12.5 MIPS DSP Microprocessor

FEATURES

Pin- and Code-Compatible DSP Microprocessors
ADSP-2100, 6.144MHz and 8.192MHz
ADSP-2100A, 10.24MHz and 12.5MHz
Separate Program and Data Buses, Extended Off-Chip
Single-Cycle Direct Access to 16K × 16 of Data Memory
Single-Cycle Direct Access to 32K × 24 of Program Memory
Dual Purpose Program Memory for Both Instruction and Data Storage
Three Independent Computational Units: ALU, Multiplier/Accumulator and Barrel Shifter
Two Independent Data Address Generators
Powerful Program Sequencer
Internal Instruction Cache
Provisions for Multiprecision Computation and Saturation Logic
Single-Cycle Instruction Execution
Multifunction Instructions
Four External Interrupts
80ns Cycle Time (ADSP-2100A)
790mW Maximum Power Dissipation (ADSP-2100A, J and K Grades)
100-Pin Grid Array, 100-Lead PQFP (JEDEC Style)

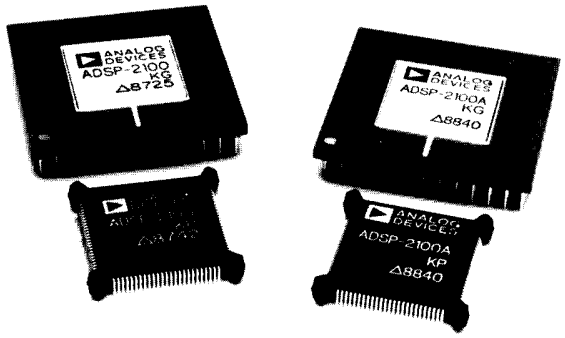
APPLICATIONS

Optimized for DSP Algorithms Including
Digital Filtering
Fast Fourier Transforms
Applications Include
Image Processing
Radar, Sonar
Speech Processing
Telecommunications

GENERAL DESCRIPTION

The ADSP-2100 and ADSP-2100A are pin- and code-compatible single-chip microprocessors optimized for digital signal processing (DSP) and other high-speed numeric processing applications. The ADSP-2100 and ADSP-2100A are both fabricated in a low-power double-layer metal CMOS process. Together, they offer a span of performance from 6MHz to 12.5MHz. All descriptions of the ADSP-2100 in the text of this data sheet refer to both the ADSP-2100A and the ADSP-2100 versions since they have identical architectures and instruction sets. Timing and electrical specifications differ as shown in those sections of the data sheet.

Both processors integrate computational units, data address generators and a program sequencer in a single device. The ADSP-2100 architecture makes efficient use of external memories for program and data storage, freeing silicon area for increased



processor performance. The resulting processor combines the functions and performance of a bit-slice/building block system with the ease of design and development support of a general purpose microprocessor.

The ADSP-2100A (K grade) operates at 12.5MHz. Every instruction executes in a single 80ns cycle. The ADSP-2100A (J and K grades) dissipates less than 790mW while the ADSP-2100 dissipates less than 475mW.

The ADSP-2100's flexible architecture and comprehensive instruction set support a high degree of operational parallelism. Because all instructions execute in a single cycle, MHz = MIPS. In one cycle the ADSP-2100 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 computational operation.

DEVELOPMENT SYSTEM

The ADSP-2100 and ADSP-2100A are supported by a complete set of tools for software and hardware system development. The Cross-Software System provides a System Builder for defining the architecture of simulated systems under development, an Assembler, a Linker and a interactive Simulator. An ANSI (draft) Standard C Compiler supports program development in this widely used programming language, producing ADSP-2100 Assembly code which may be assembled, linked and simulated with the other development system tools. A PROM Splitter generates PROM burner compatible files. An In-Circuit Emulator is available for hardware debugging.

An Evaluation Board is available for quick assessment of actual processor performance in a prepackaged hardware environment.

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 which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way; P. O. Box 9106; Norwood, MA 02062-9106 U.S.A.
Tel: 617/329-4700 **Twx: 710/394-6577**
Telex: 924491 **Cables: ANALOG NORWOODMASS**

ADDITIONAL INFORMATION

For additional information on the architecture and instruction set of the processor, refer to the *ADSP-2100 User's Manual*. For more information about programming and the Development System, refer to the *ADSP-2100 Cross-Software Manual* and the *ADSP-2100 Emulator Manual*. For examples of applications routines, refer to the *ADSP-2100 Applications Handbook, Volume 1, 2 or 3*. Manuals are available only from your local Analog Devices sales office. There is also a quarterly newsletter, *DSPatch*™, supporting Analog Devices' digital signal processing customers.

ARCHITECTURE OVERVIEW

Figure 1 is an overall block diagram of the ADSP-2100. The processor contains three independent computational units: the ALU, the multiplier/accumulator (MAC) and the Shifter. The computational units process 16-bit data directly 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 Shifter performs logical and arithmetic shifts, normalization, denormalization and derive exponent operations. The Shifter can be used to efficiently implement any degree of numeric format control, up to and including full floating point representations. The computational units are arranged side-by-side instead of serially for flexible operation sequencing. The internal result (R) bus directly connects the computational units so that the output of any unit may be the input of any unit on the next cycle.

A powerful program sequencer and two dedicated data address generators ensure efficient use of these computational units. The program sequencer generates the next instruction address. To minimize overhead cycles, the sequencer supports conditional jumps, subroutine calls and returns in a single cycle. With internal loop counters and loop stacks, the ADSP-2100 executes looped code with zero overhead; no explicit jump instructions are required to maintain the loop.

The data address generators (DAGs) handle address pointer updates. Each DAG keeps track of up to four address pointers. Whenever the pointer is used to access external data (indirect addressing), it is modified by a prespecified value. A length value may be associated with each pointer to implement automatic modulo addressing for circular buffers. With two independent DAGs, the processor can generate two addresses simultaneously for dual operand fetches.

Efficient data transfer is achieved with the use of five internal buses.

- Program Memory Address (PMA) bus
- Program Memory Data (PMD) bus
- Data Memory Address (DMA) bus
- Data Memory Data (DMD) bus
- Result (R) bus

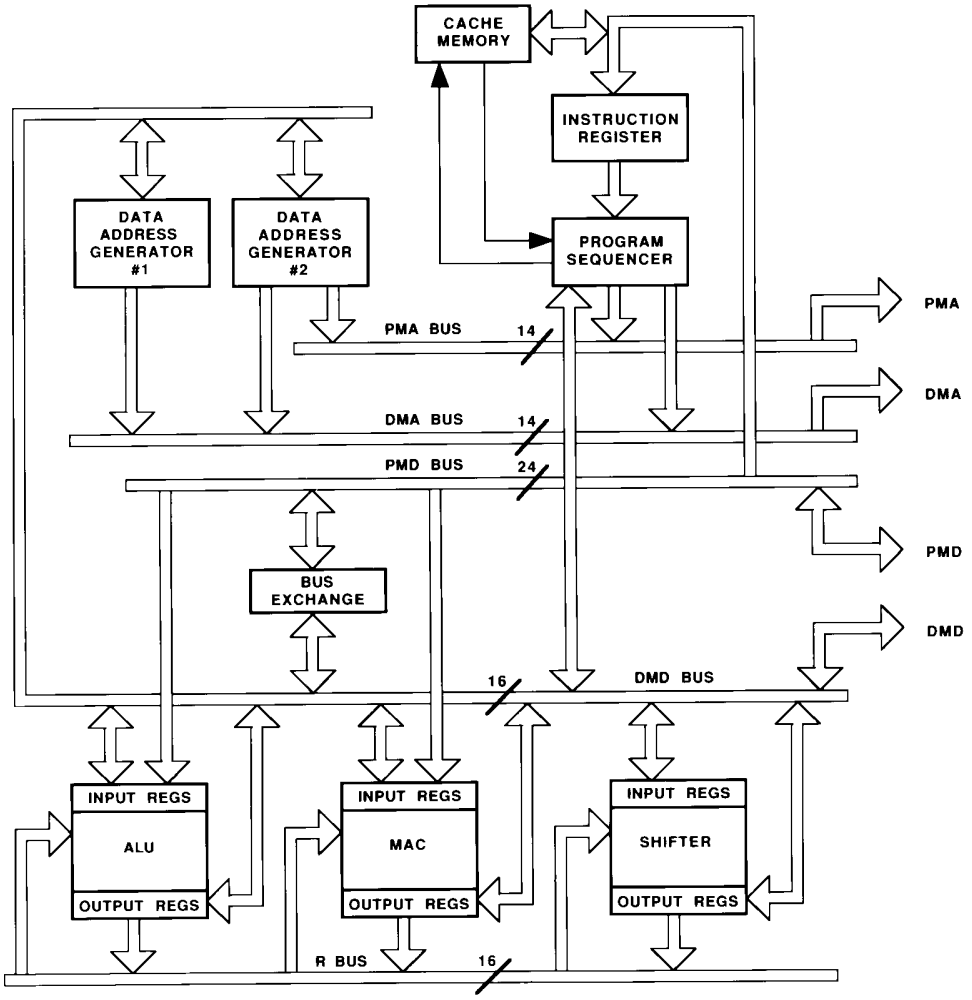


Figure 1. ADSP-2100 Block Diagram

DSPatch is a trademark of Analog Devices, Inc.

The program memory (PMD, PMA) buses and data memory (DMA, DMD) buses extend off-chip to provide direct connections to external memories. The DMD bus is the primary bus for routing data internally and to/from external data memory. The 14-bit DMA bus provides direct addressing of 16K × 16 of external memory. Although the primary function of the program memory is for storing instructions, it can also store data. In this case, the PMD bus provides a path for routing data to/from program memory, permitting dual operand fetches. The 14-bit PMA bus provides direct addressing of 16K × 24 of external memory, expandable to 32K × 24 by using the program memory data access (PMDA) signal as the 15th address line.

When a data fetch from program memory is required, an extra memory cycle is automatically appended to enable the next instruction fetch. To avoid this extra cycle, the ADSP-2100 has an internal instruction cache (16 instructions deep) which serves as an alternate source for the next instruction. The cache monitor circuit transparently determines when the cache contents are valid. When the next instruction is in the cache, no extra cycle is necessary.

The data memory interface supports slower memories and memory-mapped peripherals with wait states. The data memory acknowledge (DMACK) signal provides the necessary handshake. External devices can gain control of program or data buses independently with bus request/ grant signals ($\overline{BR}$, and $\overline{BG}$).

The ADSP-2100 can respond to four external interrupts, which are internally prioritized, maskable and independently programmable as either edge- or level-sensitive. Additional external controls are provided by the $\overline{RESET}$, $\overline{HALT}$ and TRAP signals. With both $\overline{BR}$ and $\overline{RESET}$ recognized, the ADSP-2100 idles, consuming the least possible current.

The ADSP-2100 instruction set provides flexible data moves and multifunction (data moves with a computation) instructions. Every instruction can be executed in a single processor cycle. The ADSP-2100 assembly language uses an algebraic syntax for ease of coding and readability. A comprehensive set of development tools supports program development.

A pin description and detailed discussion of each section of the ADSP-2100 follows.

Pin Description

This section summarizes the pin description of the processor by interface. In this data sheet, when groups of pins are identified with subscripts, as in PMD_{23-0}, the highest numbered pin (PMD_{23}) is the MSB.

Pin Name	Type	Function
Clocks:		
CLKIN	Input	Master input clock operating at four times the processor instruction rate. Nominally 50% duty cycle. The phases of CLKIN define the eight internal processor states making up one instruction cycle.
CLKOUT	Output	Output clock operating at the processor instruction rate with a 50% duty cycle. Synchronized to the internal processor states.
Interrupt Request Lines:		
$\overline{IRQ}_{3-0}$	Input	Interrupt Request lines that may be either edge triggered or level sensitive. Interrupts are prioritized and individually maskable.
Control Interface:		
$\overline{RESET}$	Input	Master Reset must be asserted long enough to assure proper reset. When $\overline{RESET}$ is released, execution begins at program memory location 0004.
$\overline{HALT}$	Input	Used to halt the processor. All control signals become inactive and the address and data buses are driven for observation.
TRAP	Output	Used to indicate the execution of a TRAP instruction. Remains asserted until $\overline{HALT}$ is asserted by an external device.
$\overline{BR}$	Input	Bus Request used by an external device to request control of the program and data memory interface. Upon receiving $\overline{BR}$ the processor halts execution at the completion of the current cycle and relinquishes the program and data memory interface by tristating PMA, PMD, $\overline{PMS}$, $\overline{PMWR}$, $\overline{PMRD}$, PMDA, DMA, DMD, $\overline{DMS}$, $\overline{DMRD}$ and $\overline{DMWR}$. The processor regains control when $\overline{BR}$ is released.
$\overline{BG}$	Output	Bus Grant. Acknowledges a bus request ($\overline{BR}$), indicating that the external device may take control. $\overline{BG}$ is held asserted until $\overline{BR}$ is released.
Program Memory Interface:		
PMA_{13-0}	Output	Program Memory Address Bus; tristated when $\overline{BG}$ is asserted.
PMD_{23-0}	Bidirectional	Program Memory Data Bus; tristated when $\overline{BG}$ is asserted.
$\overline{PMS}$	Output	Program Memory Select signals a program memory access on the PM interface. Usable as a chip select signal for external memories. Remains asserted on successive program memory accesses. HI only when the processor is halted or after execution of a TRAP instruction. Tristated when $\overline{BG}$ is asserted.

Program Memory Interface:

$\overline{PMRD}$	Output	Program Memory Read indicates a read operation on the PM interface. Also usable as a read strobe or output enable signal. Tristated when $\overline{BG}$ is asserted.
$\overline{PMWR}$	Output	Program Memory Write establishes the direction of data transfer on the PM interface. Also usable as a write strobe. Tristated when $\overline{BG}$ is asserted.
PMDA	Output	Program Memory Data Access used to distinguish instruction and data fetches from PM. Asserted high when data, as opposed to instruction, are accessed. Also usable as a fifteenth PM address bit. Tristated when $\overline{BG}$ is asserted.

Data Memory Interface:

DMA_{13-0}	Output	Data Memory Address Bus; tristated when $\overline{BG}$ is asserted.
DMD_{15-0}	Bidirectional	Data Memory Data Bus; tristated when $\overline{BG}$ is asserted.
$\overline{DMS}$	Output	Data Memory Select signals a Data Memory Access on the Data Memory interface. Usable as a chip select signal for external memories. Remains asserted on successive data memory accesses. HI only when the processor is halted or after execution of a TRAP instruction. Tristated when $\overline{BG}$ is asserted.
$\overline{DMRD}$	Output	Data Memory Read indicates a read operation on the Data Memory interface. Also usable as a read strobe or output enable signal. Tristated when $\overline{BG}$ is asserted.
$\overline{DMWR}$	Output	Data Memory Write indicates a write operation on the Data Memory interface. Also usable as a write strobe. Tristated when $\overline{BG}$ is asserted.
DMACK	Input	Data Memory Acknowledge signal used for asynchronous transfers across the DM interface. Indicates that data memory or memory-mapped peripherals are ready for data transfer. If DMACK is not asserted when checked by the processor, wait states are automatically generated until DMACK is asserted.

Supply Rails:

V_{DD}	Supply	Power supply rail nominally +5VDC. There are four V_{DD} pins.
GND	Ground	Power supply return. There are nine GND pins.

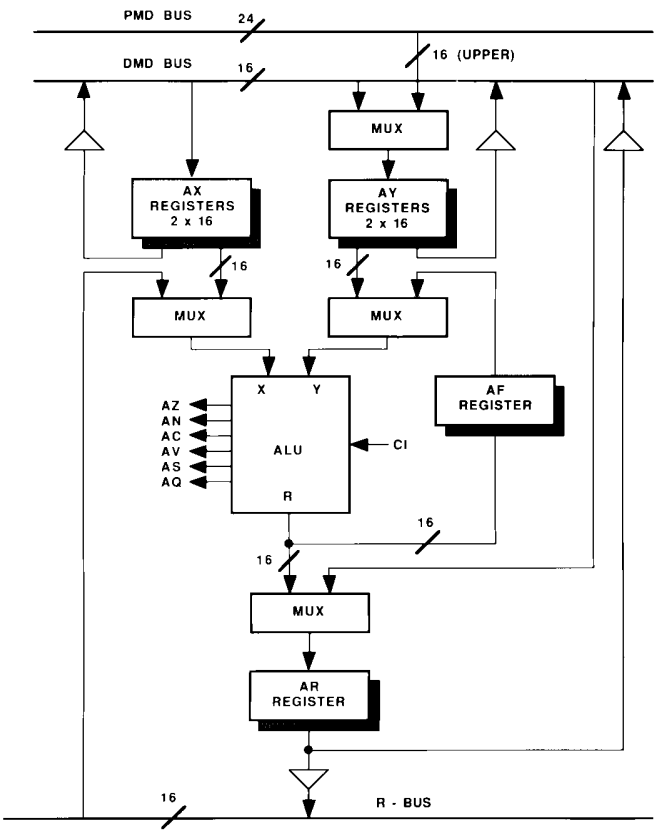


Figure 2. ALU Block Diagram

Arithmetic/Logic Unit

Figure 2 shows a block diagram of the Arithmetic/Logic Unit (ALU).

The ALU provides a standard set of general purpose arithmetic and logic functions: add, subtract, negate, increment, decrement, absolute value, AND, OR, Exclusive OR and NOT. Two divide primitives are also provided to facilitate division. The ALU takes two 16-bit inputs, X and Y, and generates one 16-bit output, R. It accepts the carry (AC) bit in the arithmetic status register (ASTAT) as the carry-in (CI) bit. The carry-in feature enables multiprecision computations. Six arithmetic status bits are generated: AZ (zero), AN (negative), AV (overflow), AC (carry), AS (sign) and AQ (quotient). These status bits are latched in ASTAT.

The X input port can be fed by either the AX register file or any result registers on the R-bus (AR, MR0, MR1, MR2, SR0, or SR1). The AX register file contains two registers, AX0 and AX1. The AX registers can be loaded from the DMD bus. The Y input port can be fed by either the AY register file or the ALU feedback (AF) register. The AY register file contains two registers, AY0 and AY1. The AY registers can be loaded from either the DMD bus or the PMD bus.

The register file outputs are dual ported so that one register can drive the ALU input while either one simultaneously drives the DMD bus. The ALU output can be latched in either the AR register or the AF register.

The AR register has a saturation capability; it can automatically output plus or minus the maximum value if an overflow or underflow occurs. The saturation mode is enabled by a bit in the mode status register (MSTAT). The AR register can drive both the R-bus and the DMD bus and can be loaded from the DMD bus.

The ALU contains a duplicate bank of registers shown in Figure 2 as a "shadow" behind the primary registers. The secondary set contains all the registers described above (AX0, AX1, AY0, AY1, AF, AR). Only one set is accessible at a time. The two sets of registers allow fast context switching for interrupt servicing. The active set is determined by a bit in MSTAT.

Multiplier/Accumulator

The multiplier/accumulator (MAC) implements high-speed multiply, multiply/add and multiply/subtract operations. Figure 3 shows a block diagram of the MAC section.

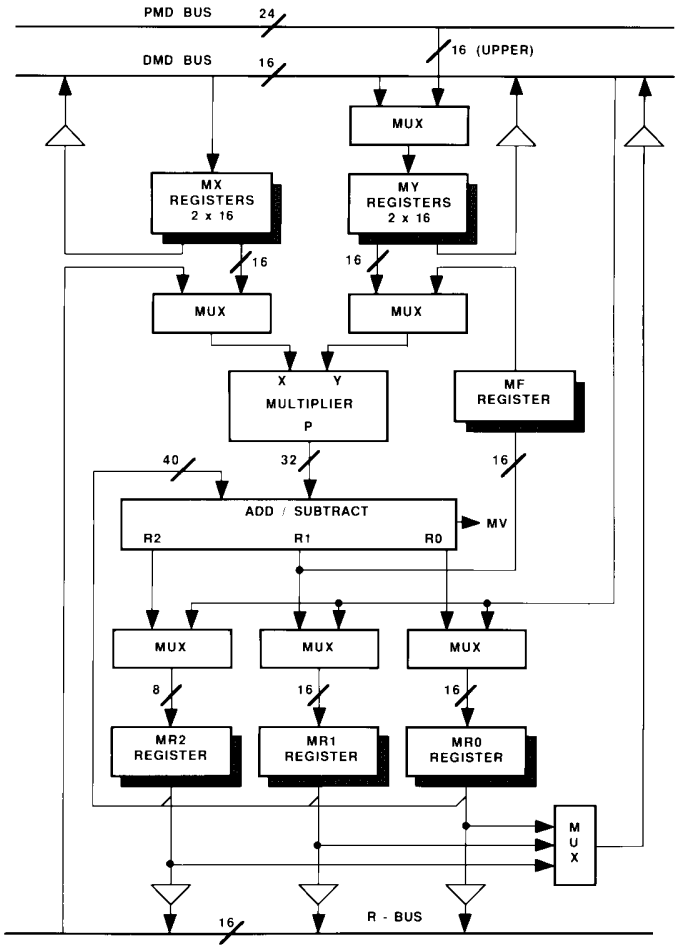


Figure 3. MAC Block Diagram

The multiplier takes two 16-bit inputs, X and Y, and generates one 32-bit output, P. The 32-bit output is routed to a 40-bit accumulator which can add or subtract the P output from the value in MR. MR is a 40-bit register which is divided into three sections: MR0 (bits 0-15), MR1 (bits 16-31), and MR2 (bits 32-39). The result of the accumulator is either loaded into the MR register or into the 16-bit MAC feedback (MF) register. The multiplier accepts the X and Y inputs in either signed or unsigned formats. The result is shifted one bit to the left automatically to remove the redundant sign bit for fractional justification. The accumulator generates one status bit, MV, which is set when the accumulator result overflows the 32-bit boundary. A saturate command is available to change the content of the MR register to the maximum or minimum 32-bit value when MV is set. The accumulator also has the capability for rounding the 40-bit result at the boundary between bit 15 and bit 16.

The MAC and ALU registers are similar. The X input port can be fed by either the MX register file (MX0, MX1) or any result registers on the R-bus (AR, MR0, MR1, MR2, SR0 or SR1).

The MX register file is readable and loadable from the DMD bus and has dual-ported outputs.

The Y input port can be fed by either the MY register file (MY0, MY1) or the MF register. The MY register file is readable from the DMD bus and readable and loadable from both the DMD and the PMD bus. Its outputs are dual ported.

The accumulator output can be latched in either the MR register or the MF register. The MR register is connected to both the R-bus and the DMD-bus. Like the ALU section, the MAC section contains two complete banks of registers (MX0, MX1, MY0, MY1, MF, MR0, MR1, MR2) to allow fast context switching.

Shifter

The Shifter gives the ADSP-2100 its unique capability to handle data formatting and numeric scaling. Figure 4 shows a block diagram of the Shifter.

The Shifter can be divided into the following components: the shifter array, the OR/PASS logic, the exponent detector and the exponent compare logic. These components give the Shifter its six basic functions: arithmetic shift, logical shift, normalization, denormalization, derive exponent and derive block exponent.

The shifter array is a 16 × 32-barrel shifter. It accepts a 16-bit input and can place it anywhere in the 32-bit output field, from off-scale right to off-scale left. The Shifter can perform arithmetic shifts (shifter output is sign-extended to the left) or logical shifts (shifter output is zero-filled to the left). The placement of the 16-bit input is determined by the control code (C) and the HI/LO reference signal. The control code can come from one of three sources: directly from the instruction (immediate arithmetic or logical shift), from the SE register (denormalization) or the negated value of the SE register (normalization). The shifter input can come from either the 16-bit SI register or any result register on the R-bus. The 32-bit output of the shifter array is fed to the OR/PASS circuit. The result can be either logically OR-ed with the current contents of the SR register or passed directly to the SR register. The SR register is divided into two 16-bit sections: SR0 (bits 0-15) and SR1 (bits 16-31).

The shifter input is also routed to the exponent detector circuitry. The exponent detector generates a value to indicate how many places the input must be up-shifted to eliminate all but one of the sign bits. This value is effectively the base 2 exponent of the number. The result of the exponent detector can be latched into the SE register (for a normalize operation) or can be sent to the exponent compare logic. The exponent compare logic compares the derived exponent with the value in the SB register and updates the SB register only when the derived exponent value is larger than the current value in the SB register. Therefore, the exponent compare logic can be used to find the largest exponent value in an array of shifter inputs.

The Shifter includes the following registers: the SI register, the SE register, the SB register and the SR register. All these registers are readable and loadable from the DMD-bus. The SR register can also drive the R-bus. Like the ALU and MAC, the Shifter contains two complete banks of registers for context switching. Each set contains all the registers described above, but only one set is accessible at a time. The active set is determined by a bit in MSTAT.

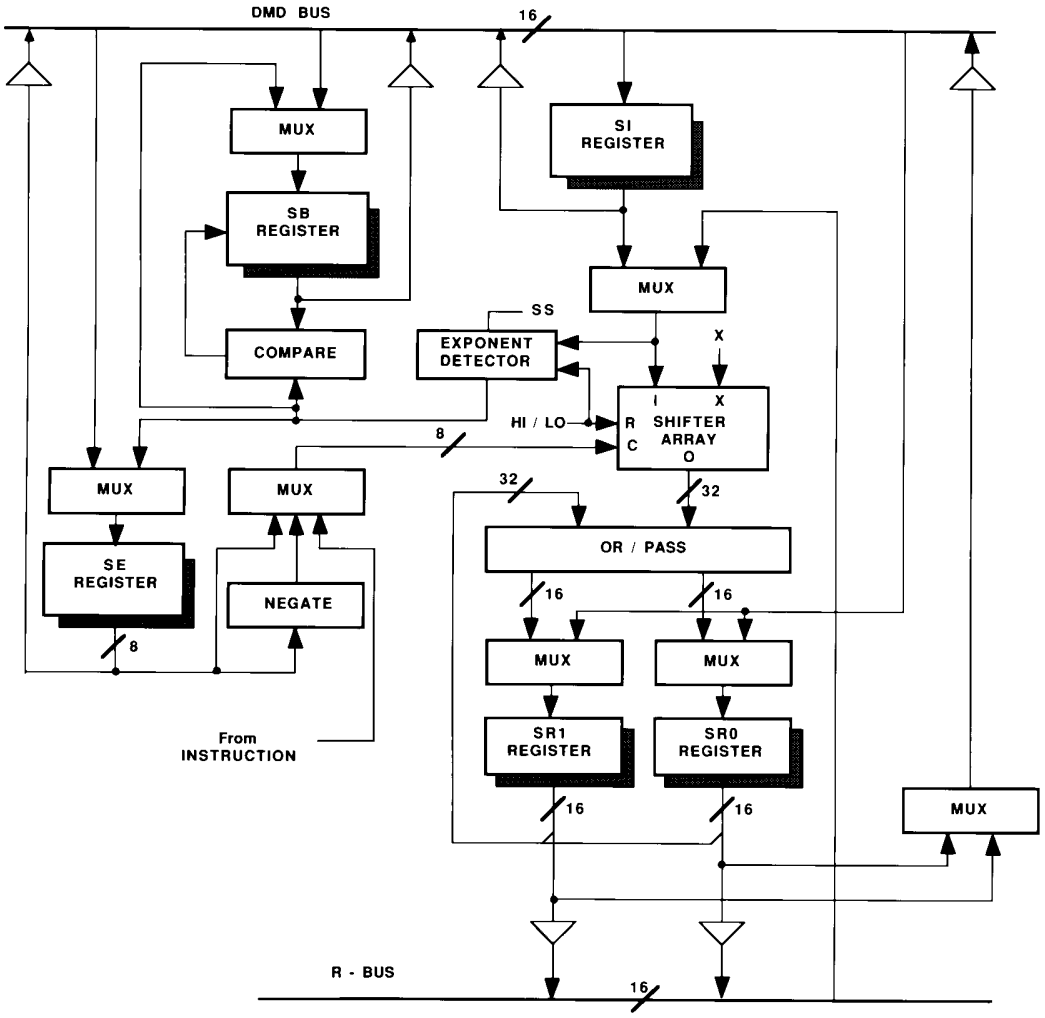


Figure 4. Shifter Block Diagram

Data Address Generators

Figure 5 shows a block diagram of a data address generator.

The data address generators (DAGs) provide indirect addressing for data stored in external memories. The processor contains two independent DAGs so that two data operands (one in program memory and one in data memory) can be addressed simultaneously. The two data address generators are identical except that DAG1 has a bit reversal option on the output and can only generate data memory addresses, while DAG2 can generate both program and data memory addresses but has no bit reversal capability.

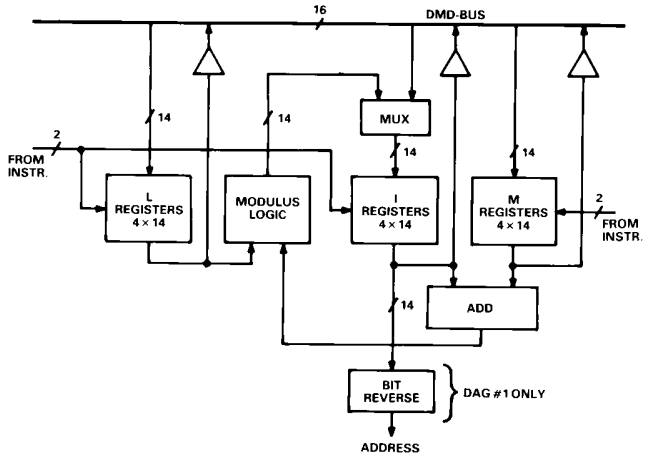


Figure 5. Data Address Generator

There are three register files in each DAG: the modify (M) register file, the indirect (I) register file, and the length (L) register file. Each of these register files contain four 14-bit registers which are readable and loadable from the DMD-bus. The I registers hold the actual addresses used to access external memory. When using the indirect addressing mode, the selected I register content is driven onto either the PMA or DMA bus. This value is post-modified by adding the content of the selected M register. The modified address is passed through the modulus logic. Associated with each I register is an L register which may contain the length of the buffer addressed by the I register. The L register and the modulus logic together enable circular buffer addressing with automatic wrap around at the buffer boundary. The modulus logic is disabled by setting the length of the associated buffer to zero.

Program Sequencer

The program sequencer incorporates powerful and flexible mechanisms for program flow control such as zero-overhead looping, single-cycle branching (both conditional and unconditional), and automatic interrupt processing. Figure 6 shows a block diagram of the program sequencer.

The sequencing logic controls the flow of the program execution. It outputs a program memory address onto the PMA bus from

one of four sources: the PC incrementer, PC stack, instruction register or interrupt controller. The next address source selector controls which of these four sources are selected based on the current instruction word and the processor status. A fifth possible source for the next program memory address is provided by DAG2 when a register indirect jump is executed.

The program counter (PC) is a 14-bit register which contains the address of the currently executing instruction. The PC output goes to the incrementer. The incremented output is selected as the next program memory address if program flow is sequential. The PC value is pushed onto the 16 × 14 PC stack when a CALL instruction is executed or when an interrupt is processed. The PC stack is popped when a return from subroutine or interrupt is executed. The PC stack is also used in zero-overhead looping.

The program sequencer section contains five status registers. These are the Arithmetic Status register (ASTAT), the Stack Status register (SSTAT), the Mode Status register (MSTAT), the Interrupt Control register (ICNTL) and the Interrupt Mask register (IMASK). These registers are described in detail in the next section.

The interrupt controller allows the processor to respond to one of four external interrupts with a minimum of overhead. The interrupts are internally prioritized and are individually maskable. Each interrupt can be set to be either edge- or level-sensitive. Depending on a bit in the interrupt control register (ICNTL), interrupt routines can either be nested, with higher priority interrupts taking precedence, or processed sequentially, with only one interrupt service active at a time. When responding to an interrupt, the status registers ASTAT, MSTAT, IMASK are pushed onto the status stack and the PC counter is loaded with the appropriate vectored address. The status stack is four levels deep to allow four levels of interrupt nesting. The stack is automatically popped when return from interrupt is executed.

The vector addresses for each interrupt are fixed at the lowest four addresses in the program memory space. Single-word, single-cycle branch instructions may be placed at these locations to transfer control to the appropriate interrupt service routine.

The down counter and the count stack implement a powerful looping mechanism. The down counter is a 14-bit register with

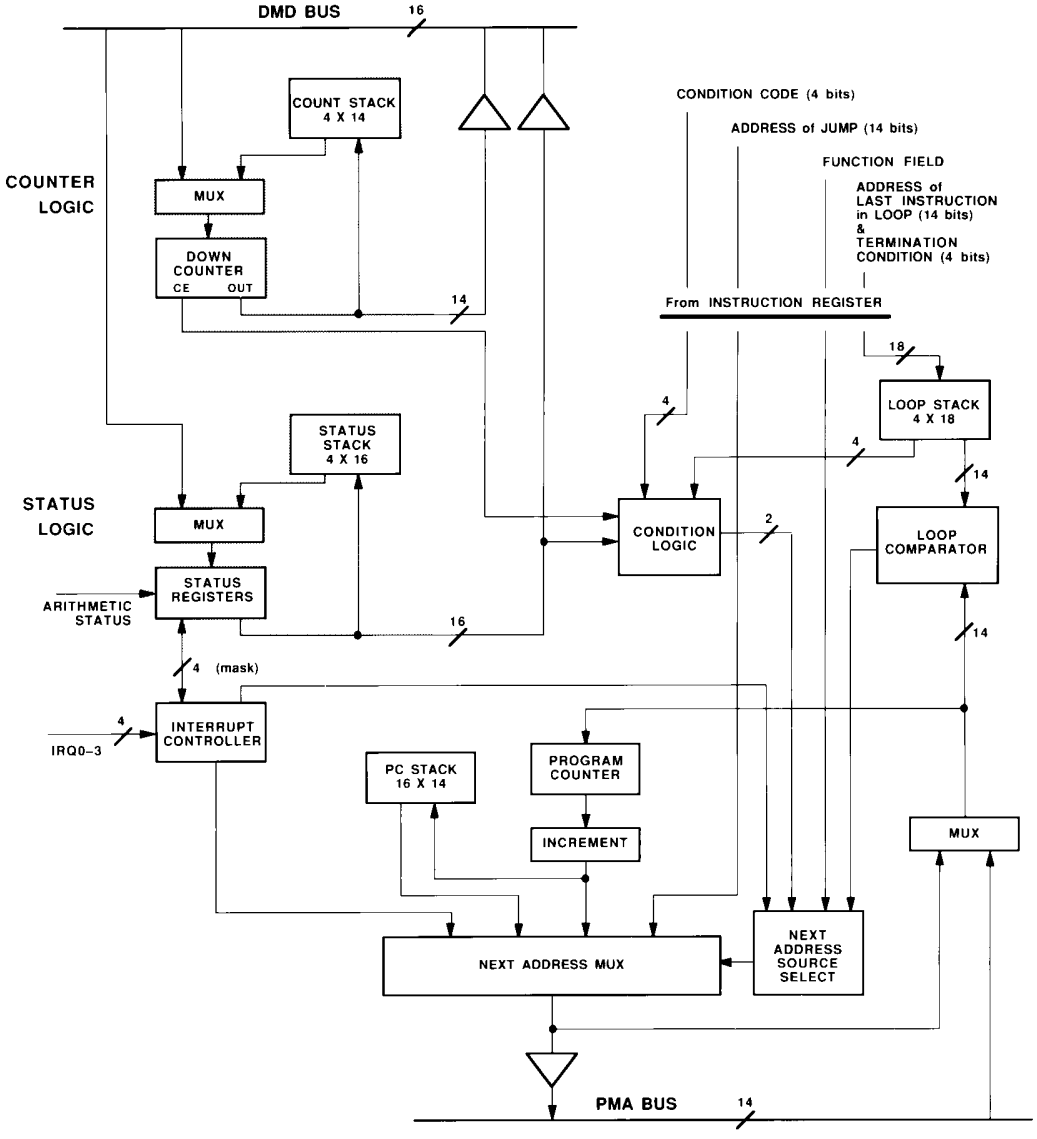


Figure 6. Program Sequencer

auto-decrement capability. It is loaded from the DMD bus with the loop count. The count is decremented every time the counter value is checked; when the count expires, the counter expired (CE) flag is set. The count stack allows the nesting of loops by storing temporarily dormant loop counts. When a new value is loaded into the counter from the DMD bus, the current counter value is automatically pushed onto the count stack as program flow enters a loop. The count stack is automatically popped whenever the CE flag is tested and is true, thereby resuming execution of the code outside the loop.

The DO UNTIL instruction executes a zero-overhead loop using the loop stack and the loop comparator. For a DO UNTIL instruction, a 14-bit termination address and a 4-bit termination condition are pushed onto the 18-bit loop stack. The address of the next instruction (which identifies the top of the loop) is pushed onto the PC stack. The loop comparator continuously compares the current PC value against the termination address on the top of the loop stack. When the termination address is detected, the processor checks if the termination condition is met. If the termination condition is not met, then the top of the PC stack is used as the next PC address, returning program flow to the beginning of the loop. If the termination condition is met, then the PC stack is popped, the current PC is incremented by one, and program flow falls out of the loop. The loop stack is four levels deep, permitting four levels of zero-overhead loop nesting.

Instruction Cache Memory

The instruction cache memory is 16 levels deep and one instruction (24 bits) wide. The cache memory maintains a short history of previously executed instructions so they can be fetched internally if they are needed again.

Every time an instruction is fetched from external memory, it is also written into the cache memory. When the program enters a loop which fits within the cache, all the instructions in the loop are stored in cache during the first pass. On subsequent passes, the instructions can be fetched from the instruction cache when a program memory data access is required. This allows the program memory to be used for data access without penalty. The ADSP-2100 then becomes, in effect, a three-bus system with two data buses and one program bus. For the multiply/accumulate operations typical of digital signal processing algorithms, this gives significant speed advantages.

Instructions are fetched from cache memory *only* when a program memory data fetch is required. The cache monitor circuit automatically keeps track of when the next instruction is contained in the cache. No maintenance or overhead is needed to store externally fetched instructions in the cache or to read previously fetched instructions from cache.

PMD-DMD Bus Exchange

The PMD-DMD bus exchange circuit couples the PMD and DMD buses. The PMD bus is 24 bits wide and the DMD bus is 16 bits wide. The upper 16 bits of PMD are connected to the DMD bus. An 8-bit register (PX) allows transfer of the full width of the PMD bus. When data is read from the PMD bus, the lower 8 bits of the PMD bus are loaded into PX. When writing to the PMD bus, the contents of PX are appended to the upper 16 bits, forming a 24-bit value. The PX register is readable and loadable from the DMD bus.

STATUS REGISTERS

The ADSP-2100 maintains five status registers, each of which can be read over the DMD bus and four of which can be written. These registers are:

ASTAT	Arithmetic Status register
SSTAT	Stack Status register (read-only)
MSTAT	Mode Status register
ICNTL	Interrupt Control register
IMASK	Interrupt Mask register

ASTAT

ASTAT is 8 bits wide and holds the status information generated by the computational sections of the processor. The bits in ASTAT are defined as follows:

0	AZ	(ALU result zero)
1	AN	(ALU result negative)
2	AV	(ALU overflow)
3	AC	(ALU carry)
4	AS	(ALU X input sign)
5	AQ	(ALU quotient flag)
6	MV	(MAC overflow)
7	SS	(Shifter input sign)

The bits which express a particular condition (AZ, AN, AV, AC, MV) are all positive sense (1 = true, 0 = false). Each of the bits are automatically updated whenever a new status is generated by an arithmetic operation. As such, each bit is affected only by a certain subset of arithmetic operations, as defined by the following table:

Status Bit	Updated on:
AZ, AN, AV, AC	Any ALU operation except division
AS	ALU absolute value operation
AQ	ALU divide operations
MV	Any MAC operation except saturate MR
SS	Shifter exponent detect operation

SSTAT

SSTAT is 8 bits wide and holds the status of the four internal stacks. The bits in SSTAT are:

0	PC Stack Empty
1	PC Stack Overflow
2	Count Stack Empty
3	Count Stack Overflow
4	Status Stack Empty
5	Status Stack Overflow
6	Loop Stack Empty
7	Loop Stack Overflow

All of the bits are positive sense (1 = true, 0 = false). The *empty* status bits indicate that the number of pop operations for the stack is greater than or equal to the number of push operations (if no stack overflow has occurred) since the last reset. The *overflow* status bits indicate that the number of push operations for the stack has exceeded the number of pop operations by an amount that is greater than the depth of the stack. When this occurs, the item(s) most recently pushed will be missing from the stack (old data is considered more important than new). The stack overflow status bits "stick" once they are set, so that subsequent pop operations have no effect on them. A processor reset must be executed to clear the stack overflow status.

MSTAT

MSTAT is a 4-bit register that defines various operating modes of the processor. The Mode Control instruction enables or disables the four operating modes. The bits in MSTAT are:

0 Data Register Bank Select
1 Bit Reverse Mode (DAG1 only)
2 ALU Overflow Latch Mode
3 AR Saturation Mode

The data register bank select bit determines which set of data registers is currently active (0 = primary, 1 = secondary). The data registers include all of the result and input registers to the ALU, MAC, and Shifter (AX0, AX1, AY0, AY1, AF, AR, MX0, MX1, MY0, MY1, MF, MR0, MR1, MR2, SB, SE, SI, SR0 and SR1). At initialization, the data register bank select bit is cleared.

The bit reverse mode, when enabled, bit-wise reverses all addresses generated by DAG1. This is most useful for reordering the input or output data in a radix-2 FFT algorithm.

The ALU overflow latch mode causes the AV (ALU overflow) status bit to "stick" once it is set. In this mode, when an ALU overflow occurs, AV will be set and remain set, even if subsequent ALU operations do not generate overflows. AV can then only be cleared by writing a zero into it from the DMD bus.

The AR saturation mode, when set, causes ALU results to be saturated to the maximum positive (H#7FFF) or negative (H#8000) values when an ALU overflow occurs.

IMASK

IMASK is four bits wide and allows the four interrupt inputs to be individually enabled or disabled. The bits in IMASK are:

0 $\overline{\text{IRQ0}}$ Enable
1 $\overline{\text{IRQ1}}$ Enable
2 $\overline{\text{IRQ2}}$ Enable
3 $\overline{\text{IRQ3}}$ Enable

The bits are all positive sense (0 = disabled, 1 = enabled). IMASK is set to zero upon a processor reset so that all interrupts are disabled initially.

ICNTL

ICNTL is a 5-bit register configuring the interrupt modes of the processor. The bits in ICNTL are:

0 $\overline{\text{IRQ0}}$ Sensitivity
1 $\overline{\text{IRQ1}}$ Sensitivity
2 $\overline{\text{IRQ2}}$ Sensitivity
3 $\overline{\text{IRQ3}}$ Sensitivity
4 Interrupt Nesting Mode

The IRQ sensitivity bits determine whether a given interrupt input is edge- or level-sensitive (0 = level-sensitive, 1 = edge-sensitive). These bits are all undefined after a processor reset.

The interrupt nesting mode determines whether nesting of interrupt service routines is allowed. When set to zero, all interrupt levels will be masked automatically when an interrupt service routine is entered. When set to one, IMASK will be set so that only equal and lower priority interrupts will be masked, permitting higher priority interrupts to interrupt the current interrupt service routine. This bit is undefined after a processor reset.

CONDITION CODES

The condition codes are used to determine whether a conditional instruction, such as a jump, trap, call, return, MAC saturation or arithmetic operation, is performed. The sixteen composite status conditions and their derivations are given in Table I. Since arithmetic status is latched into ASTAT at the end of a processor cycle, the condition logic outputs represent conditions generated on a previous cycle.

Code	Status Condition	True If:
EQ	ALU Equal Zero	AZ = 1
NE	ALU Not Equal Zero	AZ = 0
LT	ALU Less Than Zero	AN .XOR. AV = 1
GE	ALU Greater Than or Equal Zero	AN .XOR. AV = 0
LE	ALU Less Than or Equal Zero	(AN .XOR. AV) .OR. AZ = 1
GT	ALU Greater Than Zero	(AN .XOR. AV) .OR. AZ = 0
AC	ALU Carry	AC = 1
NOT AC	Not ALU Carry	AC = 0
AV	ALU Overflow	AV = 1
NOT AV	Not ALU Overflow	AV = 0
MV	MAC Overflow	MV = 1
NOT MV	Not MAC Overflow	MV = 0
NEG	ALU X Input Sign Negative	AS = 1
POS	ALU X Input Sign Positive	AS = 0
NOT CE	Not Counter Expired	CE ≠ 0
TRUE	True	Always True

Table I. Condition Codes

SYSTEM INTERFACE

Figure 7 shows a basic system configuration with the ADSP-2100.

Clock Signals

The ADSP-2100 takes a TTL-compatible clock signal, CLKIN, running at four times the basic processor cycle time as an input. Using this clock input, the processor divides the internal processor cycle into eight states, defined by the edges of the input clock. The active processor cycle consists of states 1 through 7. State 8 is a dead zone to provide a neutral stopping point for halting the processor.

A clock output (CLKOUT) signal is generated by the processor to synchronize external devices to the processor's internal cycles. CLKOUT is high during states 8, 1, 2 and 3, and low during states 4, 5, 6 and 7. Its frequency is one-fourth of that of CLKIN. Except during $\overline{\text{RESET}}$, the CLKOUT signal runs continuously.

Bus Interface

The ADSP-2100 can relinquish control of the memory buses to an external device. When the external device requires access to memory, it asserts the Bus Request ($\overline{\text{BR}}$) signal. After completing the current instruction, the processor halts program execution, tristates the PMA, PMD, $\overline{\text{PMS}}$, $\overline{\text{PMRD}}$, $\overline{\text{PMWR}}$ and PMDA output drivers and the DMA, DMD, $\overline{\text{DMS}}$, $\overline{\text{DMRD}}$ and $\overline{\text{DMWR}}$ output drivers, and asserts the Bus Grant ($\overline{\text{BG}}$) signal. When the $\overline{\text{BR}}$ signal is released, the processor re-enables the output drivers, releases the $\overline{\text{BG}}$ signal, and continues program execution from the point where it stopped.

Program Memory Interface

The Program Memory Interface supports two buses: the program memory address bus (PMA) and the program memory data bus (PMD). The 14-bit PMA bus directly addresses up to 16K words. The PMD bus is bidirectional and 24 bits wide.

Since program memory can be used for both instruction code and data storage, the Program Memory Data Access (PMDA) signal is asserted whenever data, as opposed to an instruction code, is fetched. There is no placement restriction for instruction code and data in program memory area if less than 16K words are used. Since the timing of PMDA is compatible with that of the PMA lines, it may be used as a 15th address line if desired. This effectively doubles the program memory area to 32K, which must be split into 16K dedicated to instruction codes and 16K to data.

The program memory data lines are bidirectional. The Program Memory Select ($\overline{\text{PMS}}$) signal indicates access to the Program Memory and can be used as a chip select signal. The Program Memory Write ($\overline{\text{PMWR}}$) signal indicates a write operation and can be used as a write strobe. The Program Memory Read ($\overline{\text{PMRD}}$) signal indicates a read operation and can be used as a read strobe or output enable signal.

Although the processor internal data bus is only 16 bits, the ADSP-2100 can write to the full 24-bit program memory using the PX register.

Data Memory Interface

The Data Memory Interface supports two buses: the Data Memory Address bus (DMA) and the Data Memory Data bus (DMD). The 14-bit DMA bus directly addresses up to 16K words of data. The DMD bus is bidirectional and 16 bits wide. The Data Memory Select ($\overline{\text{DMS}}$) signal indicates access to the Data Memory and can be used as a chip select signal. The Data Memory Write ($\overline{\text{DMWR}}$) signal indicates a write operation and can be used as a write strobe. The Data Memory Read ($\overline{\text{DMRD}}$) signal indicates a read operation and can be used as a read strobe or output enable signal.

The ADSP-2100 supports memory-mapped I/O, with the peripherals memory mapped into the data memory address space and accessed by the processor in the same manner as data memory.

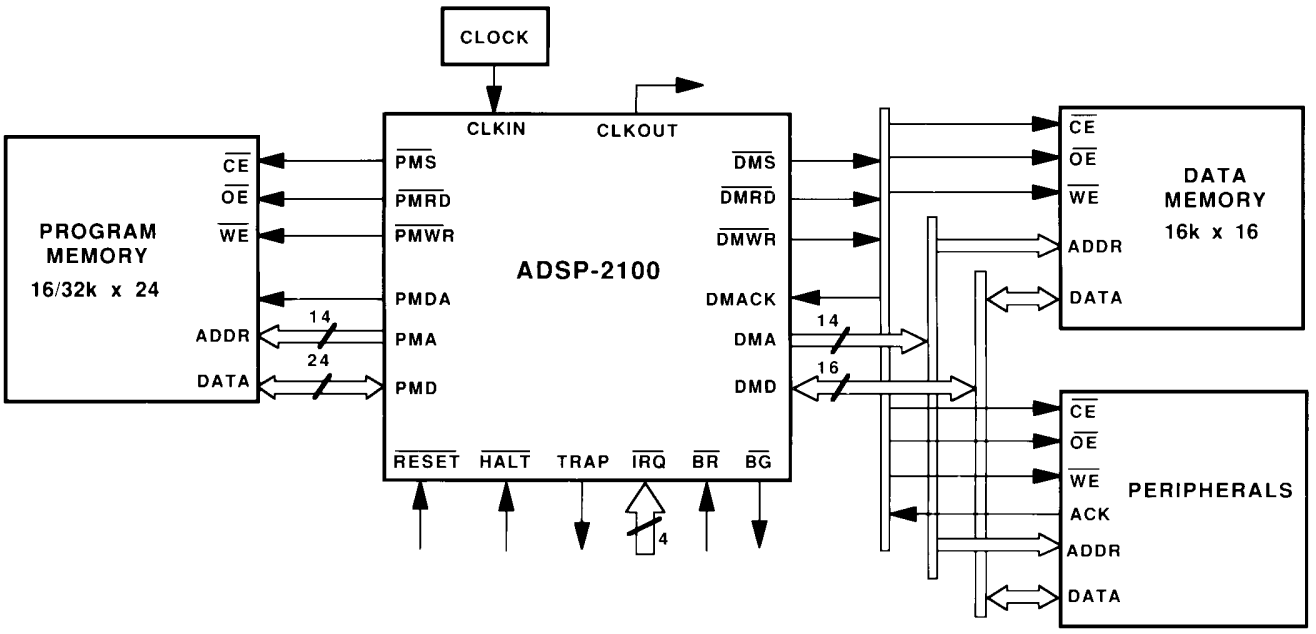


Figure 7. Basic System Configuration

To allow interfacing to slower peripherals, the data memory acknowledge (DMACK) signal is provided. The ADSP-2100 checks the status of the DMACK signal at the end of each processor cycle. If the DMACK signal is not asserted, the processor extends the current cycle by another full cycle. This extension occurs as many times as necessary until the DMACK signal is asserted and the access is completed.

Interrupt Handling

The ADSP-2100 provides four direct interrupt input pins, $\overline{IRQ_0}$ to $\overline{IRQ_3}$. Each interrupt pin corresponds to a particular interrupt priority level from 3 (highest) to 0 (lowest). The four interrupt levels are internally prioritized and individually maskable. These input pins can be programmed to be either level- or edge-sensitive.

The ADSP-2100 supports a vectored interrupt scheme: when an external interrupt is acknowledged, the processor switches program control to the interrupt vector address corresponding to the interrupt level (program memory locations 0000 to 0003). Interrupts can optionally be nested so that a higher priority interrupt can preempt the currently executing interrupt service routine.

Processor Control Interface

The processor control interface provides external control over the activity of the processor. The control signals are $\overline{RESET}$, $\overline{HALT}$ and TRAP.

The $\overline{RESET}$ signal initiates a master reset of the ADSP-2100. The $\overline{RESET}$ signal must be asserted after the chip is powered up to assure proper initialization. The master reset performs the following:

1 Initialize internal clock circuitry
2 Reset all internal stack pointers
3 Clear the cache memory monitor
4 If there is no pending bus request, PMA is driven with 0004
5 Mask all interrupts
6 Clear MSTAT register.

The $\overline{HALT}$ signal is used to suspend program execution temporarily. When $\overline{HALT}$ is asserted, the processor stops at the end of the current instruction. To ensure that the processor always halts after completion of an instruction fetch, an external fetch of the next instruction is forced even if the instruction is available from internal cache memory. Since the processor always stops after an external instruction fetch cycle, the controlling device is able to observe the instruction address where the program was stopped. The halt condition can be sustained for any length of time, during which all signals generated by the processor will remain static (maintaining the output at state 8). The processor will continue normal execution when the $\overline{HALT}$ line is released.

The TRAP signal is generated by the processor whenever a TRAP instruction is executed. Assertion of the TRAP signal indicates that the processor has stopped instruction execution just after the end of the cycle which executed the TRAP instruction. The TRAP state is identical to the $\overline{HALT}$ state, with the processor output frozen in state 8. In this case, the processor PMA bus contains the address of the instruction following the TRAP instruction. The TRAP signal remains asserted until the $\overline{HALT}$ signal is asserted externally. When the $\overline{HALT}$ signal assertion is sensed, the processor releases the TRAP signal. However, the processor remains in the halt condition until the $\overline{HALT}$ line is released.

Multiprocessor Synchronization

Even when multiple ADSP-2100s are driven from the same CLKIN signal, there is a phase ambiguity between the various processors. This ambiguity can be prevented by using a single master $\overline{RESET}$ signal synchronized to CLKIN. When the master $\overline{RESET}$ is released, all the processors begin state 5 on the same edge of CLKIN. Once initialized in this manner, the cycle states of the processors remain synchronized with each other.

INSTRUCTION SET DESCRIPTION

The ADSP-2100 assembly language uses an algebraic syntax for ease of coding and readability. The sources and destinations of computations and data movements are written explicitly in each assembly statement, eliminating cryptic assembler mnemonics. Nevertheless, every instruction assembles into a single 24-bit word and executes in a single cycle. The instructions encompass a wide variety of instruction types along with a high degree of operational parallelism. There are five basic categories of instructions: data move instructions, computational instructions, multifunction instructions, program flow control instructions and miscellaneous instructions. Each of these instruction types is described briefly. The complete instruction set is summarized in Table IV at the end of this section.

Data Move Instructions

Table II gives a list of all registers that are accessible using the data move instructions. (Only the program counter (PC), the instruction register, the arithmetic feedback register (AF) and the multiplier feedback register (MF) are not on this list.) This set of registers is denoted as *reg* in the instruction set summary given in Table IV. A subset of the *reg* group associated with the computational units, which generally hold data as opposed to address or status information, is denoted as *dreg*.

The data move instructions include transfers between internal registers, between data memories and internal registers, between program memories and internal registers, and immediate value loading of registers and data memories. The content of every *reg*

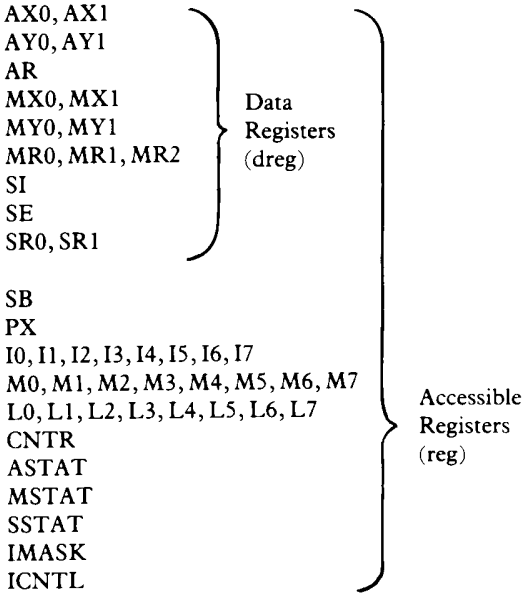


Table II. Register Classification

can also be loaded to any other *reg*. Every *reg* can be loaded with an immediate value which is the full width of the particular register being loaded.

Two addressing modes are supported for data memory transfers: direct addressing and indirect addressing. In direct addressing, the memory address is supplied from the instruction word. In indirect addressing, one of the data address generators provides the address. Using direct addressing, the content of a data memory location can be written and read by any *reg*. Using indirect addressing, the content of a data memory location can only be written and read by a *dreg*. Immediate data load to data memory is permitted with indirect addressing. Only the indirect addressing mode is supported for program memory data transfers, and contents of a program memory location can be read and written to any *dreg*.

Computational Instructions

There are three types of operations associated with the computational units: ALU operations, MAC operations and shifter operations. With few exceptions, all these computational instructions can be made conditional. (The permissible conditions are specified in Table I.) Each computational unit has a set of input registers and output registers. A list of permissible input operands and result registers for each of the units is given in Table III.

Multifunction Instructions

Multifunction instructions execute one computational operation with one or two data moves. All of the multifunction instructions utilize various combinations of the computational and data move operations described above. Since the instruction word is only 24 bits wide, only certain combinations are valid. In general, the following rules are followed.

1. Only one unconditional computational operation can be specified
2. Any memory transfer must use the indirect addressing mode
3. Data move operations can only involve data registers (*dregs*)
4. Only an ALU or a MAC operation can be specified with two operand fetches, one from program memory and one from data memory.

Program Flow Control Instructions

Program flow control instructions include JUMP, CALL, return from subroutine, return from interrupt, DO UNTIL and TRAP. All of these instructions can be made conditional. The JUMP and CALL instructions support both direct addressing, with the destination address specified by the instruction word, and indirect addressing, with the destination address specified by one of the I registers in DAG2.

Miscellaneous Instructions

Miscellaneous instructions include indirect register modify, stack control, mode control and NOP operations.

ALU

Source for X input port (xop)	Source for Y input port (yop)	Destination for output port R
AX0, AX1	AY0, AY1	AR
AR	AF	AF
MR0, MR1, MR2		
SR0, SR1		

MAC

Source for X input port (xop)	Source for Y input port (yop)	Destination for output port R
MX0, MX1	MY0, MY1	MR (MR2, MR1, MR0)
AR	MF	MF
MR0, MR1, MR2		
SR0, SR1		

Shifter

Source for Shifter input (xop)	Destination for Shifter output
SI	SR (SR1, SR0)
AR	
MR0, MR1, MR2	
SR0, SR1	

Table III. Computational Input/Output Registers

These conventions are used in Table IV:

1. All keywords are shown in capital letters.
2. Brackets enclose optional parts of the syntax.
3. Vertical lines indicate that one parameter must be chosen from those enclosed.
4. Table I defines the conditions for *condition*.
5. Table II defines the set of registers for *dreg* and *reg*.
6. Table III defines the set of registers for *xop* and *yop*.
7. <data> represents an immediate value.
8. <address> may be an immediate value or label.
9. <comp>, in a multifunction instruction, represents all legal ALU, MAC or Shifter operations with these restrictions:
   - All operations are performed unconditionally
   - Shift Immediate operations are not allowed
   - ALU division (DIVS, DIVQ) is not allowed

DATA MOVE INSTRUCTIONS

Register Move

reg = reg;

Load Register Immediate

reg = <data>;

Data Memory Read (direct address)

reg = DM(<address>);

Data Memory Read (indirect address)

dreg = DM(I0 | I1 | I2 | I3 | I4 | I5 | I6 | I7 , M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7);

Program Memory Read (indirect address)

dreg = PM(I4 | I5 | I6 | I7 , M4 | M5 | M6 | M7);

Data Memory Write (direct address)

DM(<address>) = reg;

Data Memory Write (indirect address)

DM(I0 | I1 | I2 | I3 | I4 | I5 | I6 | I7 , M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7) = dreg | <data> ;

Program Memory Write (indirect address)

PM(I4 | I5 | I6 | I7 , M4 | M5 | M6 | M7) = dreg;

COMPUTATIONAL INSTRUCTIONS: ALU

Add/Add with Carry

[IF condition] AR | AF = xop [+yop | +C | +yop+C] ;

Subtract X-Y/Subtract X-Y with Borrow

[IF condition] AR | AF = xop [-yop | -yop+C-1] ;

Subtract Y-X/Subtract Y-X with Borrow

[IF condition] AR | AF = yop [-xop | -xop+C-1] ;

AND, OR, Exclusive OR

[IF condition] AR | AF = xop AND | OR | XOR yop ;

Pass/Clear

[IF condition] AR | AF = PASS xop | yop ;

Negate

[IF condition] AR | AF = - xop | yop ;

NOT

[IF condition] AR | AF = NOT xop | yop ;

Absolute Value

[IF condition] AR | AF = ABS xop | yop ;

Increment

[IF condition] AR | AF = yop +1 ;

Decrement

[IF condition] AR | AF = yop -1 ;

Divide

DIVS yop, xop ;
DIVQ xop ;

COMPUTATIONAL INSTRUCTIONS: SHIFTER

Arithmetic Shift

[IF condition] SR = [SR OR] ASHIFT xop (HI) | (LO) ;

Logical Shift

[IF condition] SR = [SR OR] LSHIFT xop (HI) | (LO) ;

Normalize

[IF condition] SR = [SR OR] NORM xop (HI) | (LO) ;

Derive Exponent

[IF condition] SE = EXP xop (HI) | (LO) | (HIX) ;

Block Exponent Adjust

[IF condition] SB = EXPADJ xop ;

Arithmetic Shift Immediate

SR = [SR OR] ASHIFT xop BY <data> [(HI) | (LO)] ;

Logical Shift Immediate

SR = [SR OR] LSHIFT xop BY <data> [(HI) | (LO)] ;

COMPUTATIONAL INSTRUCTIONS: MAC

Multiply

[IF condition] [MR | MF] = xop*yop ([SS | SU | US | UU | RND]) ;

Multiply Accumulate

[IF condition] [MR | MF] = MR + xop*yop ([SS | SU | US | UU | RND]) ;

Multiply Subtract

[IF condition] [MR | MF] = MR − xop*yop ([SS | SU | US | UU | RND]) ;

Clear

[IF condition] [MR | MF] = 0 ;

Transfer MR

[IF condition] [MR | MF] = MR [(RND)] ;

Conditional MR Saturation

IF MV SAT MR ;

PROGRAM FLOW CONTROL INSTRUCTIONS

Jump

[IF condition] JUMP ([I4 | I5 | I6 | I7 | <address>]) ;

Call

[IF condition] CALL ([I4 | I5 | I6 | I7 | <address>]) ;

Return from Subroutine

[IF condition] RTS ;

Return from Interrupt

[IF condition] RTI ;

Do Until

DO <address> [UNTIL condition] ;

Trap

[IF condition] TRAP ;

MULTIFUNCTION INSTRUCTIONS

Computation with Memory Read

<comp> , dreg = [DM([I0 | I1 | I2 | I3 | I4 | I5 | I6 | I7] , [M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7]) | PM ([I4 | I5 | I6 | I7] , [M4 | M5 | M6 | M7])] ;

Computation with Data Register Move

<comp> , dreg = dreg ;

Computation with Memory Write

[DM([I0 | I1 | I2 | I3 | I4 | I5 | I6 | I7] , [M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7]) | PM ([I4 | I5 | I6 | I7] , [M4 | M5 | M6 | M7])] = dreg, <comp> ;

Data & Program Memory Read

AX0	= DM (	I0	,	M0	) ,	AY0	= PM (	I4	,	M4	) ;
AX1		I1		M1		AY1		I5		M5	
MX0		I2		M2		MY0		I6		M6	
MX1		I3		M3		MY1		I7		M7	

ALU/MAC Operation with Data & Program Memory Read*

<ALU>	, AX0	= DM (	I0	,	M0	) ,	AY0	= PM (	I4	,	M4	) ;
<MAC>	AX1		I1		M1		AY1		I5		M5	
	MX0		I2		M2		MY0		I6		M6	
	MX1		I3		M3		MY1		I7		M7	

*ALU Division operations not allowed.

MISCELLANEOUS INSTRUCTIONS

Stack Control

[PUSH STS] [, POP CNTR] [, POP PC] [, POP LOOP] ;
[POP]

Mode Control

[ENA BIT_REV][, ENA AV_LATCH][, ENA AR_SAT][, ENA SEC_REG];
[DIS] [DIS] [DIS] [DIS]

Modify Address Register

MODIFY	(	I0	,	M0	) ;
		I1		M1	
		I2		M2	
		I3		M3	
		I4		M4	
		I5		M5	
		I6		M6	
		I7		M7	

No Operation

NOP ;

Table IV. Instruction Set Summary

SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

Parameter		ADSP-2100/ADSP-2100A J, K, AJ, AK Grades Min	Max	S, AS, AT, AU Grades Min	Max	Unit
V_{DD}	Supply Voltage	4.75	5.25	4.50	5.50	V
T_{AMB}	Ambient Operating Temperature	0	+70	−55	+125	°C

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	ADSP-2100 J & K Grades Min	Max	S Grade Min	Max	Unit
V_{IH}	Hi-Level Input Voltage[1]	@ V_{DD} = max	2.0		2.2		
V_{IL}	Lo-Level Input Voltage[1]	@ V_{DD} = min		0.8		0.8	V
V_{OH}	Hi-Level Output Voltage[2]	@ V_{DD} = min, I_{OH} = −1mA	2.4		2.4		V
V_{OL}	Lo-Level Output Voltage[2]	@ V_{DD} = min, I_{OL} = 4mA		0.4		0.6	V
I_{IH}	Hi-Level Input Current[3]	@ V_{DD} = max, V_{IN} = max		10		10	μA
I_{IL}	Lo-Level Input Current[3]	@ V_{DD} = max, V_{IN} = 0V		10		10	μA
I_{OZH}	Tristate Leakage Current[4]	@ V_{DD} = max, V_{IN} = max[7]		10		10	μA
I_{OZL}	Tristate Leakage Current[5]	@ V_{DD} = max, V_{IN} = 0V[7]		10		10	μA
I_{OZL}	Tristate Pullup Current[6]	@ V_{DD} = max, V_{IN} = 0V[7]		150		150	μA
I_{DD}	Supply Current (Power-Down)[9]	@ V_{DD} = max, V_{IN} = 0V[6,7]		10		15	mA
I_{DD}	Supply Current (Dynamic)	@ V_{DD} = max, max clock rate[8]		90		100	mA

Parameter		Test Conditions	ADSP-2100A AJ&AK Grades Min	Max	AS Grade Min	Max	AT Grade Min	Max	AU Grade Min	Max	Unit
V_{IH}	Hi-Level Input Voltage[1]	@ V_{DD} = max	2.0		2.2		2.2		2.2		V
V_{IH}	Hi-Level Input Voltage at CLKIN	@ V_{DD} = max	2.2		2.4		2.4		2.4		V
V_{IL}	Lo-Level Input Voltage[1]	@ V_{DD} = min		0.8		0.8		0.8		0.8	V
V_{IL}	Lo-Level Input Voltage at CLKIN	@ V_{DD} = min		0.8		0.8		0.8		0.8	V
V_{OH}	Hi-Level Output Voltage[2]	@ V_{DD} = min, I_{OH} = −1mA	2.4		2.4		2.4		2.4		V
V_{OL}	Lo-Level Output Voltage[2]	@ V_{DD} = min, I_{OL} = 4mA		0.4		0.6		0.6		0.6	V
I_{IH}	Hi-Level Input Current[3]	@ V_{DD} = max, V_{IN} = max		10		10		10		10	μA
I_{IL}	Lo-Level Input Current[3]	@ V_{DD} = max, V_{IN} = 0V		10		10		10		10	μA
I_{OZH}	Tristate Leakage Current[4]	@ V_{DD} = max, V_{IN} = max[7]		10		10		10		10	μA
I_{OZL}	Tristate Leakage Current[5]	@ V_{DD} = max, V_{IN} = 0V[7]		10		10		10		10	μA
I_{OZL}	Tristate Pullup Current[6]	@ V_{DD} = max, V_{IN} = 0V[7]		180		180		180		180	μA
I_{DD}	Supply Current (Power-Down)[9]	@ V_{DD} = max, V_{IN} = 0V[6,7]		10		15		15		15	mA
I_{DD}	Supply Current (Dynamic)	@ V_{DD} = max, max clock rate[8]		150		130		180		200	mA

NOTES

[1]Applies to pins: $PMD_{0\ 23}$, $DMD_{0\ 15}$, $\overline{BR}$, $\overline{IRQ}_{0\ 3}$, DMACK, $\overline{RESET}$, $\overline{HALT}$, (48 input pins for ADSP-2100A). Includes CLKIN for ADSP-2100 (49 input pins).

[2]Applies to pins: $PMA_{0\ 13}$, $\overline{PMS}$, $PMD_{0\ 23}$, $\overline{PMRD}$, $\overline{PMWR}$, PMDA, $\overline{BG}$, $DMA_{0\ 13}$, $\overline{DMS}$, $DMD_{0\ 15}$, $\overline{DMRD}$, $\overline{DMWR}$, TRAP, CLKOUT (78 output pins).

[3]Applies to pins: $\overline{BR}$, $\overline{IRQ}_{0\ 3}$, DMACK, $\overline{RESET}$, $\overline{HALT}$, CLKIN (9 input-only pins).

[4]Applies to pins: $PMA_{0\ 13}$, $\overline{PMS}$, $PMD_{0\ 23}$, $\overline{PMRD}$, $\overline{PMWR}$, PMDA, $DMA_{0\ 13}$, $\overline{DMS}$, $DMD_{0\ 15}$, $\overline{DMRD}$, $\overline{DMWR}$ (75 tristateable pins).

[5]Applies to pins: $PMA_{0\ 13}$, PMDA, $DMA_{0\ 13}$ (29 tristateable pins w/o pullup).

[6]Applies to pins: $PMD_{0\ 23}$, $\overline{PMS}$, $\overline{PMRD}$, $\overline{PMWR}$, $DMD_{0\ 15}$, $\overline{DMS}$, $\overline{DMRD}$, $\overline{DMWR}$ (46 tristateable pins w/pullup).

[7]Additional Test Conditions: V_{IN} = 0V on $\overline{BR}$ and $\overline{RESET}$, CLKIN active, forces tristate condition.

[8]Additional Test Conditions: Outputs loaded TTL loads w/100pF capacitance, V_{IH} = 2.4V, V_{IL} = 0.4V, clock rate = max.

[9]"Power-down" refers to an idle state. While the processor does not have any special standby or low-power mode, these conditions represent the lowest power consumption state.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage . . . . . −0.3V to +7V
Input Voltage . . . . . −0.3V to V_{DD} +0.3V
Output Voltage Swing . . . . . −0.3V to V_{DD} +0.3V
Operating Temperature Range (Ambient) . . −55°C to +125°C
Storage Temperature Range . . . . . −65°C to +150°C
Lead Temperature (10sec) PGA . . . . . +300°C
Lead Temperature (5sec) PQFP . . . . . +280°C

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ORDERING INFORMATION

Part Number	Speed (MHz)	Temperature Range	Package
ADSP-2100JG	6.144	0 to +70°C	100-Pin Grid Array
ADSP-2100KG	8.192	0 to +70°C	100-Pin Grid Array
ADSP-2100AJG	10.24	0 to +70°C	100-Pin Grid Array
ADSP-2100AKG	12.50	0 to +70°C	100-Pin Grid Array
ADSP-2100JP	6.144	0 to +70°C	100-PQFP
ADSP-2100KP	8.192	0 to +70°C	100-PQFP
ADSP-2100AJP	10.24	0 to +70°C	100-PQFP
ADSP-2100AKP	12.50	0 to +70°C	100-PQFP
ADSP-2100SG	6.144	−55°C to +125°C	100-Pin Grid Array
ADSP-2100ASG	8.192	−55°C to +125°C	100-Pin Grid Array
ADSP-2100ATG	10.24	−55°C to +125°C	100-Pin Grid Array
ADSP-2100AUG	12.50	−55°C to +125°C	100-Pin Grid Array
ADSP-2100SG/883B	6.144	−55°C to +125°C	100-Pin Grid Array
ADSP-2100ASG/883B	8.192	−55°C to +125°C	100-Pin Grid Array
ADSP-2100ATG/883B	10.24	−55°C to +125°C	100-Pin Grid Array
ADSP-2100AUG/883B	12.50	−55°C to +125°C	100-Pin Grid Array

ADSP-2100/ADSP-2100A Development Tools

Part Number	Description
ADDS-2110	Cross-Software and Simulator (VAX/VMS)
ADDS-2121	Cross-Software (IBM PC/DOS)
ADDS-2122	Simulator (IBM PC/DOS)
ADDS-2123-C	Cross-Software and Simulator (Sun 2/3, Unix BSD 4.2)
ADDS-2130	C Compiler, Cross-Software and Simulator (VAX/VMS)
ADDS-2131	C Compiler, Cross-Software and Simulator (IBM PC/DOS)
ADDS-2133-C	C Compiler, Cross-Software and Simulator (Sun 2/3, Unix BSD 4.2)
ADDS-2150A-8	ADSP-2100A 8MHz In-Circuit Emulator (110V)
ADDS-2150AE-8	ADSP-2100A 8MHz In-Circuit Emulator (220V)
ADDS-2160-8	ADSP-2100A 8MHz Evaluation Board
ADDS-2169	University Package (ADDS-2131 and ADDS-2160)
ADDS-2190	Three Day ADSP-2100 Workshop (U.S.)
ADDS-2190E	Three Day ADSP-2100 Workshop (Europe)

ESD SENSITIVITY

The ADSP-2100 and ADSP-2100A feature proprietary input protection circuitry. Per Method 3015 of MIL-STD-883, the ADSP-2100 has been classified as a Class 1 device and the ADSP-2100A as a Class 2 device.

Proper ESD precautions are strongly recommended to avoid functional damage or performance degradation. Charges as high as 4000 volts readily accumulate on the human body and test equipment and discharge without detection. Unused devices must be stored in conductive foam or shunts, and the foam should be discharged to the destination socket before devices are removed. For further information on ESD precautions, refer to Analog Devices' *ESD Prevention Manual*.

SWITCHING CHARACTERISTICS

GENERAL NOTES

Use the exact timing information given. Do not attempt to derive parameters from the addition or subtraction of others. While this addition or subtraction would yield meaningful results for an individual part, the values given in this data sheet reflect statistical variations and worst cases. Consequently, you cannot meaningfully add up parameters to derive or "verify" longer times.

TIMING NOTES

Switching characteristics specify how the processor is switching its signals. The user has no control over this operation. It is dependent on the internal design. Timing requirements specify the timing of signals that the user has control over such as the placement of data on the DMD bus as input for a read operation.

Timing requirements are used by a designer to guarantee that the processor operates correctly with another device while switching characteristics inform the designer what the device is doing under any given circumstance. Switching characteristics are also referenced to ensure that any timing requirement of a device connected to the processors (such as a memory) is satisfied.

MEMORY REQUIREMENTS

This chart links common memory device specification names and ADSP-2100/ADSP-2100A timing parameters for your convenience.

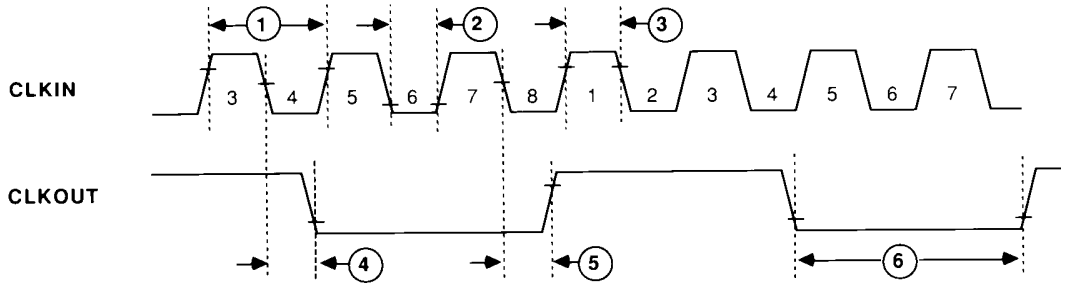
Parameter Number	Parameter Name	Common Memory Device Specification Name
41	PMA Valid to $\overline{\text{PMWR}}$ Low	Address Set Up to Write Start
79	DMA Valid to $\overline{\text{DMWR}}$ Low	Address Set Up to Write Start
42	$\overline{\text{PMWR}}$ High to PMA Invalid	Address Hold Time
80	$\overline{\text{DMWR}}$ High to DMA Invalid	Address Hold Time
55	PMD Out Valid to $\overline{\text{PMWR}}$ High	Data Set Up Time
91	DMD Out Valid to $\overline{\text{DMWR}}$ High	Data Set Up Time
54	$\overline{\text{PMWR}}$ High to PMD Out Invalid	Data Hold Time
90	$\overline{\text{DMWR}}$ High to DMD Out Invalid	Data Hold Time
58	$\overline{\text{PMRD}}$ Low to PMD Input Valid	$\overline{\text{OE}}$ to Data Valid
94	$\overline{\text{DMRD}}$ Low to DMD Input Valid	$\overline{\text{OE}}$ to Data Valid
59	PMA Valid to PMD Input Valid	Address Access Time
95	DMA Valid to DMD Input Valid	Address Access Time
41 + 40	PMA Valid to $\overline{\text{PMWR}}$ Low + $\overline{\text{PMWR}}$ Width Low	Address Set Up to Write End
79 + 78	DMA Valid to $\overline{\text{DMWR}}$ Low + $\overline{\text{DMWR}}$ Width Low	Address Set Up to Write End

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

ADSP-2100 Clock Signals		Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Timing Requirements										
1	CLKIN Period[1]	A	40.5		30.5		40.5		ns	
2	CLKIN Width Low	A	11		8		11		ns	
3	CLKIN Width High	A	18		12		18		ns	
Switching Characteristics										
4	CLKIN Low (3-4) to CLKOUT Low	B	13	34	13	29	11	34	ns	
5	CLKIN Low (7-8) to CLKOUT High	B	6	24	6	20	5	24	ns	
6	CLKOUT Width Low	A	60		45		60		ns	4

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

	ADSP-2100A Clock Signals	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Timing Requirements														
1	CLKIN Period[1]	A	24.4		20		30.5		24.4		20		ns	
2	CLKIN Width Low	A	7		4		8		7		4		ns	
3	CLKIN Width High	A	9		8		12		9		8		ns	
Switching Characteristics														
4	CLKIN Low (3-4) to CLKOUT Low	B		24		22		29		24		22	ns	
5	CLKIN Low (7-8) to CLKOUT High	B		20		18		20		20		18	ns	
6	CLKOUT Width Low	A	36		28		45		36		28		ns	4



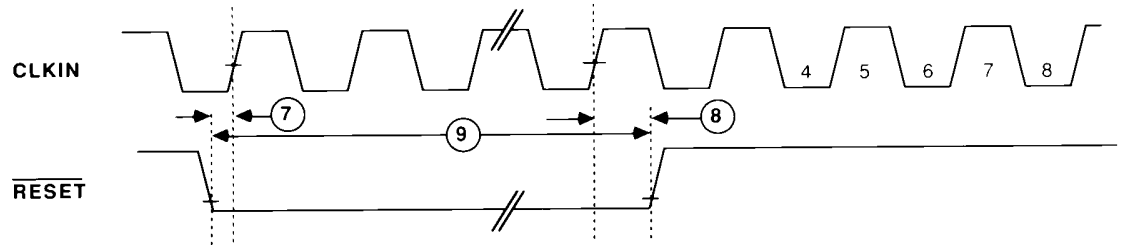
NOTE
The Processor Cycle is Divided into 8 Internal States Determined by the Rising and Falling Edges of CLKIN. CLKOUT is Synchronized to the Processor States as Shown Above.

Figure 8. Clock Signals

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

	ADSP-2100 Control Signals	Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Timing Requirements										
7	$\overline{\text{RESET}}$ Low to CLKIN High	B	2		2		2		ns	
8	CLKIN High to $\overline{\text{RESET}}$ High	B	6	36	4	26	6	36	ns	2 (max only)
9	$\overline{\text{RESET}}$ Width Low	A	162		122		170		ns	8

	ADSP-2100A Control Signals	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Timing Requirements														
7	$\overline{\text{RESET}}$ Low to CLKIN High	B	2		2		2		2		2		ns	
8	CLKIN High to $\overline{\text{RESET}}$ High	B	4	20	4	16	6	26	4	20	4	16	ns	2 (max only)
9	$\overline{\text{RESET}}$ Width Low	A	98		80		128		98		80		ns	8



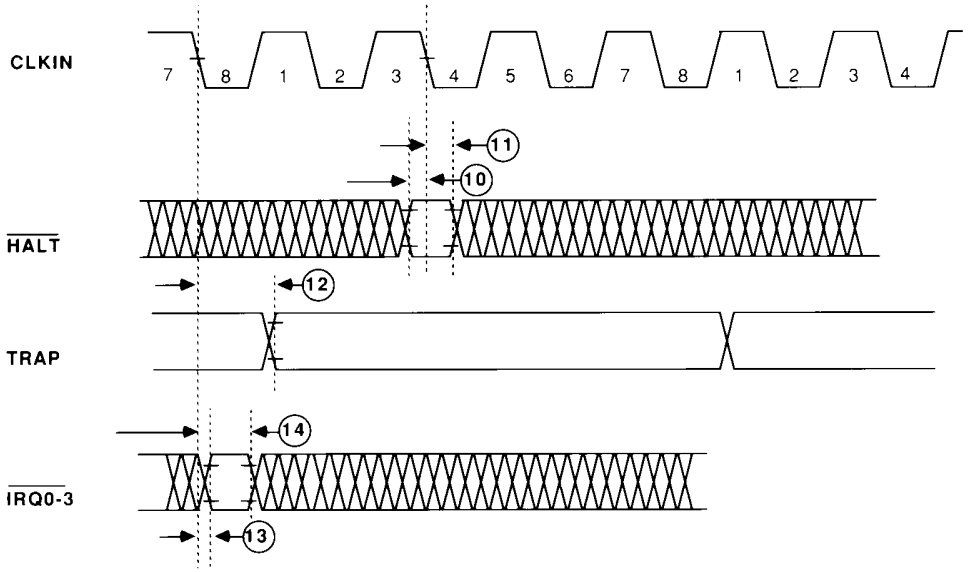
NOTE
The Reset signal determines the phase of the processor cycle.
The processor starts from state 4 after the release of the Reset signal.

Figure 9. $\overline{\text{RESET}}$ Signal

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

ADSP-2100 Control Signals		Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Timing Requirements										
10	$\overline{\text{HALT}}$ Valid to CLKIN Low (3-4)	B	0		0		0		ns	
11	CLKIN Low (3-4) to $\overline{\text{HALT}}$ Invalid	B	12		10		12		ns	
Switching Characteristics										
12	CLKIN Low (7-8) to TRAP Valid	B		25		20		25	ns	
Interrupts										
Timing Requirements										
13	CLKIN Low (7-8) to $\overline{\text{IRQ}}$ Valid	B		2		2		1	ns	
14	CLKIN Low (7-8) to $\overline{\text{IRQ}}$ Invalid	B	21		17		21		ns	

ADSP-2100A Control Signals		Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Timing Requirements														
10	$\overline{\text{HALT}}$ Valid to CLKIN Low (3-4)	B	2		2		2		2		2		ns	
11	CLKIN Low (3-4) to $\overline{\text{HALT}}$ Invalid	B	10		8		10		10		8		ns	
Switching Characteristics														
12	CLKIN Low (7-8) to TRAP Valid	B		18		16		20		18		16	ns	
Interrupts														
Timing Requirements														
13	CLKIN Low (7-8) to $\overline{\text{IRQ}}$ Valid	B		1		1		1		1		1	ns	
14	CLKIN Low (7-8) to $\overline{\text{IRQ}}$ Invalid	B	14		14		17		14		14		ns	



NOTE
The Control Signals are Shown in Relationship to the Processor States in Which They are Recognized or Asserted as Defined by CLKIN. There is No Implied Relationship between HALT, TRAP, and IRQ_{0-3}.

Figure 10. Control Signals

Notes 1 and 2 and information about explaining the Derating Factors and Test Codes appear on page 32.

ADSP-2100 Bus Request Asserted		Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Timing Requirements										
15	$\overline{BR}$ Valid to CLKIN Low (3-4)	B	1		1		1		ns	
16	CLKIN Low (3-4) to $\overline{BR}$ Invalid	B	10		7		10		ns	
Switching Characteristics										
17	CLKIN Low (3-4) to $\overline{BG}$ Low	B		38		30		38	ns	
19	$\overline{BG}$ Low to xMxx Disable[2]	D		22		17		22	ns	

ADSP-2100A Bus Request Asserted		Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Timing Requirements														
15	$\overline{BR}$ Valid to CLKIN Low (3-4)	B	4		4		1		4		4		ns	
16	CLKIN Low (3-4) to $\overline{BR}$ Invalid	B	4		4		7		4		4		ns	
Switching Characteristics														
17	CLKIN Low (3-4) to $\overline{BG}$ Low	B		26		24		30		26		24	ns	
19	$\overline{BG}$ Low to xMxx Disable[2]	D		16		16		17		16		16	ns	

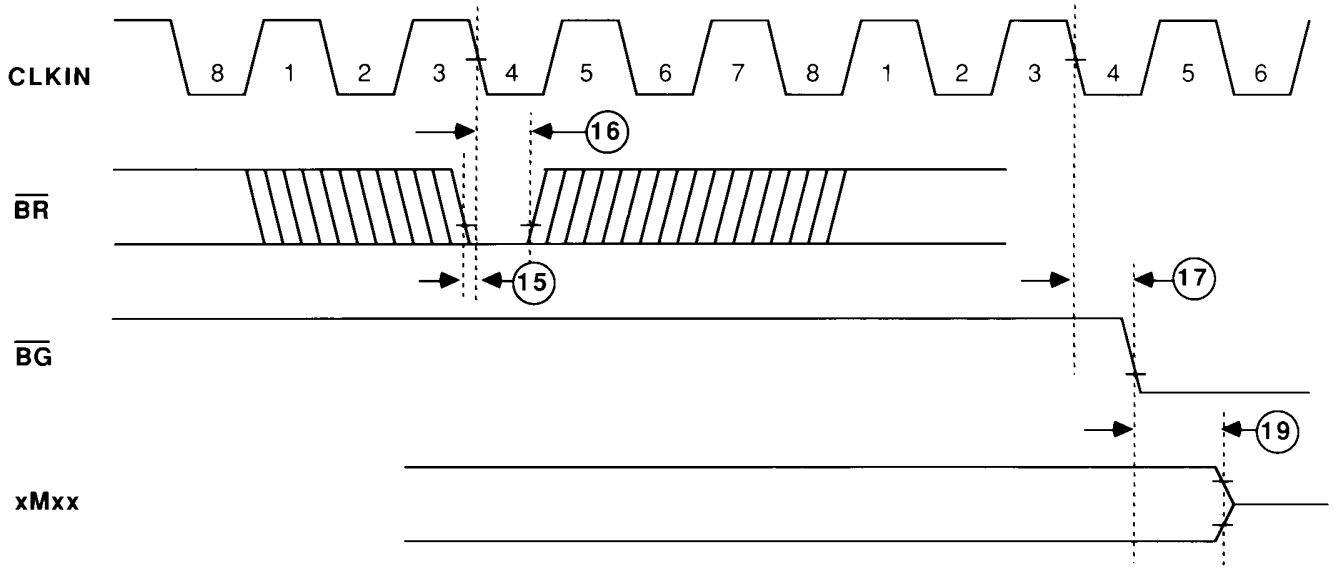


Figure 11. Bus Request Asserted

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

ADSP-2100 Bus Request Negated	Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Timing Requirements									
15 $\overline{BR}$ Valid to CLKIN Low (3-4)	B	1		1		1		ns	
16 CLKIN Low (3-4) to $\overline{BR}$ Invalid	B	10		7		10		ns	
Switching Characteristics									
18 CLKIN Low (7-8) to $\overline{BG}$ High	B		31		25		31	ns	
20 xMxx Enable to $\overline{BG}$ High[2]	F		12		10		12	ns	

ADSP-2100A Bus Request Negated	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Timing Requirements													
15 $\overline{BR}$ Valid to CLKIN Low (3-4)	B	4		4		1		4		4		ns	
16 CLKIN Low (3-4) to $\overline{BR}$ Invalid	B	4		4		7		4		4		ns	
Switching Characteristics													
18 CLKIN Low (7-8) to $\overline{BG}$ High	B		24		20		25		24		20	ns	
20 xMxx Enable to $\overline{BG}$ High[2]	F		10		8		10		10		8	ns	

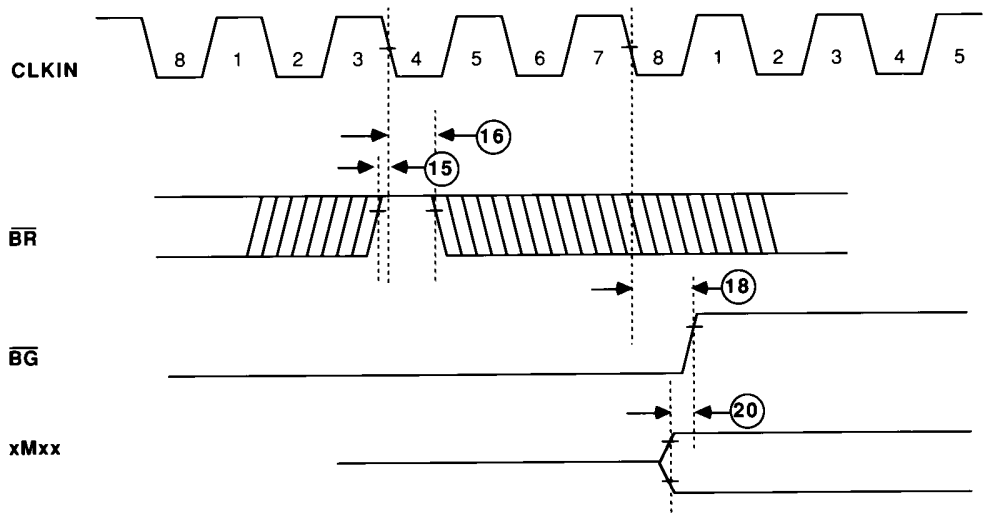
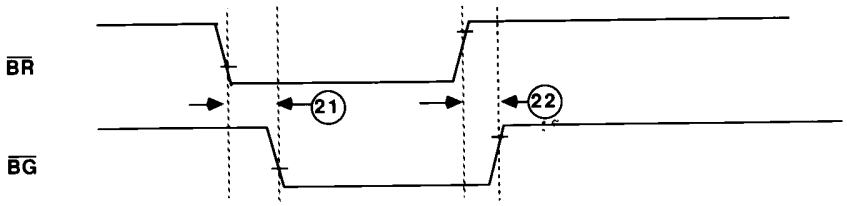


Figure 12. Bus Request Negated

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

ADSP-2100 Bus Request/Grant with $\overline{\text{RESET}}$ Low	Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Switching Characteristics									
21 $\overline{\text{BR}}$ Low to $\overline{\text{BG}}$ Low during reset	A		28		23		28	ns	
22 $\overline{\text{BR}}$ High to $\overline{\text{BG}}$ High during reset	A		21		18		21	ns	

ADSP-2100A Bus Request/Grant with $\overline{\text{RESET}}$ Low	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Switching Characteristics													
21 $\overline{\text{BR}}$ Low to $\overline{\text{BG}}$ Low during reset	A		18		16		23		18		16	ns	
22 $\overline{\text{BR}}$ High to $\overline{\text{BG}}$ High during reset	A		16		14		18		16		14	ns	



NOTE
During Reset, the Processor Bus Ignores the CLKIN Signal and Therefore the Bus Request/Grant Signals Operate Asynchronously.

Figure 13. Bus Request/Grant with $\overline{\text{RESET}}$ Low

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

	ADSP-2100 Program Memory Read	Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
	Switching Characteristics									
31	$\overline{PMRD}$ Width Low	A	60		45		60		ns	4
32	PMA Valid to $\overline{PMRD}$ Low	A	18		11		18		ns	3
33	$\overline{PMRD}$ High to PMA Invalid	A	20		16		20		ns	1
34	PMDA Valid to $\overline{PMRD}$ Low	A	41		31		41		ns	3
35	$\overline{PMRD}$ High to PMDA Invalid	A	23		18		22		ns	1
36	$\overline{PMS}$ Valid to $\overline{PMRD}$ Low	A	55		40		55		ns	3
37	$\overline{PMRD}$ High to $\overline{PMS}$ Invalid	A	16		12		16		ns	1
	Timing Requirements									
58	$\overline{PMRD}$ Low to PMD Input Valid	A		45		37		45	ns	4
59	PMA Valid to PMD Input Valid	A		57		50		57	ns	7
60	$\overline{PMS}$ Valid to PMD Input Valid	A		90		65		90	ns	7
97	$\overline{PMRD}$ High to PMD Input Invalid	A	0		0		0		ns	

	ADSP-2100A Program Memory Read	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
	Switching Characteristics													
31	$\overline{PMRD}$ Width Low	A	36		28		45		36		28		ns	4
32	PMA Valid to $\overline{PMRD}$ Low	A	6		4		14		6		4		ns	3
33	$\overline{PMRD}$ High to PMA Invalid	A	8		6		10		8		6		ns	1
34	PMDA Valid to $\overline{PMRD}$ Low	A	20		18		24		20		15		ns	3
35	$\overline{PMRD}$ High to PMDA Invalid	A	10		10		12		10		10		ns	1
36	$\overline{PMS}$ Valid to $\overline{PMRD}$ Low	A	32		26		40		32		26		ns	3
37	$\overline{PMRD}$ High to $\overline{PMS}$ Invalid	A	8		6		8		8		6		ns	1
	Timing Requirements													
58	$\overline{PMRD}$ Low to PMD Input Valid	A		28		20		33		28		18	ns	4
59	PMA Valid to PMD Input Valid	A		46		32		50		46		32	ns	7
60	$\overline{PMS}$ Valid to PMD Input Valid	A		50		45		65		50		35	ns	7
97	$\overline{PMRD}$ High to PMD Input Invalid	A	0		0		0		0		0		ns	

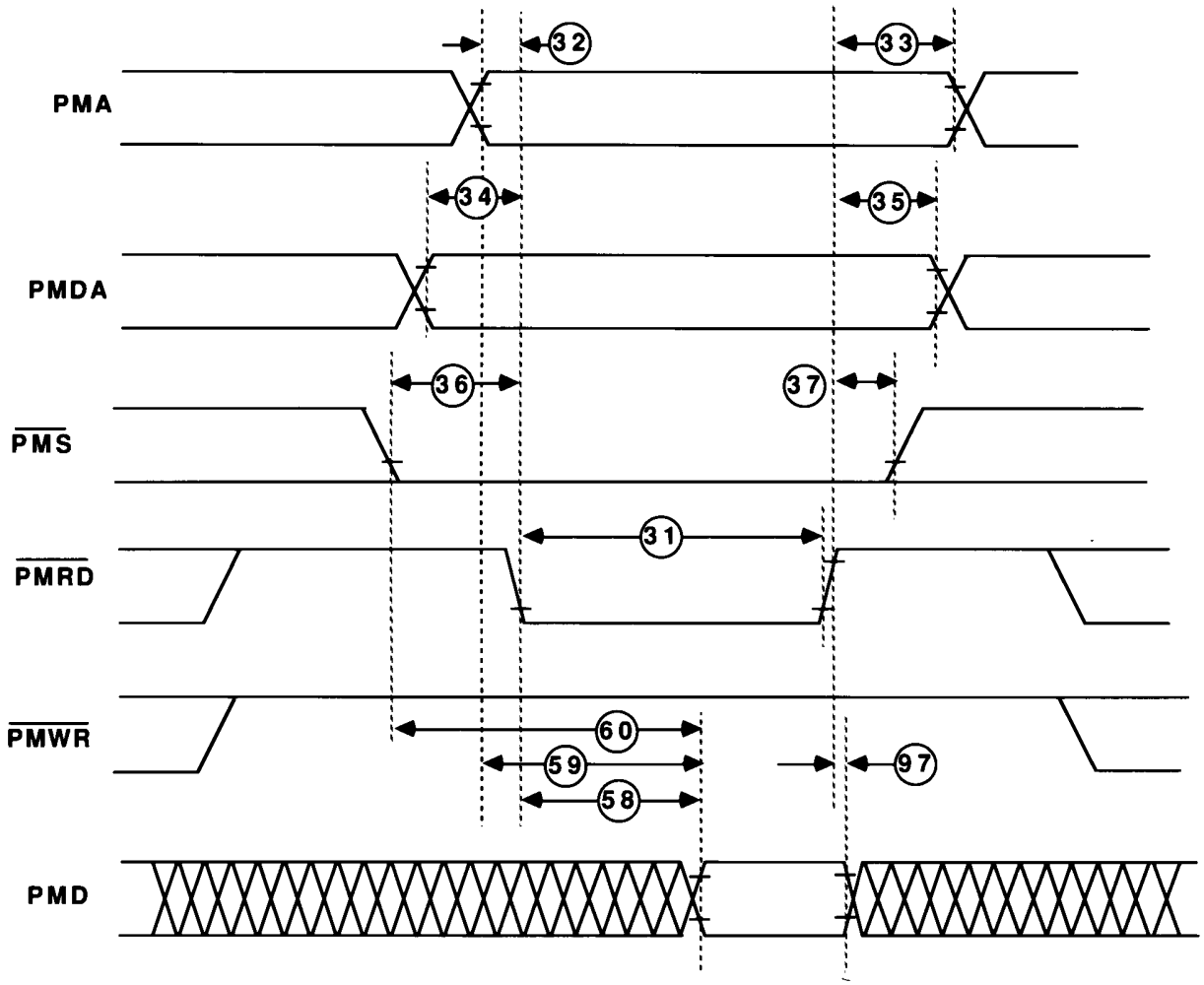


Figure 14. Program Memory Read

Notes 1 and 2 and a table explaining the Derating Factors and Test Codes appear on page 32.

	ADSP-2100 Program Memory Write	Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Switching Characteristics										
40	$\overline{\text{PMWR}}$ Width Low	A	60		45		60		ns	4
41	PMA Valid to $\overline{\text{PMWR}}$ Low	A	16		10		16		ns	3
42	$\overline{\text{PMWR}}$ High to PMA Invalid	A	19		15		19		ns	1
43	PMDA Valid to $\overline{\text{PMWR}}$ Low	A	39		29		39		ns	3
44	$\overline{\text{PMWR}}$ High to PMDA Invalid	A	20		16		21		ns	1
45	$\overline{\text{PMS}}$ Valid to $\overline{\text{PMWR}}$ Low	A	54		40		54		ns	3
46	$\overline{\text{PMWR}}$ High to $\overline{\text{PMS}}$ Invalid	A	15		11		14		ns	1
51	$\overline{\text{PMWR}}$ Low to PMD Out Enable	F	15		10		15		ns	1
52	$\overline{\text{PMWR}}$ High to PMD Out Disable	D		43		37		43	ns	1
53	$\overline{\text{PMWR}}$ Low to PMD Out Valid	A		40		32		40	ns	1
54	$\overline{\text{PMWR}}$ High to PMD Out Invalid	A	23		18		21		ns	1
55	PMD Out Valid to $\overline{\text{PMWR}}$ High	A	33		25		33		ns	3

	ADSP-2100A Program Memory Write	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Switching Characteristics														
40	$\overline{\text{PMWR}}$ Width Low	A	36		28		45		36		28		ns	4
41	PMA Valid to $\overline{\text{PMWR}}$ Low	A	8		4		12		8		4		ns	3
42	$\overline{\text{PMWR}}$ High to PMA Invalid	A	8		6		10		8		6		ns	1
43	PMDA Valid to $\overline{\text{PMWR}}$ Low	A	20		16		28		20		16		ns	3
44	$\overline{\text{PMWR}}$ High to PMDA Invalid	A	10		8		12		10		8		ns	1
45	$\overline{\text{PMS}}$ Valid to $\overline{\text{PMWR}}$ Low	A	32		26		40		32		26		ns	3
46	$\overline{\text{PMWR}}$ High to $\overline{\text{PMS}}$ Invalid	A	6		4		8		6		4		ns	1
51	$\overline{\text{PMWR}}$ Low to PMD Out Enable	F	8		6		8		8		6		ns	1
52	$\overline{\text{PMWR}}$ High to PMD Out Disable	D		32		29		38		32		29	ns	1
53	$\overline{\text{PMWR}}$ Low to PMD Out Valid	A		29		26		32		29		26	ns	1
54	$\overline{\text{PMWR}}$ High to PMD Out Invalid	A	10		8		12		10		8		ns	1
55	PMD Out Valid to $\overline{\text{PMWR}}$ High	A	16		13		25		16		13		ns	3

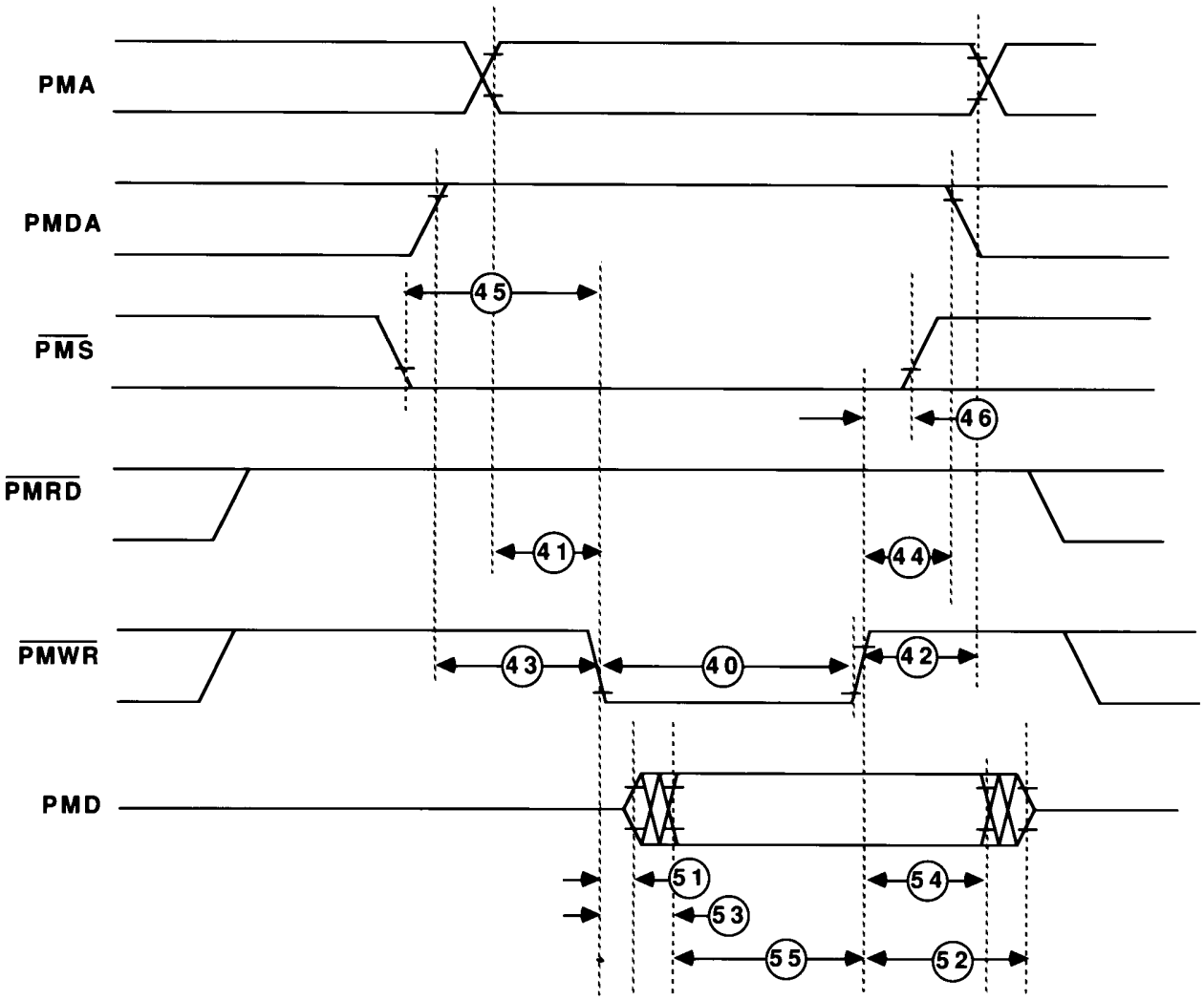


Figure 15. Program Memory Write

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

ADSP-2100 Data Memory Read	Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Switching Characteristics									
67 $\overline{\text{DMRD}}$ Width Low	A	60		45		60		ns	4
68 DMA Valid to $\overline{\text{DMRD}}$ Low	A	21		16		21		ns	3
69 $\overline{\text{DMRD}}$ High to DMA Invalid	A	19		15		19		ns	1
70 $\overline{\text{DMS}}$ Valid to $\overline{\text{DMRD}}$ Low	A	35		27		35		ns	3
71 $\overline{\text{DMRD}}$ High to $\overline{\text{DMS}}$ Invalid	A	22		18		21		ns	1
Timing Requirements									
74 $\overline{\text{DMRD}}$ Low to DMACK Valid	A	0	31	0	21	0	31	ns	3
75 DMA Valid to DMACK Valid	A	0	57	0	42	0	57	ns	6
94 $\overline{\text{DMRD}}$ Low to DMD Input Valid	A		57		41		55	ns	4
95 DMA Valid to DMD Input Valid	A		82		61		79	ns	7
96 $\overline{\text{DMS}}$ Valid to DMD Input Valid	A		96		70		96	ns	7
98 $\overline{\text{DMRD}}$ High to DMD Input Invalid	A	0		0		0		ns	
103 CLKOUT High to DMACK Invalid	A	0	60	0	45	0	60	ns	4

ADSP-2100A Data Memory Read	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Switching Characteristics													
67 $\overline{\text{DMRD}}$ Width Low	A	36		28		45		36		28		ns	4
68 DMA Valid to $\overline{\text{DMRD}}$ Low	A	6		4		14		6		4		ns	3
69 $\overline{\text{DMRD}}$ High to DMA Invalid	A	8		6		10		8		6		ns	1
70 $\overline{\text{DMS}}$ Valid to $\overline{\text{DMRD}}$ Low	A	18		14		27		18		14		ns	3
71 $\overline{\text{DMRD}}$ High to $\overline{\text{DMS}}$ Invalid	A	8		6		10		8		6		ns	1
Timing Requirements													
74 $\overline{\text{DMRD}}$ Low to DMACK Valid	A	0	16	0	10	0	21	0	16	0	10	ns	3
75 DMA Valid to DMACK Valid	A	0	30	0	20	0	42	0	30	0	20	ns	6
94 $\overline{\text{DMRD}}$ Low to DMD Input Valid	A		30		20		37		28		18	ns	4
95 DMA Valid to DMD Input Valid	A		48		32		59		46		32	ns	7
96 $\overline{\text{DMS}}$ Valid to DMD Input Valid	A		52		45		67		50		35	ns	7
98 $\overline{\text{DMRD}}$ High to DMD Input Invalid	A	0		0		0		0		0		ns	
103 CLKOUT High to DMACK Invalid	A	0	36	0	28	0	45	0	36	0	28	ns	4

NOTE ON GENERATING WAIT STATES

See the application note "Wait State Generation on the ADSP-2100/2100A" for information on using DMACK to generate wait states.

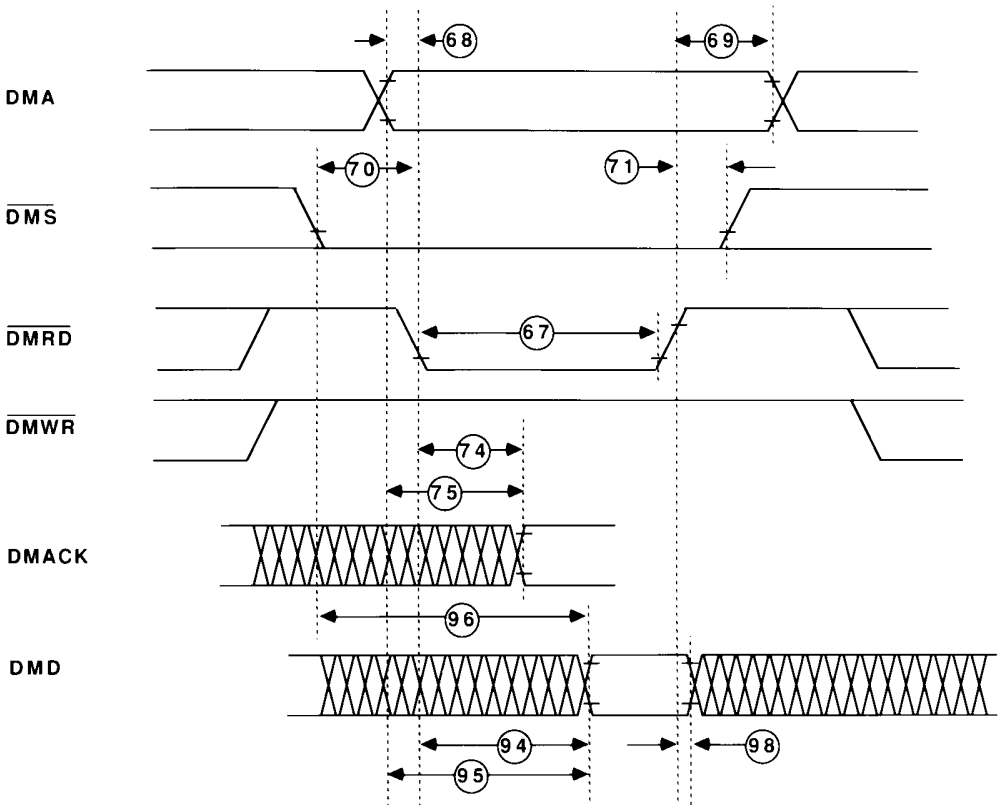


Figure 16a. Data Memory Read

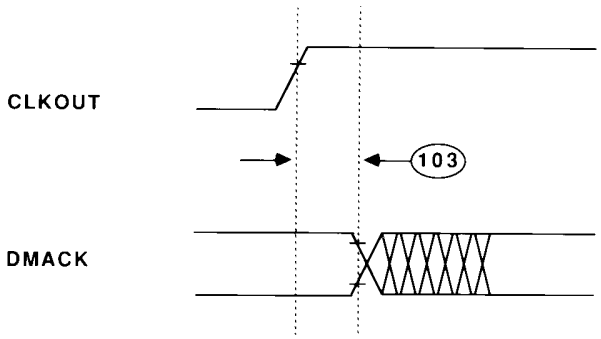


Figure 16b. Data Memory Wait States Extended with DMACK

Notes 1 and 2 and information about the Derating Factors and Test Codes appear on page 32.

ADSP-2100 Data Memory Write	Test Code	J Grade Min	J Grade Max	K Grade Min	K Grade Max	S Grade Min	S Grade Max	Units	Derating Factor
Switching Characteristics									
78 $\overline{\text{DMWR}}$ Width Low	A	60		45		60		ns	4
79 DMA Valid to $\overline{\text{DMWR}}$ Low	A	24		17		24		ns	3
80 $\overline{\text{DMWR}}$ High to DMA Invalid	A	20		15		19		ns	1
81 $\overline{\text{DMS}}$ Valid to $\overline{\text{DMWR}}$ Low	A	37		28		37		ns	3
82 $\overline{\text{DMWR}}$ High to $\overline{\text{DMS}}$ Invalid	A	22		19		22		ns	1
87 $\overline{\text{DMWR}}$ Low to DMD Out Enable	F	14		9		14		ns	1
88 $\overline{\text{DMWR}}$ High to DMD Out Disable	D		40		35		40	ns	1
89 $\overline{\text{DMWR}}$ Low to DMD Out Valid	A		38		32		38	ns	1
90 $\overline{\text{DMWR}}$ High to DMD Out Invalid	A	21		16		19		ns	1
91 DMD Out Valid to $\overline{\text{DMWR}}$ High	A	33		21		33		ns	3
Timing Requirements									
75 DMA Valid to DMACK Valid	A	0	57	0	42	0	57	ns	6
99 $\overline{\text{DMWR}}$ Low to DMACK Valid	A	0	31	0	21	0	31	ns	3
103 CLKOUT High to DMACK Invalid	A	0	60	0	45	0	60	ns	4

ADSP-2100A Data Memory Write	Test Code	AJ Grade Min	AJ Grade Max	AK Grade Min	AK Grade Max	AS Grade Min	AS Grade Max	AT Grade Min	AT Grade Max	AU Grade Min	AU Grade Max	Units	Derating Factor
Switching Characteristics													
78 $\overline{\text{DMWR}}$ Width Low	A	36		28		45		36		28		ns	4
79 DMA Valid to $\overline{\text{DMWR}}$ Low	A	8		4		17		8		4		ns	3
80 $\overline{\text{DMWR}}$ High to DMA Invalid	A	8		6		10		8		6		ns	1
81 $\overline{\text{DMS}}$ Valid to $\overline{\text{DMWR}}$ Low	A	20		16		28		20		16		ns	3
82 $\overline{\text{DMWR}}$ High to $\overline{\text{DMS}}$ Invalid	A	6		4		8		6		4		ns	1
87 $\overline{\text{DMWR}}$ Low to DMD Out Enable	F	8		6		8		8		6		ns	1
88 $\overline{\text{DMWR}}$ High to DMD Out Disable	D		32		29		38		32		29	ns	1
89 $\overline{\text{DMWR}}$ Low to DMD Out Valid	A		29		26		32		29		26	ns	1
90 $\overline{\text{DMWR}}$ High to DMD Out Invalid	A	10		8		12		10		8		ns	1
91 DMD Out Valid to $\overline{\text{DMWR}}$ High	A	18		13		25		16		13		ns	3
Timing Requirements													
75 DMA Valid to DMACK Valid	A	0	30	0	20	0	42	0	30	0	20	ns	6
99 $\overline{\text{DMWR}}$ Low to DMACK Valid	A	0	16	0	10	0	20	0	16	0	10	ns	3
103 CLKOUT High to DMACK Invalid	A	0	36	0	28	0	45	0	36	0	28	ns	4

NOTE ON GENERATING WAIT STATES

See the application note "Wait State Generation on the ADSP-2100/2100A" for information on using DMACK to generate wait states.

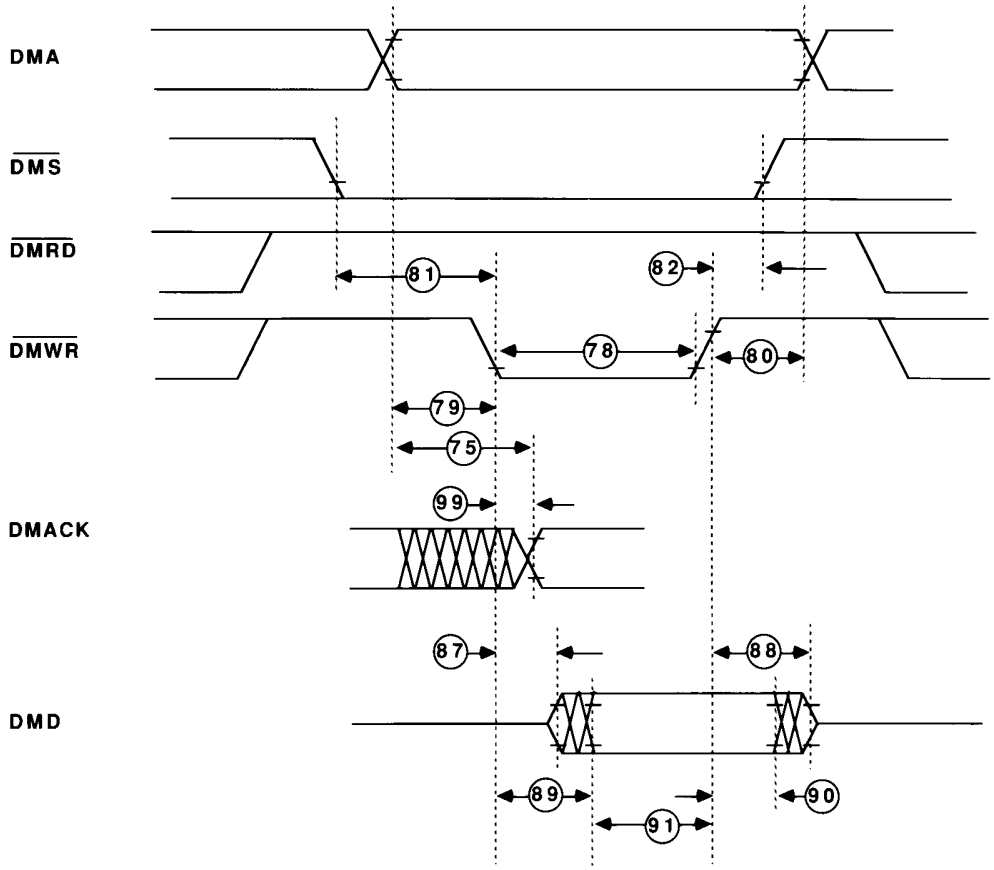


Figure 17a. Data Memory Write

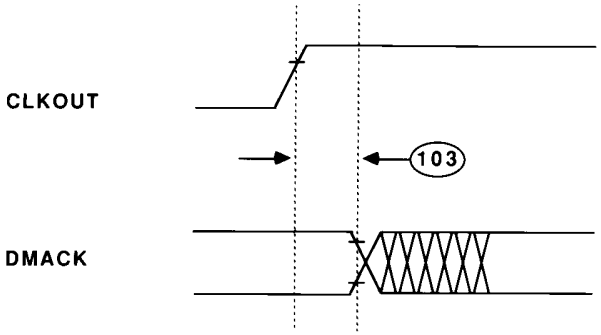


Figure 17b. Data Memory Wait States Extended with DMACK

NOTES

[1]Rise and fall times ≤4ns for ADSP-2100A, 5ns for ADSP-2100.

[2]"xMxx" refers to PMA_{0-13}, $\overline{PMS}$, $\overline{PMRD}$, $\overline{PMWR}$, PMDA, DMA_{0-13}, $\overline{DMS}$, $\overline{DMRD}$ and $\overline{DMWR}$.

TEST CODES

Code	Test Type	Level Reference
A	Inputs, Outputs	Low = 0.8V, High = 2.0V
B	CLKIN	1.5V
	to/from	
	Inputs, Outputs	Low = 0.8V, High = 2.0V
D	Output	Low = 0.8V, High = 2.0V
	to	
	Output Disable	Low = V_{OL} + 0.5V, High = V_{OH} − 0.5V
F	Output	Low = 0.8V, High = 2.0V
	to/from	
	Output Enable	Low = VT − 0.1V, High = VT + 0.1V

VT = 1.5V, the voltage to which tristated outputs are forced.

DERATING FACTOR

The value **N** in the Derating Column shows, for each timing parameter affected, how many of the eight internal clock states are used by this timing parameter; **N**, therefore, ranges between 1 and 8. The formula for changing any individual parameter T uses timing parameter number one, CLKIN Period, shown as P#1:

$$T_{new} = T_{old} + N\ ((P\#1_{new} - P\#1_{old})\ /2)$$

You determine the new value of P#1 based on the derating you wish to accomplish. If no **N** value is given for derating, that timing parameter does not change with clock changes.

CAPACITANCE IN PGA PACKAGE

Input capacitance	C_{IN}	10pF typical
Output capacitance	C_{OUT}	10pF typical

Note that output-only pads (PMA_{13-0}, PMDA and DMA_{13-0}) and bidirectional pads (PMD_{23-0} and DMD_{15-0}) have 50kΩ (typical) pull-up resistors between the output and V_{DD} present when the output driver is off.

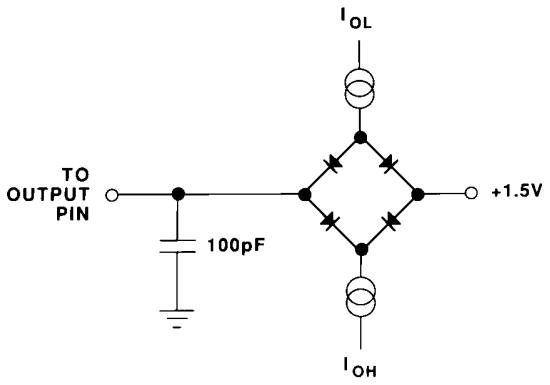


Figure 18. Normal Load for ac Measurements

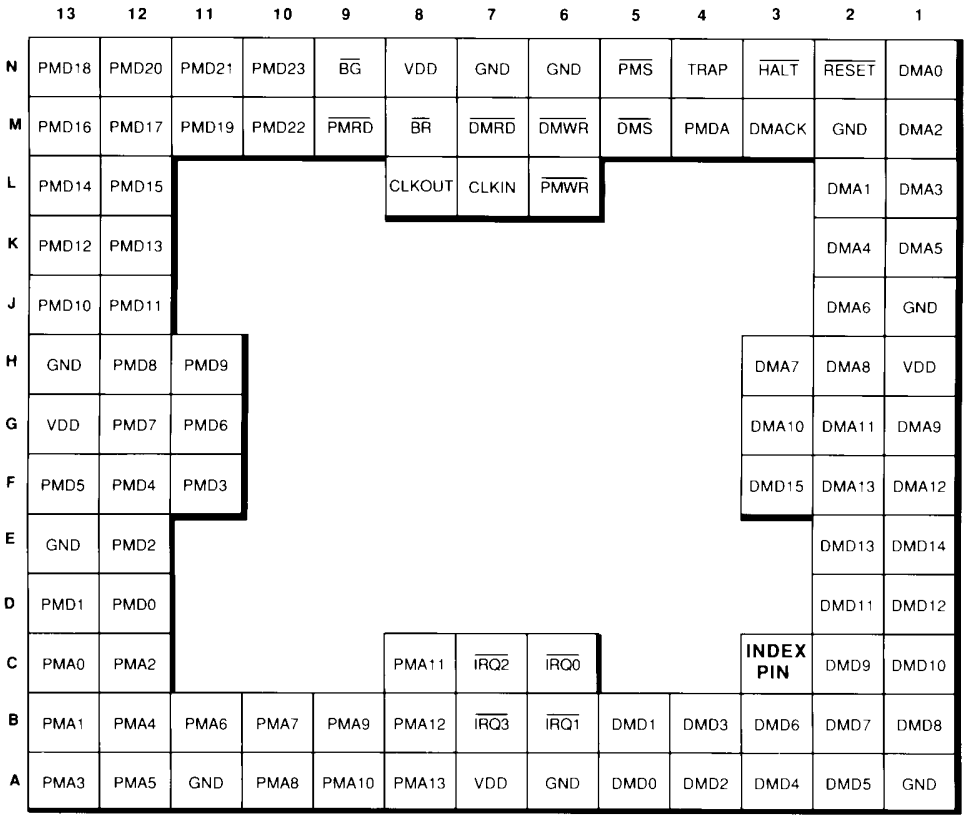


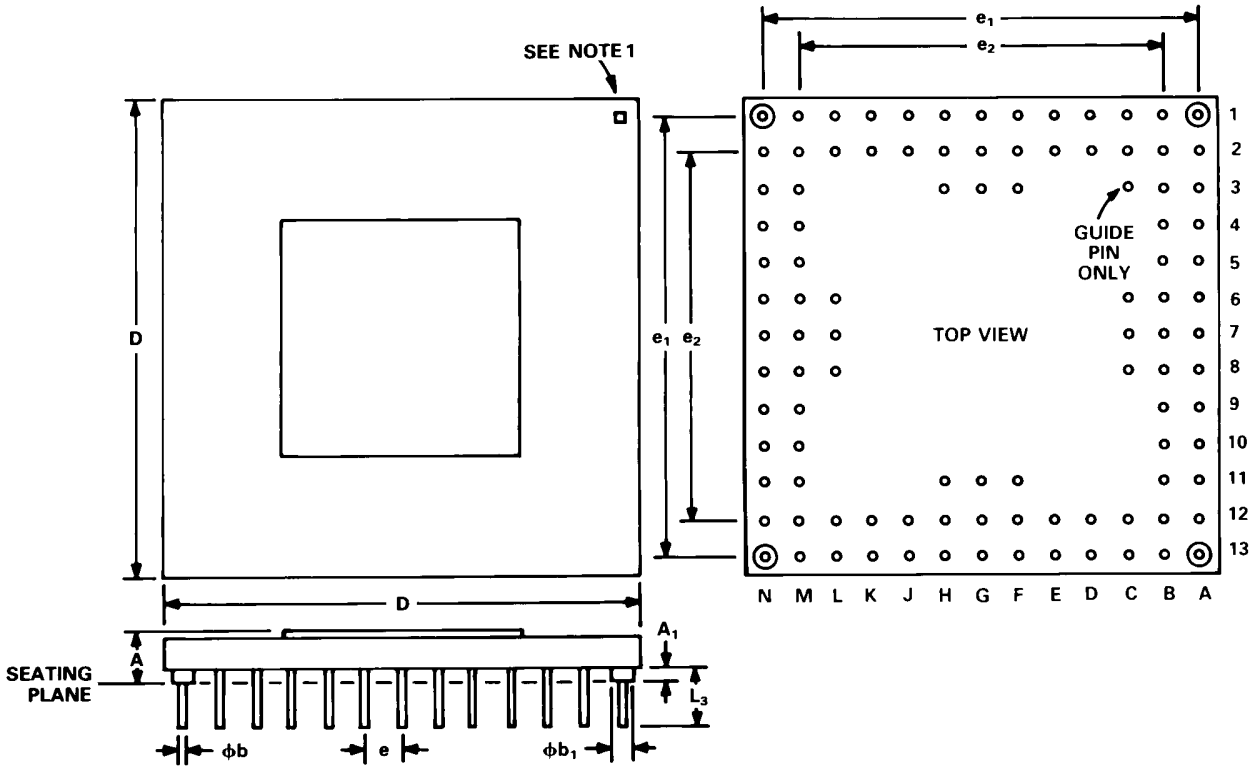
Figure 19. ADSP-2100 Pins, Top View, Pins Down

Function	Location	Function	Location	Function	Location	Function	Location
V_{DD}	A7	PMA1	B13	PMD12	K13	DMA9	G1
V_{DD}	G13	PMA2	C12	PMD13	K12	DMA10	G3
V_{DD}	H1	PMA3	A13	PMD14	L13	DMA11	G2
V_{DD}	N8	PMA4	B12	PMD15	L12	DMA12	F1
GND	A1	PMA5	A12	PMD16	M13	DMA13	F2
GND	A6	PMA6	B11	PMD17	M12	DMD0	A5
GND	A11	PMA7	B10	PMD18	N13	DMD1	B5
GND	E13	PMA8	A10	PMD19	M11	DMD2	A4
GND	H13	PMA9	B9	PMD20	N12	DMD3	B4
GND	J1	PMA10	A9	PMD21	N11	DMD4	A3
GND	M2	PMA11	C8	PMD22	M10	DMD5	A2
GND	N6	PMA12	B8	PMD23	N10	DMD6	B3
GND	N7	PMA13	A8	$\overline{PMS}$	N5	DMD7	B2
CLKIN	L7	PMD0	D12	$\overline{PMWR}$	L6	DMD8	B1
CLKOUT	L8	PMD1	D13	$\overline{PMRD}$	M9	DMD9	C2
$\overline{BR}$	M8	PMD2	E12	PMDA	M4	DMD10	C1
$\overline{BG}$	N9	PMD3	F11	DMA0	N1	DMD11	D2
$\overline{IRQ0}$	C6	PMD4	F12	DMA1	L2	DMD12	D1
$\overline{IRQ1}$	B6	PMD5	F13	DMA2	M1	DMD13	E2
$\overline{IRQ2}$	C7	PMD6	G11	DMA3	L1	DMD14	E1
$\overline{IRQ3}$	B7	PMD7	G12	DMA4	K2	DMD15	F3
$\overline{RESET}$	N2	PMD8	H12	DMA5	K1	$\overline{DMS}$	M5
TRAP	N4	PMD9	H11	DMA6	J2	$\overline{DMWR}$	M6
$\overline{HALT}$	N3	PMD10	J13	DMA7	H3	$\overline{DMRD}$	M7
INDEX PIN	NC	PMD11	J12	DMA8	H2	DMACK	M3
PMA0	C13						

Table V. ADSP-2100 Pins by Function – G-100A

ADSP-2100 MECHANICAL INFORMATION
100-PIN GRID ARRAY

Dimensions shown in inches and (mm).



SYMBOL	INCHES		MILLIMETERS		NOTES
	MIN	MAX	MIN	MAX	
A		0.169		4.29	3
A_1	0.025	0.055	0.64	1.40	3
ϕb	0.016	0.020	0.41	0.51	8
ϕb_1	0.040	0.055	1.02	1.40	2,8
D	1.308	1.332	33.22	33.83	4,9
e_1	1.188	1.212	30.18	30.78	7
e_2	0.988	1.024	25.10	26.01	7
e	0.095	0.105	2.41	2.67	5
L_3	0.165	0.190	4.19	4.83	

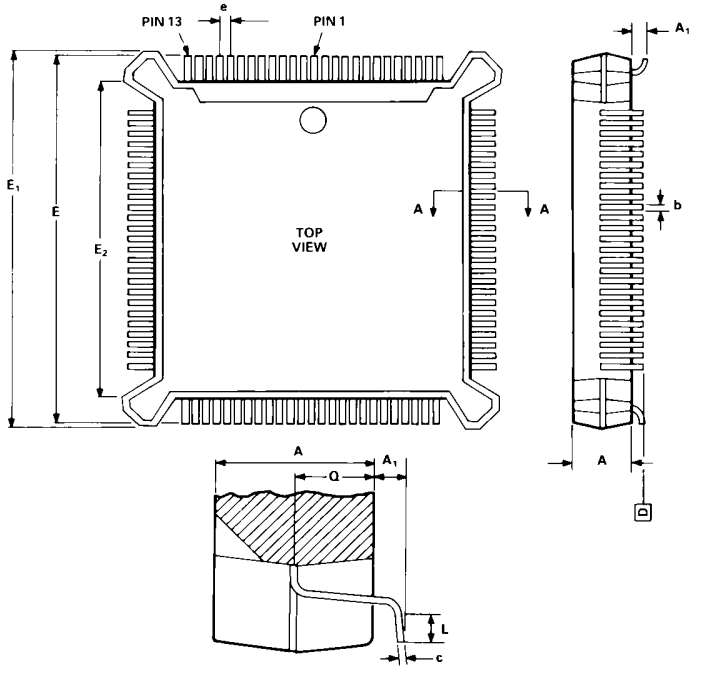
NOTES

1. Index area; a notch or a lead one identification mark is located adjacent to lead one.
2. The minimum limit for dimension ϕb_1 may be 0.023" (0.58mm) for all four corner leads only.
3. Dimension shall be measured from the seating plane to the base plane.
4. This dimension allows for off-center lid, meniscus and glass overrun.
5. The basic pin spacing is 0.100" (2.54mm) between centerlines.
6. Applies to all four corners.
7. Lead center when α is 0°; e_1 shall be measured at the centerline of the leads.
8. All leads – increase maximum limit by 0.003" (0.08mm) measured at the center of the flat, when hot solder dip lead finish is applied.
9. All four sides.
10. Gold plating 50μ inches over 100μ inches ref. Thickness of nickel.

PIN	FUNCTION	PIN	FUNCTION	PIN	FUNCTION	PIN	FUNCTION	PIN	FUNCTION
1	PMD6	21	PMA10	41	DMD9	61	DMA2	81	$\overline{BG}$
2	V_{DD}	22	PMA11	42	DMD10	62	DMA1	82	$\overline{PMRD}$
3	PMD5	23	PMA12	43	DMD11	63	DMA0	83	PMD23
4	PMD4	24	PMA13	44	DMD12	64	GND	84	PMD22
5	PMD3	25	$\overline{IRQ3}$	45	DMD13	65	$\overline{RESET}$	85	PMD21
6	GND	26	$\overline{IRQ2}$	46	DMD14	66	DMACK	86	PMD20
7	PMD2	27	V_{DD}	47	DMD15	67	$\overline{HALT}$	87	PMD19
8	PMD1	28	GND	48	DMA13	68	PMDA	88	PMD18
9	PMD0	29	$\overline{IRQ1}$	49	DMA12	69	TRAP	89	PMD17
10	PMA0	30	$\overline{IRQ0}$	50	DMA11	70	$\overline{DMS}$	90	PMD16
11	PMA1	31	DMD0	51	DMA10	71	$\overline{PMS}$	91	PMD15
12	PMA2	32	DMD1	52	DMA9	72	$\overline{PMWR}$	92	PMD14
13	PMA3	33	DMD2	53	V_{DD}	73	$\overline{DMWR}$	93	PMD13
14	PMA4	34	DMD3	54	DMA8	74	GND	94	PMD12
15	PMA5	35	DMD4	55	DMA7	75	$\overline{DMRD}$	95	PMD11
16	PMA6	36	DMD5	56	GND	76	CLKIN	96	PMD10
17	GND	37	DMD6	57	DMA6	77	GND	97	PMD9
18	PMA7	38	GND	58	DMA5	78	V_{DD}	98	PMD8
19	PMA8	39	DMD7	59	DMA4	79	$\overline{BR}$	99	GND
20	PMA9	40	DMD8	60	DMA3	80	CLKOUT	100	PMD7

Table VI. ADSP-2100 Pins by Function – P-100 and F-100A

P-100
Plastic Quad Flat Pack
(JEDEC Style)



SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.160	0.180	4.06	4.57
A_1	0.020	0.040	0.51	1.02
b	0.010	0.013	0.25	0.33
c	0.006	0.008	0.15	0.20
E	0.875	0.885	22.23	22.48
E_1	0.897	0.903	22.78	22.94
E_2	0.747	0.753	18.97	19.13
e	0.020	0.030	0.51	0.76
L	0.020	0.030	0.51	0.76
Q	0.065	0.075	1.65	1.91
⌭		0.008		0.20

C1235c-8-5/93

PRINTED IN U.S.A.