1 INTRODUCTION

Overview

This hardware specification contains architectural information required for designing of TigerSHARC® DSP-based systems. A separate document, the *TigerSHARC® DSP Instruction Set Specification*, contains the instruction set description required for programming TigerSHARC®-based systems. In addition to this manual, hardware designers should refer to the *TigerSHARC® Data Sheet* for timing, electrical, and package specifications.

The TigerSHARC® 128-bit digital signal processor is a high-performance next-generation version of the ADSP-2106x SHARC. The TigerSHARC® DSP sets a new standard of performance for digital signal processors, combining multiple computation units for floating-point and fixed-point processing as well as very wide word widths. The TigerSHARC® DSP maintains a 'system-on-a-chip' scalable computing design philosophy, including a 6-Mbit, on-chip SRAM, integrated I/O peripherals, a host processor interface, DMA controllers, link ports and shared-bus connectivity for glueless multiprocessing.

Besides providing unprecedented performance in DSP applications in raw MFLOPS and MIPS, the TigerSHARC® DSP boosts performance measures such as MFLOPS/Watt and MFLOPS/square inch in multiprocessing applications.

The processor operates with a two-cycle arithmetic pipeline. The branch pipeline is two to six cycles and, because of this deep pipeline, there is a

Branch Target Buffer (BTB) to reduce branch delay. The two identical computation units support both floating-point and fixed-point arithmetic.

High performance is facilitated by the ability to execute up to four 32-bit wide instructions per cycle. The TigerSHARC® processor uses a variation of a *static superscalar* architecture to allow the programmer to specify which instructions are executed in parallel in each cycle. The instructions do not have to be aligned in memory so that program memory is not wasted.

The 6-Mbit internal memory is divided into three 128-bit wide memory blocks. Each of the three internal address/data bus pairs connect to one of the three memory blocks. These memory blocks can be used for triple accesses every cycle where each memory block can access up to four, 32-bit words in a cycle.

The external port cluster bus is 64 bits wide. The high I/O bandwidth complements the high processing speeds of the core. To facilitate the high clock rate, the TigerSHARC® DSP uses a pipelined external bus with programmable pipeline depth for interprocessor communications and for Synchronous SRAM and DRAM (SSRAM and SDRAM).

The four link ports support point-to-point high bandwidth data transfer. Link ports have hardware supported, two-way communication.

Figure 1-1 on page 1-4 illustrates the microarchitecture of the TigerSHARC® DSP. This detailed block diagram shows the following architectural features:

- Dual computation blocks, X and Y, each consisting of a multiplier, ALU, shifter and a 32-word register file.

- Dual integer ALUs, J and K, each containing a 32-bit ALU and 32-word register file.

- Program Sequencer, which controls the program flow. The Program Sequencer contains an instruction alignment buffer (IAB) and a Branch Target Buffer (BTB).
- Three 128-bit buses that provide high bandwidth connectivity between all blocks of 48 bytes per cycle.
- External port interface that includes the Host Interface, SDRAM controller, static pipelined interface, four DMA channels, four link ports, each with two DMA channels, and multiprocessing support.
- Three internal memory blocks, M0, M1, and M2, each 128 bits wide.
- Debug features
- JTAG test access port

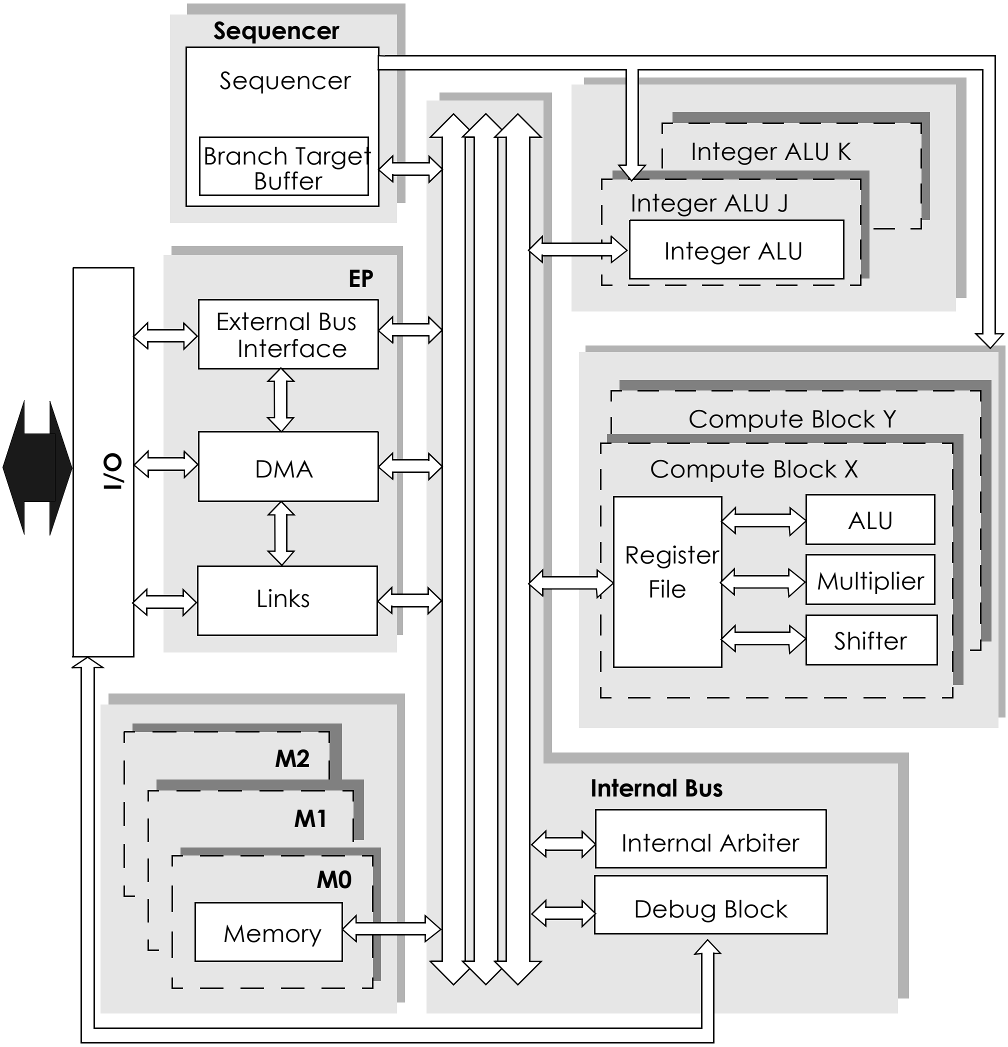

Figure 1-1. Chip Level Block Diagram

The TigerSHARC® DSP external port provides an interface to external memory, memory-mapped I/O, host processor, and additional Tiger-SHARC® DSPs. The external port performs external bus arbitration as well as supplying control signals to shared, global memory and I/O devices.

Figure 1-2 illustrates a typical single-processor system. Multiprocessor systems are illustrated in Figure 1-3 on page 1-6, and discussed later in "Multiprocessing" on page 6-74.

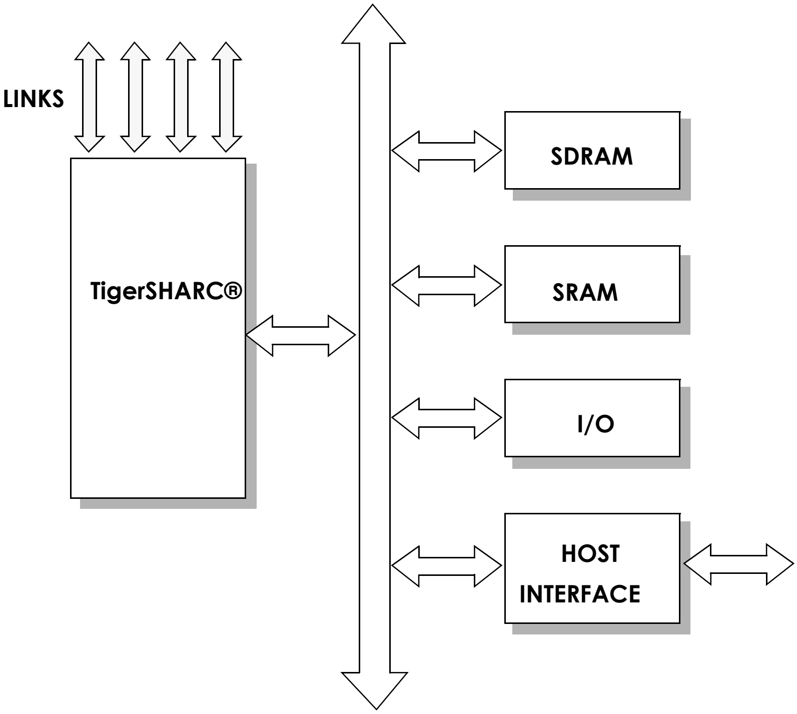

Figure 1-2. Single Processor Configuration

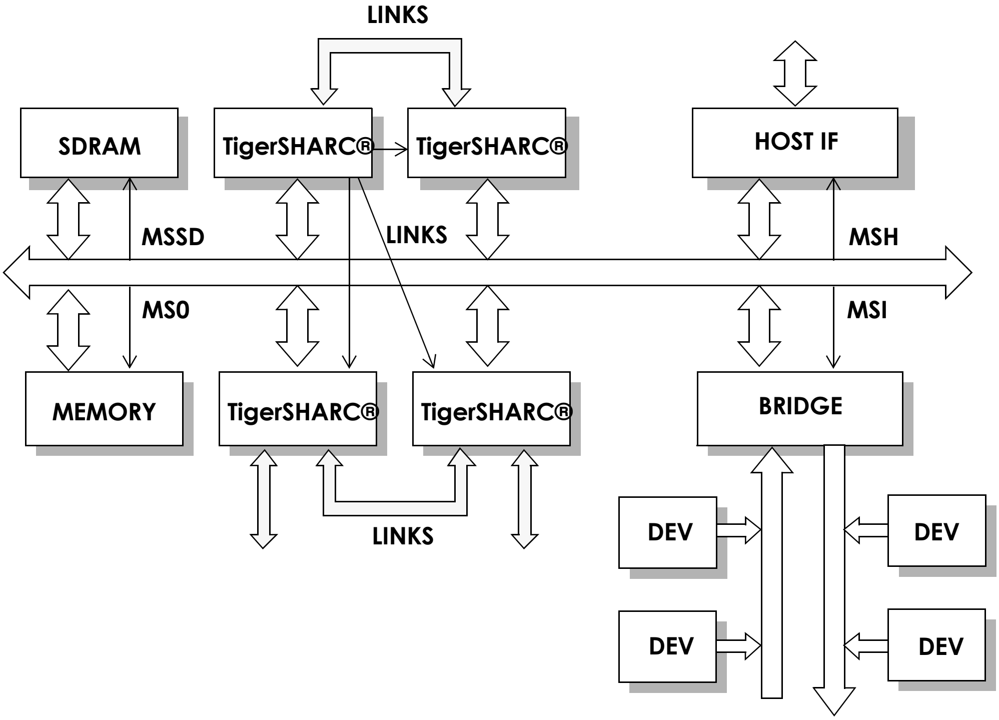

Figure 1-3. Multiprocessing Cluster Configurations

Key Architectural Features

The TigerSHARC® DSP's key architectural features include the following:

- Parallel operations
- Internal memories
- Quad instruction execution
- Scalability and multiprocessing

These features are outlined in the following subsections.

Parallel Operations

During compute-intensive operations, one or both integer ALUs compute or generate addresses for fetching up to two quad operands from two memory blocks, while the program sequencer simultaneously fetches the next quad instruction from the third memory block. In parallel, the computation units can operate on previously-fetched operands while the sequencer prepares for a branch.

While the core processor is performing the steps described above, the DMA channels are able to replenish the internal memories in the background with quad data from either the external port or the link ports.

Core Processor Specifications

The processing core of the TigerSHARC® DSP reaches exceptionally high performance due to the following:

- Computation pipeline
- Dual computation units

- Execution of up to four instructions per cycle
- Access of up to eight words per cycle from memory

The two computation units perform up to six floating-point or 24 fixed-point operations per cycle.

Each multiplier and ALU unit can execute four 16-bit fixed-point operations per cycle (SIMD). This boosts performance of critical imaging and signal processing applications that use fixed-point data.

Internal Memories

The on-chip memory consists of three blocks of two Mbits each. Each block is 128 bits (four words) wide, which provides high bandwidth sufficient to support both computation units, the instruction stream and external I/O, even in very intensive operations. The TigerSHARC® DSP provides access to a program and two data operands without memory or bus constraints. The memory blocks can store instructions and data interchangeably.

Quad Instruction Execution

The TigerSHARC® DSP can execute up to four instructions per cycle from a single memory block, due to the 128-bit wide access per cycle. The ability to execute several instructions in a single cycle arises from a "static superscalar" architectural concept. Static superscalar is not strictly a superscalar architecture because the instructions executed in each cycle are specified by the programmer or by the compiler, and not by the chip hardware. There is also no instruction re-ordering. Register dependencies are, however, examined by the hardware and stalls are generated where required. Code is fully compacted in memory with no alignment restrictions for instruction lines.

Quad Data Access

Instructions specify if one, two, or four words are to be loaded or stored. Generally, quad words must be aligned on a quad word boundary, and long words aligned on a long word boundary. Meeting this requirement, however, is not necessary when loading data to computation units via the data alignment buffer (DAB), because the DAB can align quad words that are not aligned in memory.

Up to four data words from each memory block can be supplied to each computation unit, meaning that new data are not required on every cycle, leaving alternate cycles for I/O to the memories. This is beneficial in applications with high I/O requirements since it allows the I/O to occur without degrading core processor performance.

Scalability and Multiprocessing

Like its predecessor, the SHARC, the TigerSHARC® DSP is designed for multiprocessing applications. The primary multiprocessing architecture supported is a cluster of up to eight TigerSHARC® DSPs that share a common bus, a global memory, and an interface to either a host processor or to other clusters. This is discussed in “Multiprocessing” on page 6-74. In large multiprocessing systems this cluster can be considered as an element and connected in configurations such as toroid, mesh, tree, crossbar, or others. The system designer can provide a custom interconnect method or use the on-chip link ports.

The TigerSHARC® processor improves on most of the multiprocessing capabilities of the SHARC DSPs. These capabilities include the following:

- On-chip bus arbitration for glueless multiprocessing
- Globally-accessible internal memory and registers
- Semaphore support
- Powerful, in-circuit multiprocessing emulation

System Level Enhancements

The TigerSHARC® DSP includes several enhancements that simplify system development. The enhancements lie in three key areas:

- Architectural features supporting high-level languages and operating systems
- IEEE 1149.1 JTAG serial scan path and on-chip emulation features
- Support of IEEE floating-point formats

High Level Languages

The TigerSHARC® processor architecture has several features that directly support high-level language compilers and operating systems:

- Simple, orthogonal instruction set allowing the compiler to use the multi-instruction slots efficiently
- General-purpose data and IALU register files
- 32- and 40-bit floating point
- 8-, 16-, 32-, and 64-bit integer data types
- 32-bit (4 gigaword) address space
- Immediate address modify fields
- Easily-supported, relocatable code and data
- Fast save and restore of processor registers onto internal memory stacks

Serial Scan and Emulation Features

The TigerSHARC® DSP supports the IEEE standard P1149.1 Joint Test Action Group (JTAG) standard for system test. This standard defines a method for serially scanning the I/O status of each component in a system. The JTAG serial port is also used by the TigerSHARC® DSP EZ-ICE to gain access to the processor's on-chip emulation features.

IEEE Formats

The TigerSHARC® processor is compatible with the IEEE single-precision floating-point data format in all respects, except for the following:

- The TigerSHARC® DSP does not provide inexact flags.
- NAN inputs generate an invalid exception and return a quiet NAN.
- Denormal operands are flushed to zero when input to a computation unit and do not generate an underflow exception. Any denormal or underflow result from an arithmetic operation is flushed to zero and an underflow exception is generated.
- Round-to-nearest and round-towards-zero are supported. Round-to-infinity is not supported.

TigerSHARC® Core Architecture Blocks

The following sections summarize the features of the TigerSHARC® DSP architecture. These features are described in greater detail in the following sections.

Compute Blocks

The TigerSHARC® core processor contains two computation units known as compute blocks. Each compute block contains a register file and three independent computation units: an ALU, a multiplier, and a shifter. For meeting a wide variety of processing needs, the computation units process data in several fixed- and floating-point formats:

- *Fixed-point format*
  64 bits (long), 32 bits (word), 16 bits (short) and byte. For short fixed-point arithmetic, quad parallel operations on quad-aligned data allow fast processing of array data. Byte operations are also supported for octal aligned data.
- *Floating-point format*
  Single floating-point and 40-bit floating-point operations are single or extended precision. The single floating-point format is the standard IEEE format, whereas the 40-bit extended-precision format occupies a double word (64 bits) and has eight additional LSBs of mantissa for greater accuracy.

ALU

The ALU performs a standard set of arithmetic and logic operations in both fixed-point and floating-point formats.

Multiplier

The multiplier performs floating-point and fixed-point multiplication as well as fixed-point multiply-and-accumulate.

Shifter

The shifter performs logical and arithmetic shifts, bit manipulation, field deposit and extraction.

Register File

A general-purpose, multiport 32-word data register file in each compute block is used for transferring data between the computation units and the data buses, and for storing intermediate results. All of these registers can be accessed as single-, dual-, or quad-aligned registers.

Execution Flow

The computation units perform single-cycle operations with a two-cycle computation pipeline, meaning that results are available for use two cycles after the operation executes. Hardware causes a stall if a result is not available in a given cycle (register dependency check). A maximum of two computation instructions per compute block can be issued in each cycle, instructing the ALU, multiplier or shifter to perform independent, simultaneous operations.

Integer ALUs

The IALUs provide memory addresses when data are transferred between memory and registers. The IALUs allow computational operations to execute with maximum efficiency since the computation units can be devoted exclusively to processing data. Dual IALUs enable simultaneous addresses for multiple operand reads or writes.

Each IALU has a multiport, 32-word register file. Operations in the IALU are not pipelined. The IALUs also support pre-modify with no update, and post-modify with update address generation as well as circular buffer implementation in hardware.

For indirect addressing, one of the registers in the register file can be modified by another register in the file, or by an immediate 8- or 32-bit value, either before (pre-modify) or after (post-modify) the access. For circular buffer addressing, a length value can be associated with the first four registers to perform automatic modulo addressing for circular data buffers; the circular buffers can be located at arbitrary boundaries in memory. Circular buffers allow efficient implementation of delay lines and other data structures, commonly used in digital filters and Fourier transformations. The

TigerSHARC® processor's circular buffers automatically handle address pointer wraparounds, reducing overhead and simplifying implementation.

Program Sequencer

The program sequencer supplies instruction addresses to memory and, together with the IALUs, allows computational operations to execute with maximum efficiency. It supports efficient branching using the Branch Target Buffer (BTB), which reduces branch delays for conditional and unconditional instructions.

The TigerSHARC® DSP has four general-purpose external interrupts, IRQ_{3-0}. The processor also has internally-generated interrupts for the two timers, DMA channels, link ports, arithmetic exceptions, multiprocessor vector interrupts, and user-defined software interrupts. Interrupts can be nested through instruction commands. The interrupts have a short latency, and do not abort instructions that are currently executing. Interrupts vector directly to a user-supplied address in the interrupt table register file.

Internal Buses

The processor core has three buses, each connected to one of the internal memories. These buses are 128 bits wide to allow up to four instructions or four aligned data words to be transferred in each cycle on each bus. External ports and the on-chip system elements of the other link ports also use these buses to access memory. Only one access to each memory block is allowed in each cycle, so DMA or external port transfers must compete with core accesses on the same block. However, because of the large bandwidth available from each block, not all the memory bandwidth can be used by the core units, leaving some bandwidth available for use by transfers by DMA or other bus interface masters.

Quad Data Accesses

Each move instruction specifies whether a single, dual or quad word is accessed from each memory block. Two memory blocks can be accessed on each cycle because of the two IALUs. Long word accesses can be used to supply two aligned words to one compute block or one aligned word to each compute block. Quad word accesses may be used to supply four aligned words to one compute block or two aligned words to each compute block. This is useful in applications that use complex (real/imaginary) data, or parallel data sets that can be aligned in memory. It is also used for fast save/restore of context during C calls or interrupts.

Internal Memory

The TigerSHARC® processor contains three two-Mbit blocks of on-chip, 128-bit wide SRAM.

Each memory block is organized as 64K words of 32-bits each. The accesses are pipelined to meet one clock cycle access time needed by the core, DMA, or the external bus. Each access can be up to four words. Memories and their associated buses must be shared among the compute blocks, the IALUs, the sequencer, the external port, and the link ports. In general, if during a particular cycle more than one unit in the processor attempts to access the same memory, one of the competing units is granted access, while the other is held off for further arbitration until the following cycle. See “Bus Arbitration Protocol” on page 6-78. The very high bandwidth of the internal buses insures that this type of conflict only has a small effect on performance.

An important benefit of large on-chip memory is the high levels of determinism in execution time that the system designer can realize by managing the movement of data on- and off-chip with DMA. Predictable and deterministic execution time is a central requirement in DSP and real-time systems.

External Port

External Bus and Host Interface

The TigerSHARC® DSP external port (EP) provides an interface between the core processor and the 32/64-bit parallel external bus. It contains FIFOs that maintain the throughput of an external bus that is shared with multiple processors and peripherals—each of which may operate at speeds other than that of the core.

The most effective way to access external data in the TigerSHARC® processor is through the DMA. This runs in the background, allowing the core to continue processing while new data are read in or processed data are written out. Multiple DMA data streams can occur simultaneously, and the use of FIFOs helps to maintain throughput in the system.

Burst accesses are provided through the BRST pin, which allows a slave device on the bus to accept the first address and then automatically increment that address as successive data words arrive. This implements a shorthand DMA transfer, since no length information is required.

External Memory

The TigerSHARC® DSP external port provides the processor interface to off-chip memory and peripherals. The off-chip memory and peripherals are included in the TigerSHARC® processor unified address space. The separate on-chip buses are multiplexed at the external port to create an external system bus with a single 32-bit address bus and a single 64-bit data bus. External memory and devices can be either 32 or 64 bits wide. The TigerSHARC® DSP automatically packs external data into either 32-, 64- or 128-bit word widths, the latter being more efficient for reducing memory access conflicts.

On-chip decoding of high-order address lines (to generate memory block select signals) facilitates addressing of external memory devices. Separate control lines are also generated for simplified addressing of page-mode DRAM.

The TigerSHARC® DSP uses the address on the external port bus to pipeline the data. This allows interfacing to synchronous DRAM and speeds up interprocessor accesses. An option allows asynchronous operation for slower devices.

External data can be accessed by DMA channels or by the core. For core accesses, the read latency can be significant—eight or more cycles. The core provides I/O buffering by stalling if the data are accessed before they are loaded in a universal register (Ureg).

Programmable memory wait states permit peripherals with different pipeline delay cycles, access, hold, and disable time requirements.

External shared memory resources are assigned between processors by using simple semaphore operations.

Multiprocessing

The TigerSHARC® DSP offers the following features tailored to multiprocessing systems:

- The unified address space allows direct interprocessor accesses of each TigerSHARC® processor internal memory and resources.
- Distributed bus arbitration logic is included on-chip for simple, glueless connection of systems containing up to eight TigerSHARC® DSPs and a host processor.
- Bus arbitration rotates, except for host requests that always hold the highest priority.
- Processor bus lock allows indivisible read-modify-write sequences for semaphores.

- A vector interrupt capability is provided for interprocessor commands.
- Broadcast writes allow simultaneous transmissions of data to all TigerSHARC® DSPs.

Host Interface

Connecting a host processor to a cluster of TigerSHARC® DSPs is simplified by the memory-mapped nature of the interface bus and the availability of special host bus request signals.

A host that is able to access a pipelined memory interface can be easily connected to the parallel TigerSHARC® DSP bus. All the internal memory, Uregs, and resources within the TigerSHARC® DSP, such as the DMA control registers and the internal memory, are accessible to the host.

The host interface is through the TigerSHARC® DSP external address and data bus, with additional lines being provided for host control. The protocol is similar to the standard TigerSHARC® DSP pipelined bus protocol.

The host becomes bus master of the cluster by asserting the Host Bus Request (HBR) signal. Host Bus Grant (HBG) is returned by the TigerSHARC® processors when the bus becomes available. The host interface is synchronous, and can be delayed a number of cycles to allow slow host access. The host can also access external memory directly.

All DMA channels are accessible to the host interface, allowing code and data transfers to be accomplished with low software overhead. The host can directly read and write the internal memory of the TigerSHARC® DSP and can access the DMA channel setup. Vector interrupt support is provided for efficient execution of host commands and burst-mode transfers.

DMA Controller

The TigerSHARC® DSP on-chip DMA controllers allows zero-overhead data transfers without processor intervention. The TigerSHARC® processor can simultaneously fetch instructions and access two memories for data without relying on data or instruction caches. The DMA controllers operate independently of the processor core, supplying addresses for internal and external memory access. The DMA channels, therefore, are not part of the core processor from a programming point of view.

Both code and data can be downloaded to the TigerSHARC® DSP using DMA transfers, which can occur between the following:

- TigerSHARC® DSP internal memory and external memory, external peripherals or a host processor
- External memory and external peripheral devices
- External memory and link ports or between two link ports

Six DMA channels are available on the TigerSHARC® processor for data transfers through the External Port. Eight DMA channels are available for link data transfers (two per link).

Asynchronous off-chip peripherals can control any one of four DMA channels using DMA request lines (`DMAR[3:0]`). Other DMA features include fly-by (for channel 0 only), interrupt generation upon completion of DMA transfers, and DMA chaining for automatically linked DMA transfers.

Link Ports

The TigerSHARC® DSP has four 8-bit link ports that provide additional I/O capabilities in multiprocessing systems. The link ports have the following characteristics:

- Link clock speed is selectable as either 1/8, 1/4, 1/3, or 1/2 of internal clock frequency.
- Link port data are packed into 128-bit words for DMA transfer to on- or off-chip memory.
- Each link port has its own buffer registers.
- Link port transfers are controlled by clock/acknowledge handshaking.
- Link ports support bidirectional transfer and flow through and transfers to/from the external port or other links.

Programming Model

Instruction Set

The TigerSHARC® DSP instruction set provides a wide variety of programming capabilities. The execution of up to four instructions in parallel enables the use of simultaneous computations with data transfers and branching or looping. These operations can be in any combinations with few restrictions.

The IALU provides flexibility in moving data as normal, long, or quad words. Every instruction can execute with a throughput of one per cycle and with one or two cycles of latency. IALU instructions execute with a single-cycle of latency while computation units have two cycles of latency. The processor implements a static branch prediction mechanism—correctly predicted branches incur no overhead cycles; incorrectly predicted branches incur a penalty of three to six cycles.

The TigerSHARC® DSP assembly language is based on an algebraic syntax for easy coding and readability. A comprehensive set of development tools supports program development.

Relative Addresses For Relocation

Most instructions in the TigerSHARC® DSP support PC relative branches to allow code to be relocated easily. Also, most data references allow programs to access data blocks relative to a base register.

Conditional Execution

All instructions can be executed conditionally. The condition field exists in one of the instructions in an instruction line, and the remaining instructions in that line are executed or not depending on the outcome of the condition.

Internal Transfer

Most registers of the TigerSHARC® DSP are classified as universal registers (Uregs). Instructions are provided for transferring data between any two Uregs, between a Ureg and memory, or for the immediate load of a Ureg. This includes control registers and status registers, as well as the data registers in the universal register files. These transfers occur with the same timing as internal memory load/store.

Context Switching

The TigerSHARC® DSP provides the ability to save and restore up to eight registers per cycle onto a stack in two internal memory blocks when using load/store instructions. This fast save/restore capability permits efficient interrupts and fast context switching. It also allows the TigerSHARC® processor to dispense with on-chip PC stack or alternate registers for register files or status registers.

Nested Call and Interrupt

Nested call and interrupt return addresses (along with other registers as needed) are saved by specific instructions onto the on-chip memory stack, allowing more generality when used by high level languages. Non-nested calls and interrupts do not need to save the return address in internal memory, making these more efficient for short, non-nested routines.

Branch Target Buffer

The TigerSHARC® DSP has an eight-cycle deep pipeline. The branch penalty in a deeply pipelined processor such as the TigerSHARC® DSP can be compensated for by the use of a Branch Target Buffer (BTB) and branch prediction.

The branch target address is stored in the BTB. When the address of a jump instruction or tag address (which in most cases is specified by the user to be taken) is recognized, the corresponding jump address is read from the BTB and is used as the jump address on the next cycle. The latency of a jump is reduced from three to six wasted cycles to zero wasted cycles. If this address is not stored in the BTB, the instruction must be fetched from memory.

Incorrectly predicted branches are expensive in terms of wasted cycles. It is best to use conditional instructions instead of branches whenever possible. All TigerSHARC® processor instructions are conditional.

Other control flow instructions also use the BTB to speed up these types of branches. The instructions are interrupt return, call return, and computed jump instructions.

Booting

The internal memory of the TigerSHARC® DSP can be loaded from an 8-bit EPROM using a boot mechanism at system powerup. It can also boot using another master or through one of the link ports. Selection of the boot source is controlled by external pins.

Miscellaneous

Timers

The TigerSHARC® DSP has two programmable interval timers that provide periodic interrupt generation. When enabled, the timers decrement a 32-bit count register every cycle. When this count register reaches zero, the TigerSHARC® processor generates an interrupt and asserts TMR0E output (for timer zero only). The count register is automatically reloaded from a 32-bit period register and the count resumes immediately.

Clock Domains

There are two major clock domains in the TigerSHARC® DSP, driven by two input clocks—the local clock (LCLK) and the system clock (SCLK).

The AC specification and bus interface are defined in reference to the SCLK. (See the *TigerSHARC® Data Sheet* for the full AC specification.) The internal SCLK is phase locked to the SCLK input by a Phase Locked Loop (PLL).

The LCLK is an input to the internal clock driver—CCLK. The CCLK is the internal clock of the core, internal buses, memory, links, and most of the chip's internal parts. The CCLK is generated by a PLL from LCLK and is phase-locked. The LCLKRAT pins define the clock multiplication of LCLK to CCLK. The clock multiplication can be 2, 2.5, 3, 3.5, 4, 5, and 6.

Systems must connect both LCLK and SCLK to the same clock source. Using an integer LCLKRAT (2, 3, 4, 5, or 6) will guarantee predictable cycle-by-cycle operation (important for SIMD operation and for fault-tolerant systems).

About this Document

The *TigerSHARC DSP Hardware Specification* is intended for designers and others who want to understand the functionality and design of the TigerSHARC® DSP:

- **Chapter 1: Introduction** (*this chapter*) provides an architectural overview to the TigerSHARC® DSP.
- **Chapter 2: I/O Pins** lists the TigerSHARC® DSP external bus pins and briefly describes their functionality.
- **Chapter 3: Memory and Register Map** defines the memory map of each element in the system. The memory space defines the location of each element on the TigerSHARC® DSP.
- **Chapter 4: Core Controls** discusses clocking inputs, including the three different types of operating modes in which the TigerSHARC® DSP can operate and the Boot modes from which the TigerSHARC® processor initiates.
- **Chapter 5: Interrupts** discusses the various types of interrupts TigerSHARC® DSP supports, some of which are internally generated and some externally generated.
- **Chapter 6: Cluster Bus** focuses on the external bus interface of the TigerSHARC® DSP, which includes the bus arbitration logic and the external address, data and control buses.
- **Chapter 7: Direct Memory Access** describes how the TigerSHARC® DSP's on-chip DMA controller acts as a machine for transferring data without core interruption.
- **Chapter 8: Link Ports** describes how link ports provide point-to-point communications between TigerSHARC® DSPs in a system, or can also be used to interface any other device that is designed to work in the same protocol.

- **Chapter 9: Debug Functionality** describes features of the TigerSHARC® DSP useful for performing both software debugging and services usually found in Operating System (OS) Kernels.

This specification is a companion document to the *TigerSHARC® DSP Instruction Set Specification*.

Additional Literature

The following publications can be ordered from any Analog Devices sales office:

- *TigerSHARC® DSP Data Sheet*
- *TigerSHARC® DSP Instruction Set Specification*
- *TigerSHARC® Family Hardware & Software Development Tools Data Sheet*
- *TigerSHARC® Family Assembler Tools & Simulator Manual*
- *TigerSHARC® Family C Tools Manual*
- *TigerSHARC® Family C Runtime Library Manual*