ADSP-TS001 TigerSHARC™ DSP

1.2 Billion MACs-per-Second Static Superscalar DSP

KEY FEATURES:

STATIC SUPERSCALAR ARCHITECTURE OPTIMIZED FOR TELECOMMUNICATIONS INFRASTRUCTURE

- Eight 16-bit MACs/cycle with 40-bit accumulation
- Two 32-bit MACs/cycle with 80-bit accumulation
- Only one half a cycle required per Add, Compare, Select (ACS) sequence for Viterbi algorithm
- Add-subtract instruction and bit reversal in hardware for FFTs
- IEEE-floating point compatible

HIGHLY INTEGRATED

- 6 Mbit on-chip SRAM
- Glueless multiprocessing
- Four Link Ports – 600 MBytes/sec transfer rate
- 64-bit external port – 600 MBytes/sec
- 14 DMA channels

FLEXIBLE PROGRAMMING IN ASSEMBLY AND C LANGUAGES

- User-defined partitioning between program and data memory
- 128 general-purpose registers
- Algebraic assembly language syntax
- Optimizing C compiler
- VisualDSP tools support
- Single-instruction, multiple-data (SIMD) instructions, or direct issue capability
- Predicated execution
- Fully interruptible with full computation performance

OVERVIEW

The ADSP-TS001 TigerSHARC DSP targets infrastructure equipment with a new level of integration and the unique ability to process 8-, 16- and 32-bit fixed-point and floating-point data types on a single chip. Each of these data types is critical to the next generation of telecommunications protocols currently under development, including IMT-2000 (also known as 3G wireless) and xDSL (Digital Subscriber Line).

In one chip, ADI has integrated six megabits of SRAM (Synchronous Random Access Memory), fixed- and floating-point cores, four bidirectional link ports, a 64-bit external port, fourteen DMA (Direct Memory Access) channels and 128 general-purpose registers. For large scale applications that require clusters of DSPs, ADI has integrated its patented link port technology, enabling direct chip-to-chip connections without the need for complex external circuitry.
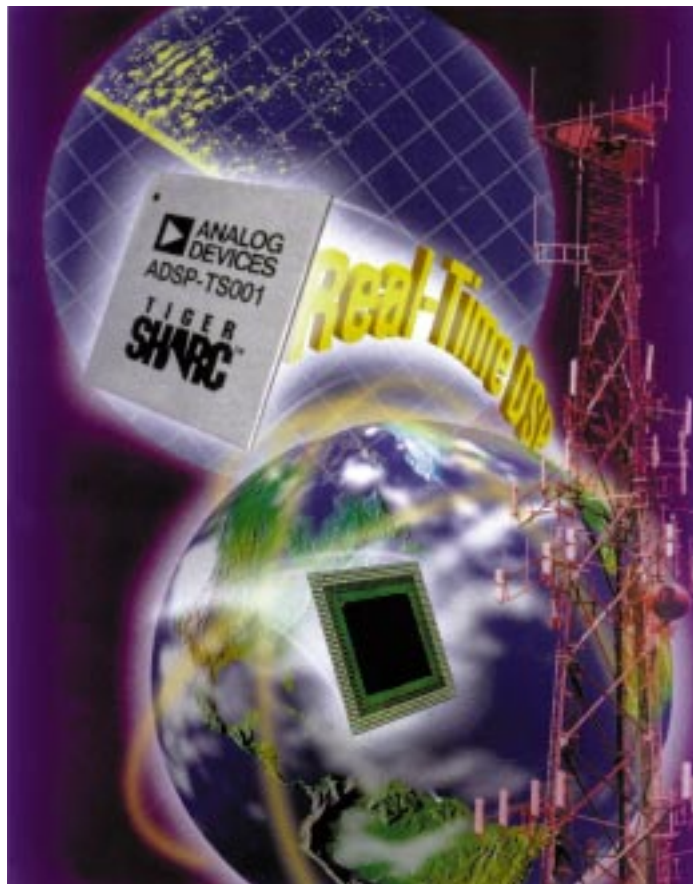
The ADSP-TS001 TigerSHARC DSP processes fixed- and floating-point data on a single chip. It executes 1.2 billion 40-bit MACs per second and achieves the world's highest floating-point DSP performance.

TIGER SHARC™

ANALOG DEVICES

STATIC SUPERSCALAR ARCHITECTURE

The new TigerSHARC DSP architecture blends best practices in microprocessor design to enable the highest performance, programmable DSP for real-time systems.

The TigerSHARC DSP employs a static superscalar architecture. It incorporates many aspects of conventional superscalar processors, including a load/store architecture, branch prediction, and a large, interlocked register file. Up to four instructions can be executed in parallel in each cycle. The term "*static superscalar*" is applied because instruction-level parallelism is determined prior to run-time and encoded in the program.

All the registers are interlocked, supporting a simple programming model that is independent of the implementation latencies and is fully interruptible. Branch prediction is supported via a 128-bit entry Branch Target Buffer (BTB) that reduces branch latency.

EIGHT MACs/CYCLE

There are two computation blocks (Processing Elements X and Y) in the TigerSHARC DSP architecture, each containing a multiplier, ALU, and
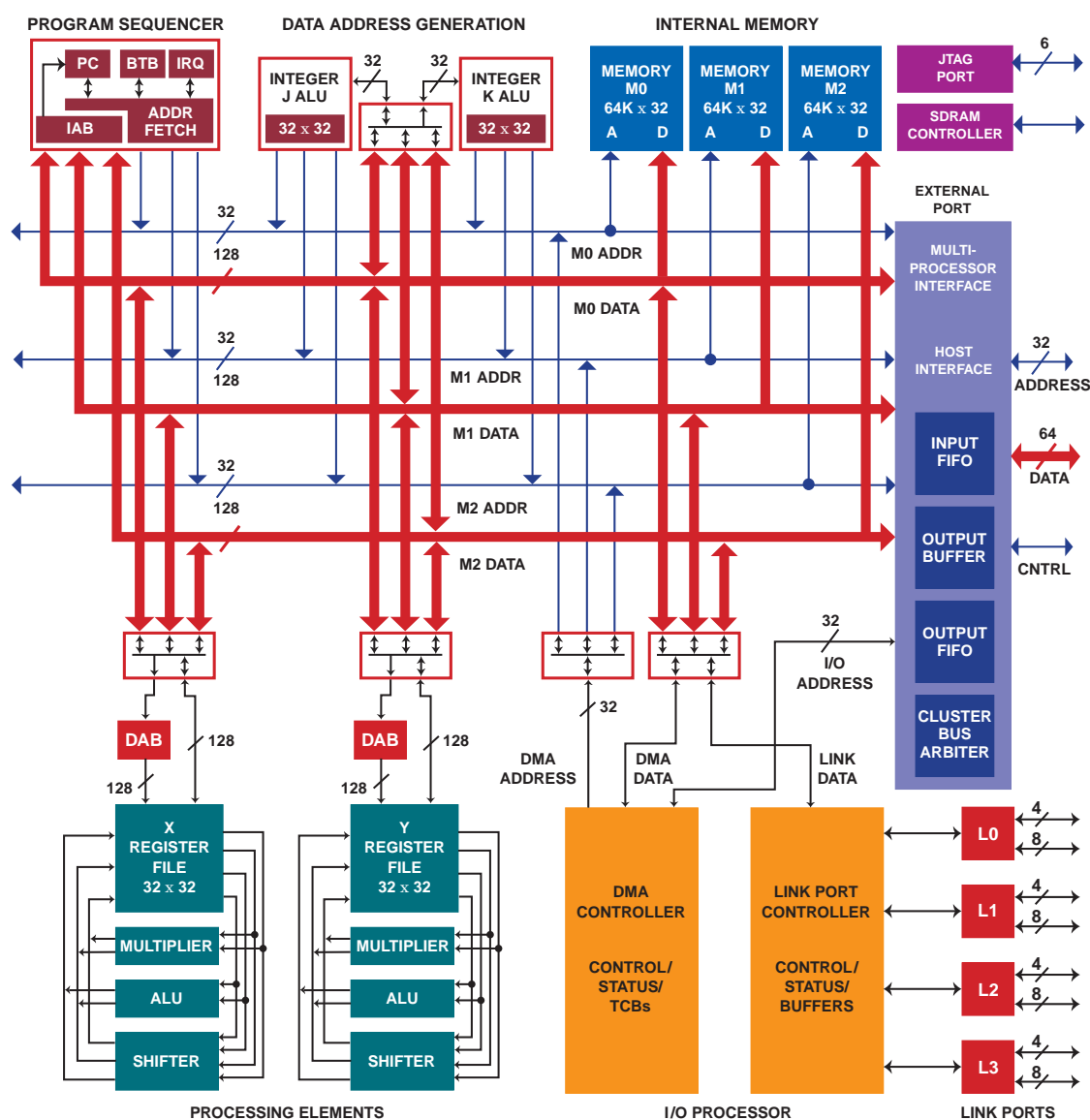
ADSP-TS001 TigerSHARC DSP Block Diagram

64-bit shifter. With the resources in these blocks, it is possible to execute eight 40-bit MACs on 16-bit data, two 40-bit MACs on 16-bit complex data, or two 80-bit MACs on 32-bit data, in a single cycle. With 8-bit data types, the architecture executes 16 operations per cycle.

TigerSHARC DSP is a register-based load/store architecture, where each computation block has access to a fully orthogonal 32-word register file.

MEMORY ARCHITECTURE

The TigerSHARC DSP features a short-vector memory architecture organized internally in three 128-bit wide banks. Quad (four words, 32 bits each), long (two words, 32 bits each), and normal word accesses move data from the memory banks to the register files for operations. In a given cycle, four 32-bit instruction words can be fetched, and 256 bits of data can be loaded to the register files or stored into memory. Data in 8-, 16-, and 32-bit words can be stored in contiguous, packed memory. Internal and external memories are organized in a unified memory map. The partition between program memory and data memory is user-determined. The internal memory bandwidth for data and instructions is 7.2 GBytes/second.

DEVELOPMENT TOOLS: VisualDSP® Environment

TheVisualDSP Integrated Development Environment (IDE) provides the interface to a complete suite of tools, including optimizing C compiler, assembler, linker, cycle-accurate simulator, and debugger.

White Mountain DSP emulators provide easier and more cost-effective methods for engineers to develop and optimize DSP systems, shortening product development cycles for faster time-to-market. The EZ-Kit provides an easy way to investigate the power of the TigerSHARC DSP family and begin to develop applications. This system consists of a PCI evaluation board developed for Analog Devices by Spectrum Signal Processing.

The TigerSHARC DSP platform offers designers a flexible development environment that supports both C and assembly programming. TigerSHARC DSP features robust and efficient C compiler tools, achieving up to 70% compiler efficiency. For time-critical inner loops, DSP programmers turn to the machine's assembly language to produce the highest performance code. The TigerSHARC DSP platform, despite its sophisticated architecture, is practical to program in assembly, with features such as easy-to-learn algebraic assembly language syntax, predictable 2-Cycle Delay for all computations, 128 fully interlocked, general-purpose registers, and branch prediction.

ADSP-TS001 TigerSHARC DSP BENCHMARKS

Peak Rates at 150 MHz

16-bit Performance	1.2 Billion MACs/ second
32-bit Fixed-Point Performance	300 Million MACs/ second
32-bit Floating-Point Performance	900 MFLOPS

16-Bit Algorithms	Execution Time at 150 MHz	Clock Cycles
256-pt. Complex FFT (Radix 2) (includes bit reversal)	7.3 µs	1100
50-Tap FIR on 1024 Input	48 µs	7200
Single FIR MAC	0.93 ns	0.14
Single Complex FIR MAC	3.80 ns	0.57

32-Bit Algorithms		
1024-pt. Complex FFT (Radix 2) (includes bit reversal)	69 µs	10,300
50-Tap FIR on 1024 Input	184 µs	27,500
Single FIR MAC	3.7 ns	0.55

INSTRUCTION SET SUMMARY

The instruction set directly supports all DSP, image, and video processing arithmetic types including signed, unsigned, fractional, and integer data types. There is optional saturation (clipping) arithmetic for all cases.

The following table represents a subset of the compute block and load/store instructions of the TigerSHARC DSP. It is not a complete list, and does not include Integer ALU instructions, control flow instructions, or other miscellaneous instructions. All arithmetic fixed-point instructions support integer and fractional data representations, with optional saturation (clipping). Instructions are provided to support transformations between data types. All listed instructions can be executed in Processing Element X or Y, or both.

TigerSHARC INSTRUCTION SET SUMMARY

Word Width					Instruction Description	Syntax Example
B	S	N	L	F		
					ALU Operations	
x	x	x	x	x	Add or subtract two operands	R0 = R1 - R2
x	x	x	x	x	Absolute Value of the sum or difference of two operands	R0 = ABS(R1 - R2)
x	x	x	x	x	Sum or difference of two operands, divide result by 2	R0 = (R1 - R2) / 2
x	x	x	x	x	MIN or MAX	R0 = MIN (R1, R2)
x	x	x	x		Increment or Decrement	R0 = INC R1
x	x	x	x		Add (or Subtract) with carry	R0 = R1 + R2 + CI
x	x	x	x	x	Compare	R0 = COMP (R1, R2)
x	x	x	x	x	Clip	R0 = CLIP R1 BY R2
x	x	x	x	x	Absolute value	R0 = ABS R1
x	x	x	x	x	Negate	R0 = -R1
			x	x	Logical (AND, OR, XOR, NOT, AND-NOT)	R0 = R1 AND R2
x	x	x			Expand	R1:0 = EXPAND sR1
			x	x	Compact	sR1 = COMPACT R1:0
x	x				Merge	sR1:0 = MERGE R2, R3
x	x				Sideways Sum	R0 = SUM sR1
			x	x	Count ones	R0 = ONES R1
x	x				Sum of Absolute Values of Differences	PR0 += ABS (sR1:0 - sR3:2)
x	x	x	x	x	Dual add-subtract	R0 = R1 + R2, R3 = R1 - R2
				x	Reciprocal seed	FR0 = RECIPS R3
				x	Reciprocal square root seed	FR0 = RSQRTS R3
					Shifter Operations	
x	x	x	x		Logical or arithmetic shift by operand or immediate value	R0 = ASHIFT R1 BY R2
			x	x	Rotate by operand or immediate value	R0 = ROT R1 BY R2
			x	x	Field Deposit / Extract	R0 += FDEP R1 BY R2
			x	x	Apply mask	R0 += MASK R1 BY R2
			x	x	Bit manipulation instructions: set, clear, toggle, test	R0 = BSET R1 BY R2
					Multiplier Operations	
	x	x		x	Multiply	R0 = R1 * R2
	x	x			Multiply-accumulate	MR1:0 += R1 * R2
	x				Complex MAC	MR1:0 += R1 ** R2
	x	x			Compact	R0 = COMPACT MR1:0
					Memory Load and Store Operations	
128-bit move					Quad word Load/store Direct	xR3:0 = Q[mem]
128-bit move					Quad word Load/store Broadcast	R3:0 = Q[mem]
128-bit move					Quad word Load/store Split	R3:2 = Q[mem]
32- or 64-bit move					Normal or Long load/store	R1 = [mem]

ANALOG DEVICES