment includes an easy to use assembler (which is based on an algebraic syntax), an archiver (librarian/library builder), a linker, a loader, a cycle-accurate instruction-level simulator, a C/C++ compiler, and a C/C++ runtime library that includes DSP and mathematical functions. A key point for these tools is C/C++ code efficiency. The compiler has been developed for efficient translation of C/C++ code to DSP assembly. The DSP has architectural features that improve the efficiency of compiled C/C++ code.

The VisualDSP++ debugger has a number of important features. Data visualization is enhanced by a plotting package that offers a significant level of flexibility. This graphical representation of user data enables the programmer to quickly determine the performance of an algorithm. As algorithms grow in com-

plexity, this capability can have a significant influence on the design development schedule by increasing productivity. Statistical profiling enables the programmer to nonintrusively poll the processor as it is running the program. This feature, unique to VisualDSP++, enables the software developer to passively gather important code execution metrics without interrupting the real-time characteristics of the program. Essentially, the developer can identify bottlenecks in software quickly and efficiently. By using the profiler, the programmer can focus on those areas in the program that impact performance and take corrective action.

Debugging both C/C++ and assembly programs with the VisualDSP++ debugger, programmers can:

- View mixed C/C++ and assembly code (interleaved source and object information)
- Insert breakpoints
- Set conditional breakpoints on registers, memory, and stacks
- Trace instruction execution
- Perform linear or statistical profiling of program execution
- Fill, dump, and graphically plot the contents of memory
- Perform source level debugging
- Create custom debugger windows

The VisualDSP++ IDE lets programmers define and manage DSP software development. Its dialog boxes and property pages let programmers configure and manage all of the SHARC development tools, including the color syntax highlighting in the VisualDSP++ editor. This capability permits programmers to:

- Control how the development tools process inputs and generate outputs
- Maintain a one-to-one correspondence with the command line switches of the tool

The VisualDSP++ Kernel (VDK) incorporates scheduling and resource management tailored specifically to address the memory and timing constraints of DSP programming. These capabilities enable engineers to develop code more effectively, eliminating the need to start from the very beginning, when developing new application code. The VDK features include threads, critical and unscheduled regions, semaphores, events, and device flags. The VDK also supports priority-based, preemptive, cooperative, and time-sliced scheduling approaches. In addition, the VDK was designed to be scalable. If the application does not use a specific feature, the support code for that feature is excluded from the target system.

Because the VDK is a library, a developer can decide whether to use it or not. The VDK is integrated into the VisualDSP++ development environment, but can also be used via standard command line tools. When the VDK is used, the development environment assists the developer with many error-prone tasks and assists in managing system resources, automating the generation of various VDK-based objects, and visualizing the system state, when debugging an application that uses the VDK.

# ADSP-21992

VCSE is Analog Devices technology for creating, using, and reusing software components (independent modules of substantial functionality) to quickly and reliably assemble software applications. The user can also download components from the Web, drop them into the application and publish component archives from within VisualDSP++. VCSE supports component implementation in C/C++ or assembly language.

Use the Expert Linker to visually manipulate the placement of code and data on the embedded system, view memory utilization in a color-coded graphical form, easily move code and data to different areas of the DSP or external memory with the drag of the mouse, and examine runtime stack and heap usage. The Expert Linker is fully compatible with existing linker definition file (LDF), allowing the developer to move between the graphical and textual environments.

Analog Devices DSP emulators use the IEEE 1149.1 JTAG test access port of the ADSP-21992 processor to monitor and control the target board processor during emulation. The emulator provides full speed emulation, allowing inspection and modification of memory, registers, and processor stacks. Nonintrusive in-circuit emulation is assured by the use of the processor JTAG interface—target system loading and timing are not affected by the emulator.

In addition to the software and hardware development tools available from Analog Devices, third parties provide a wide range of tools supporting the SHARC processor family. Hardware tools include SHARC DSP PC plug-in cards. Third-party software tools include DSP libraries, real-time operating systems, and block diagram design tools.

## DESIGNING AN EMULATOR-COMPATIBLE DSP BOARD

The Analog Devices family of emulators are tools that every DSP developer needs to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG test access port (TAP) on each JTAG DSP. The emulator uses the TAP to access the internal features of the DSP, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The DSP must be halted to send data and commands, but once an operation has been completed by the emulator, the DSP system is set running at full speed with no impact on system timing.

To use these emulators, the target board must include a header that connects the DSP JTAG port to the emulator.

For details on target board design issues including mechanical layout, single processor connections, multiprocessor scan chains, signal buffering, signal termination, and emulator pod logic, see the *EE-68: JTAG Emulation Technical Reference* on the Analog Devices website ([www.analog.com](http://www.analog.com))—use site search on “EE-68.” This document is updated regularly to keep pace with improvements to emulator support.

## ADDITIONAL INFORMATION

This data sheet provides a general overview of the ADSP-21992 architecture and functionality. For detailed information on the ADSP-21992 embedded DSP core architecture, instruction set,

communications ports and embedded control peripherals, refer to the *ADSP-2199x Mixed Signal DSP Controller Hardware Reference Manual*.

## PIN FUNCTION DESCRIPTIONS

ADSP-21992 pin definitions are listed in [Table 4](#). All ADSP-21992 inputs are asynchronous and can be asserted asynchronously to CLKIN (or to TCK for  $\overline{\text{TRST}}$ ).

Unused inputs should be tied or pulled to  $V_{\text{DDEXT}}$  or GND, except for ADDR21–0, DATA15–0, PF7–0, and inputs that have internal pull-up or pull-down resistors ( $\overline{\text{TRST}}$ , BMODE0, BMODE1, BMODE2, BYPASS, TCK, TMS, TDI, PWMPOL, PWMSR, and RESET)—these pins can be left floating. These pins have a logic level hold circuit that prevents input from

floating internally.  $\overline{\text{PWMTRIP}}$  has an internal pull-down, but should not be left floating to avoid unnecessary PWM shutdowns.

The following symbols appear in the Type column of [Table 4](#): G = ground, I = input, O = output, P = power supply, B = bidirectional, T = three-state, D = digital, A = analog, CKG = clock generation pin, PU = internal pull-up, PD = internal pull-down, and OD = open drain.

**Table 4. Pin Descriptions**

Name	Type	Function
A19–A0	D, OT	External Port Address Bus
D15–D0	D, BT	External Port Data Bus
$\overline{\text{RD}}$	D, OT	External Port Read Strobe
$\overline{\text{WR}}$	D, OT	External Port Write Strobe
ACK	D, I	External Port Access Ready Acknowledge
$\overline{\text{BR}}$	D, I, PU	External Port Bus Request
$\overline{\text{BG}}$	D, O	External Port Bus Grant
$\overline{\text{BGH}}$	D, O	External Port Bus Grant Hang
$\overline{\text{MS0}}$	D, OT	External Port Memory Select Strobe 0
$\overline{\text{MS1}}$	D, OT	External Port Memory Select Strobe 1
$\overline{\text{MS2}}$	D, OT	External Port Memory Select Strobe 2
$\overline{\text{MS3}}$	D, OT	External Port Memory Select Strobe 3
$\overline{\text{IOMS}}$	D, OT	External Port IO Space Select Strobe
$\overline{\text{BMS}}$	D, OT	External Port Boot Memory Select Strobe
CLKIN	D, I, CKG	Clock Input/Oscillator Input/Crystal Connection 0
XTAL	D, O, CKG	Oscillator Output/Crystal Connection 1
CLKOUT	D, O	Clock Output (HCLK)
BYPASS	D, I, PU	PLL Bypass Mode Select
$\overline{\text{RESET}}$	D, I, PU	Processor Reset Input
$\overline{\text{POR}}$	D, O	Power on Reset Output
BMODE2	D, I, PU	Boot Mode Select Input 2
BMODE1	D, I, PD	Boot Mode Select Input 1
BMODE0	D, I, PU	Boot Mode Select Input 0
TCK	D, I	JTAG Test Clock
TMS	D, I, PU	JTAG Test Mode Select
TDI	D, I, PU	JTAG Test Data Input
TDO	D, OT	JTAG Test Data Output
$\overline{\text{TRST}}$	D, I, PU	JTAG Test Reset Input
$\overline{\text{EMU}}$	D, OT, PU	Emulation Status
VIN0	A, I	ADC Input 0
VIN1	A, I	ADC Input 1
VIN2	A, I	ADC Input 2
VIN3	A, I	ADC Input 3
VIN4	A, I	ADC Input 4
VIN5	A, I	ADC Input 5
VIN6	A, I	ADC Input 6
VIN7	A, I	ADC Input 7
ASHAN	A, I	Inverting SHA_A Input

# ADSP-21992

Table 4. Pin Descriptions (Continued)

Name	Type	Function
BSHAN	A, I	Inverting SHA_B Input
CAPT	A, O	Noise Reduction Pin
CAPB	A, O	Noise Reduction Pin
VREF	A, I, O	Voltage Reference Pin (Mode Selected by State of SENSE)
SENSE	A, I	Voltage Reference Select Pin
CML	A, O	Common-Mode Level Pin
CONVST	D, I	ADC Convert Start Input
CANRX	D, I	Controller Area Network (CAN) Receive
CANTX	D, OT	Controller Area Network (CAN) Transmit
PF15	D, BT, PD	General-Purpose IO15
PF14	D, BT, PD	General-Purpose IO14
PF13	D, BT, PD	General-Purpose IO13
PF12	D, BT, PD	General-Purpose IO12
PF11	D, BT, PD	General-Purpose IO11
PF10	D, BT, PD	General-Purpose IO10
PF9	D, BT, PD	General-Purpose IO9
PF8	D, BT, PD	General-Purpose IO8
PF7/SPISEL7	D, BT, PD	General-Purpose IO7/SPI Slave Select Output 7
PF6/SPISEL6	D, BT, PD	General-Purpose IO6/SPI Slave Select Output 6
PF5/SPISEL5	D, BT, PD	General-Purpose IO5/SPI Slave Select Output 5
PF4/SPISEL4	D, BT, PD	General-Purpose IO4/SPI Slave Select Output 4
PF3/SPISEL3	D, BT, PD	General-Purpose IO3/SPI Slave Select Output 3
PF2/SPISEL2	D, BT, PD	General-Purpose IO2/SPI Slave Select Output 2
PF1/SPISEL1	D, BT, PD	General-Purpose IO1/SPI Slave Select Output 1
PF0/SPISS	D, BT, PD	General-Purpose IO0/SPI Slave Select Input 0
SCK	D, BT	SPI Clock
MISO	D, BT	SPI Master In Slave Out Data
MOSI	D, BT	SPI Master Out Slave In Data
DT	D, OT	SPORT Data Transmit
DR	D, I	SPORT Data Receive
RFS	D, BT	SPORT Receive Frame Sync
TFS	D, BT	SPORT Transmit Frame Sync
TCLK	D, BT	SPORT Transmit Clock
RCLK	D, BT	SPORT Receive Clock
EIA	D, I	Encoder A Channel Input
EIB	D, I	Encoder B Channel Input
EIZ	D, I	Encoder Z Channel Input
EIS	D, I	Encoder S Channel Input
AUX0	D, O	Auxiliary PWM Channel 0 Output
AUX1	D, O	Auxiliary PWM Channel 1 Output
AUXTRIP	D, I, PD	Auxiliary PWM Shutdown Pin
TMR2	D, BT	Timer 0 Input/Output Pin
TMR1	D, BT	Timer 1 Input/Output Pin
TMR0	D, BT	Timer 2 Input/Output Pin
AH	D, O	PWM Channel A HI PWM
AL	D, O	PWM Channel A LO PWM
BH	D, O	PWM Channel B HI PWM
BL	D, O	PWM Channel B LO PWM
CH	D, O	PWM Channel C HI PWM
CL	D, O	PWM Channel C LO PWM

Table 4. Pin Descriptions (Continued)

<b>Name</b>	<b>Type</b>	<b>Function</b>
PWMSYNC	D, BT	PWM Synchronization
PWMPOL	D, I, PU	PWM Polarity
PWMTRIP	D, I, PD	PWM Trip
PWMSR	D, I, PU	PWM SR Mode Select
AVDD (2 pins)	A, P	Analog Supply Voltage
AVSS (2 pins)	A, G	Analog Ground
VDDINT (6 pins)	D, P	Digital Internal Supply
VDDEXT (10 pins)	D, P	Digital External Supply
GND (16 pins)	D, G	Digital Ground

# ADSP-21992

## SPECIFICATIONS

Specifications subject to change without notice.

### OPERATING CONDITIONS

**Table 5. Recommended Operating Conditions—ADSP-21992BBC**

Parameter	Conditions	Min	Typ	Max	Unit
V <sub>DDINT</sub>	Internal (Core) Supply Voltage	2.375	2.5	2.625	V
V <sub>DDEXT</sub>	External (I/O) Supply Voltage	3.135	3.3	3.465	V
AVDD	Analog Supply Voltage	2.375	2.5	2.625	V
CCLK	DSP Instruction Rate, Core Clock	0		150	MHz
HCLK <sup>1, 2</sup>	Peripheral Clock Rate	0		75	MHz
CLKIN <sup>3</sup>	Input Clock Frequency	0		150	MHz
T <sub>JUNC</sub> <sup>4</sup>	Silicon Junction Temperature			140	°C
T <sub>AMB</sub>	Ambient Operating Temperature	-40		+85	°C

<sup>1</sup>The HCLK frequency may be made to appear at the dedicated CLKOUT pin of the device. For low power operation, however, the CLKOUT pin can be disabled.

<sup>2</sup>The peripherals operate at the HCLK rate, which may be selected to be equal to CCLK or CCLK ÷ 2, up to a maximum of a 75 MHz HCLK for the ADSP-21992BBC.

<sup>3</sup>In order to attain the correct CCLK and HCLK values, the input clock frequency or crystal frequency depends on the internal operation of the clock generation PLL circuit and the associated frequency ratio.

<sup>4</sup>The maximum junction temperature is limited to 140°C in order to meet all of the electrical specifications. It is ultimately the responsibility of the user to ensure that the power dissipation of the ADSP-21992 (including all dc and ac loads) is such that the maximum junction temperature limit of 140°C is not exceeded.

**Table 6. Recommended Operating Conditions—ADSP-21992YBC**

Parameter	Conditions	Min	Typ	Max	Unit
V <sub>DDINT</sub>	Internal (Core) Supply Voltage	2.375	2.5	2.625	V
V <sub>DDEXT</sub>	External (I/O) Supply Voltage	3.135	3.3	3.465	V
AVDD	Analog Supply Voltage	2.375	2.5	2.625	V
CCLK	DSP Instruction Rate, Core Clock	0		150	MHz
HCLK <sup>1, 2</sup>	Peripheral Clock Rate	0		75	MHz
CLKIN <sup>3</sup>	Input Clock Frequency	0		150	MHz
T <sub>JUNC</sub> <sup>4</sup>	Silicon Junction Temperature			140	°C
T <sub>AMB</sub>	Ambient Operating Temperature	-40		+125	°C

<sup>1</sup>The HCLK frequency may be made to appear at the dedicated CLKOUT pin of the device. For low power operation, however, the CLKOUT pin can be disabled.

<sup>2</sup>The peripherals operate at the HCLK rate, which may be selected to be equal to CCLK or CCLK ÷ 2, up to a maximum of an 75 MHz HCLK for the ADSP-21992YBC.

<sup>3</sup>In order to attain the correct CCLK and HCLK values, the input clock frequency or crystal frequency depends on the internal operation of the clock generation PLL circuit and the associated frequency ratio.

<sup>4</sup>The maximum junction temperature is limited to 140°C in order to meet all of the electrical specifications. It is ultimately the responsibility of the user to ensure that the power dissipation of the ADSP-21992 (including all dc and ac loads) is such that the maximum junction temperature limit of 140°C is not exceeded.



**Table 7. Recommended Operating Conditions—ADSP-21992BST**

Parameter	Conditions	Min	Typ	Max	Unit
V <sub>DDINT</sub>	Internal (Core) Supply Voltage	2.375	2.5	2.625	V
V <sub>DDEXT</sub>	External (I/O) Supply Voltage	3.135	3.3	3.465	V
AVDD	Analog Supply Voltage	2.375	2.5	2.625	V
CCLK	DSP Instruction Rate, Core Clock	0		160	MHz
HCLK <sup>1, 2</sup>	Peripheral Clock Rate	0		80	MHz
CLKIN <sup>3</sup>	Input Clock Frequency	0		160	MHz
T <sub>JUNC</sub> <sup>4</sup>	Silicon Junction Temperature			140	°C
T <sub>AMB</sub>	Ambient Operating Temperature	-40		+85	°C

<sup>1</sup>The HCLK frequency may be made to appear at the dedicated CLKOUT pin of the device. For low power operation, however, the CLKOUT pin can be disabled.

<sup>2</sup>The peripherals operate at the HCLK rate, which may be selected to be equal to CCLK or CCLK÷2, up to a maximum of a 80 MHz HCLK for the ADSP-21992BST.

<sup>3</sup>In order to attain the correct CCLK and HCLK values, the input clock frequency or crystal frequency depends on the internal operation of the clock generation PLL circuit and the associated frequency ratio.

<sup>4</sup>The maximum junction temperature is limited to 140°C in order to meet all of the electrical specifications. It is ultimately the responsibility of the user to ensure that the power dissipation of the ADSP-21992 (including all dc and ac loads) is such that the maximum junction temperature limit of 140°C is not exceeded.

**Table 8. Recommended Operating Conditions—ADSP-21992YST**

Parameter	Conditions	Min	Typ	Max	Unit
V <sub>DDINT</sub>	Internal (Core) Supply Voltage	2.375	2.5	2.625	V
V <sub>DDEXT</sub>	External (I/O) Supply Voltage	3.135	3.3	3.465	V
AVDD	Analog Supply Voltage	2.375	2.5	2.625	V
CCLK	DSP Instruction Rate, Core Clock	0		100	MHz
HCLK <sup>1, 2</sup>	Peripheral Clock Rate	0		50	MHz
CLKIN <sup>3</sup>	Input Clock Frequency	0		100	MHz
T <sub>JUNC</sub> <sup>4</sup>	Silicon Junction Temperature			140	°C
T <sub>AMB</sub>	Ambient Operating Temperature	-40		+125	°C

<sup>1</sup>The HCLK frequency may be made to appear at the dedicated CLKOUT pin of the device. For low power operation, however, the CLKOUT pin can be disabled.

<sup>2</sup>The peripherals operate at the HCLK rate, which may be selected to be equal to CCLK or CCLK÷2, up to a maximum of an 50 MHz HCLK for the ADSP-21992YST.

<sup>3</sup>In order to attain the correct CCLK and HCLK values, the input clock frequency or crystal frequency depends on the internal operation of the clock generation PLL circuit and the associated frequency ratio.

<sup>4</sup>The maximum junction temperature is limited to 140°C in order to meet all of the electrical specifications. It is ultimately the responsibility of the user to ensure that the power dissipation of the ADSP-21992 (including all dc and ac loads) is such that the maximum junction temperature limit of 140°C is not exceeded.

# ADSP-21992

**Table 9. Electrical Characteristics—ADSP-21992BBC**

Parameter	Conditions	Test Conditions	Min	Typ	Max	Unit
V <sub>IH</sub>	High Level Input Voltage <sup>1</sup>	@ V <sub>DDEXT</sub> = Maximum	2.0		V <sub>DDEXT</sub>	V
V <sub>IH</sub>	High Level Input Voltage <sup>2</sup>	@ V <sub>DDEXT</sub> = Maximum	2.2		V <sub>DDEXT</sub>	V
V <sub>IL</sub>	High Level Input Voltage <sup>1,2</sup>	@ V <sub>DDEXT</sub> = Minimum			0.8	V
V <sub>OH</sub>	High Level Output Voltage <sup>3</sup>	@ V <sub>DDEXT</sub> = Minimum, I <sub>OH</sub> = -0.5 mA	2.4			V
V <sub>OL</sub>	Low Level Output Voltage <sup>3</sup>	@ V <sub>DDEXT</sub> = Minimum, I <sub>OL</sub> = 2.0 mA			0.4	V
I <sub>IH</sub>	High Level Input Current <sup>4</sup>	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 3.6 V			10	μA
I <sub>IH</sub>	High Level Input Current <sup>5</sup>	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 3.6 V			150	μA
I <sub>IH</sub>	High Level Input Current <sup>6</sup>	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 3.6 V			10	μA
I <sub>IL</sub>	Low Level Input Current	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 0 V			10	μA
I <sub>IL</sub>	Low Level Input Current	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 0 V			10	μA
I <sub>IL</sub>	Low Level Input Current	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 0 V			150	μA
I <sub>OZH</sub>	Three-State Leakage Current <sup>7</sup>	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 3.6 V			10	μA
I <sub>OZL</sub>	Three-State Leakage Current <sup>7</sup>	@ V <sub>DDINT</sub> = Maximum, V <sub>IN</sub> = 0 V			10	μA
C <sub>I</sub>	Input Pin Capacitance	f <sub>IN</sub> = 1 MHz		10		pF
C <sub>O</sub>	Output Pin Capacitance	f <sub>IN</sub> = 1 MHz		10		pF
I <sub>DD-PEAK</sub>	Supply Current (Internal) <sup>8,9</sup>			190	325	mA
I <sub>DD-TYP</sub>	Supply Current (Internal) <sup>8</sup>			155	275	mA
I <sub>DD-IDLE</sub>	Supply Current (Idle) <sup>8</sup>			145	250	mA
I <sub>DD-STOPCLK</sub>	Supply Current (Power-Down) <sup>8,10</sup>			60	125	mA
I <sub>DD-STOPALL</sub>	Supply Current (Power-Down) <sup>8,11</sup>			12	40	mA
I <sub>DD-PDOWN</sub>	Supply Current (Power-Down) <sup>8,12</sup>			6	30	mA
I <sub>AVDD</sub>	Analog Supply Current <sup>13</sup>			46	65	mA
I <sub>AVDD-ADCOFF</sub>	Analog Supply Current <sup>12</sup>			5	15	mA

<sup>1</sup> Applies to all input and bidirectional pins.

<sup>2</sup> Applies to input pins CLKIN, RESET, TRST.

<sup>3</sup> Applies to all output and bidirectional pins.

<sup>4</sup> Applies to all input only pins.

<sup>5</sup> Applies to input pins with internal pull-down.

<sup>6</sup> Applies to input pins with internal pull-up.

<sup>7</sup> Applies to three-stateable pins.

<sup>8</sup> The I<sub>DD</sub> supply currents are affected by the operating frequency of the device. The guaranteed numbers are based on an assumed CCLK = 150 MHz, HCLK = 75 MHz for the ADSP-21992BBC. I<sub>DD</sub> refers only to the current consumption on the internal power supply lines (V<sub>DDINT</sub>). The current consumption at the I/O on the V<sub>DDEXT</sub> power supply is very much dependent on the particular connection of the device in the final system.

<sup>9</sup> I<sub>DD-PEAK</sub> represents worst-case processor operation and is not sustainable under normal application conditions. Actual internal power measurements made using typical applications are less than specified. Measured at V<sub>DDINT</sub> = maximum.

<sup>10</sup> IDLE denotes the current consumption during execution of the IDLE instruction. Measured at V<sub>DDINT</sub> = maximum.

<sup>11</sup> I<sub>DD-PDOWN</sub> represents the processor operation in full power-down mode with both core and peripheral clocks disabled. Measured at V<sub>DDINT</sub> = maximum.

<sup>12</sup> I<sub>AVDD</sub> represents the power consumption of the analog system. Measured at AVDD = maximum.

<sup>13</sup> The responsibility lies with the user to ensure that the device is operated in such a manner that the maximum allowable junction temperature is not exceeded.