Introduction ■ 1

1.1 OVERVIEW

The ADSP-2100 family is a collection of programmable single-chip microprocessors that share a common base architecture optimized for digital signal processing (DSP) and other high-speed numeric processing applications. The various family processors differ principally in the type of on-chip peripherals they add to the base architecture. On-chip memory, a timer, serial port(s), and parallel ports are available in different members of the family. In addition, the ADSP-21msp58/59 processors include an on-chip analog interface for voiceband signal conversion.

This manual provides the information necessary to understand and evaluate the processors' architecture, and to determine which device best meets your needs for a particular application. Together with the data sheets describing the individual devices, this manual provides all the information required to design a DSP system. Complete reference material for programmers is also included.

1.1.1 Functional Units

Table 1.1 on the following page lists the main functional units of the ADSP-21xx architecture, and shows which functions are included on each of the processors.

- *Computational Units*—Every processor in the ADSP-2100 family contains three independent, full-function computational units: an arithmetic/logic unit (ALU), a multiplier/accumulator (MAC) and a barrel shifter. The computational units process 16-bit data directly and also provide hardware support for multiprecision computations.

- *Data Address Generators & Program Sequencer*—Two dedicated address generators and a program sequencer supply addresses for on-chip or external memory access. The sequencer supports single-cycle conditional branching and executes program loops with zero overhead. Dual data address generators allow the processor to generate simultaneous addresses for dual operand fetches. Together the sequencer and data address generators keep the computational units continuously working, maximizing throughput.

Feature	2101	2103	2105	2115	2111	2171	2173	2181	2183	21msp58
Arithmetic/Logic Unit	●	●	●	●	●	●	●	●	●	●
Multiply/Accumulator	●	●	●	●	●	●	●	●	●	●
Shifter	●	●	●	●	●	●	●	●	●	●
Data Address Generators	●	●	●	●	●	●	●	●	●	●
Program Sequencer	●	●	●	●	●	●	●	●	●	●
Data Memory RAM	1K	1K	512	512	1K	2K	2K	16K	16K	2K
Program Memory RAM	2K	2K	1K	1K	2K	2K	2K	16K	16K	2K
Timer	●	●	●	●	●	●	●	●	●	●
Serial Port 0 (Multichannel)	●	●	–	●	●	●	●	●	●	●
Serial Port 1	●	●	●	●	●	●	●	●	●	●
Host Interface Port	–	–	–	–	●	●	●	–	–	●
DMA Ports	–	–	–	–	–	–	–	●	●	–
Analog Interface	–	–	–	–	–	–	–	–	–	●
Supply Voltage	5V	3.3V	5V	5V	5V	5V	3.3V	5V	3.3V	5V
Instruction Rate (MIPS)	20	10	13.8	20	20	33	20	33	33	26

Table 1.1 ADSP-2100 Family Processor Features & On-Chip Peripherals

- *Memory*—The ADSP-2100 family uses a modified Harvard architecture in which data memory stores data, and program memory stores both instructions and data. All ADSP-2100 family processors contain on-chip RAM that comprises a portion of the program memory space and data memory space. The speed of the on-chip memory allows the processor to fetch two operands (one from data memory and one from program memory) and an instruction (from program memory) in a single cycle.

- *Serial Ports*—The serial ports (SPORTs) provide a complete serial interface with hardware companding for data compression and expansion. Both μ-law and A-law companding are supported. The SPORTs interface easily and directly to a wide variety of popular serial devices. Each SPORT can generate a programmable internal clock or accept an external clock. SPORT0 includes a multichannel option.

- *Timer*—A programmable timer/counter with 8-bit prescaler provides periodic interrupt generation.

- *Host Interface Port*—The Host Interface Port (HIP) allows direct connection (with no glue logic) to a host processor. The HIP is made up of 16 data pins and 11 control pins. The HIP is extremely flexible and has provisions to allow simple interface to a variety of host processors. For example, the Motorola 68000, the Intel 8051, or another ADSP-2100 family processor can be easily connected to the HIP.

- *DMA Ports*—The ADSP-2181's Internal DMA Port (IDMA) and Byte DMA Port (BDMA) provide efficient data transfers to and from internal memory. The IDMA port has a 16-bit multiplexed address and data bus and supports 24-bit program memory. The IDMA port is completely asynchronous and can be written to while the ADSP-2181 is operating at full speed. The byte memory DMA port allows boot loading and storing of program instructions and data.

- *Analog Interface*—The ADSP-21msp58/59 processors include on-chip circuitry for mixed analog and digital signal processing. This circuitry includes an analog-to-digital converter (ADC), a digital-to-analog converter (DAC), analog and digital filters, and a parallel interface to the processor's core. The converters use sigma-delta technology to capture data samples from a highly oversampled signal.

The ADSP-2100 family architecture exhibits a high degree of parallelism, tailored to DSP requirements. In a single cycle, any device in the family can:

- Generate the next program address.
- Fetch the next instruction.
- Perform one or two data moves.
- Update one or two data address pointers.
- Perform a computation.

In that same cycle, processors which have the relevant functional units can also:

- Receive and/or transmit data via the serial port(s).
- Receive and/or transmit data via the host interface port.
- Receive and/or transmit data via the DMA ports.
- Receive and/or transmit data via the analog interface.

1.1.2 Memory And System Interface

In each ADSP-21xx processor, four on-chip buses connect internal memory with the other functional units: Data Memory Address bus, Data Memory Data bus, Program Memory Address bus, and Program Memory Data bus. A single external address bus and and a single external data bus are extended off-chip; these buses can be used for either program or data memory access.

External devices can gain control of the processor's buses with the bus request and grant signals ($\overline{BR}$, $\overline{BG}$). The ADSP-21xx processors can continue running while the buses are granted to another device, as long as an external memory operation is not required.

The ADSP-21xx processors support memory-mapped peripherals with programmable wait state generation.

Boot circuitry provides for loading on-chip program memory automatically after reset. This can be done either through the memory interface from a single low-cost EPROM, through the host interface port from a host processor, or through the BDMA port of the ADSP-2181. Multiple programs can be selected and loaded with no additional hardware.

ADSP-2100 family processors differ in their response to interrupts. In all cases, however, the program sequencer allows the processor to respond with minimum latency. Interrupts can be nested with no additional latency. External interrupts can be configured as edge- or level-sensitive. Internal interrupts can be generated from the timer, the host interface port, the serial ports, and the BDMA port.

1.1.3 Instruction Set

The ADSP-2100 family shares a single unified instruction set designed for upward compatibility with higher-integration devices. The ADSP-2171, ADSP-2181, and ADSP-21msp58/59 processors have a number of additional and enhanced instructions.

The ADSP-2100 family instruction set provides flexible data moves. Multifunction instructions combine one or more data moves with a computation. Every instruction can be executed in a single processor cycle. The assembly language uses an algebraic syntax for readability and ease of coding. A comprehensive set of software and hardware tools supports program development.

1.1.4 DSP Performance

Signal processing applications make special performance demands which distinguish DSP architectures from other microprocessor and microcontroller architectures. Not only must instruction execution be fast, but DSPs must also perform well in each of the following areas:

- *Fast and Flexible Arithmetic*—The ADSP-2100 family base architecture provides single-cycle computation for multiplication, multiplication with accumulation, arbitrary amounts of shifting, and standard arithmetic and logical operations. In addition, the arithmetic units allow for any sequence of computations so that a given DSP algorithm can be executed without being reformulated.

- *Extended Dynamic Range*—Extended sums-of-products, common in DSP algorithms, are supported in the multiply/accumulate units of the ADSP-2100 family. A 40-bit accumulator provides eight bits of protection against overflow in successive additions to ensure that no loss of data or range occurs; 256 overflows would have to occur before any data is lost. Special instructions are provided for implementing block floating-point scaling of data.

- *Single-Cycle Fetch of Two Operands*—In extended sums-of-products calculations, two operands are needed on each cycle to feed the calculation. All members of the ADSP-2100 family are able to sustain two-operand data throughput, whether the data is stored on-chip or off.

- *Hardware Circular Buffers*—A large class of DSP algorithms, including digital filters, requires circular data buffers. The ADSP-2100 family base architecture includes hardware to handle address pointer wraparound, simplifying the implementation of circular buffers both on- and off-chip, and reducing overhead (thereby improving performance).

- *Zero-Overhead Looping and Branching*—DSP algorithms are repetitive and are most logically expressed as loops. The program sequencer in the ADSP-2100 family supports looped code with zero overhead, combining excellent performance with the clearest program structure. Likewise, there are no overhead penalties for conditional branches.

1.2 CORE ARCHITECTURE

This section describes the core architecture of the ADSP-2100 family, as shown in Figure 1.1. Each component of the core architecture is described in detail in different chapters of this manual, as shown below:

Arithmetic/logic unit (ALU)	Chapter 2, *Computation Units*
Multiplier/accumulator (MAC)	Chapter 2, *Computation Units*
Barrel shifter	Chapter 2, *Computation Units*
Program sequencer	Chapter 3, *Program Control*
Status registers and stacks	Chapter 3, *Program Control*
Two data address generators (DAGs)	Chapter 4, *Data Transfer*
PMD-DMD bus exchange (PX registers)	Chapter 4, *Data Transfer*

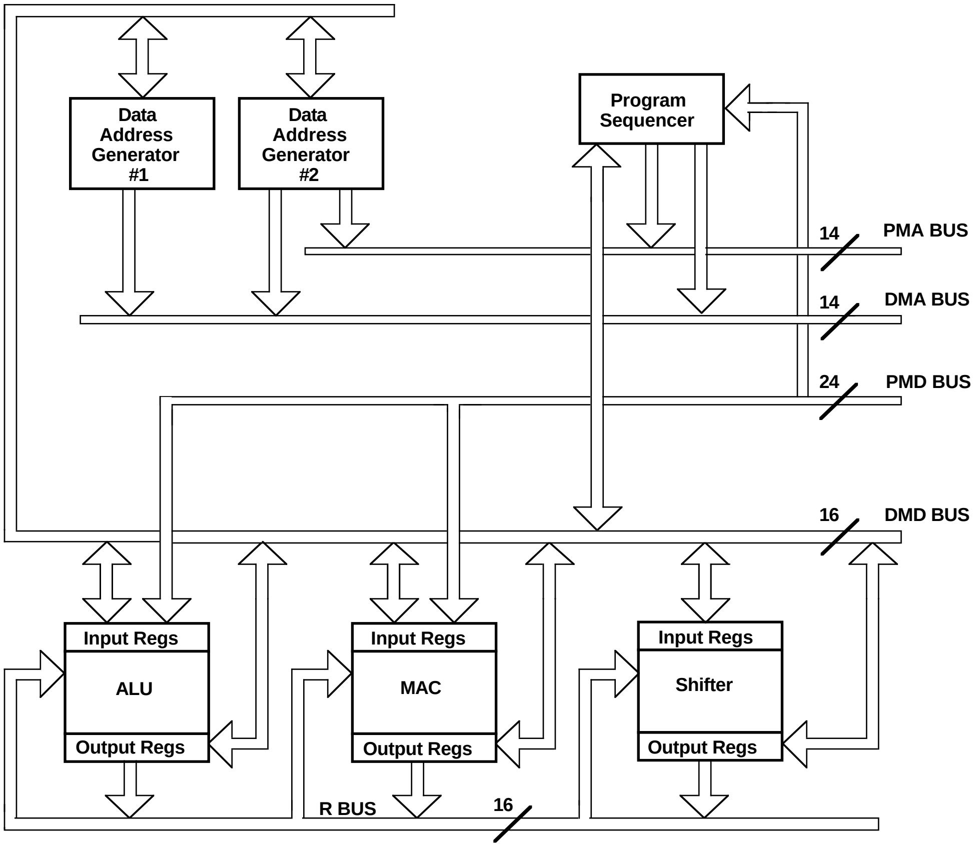

Figure 1.1 Base Architecture

1.2.1 Computational Units

Every processor in the ADSP-2100 family contains three independent, full-function computational units: an arithmetic/logic unit (ALU), a multiplier/accumulator (MAC) and a barrel shifter. The computation units process 16-bit data directly and provide hardware support for multiprecision computation as well.

The ALU performs a standard set of arithmetic and logic operations in addition to division primitives. The MAC performs single-cycle multiply, multiply/add and multiply/subtract operations. The shifter performs logical and arithmetic shifts, normalization, denormalization, and derive-exponent operations. The shifter implements numeric format control including multiword floating-point representations. The computational units are arranged side-by-side instead of serially so that the output of any unit may be the input of any unit on the next cycle. The internal result (R) bus directly connects the computational units to make this possible.

All three units contain input and output registers which are accessible from the internal data memory data (DMD) bus. Computational operations generally take their operands from input registers and load the result into an output register. The registers act as a stopover point for data between memory and the computational circuitry. This feature introduces one level of pipelining on input, and one level on output. The R bus allows the result of a previous computation to be used directly as the input to another computation. This avoids excessive pipeline delays when a series of different operations are performed.

1.2.2 Address Generators & Program Sequencer

Two dedicated data address generators and a powerful program sequencer ensure efficient use of the computational units. The data address generators (DAGs) provide memory addresses when memory data is transferred to or from the input or output registers. Each DAG keeps track of up to four address pointers. When a pointer is used for indirect addressing, it is post-modified by a value in a specified register. With two independent DAGs, the processor can generate two addresses simultaneously for dual operand fetches.

A length value may be associated with each pointer to implement automatic modulo addressing for circular buffers. (The circular buffer feature is also used by the serial ports for automatic data transfers. Refer to the Serial Ports chapter for additional information.)

DAG1 can supply addresses to data memory only; DAG2 can supply addresses to either data memory or program memory. When the appropriate mode bit is set in the mode status register (MSTAT), the output address of DAG1 is bit-reversed before being driven onto the address bus. This feature facilitates addressing in radix-2 Fast Fourier Transform (FFT) algorithms.

The program sequencer supplies instruction addresses to the program memory. The sequencer is driven by the instruction register which holds the currently executing instruction. The instruction register introduces a single level of pipelining into the program flow. Instructions are fetched and loaded into the instruction register during one processor cycle, and executed during the following cycle while the next instruction is prefetched. To minimize overhead cycles, the sequencer supports conditional jumps, subroutine calls and returns in a single cycle. With an internal loop counter and loop stack, the processor executes looped code with zero overhead. No explicit jump instructions are required to loop.

1.2.3 Buses

The processors have five internal buses. The program memory address (PMA) and data memory address (DMA) buses are used internally for the addresses associated with program and data memory. The program memory data (PMD) and data memory data (DMD) buses are used for the data associated with the memory spaces. The buses are multiplexed into a single external address bus and a single external data bus; the $\overline{\text{BMS}}$, $\overline{\text{DMS}}$ and $\overline{\text{PMS}}$ signals select the different address spaces. The R bus transfers intermediate results directly between the various computational units.

The PMA bus is 14 bits wide allowing direct access of up to 16K words of mixed instruction code and data. The PMD bus is 24 bits wide to accommodate the 24-bit instruction width.

The DMA bus is 14 bits wide allowing direct access of up to 16 K words of data. The data memory data (DMD) bus is 16 bits wide. The DMD bus provides a path for the contents of any register in the processor to be transferred to any other register or to any data memory location in a single cycle. The data memory address comes from two sources: an absolute value specified in the instruction code (direct addressing) or the output of a data address generator (indirect addressing). Only indirect addressing is supported for data fetches from program memory.

The program memory data (PMD) bus can also be used to transfer data to and from the computational units through direct paths or via the PMD-DMD bus exchange unit. The PMD-DMD bus exchange unit permits data to be passed from one bus to the other. It contains hardware to overcome the 8-bit width discrepancy between the two buses, when necessary.

1.3 ON-CHIP PERIPHERALS

This section describes the additional functional units which are included in various processors of the ADSP-2100 family.

1.3.1 Serial Ports

Most ADSP-21xx processors have two bidirectional, double-buffered serial ports (SPORTs) for serial communications. The SPORTs are synchronous and use framing signals to control data flow. Each SPORT can generate its serial clock internally or use an external clock. The framing sync signals may be generated internally or by an external device. Word lengths may vary from three to sixteen bits. One serial port, SPORT0, has a multichannel capability that allows the receiving or transmitting of arbitrary data words from a 24-word or 32-word bitstream. The other serial port, SPORT1, may optionally be configured as two additional external interrupt pins ($\overline{\text{IRQ1}}$ and $\overline{\text{IRQ0}}$)and the Flag Out (FO) and Flag In (FI) pins.

1.3.2 Timer

The programmable interval timer provides periodic interrupt generation. An 8-bit prescaler register allows the timer to decrement a 16-bit count register over a range from each cycle to every 256 cycles. An interrupt is generated when this count register reaches zero. The count register is automatically reloaded from a 16-bit period register and the count resumes immediately.

1.3.3 Host Interface Port (ADSP-2111, ADSP-2171, ADSP-21msp5x)

The host interface port (HIP) is a parallel I/O port that allows for an easy connection to a host processor. Through the HIP, an ADSP-21xx DSP can be used as a memory-mapped peripheral of the host. The HIP operates in parallel with and asynchronous to the ADSP-21xx's computational core and internal memory. The host interface port consists of registers through which the ADSP-21xx and the host processor pass data and status information. The HIP can be configured for: an 8-bit data bus or 16-bit data bus; a multiplexed address/data bus or separate address and data buses; and separate read and write strobes or a read/write strobe and a data strobe.

1.3.4 DMA Ports (ADSP-2181)

The ADSP-2181 contains two DMA ports, and Internal DMA Port and a Byte DMA Port. The IDMA port provides an efficient means of communication between a host system and the DSP. The port is used to access the on-chip program memory and data memory of the DSP with only one cycle per word of overhead. The IDMA port has a 16-bit multiplexed address and data bus and supports 24-bit program memory. The IDMA port is completely asynchronous and can be written to while the ADSP-2181 is operating at full speed.

The internal memory address is latched and then automatically incremented after each IDMA transaction. An external device can therefore access a block of sequentially addressed memory by specifying only the starting address of the block.

The byte memory DMA controller allows loading and storing of program instructions and data using the byte memory space. The BDMA circuit is able to access the byte memory space while the processor is operating normally and steals only one processor cycle per 8-, 16- or 24-bit word transferred.

1.3.5 Analog Interface

The analog interface of the ADSP-21msp58/59 consists of input amplifiers and a 16-bit sigma-delta analog-to-digital converter (ADC) as well as a sigma-delta digital-to-analog converter (DAC) and a differential output amplifier.

1.4 ADSP-2100 FAMILY DEVELOPMENT TOOLS

The ADSP-2100 family is supported with a complete set of software and hardware development tools. The ADSP-2100 Family Development System includes software utilities for program development and EZ Tools for hardware/software debugging.

The Development Software includes:

- *System Builder*—The System Builder defines the architecture of your hardware system. This includes the specification of the amount of external memory available and any memory-mapped I/O ports.

- *Assembler*—The Assembler assembles the source code and data modules as well as supporting the high-level syntax of the instruction set. In addition to supporting a full range of system diagnostics, the Assembler provides flexible macro processing, include files, and modular code development.

- *Linker*—The Linker links separately assembled modules. It maps the linked code and data output to the target system hardware, as specified by the System Builder output.

- *Simulator*—The Simulator performs an interactive, instruction-level simulation of the hardware configuration described by the System Builder. It flags illegal operations and supports full symbolic assembly and disassembly.

- *PROM Splitter*—This module reads the Linker output and generates PROM programmer compatible files.

- *C Compiler*—The C Compiler reads ANSI C source and outputs ADSP-2100 family source code ready to be assembled. It also supports inline assembler code.

The EZ-ICE® emulators provide hardware-based debugging of ADSP-21xx systems. The EZ-ICEs perform stand-alone, in-circuit emulation with little or no degradation in processor performance.

The EZ-LAB® evaluation boards are low-cost, basic hardware platforms for running example applications.

For additional information on the development tools, refer to the *ADSP-2100 Family Development Tools Data Sheet.*

1.5 ORGANIZATION OF THIS MANUAL

This manual is organized as follows.

Chapters 2, 3, and 4 describe the core architectural features shared by all members of the ADSP-2100 family:

- Chapter 2, "Computational Units," describes the functions and internal organization of the arithmetic/logic unit (ALU), the multiplier/accumulator (MAC), and the barrel shifter.

- Chapter 3, "Program Control," describes the program sequencer, interrupt controller and status and condition logic.

- Chapter 4, "Data Transfer," describes the data address generators (DAGs) and the PMD-DMD bus exchange unit.

Chapters 5, 6, 7, and 8 describe the additional functional units included in different members of the ADSP-2100 family. (See Table 1.1 for a list of the functions included in each device.)

- Chapter 5, "Serial Ports," describes the serial ports, SPORT0 and SPORT1.

- Chapter 6, "Timer," explains the programmable interval timer.

- Chapter 7, "Host Interface Port," describes the operation of the host interface port, including boot loading and software reset.

- Chapter 8, "Analog Interface," describes the operation and the internal architecture of the ADSP-21msp58/59's analog interface.

Chapters 9 and 10 describe the behavior of the ADSP-21xx processors from the point of view of external memory and control logic:

- Chapter 9, "System Interface," discusses the issue of system clocking, and describes the processors' control interface, the software reboot function, and the powerdown mode.

- Chapter 10, “Memory Interface,” describes the data and program memory spaces. This chapter describes both internal and external memory, including the use of boot memory space. A special section is devoted to the ADSP-2181, since its memory interface differs from that of the other family processors.

Chapter 11, “DMA Ports,” describes the operation of the ADSP-2181’s IDMA and BDMA features.

Chapter 12, “Programming Model,” gives a functional description of the processor resources—such as registers—as they appear in software.

Chapter 13, “Hardware Examples,” gives examples of system designs using the ADSP-21xx processors. Each example illustrates the solution to a different system design issue, using block diagrams, explanatory text, and programs or timing diagrams as needed.

Chapter 14, “Software Examples,” provides illustrative code for some important DSP and numerical algorithms.

Chapter 15, “Instruction Set Reference,” provides a detailed description of each ADSP-21xx instruction.

The Appendices provide reference material and further details on specific issues:

- Appendix A, “Instruction Coding,” gives the complete set of opcodes and specifies the bit patterns for choices within each field of the instruction word.

- Appendix B, “Division Exceptions,” describes signed and unsigned division.

- Appendix C, “Numeric Formats,” describes the fixed-point numerical formats directly supported by the ADSP-2100 family, discusses block floating-point arithmetic, and tells how to handle the results of multiplication for operands of various formats.

- Appendix D, “Interrupt Vector Addresses,” lists the interrupt vectors of each family processor.

- Appendix E, “Control/Status Registers,” summarizes the processors’ control and status registers.