ANALOG DEVICES

32-Bit IEEE Floating-Point DSP Microprocessor

ADSP-21010

FEATURES

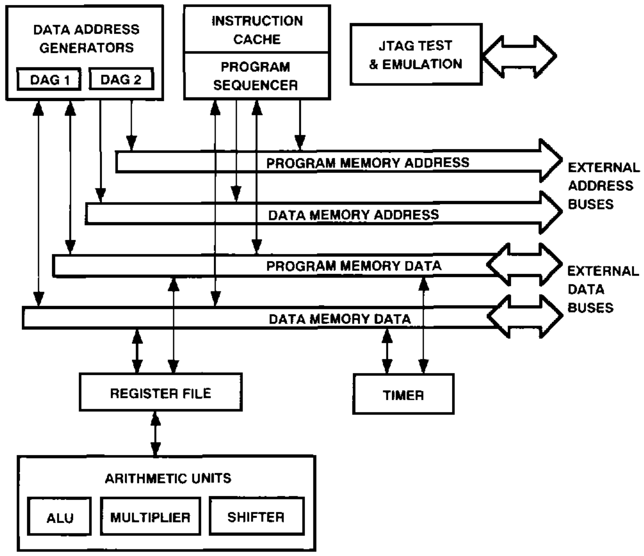
Superscalar IEEE Floating-Point Processor
Off-Chip Harvard Architecture Maximizes Signal Processing Performance
80 ns, 12.5 MIPS Instruction Rate, Single-Cycle Execution
37.5 MFLOPS Peak, 25 MFLOPS Sustained Performance
1024-Point Complex FFT Benchmark: 1.54 ms
Divide (y/x): 480 ns
Inverse Square Root ($1/\sqrt{x}$): 720 ns
32-Bit Single-Precision IEEE Floating-Point Data Format
32-Bit Fixed-Point Format, Integer and Fractional, with 80-Bit Accumulators
IEEE Exception Handling with Interrupt on Exception
Three Independent Computation Units: Multiplier, ALU, and Barrel Shifter
Dual Data Address Generators with Indirect, Immediate, Modulo, and Bit Reverse Addressing Modes
Two Off-Chip Memory Transfers in Parallel with Instruction Fetch and Single-Cycle Multiply & ALU Operations
Multiply with Add & Subtract for FFT Butterfly Computation
Efficient Program Sequencing with Zero-Overhead Looping: Single-Cycle Loop Setup & Exit
Single-Cycle Register File Context Switch
55 ns External RAM Access Time for Zero-Wait-State, 80 ns Instruction Execution
IEEE JTAG Standard 1149.1 Test Access Port and On-Chip Emulation Circuitry
304-Lead PQFP Surface-Mount Package

GENERAL DESCRIPTION

The ADSP-21010 is the low-cost member of Analog Devices' family of single-chip IEEE floating-point processors optimized for digital signal processing applications. It is a 12.5 MIPS, 32-bit-only version of the ADSP-21020 IEEE Floating-Point DSP Microprocessor. The ADSP-21010 is source- and object-code compatible with the ADSP-21020 for all instructions using 32-bit data operands, but does not support the 40-bit data mode of the ADSP-21020.

Fabricated in a high-speed, low-power CMOS process, the ADSP-21010 has an 80 ns instruction cycle time. With a high-performance on-chip instruction cache, the ADSP-21010 can execute every instruction in a single cycle.

FUNCTIONAL BLOCK DIAGRAM



The ADSP-21010 features:

- **Independent Parallel Computation Units**
  The arithmetic/logic unit (ALU), multiplier and shifter perform single-cycle instructions. The units are architecturally arranged in parallel, maximizing computational throughput. A single multifunction instruction executes parallel ALU and multiplier operations. These computation units support IEEE 32-bit single-precision floating-point and 32-bit fixed-point data formats.
- **Data Register File**
  A general-purpose data register file is used for transferring data between the computation units and the data buses, and for storing intermediate results. This 10-port (16-register) register file, combined with the ADSP-21010's Harvard architecture, allows unconstrained data flow between computation units and off-chip memory.
- **Single-Cycle Fetch of Instruction and Two Operands**
  The ADSP-21010 uses a modified Harvard architecture in which data memory stores data and program memory stores both instructions and data. Because of its separate program and data memory buses and on-chip instruction cache, the processor can simultaneously fetch an operand from data memory, an operand from program memory, and an instruction from the cache, all in a single cycle.
- **Memory Interface**
  Addressing of external memory devices by the ADSP-21010 is facilitated by on-chip decoding of high-order address lines to generate memory bank select signals. Separate control lines are also generated for simplified addressing of page-mode DRAM.

REV. A

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 617/329-4700 Fax: 617/326-8703

The ADSP-21010 provides programmable memory wait states, and external memory acknowledge controls allow interfacing to peripheral devices with variable access times.

- **Instruction Cache**
  The ADSP-21010 includes a high performance instruction cache that enables three-bus operation for fetching an instruction and two data values. The cache is selective—only the instructions whose fetches conflict with program memory data accesses are cached. This allows full-speed execution of core, looped operations such as digital filter multiply-accumulates and FFT butterfly processing.
- **Hardware Circular Buffers**
  The ADSP-21010 provides hardware to implement circular buffers in memory, which are common in digital filters and Fourier transform implementations. It handles address pointer wraparound, reducing overhead (thereby increasing performance) and simplifying implementation. Circular buffers can start and end at any location.
- **Flexible Instruction Set**
  The ADSP-21010's 48-bit instruction word accommodates a variety of parallel operations, for concise programming. For example, the ADSP-21010 can conditionally execute a multiply, an add, a subtract and a branch in a single instruction.

DEVELOPMENT SYSTEM

The ADSP-21010 is supported with a complete set of software and hardware development tools. The ADSP-21000 Family Development Software supports all processors in the ADSP-21000 Family, and the ADSP-21020 EZ-LAB and ADSP-21020 EZ-ICE are used to develop ADSP-21010 systems.

- **Assembler**
  Creates relocatable, COFF (Common Object File Format) object files from ADSP-21xxx assembly source code. It accepts standard C preprocessor directives for conditional assembly and macro processing. The algebraic syntax of the ADSP-21xxx assembly language facilitates coding and debugging of DSP algorithms.
- **Linker/Librarian**
  The Linker processes separately assembled object files and library files to create a single executable program. It assigns memory locations to code and to data in accordance with a user-defined architecture file that describes the memory and I/O configuration of the target system. The Librarian allows you to group frequently used object files into a single library file that can be linked with your main program.
- **ADSP-21020/ADSP-21010 Simulator**
  The Simulator performs interactive, instruction-level simulation of ADSP-21xxx code within the hardware configuration described by a system architecture file. It flags illegal operations and supports full symbolic disassembly. It provides an easy-to-use, window-based, graphical user interface that is identical to the one used by the EZ-ICE Emulator. Commands are accessed from pull-down menus with a mouse.
- **PROM Splitter**
  Formats an executable file into files that can be used with an industry-standard PROM programmer.
- **C Compiler and Runtime Library**
  The C Compiler complies with ANSI specifications and has been validated to the widely used Plum-Hall™ Validation Suite as well as the Perennial Validation Suite™. It takes advantage of the ADSP-21010's high-level language architectural features and incorporates optimizing algorithms to speed up the execution of code. It includes an extensive runtime library with over 100 standard and DSP-specific functions.
- **C Source Level Debugger**
  A full-featured C source level debugger that works with the simulator or EZ-ICE emulator to allow debugging of assembler source, C source, or mixed assembler and C.
- **DSP/C™ Compiler** (Available 1st half 1993)
  Supports ANSI Standard (X3J11.1) Numerical C as defined by the Numeric C Extensions Group. The DSP/C™ Compiler accepts C source input containing Numerical C extensions for array selection, vector math operations, complex data types, circular pointers, and variably dimensioned arrays, and outputs ADSP-21xxx assembly language source code.
- **ADSP-21020/ADSP-21010 EZ-LAB® Evaluation Board**
  The EZ-LAB Evaluation Board is a general-purpose, standalone ADSP-21020 system that includes 32K words of program memory and 32K words of data memory as well as analog I/O. A PC RS-232 download path enables the user to download and run programs directly on the EZ-LAB. In addition, it may be used in conjunction with the EZ-ICE Emulator to provide a powerful software debug environment.
- **ADSP-21020/ADSP-21010 EZ-ICE® Emulator**
  This in-circuit emulator provides the system designer with a PC-based development environment that allows nonintrusive access to the ADSP-21010's internal registers through the processor's 5-pin JTAG Test Access Port. This use of on-chip emulation circuitry enables reliable, full-speed performance in any target. The emulator has the same graphical user interface as the simulator, allowing an easy transition from software to hardware debug. (See "Target System Requirements For Use of EZ-ICE Emulator" on page 23.)

ADDITIONAL INFORMATION

This data sheet provides a general overview of ADSP-21010 functionality. For additional information on the architecture and instruction set of the processor, refer to the *ADSP-21020 User's Manual*. For development system and programming reference information, refer to the *ADSP-21000 Family Development Software Manuals* and the *ADSP-21020 Programmer's Quick Reference*. Applications code listings and benchmarks for key DSP algorithms are available on the DSP Applications BBS; call (617) 461-4258, 8 data bits, no parity, 1 stop bit, 300/1200/2400/9600 baud.

ARCHITECTURE OVERVIEW

Figure 1 shows a block diagram of the ADSP-21010. The processor features:

- Three Computation Units (ALU, Multiplier, and Shifter) with a Shared Data Register File
- Two Data Address Generators (DAG 1, DAG 2)
- Program Sequencer with Instruction Cache
- 32-Bit Timer
- Memory Buses and Interface
- JTAG Test Access Port and On-Chip Emulation Support

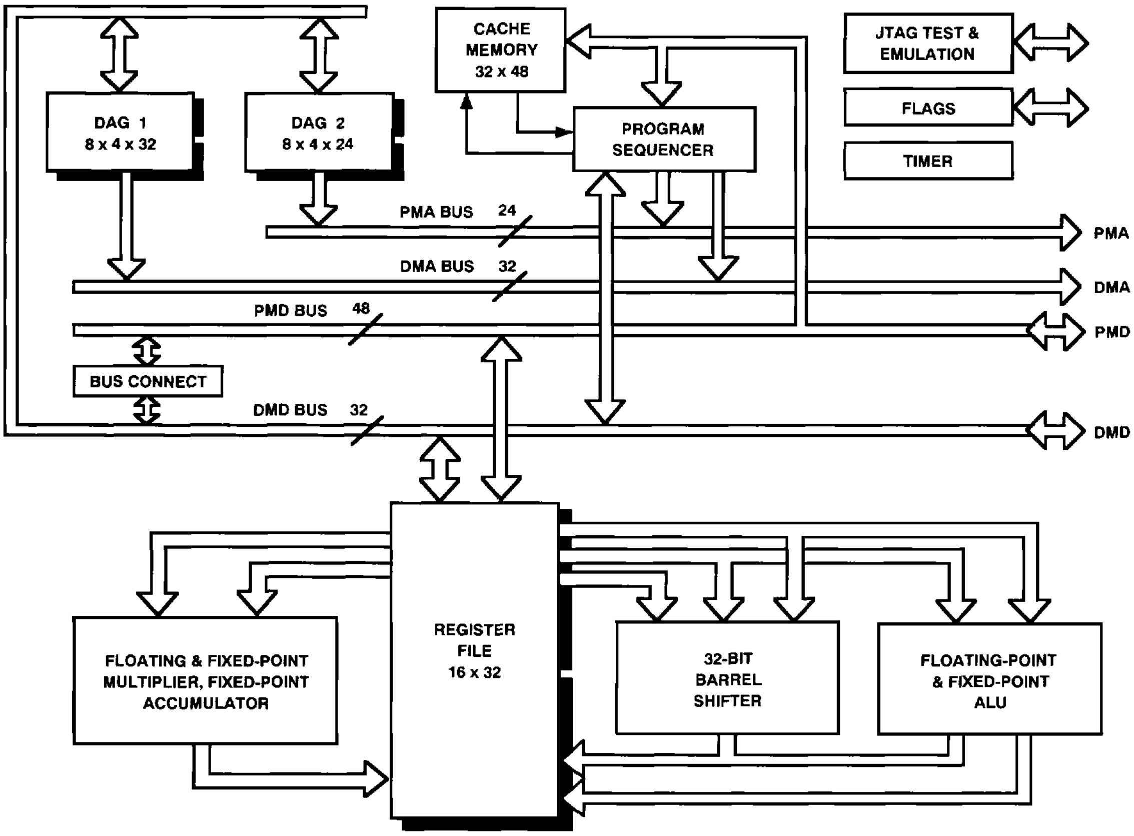


Figure 1. ADSP-21010 Block Diagram

Computation Units

The ADSP-21010 contains three independent computation units: an ALU, a multiplier with fixed-point accumulator, and a shifter. The computation units process data in two formats: 32-bit fixed-point and 32-bit floating-point. The floating-point operations are single-precision IEEE-compatible (IEEE Standard 754/854). The ADSP-21010 must be operated in 32-bit single-precision mode: the RND32 bit of the MODE1 register must be set to 1 at power up.

The multiplier performs floating-point and fixed-point multiplication as well as fixed-point multiply/add and multiply/subtract operations. Integer products are 64 bits wide, and the accumulator is 80 bits wide. The ALU performs 45 standard arithmetic and logic operations, supporting both fixed-point and floating-point formats. The shifter performs 19 different operations on 32-bit operands. These operations include logical and arithmetic shifts, bit manipulation, field deposit, and extract and derive exponent operations.

The computation units perform single-cycle operations; there is *no* computation pipeline. The three units are connected in parallel rather than serially, via multiple-bus connections with the 10-port data register file. The output of any computation unit may be used as the input of any unit on the next cycle. In a *multifunction* computation, the ALU and multiplier perform independent, simultaneous operations.

Data Register File

The ADSP-21010's general-purpose data register file is used for transferring data between the computation units and the data buses, and for storing intermediate results. The register file has two sets (primary and alternate) of sixteen 32-bit registers each, for fast context switching.

With a large number of buses connecting the registers to the computation units, data flow between computation units and from/to off-chip memory is unconstrained and free from bottlenecks. The 10-port register file and Harvard architecture of the ADSP-21010 allow the following nine data transfers to be performed every cycle:

- Off-chip read/write of two operands to or from the register file
- Two operands supplied to the ALU
- Two operands supplied to the multiplier
- Two results received from the ALU and multiplier (three, if the ALU operation is a combined addition/subtraction)

The processor's 48-bit orthogonal instruction word supports fully parallel data transfer and arithmetic operations in the same instruction.

Address Generators and Program Sequencer

Two dedicated address generators and a program sequencer supply addresses for memory accesses. Because of this, the computation units need never be used to calculate addresses. Because of its instruction cache, the ADSP-21010 can simultaneously fetch an instruction and data values from both off-chip program memory and off-chip data memory in a single cycle.

The data address generators (DAGs) provide memory addresses when external memory data is transferred over the parallel memory ports to or from internal registers. Dual data address generators enable the processor to output two simultaneous addresses for dual operand reads and writes. DAG 1 supplies 32-bit addresses to data memory. DAG 2 supplies 24-bit addresses to program memory for program memory data accesses.

Each DAG keeps track of up to eight address pointers, eight modifiers, eight buffer length values and eight base values. A pointer used for indirect addressing can be modified by a value in a specified register, either before (premodify) or after (post-modify) the access. To implement automatic modulo addressing for circular buffers, the ADSP-21010 provides buffer length registers that can be associated with each pointer. Base values for pointers allow circular buffers to be placed at arbitrary locations. Each DAG register has an alternate register that can be activated for fast context switching.

The program sequencer supplies instruction addresses to program memory. It controls loop iterations and evaluates conditional instructions. To execute looped code with zero overhead, the ADSP-21010 maintains an internal loop counter and loop stack. No explicit jump or decrement instructions are required to maintain the loop.

The ADSP-21010 derives its high clock rate from pipelined *fetch, decode* and *execute* cycles. Approximately 70% of the machine cycle is available for memory accesses; consequently, ADSP-21010 systems can be built using slower and therefore less expensive memory chips.

Instruction Cache

The program sequencer includes a high performance, selective instruction cache that enables three-bus operation for fetching an instruction and two data values. This two-way, set-associative cache holds 32 instructions. The cache is selective—only the instructions whose fetches conflict with program memory data accesses are cached, so the ADSP-21010 can perform a program memory data access and can execute the corresponding instruction in the same cycle. The program sequencer fetches the instruction from the cache instead of from program memory, enabling the ADSP-21010 to simultaneously access data in both program memory and data memory.

Context Switching

Many of the ADSP-21010's registers have alternate register sets that can be activated during interrupt servicing to facilitate a fast context switch. The data registers in the register file, DAG registers and the multiplier result register all have alternate sets. Registers active at reset are called *primary* registers; the others are called *alternate* registers. Bits in the MODE1 control register determine which registers are active at any particular time.

The primary/alternate select bits for each half of the register file (top eight or bottom eight registers) are independent. Likewise, the top four and bottom four register sets in each DAG have independent primary/alternate select bits. This scheme allows passing of data between contexts.

Interrupts

The ADSP-21010 has four external hardware interrupts, nine internally generated interrupts, and eight software interrupts. For the external interrupts and the internal timer interrupt, the ADSP-21010 automatically stacks the arithmetic status and mode (MODE1) registers when servicing the interrupt, allowing five nesting levels of fast service for these interrupts.

An interrupt can occur at any time while the ADSP-21010 is executing a program. Internal events that generate interrupts include arithmetic exceptions, which allow for fast trap handling and recovery.

Timer

The programmable interval timer provides periodic interrupt generation. When enabled, the timer decrements a 32-bit count register every cycle. When this count register reaches zero, the ADSP-21010 generates an interrupt and asserts its TIMEXP output. The count register is automatically reloaded from a 32-bit period register and the count resumes immediately.

System Interface

Figure 2 shows an ADSP-21010 basic system configuration.

The external memory interface supports memory-mapped peripherals and slower memory with a user-defined combination of programmable wait states and hardware acknowledge signals. Both the program memory and data memory interfaces support addressing of page-mode DRAMs.

The ADSP-21010's internal functions are supported by four internal buses: the program memory address (PMA) and data memory address (DMA) buses are used for addresses associated with program and data memory. The program memory data (PMD) and data memory data (DMD) buses are used for data associated with the two memory spaces. These buses are extended off chip. Four data memory select (DMS) signals select one of four user-configurable banks of data memory. Similarly, two program memory select (PMS) signals select between two user-configurable banks of program memory. All banks are independently programmable for 0–7 wait states.

The PX registers permit passing data between program memory and data memory spaces. They provide a bridge between the 48-bit PMD bus and the 32-bit DMD bus or between the 32-bit register file and the PMD bus.

The PMA bus is 24 bits wide allowing direct access of up to 16M words of mixed instruction code and data. The PMD is 48 bits wide to accommodate the 48-bit instruction width. For access of 32-bit data the lower 16 bits are ignored.

The DMA bus is 32 bits wide allowing direct access of up to 4 Gigawords of data. The DMD bus is 32 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 external data memory location in a single cycle. The data memory address comes from one of two sources: an absolute value specified in the instruction code (direct addressing) or the output of a data address generator (indirect addressing).

External devices can gain control of the processor's memory buses from the ADSP-21010 by means of the bus request/grant signals ($\overline{BR}$ and $\overline{BG}$). To grant its buses in response to a bus request, the ADSP-21010 halts internal operations and places its program and data memory interfaces in a high impedance state. In addition, three-state controls ($\overline{DMTS}$ and $\overline{PMTS}$) allow an external device to place either the program or data memory interface in a high impedance state without affecting the other interface and without halting the ADSP-21010 unless it requires a memory access from the affected interface. The three-state controls make it easy for an external cache controller to hold the ADSP-21010 off the bus while it updates an external cache memory.

JTAG Test and Emulation Support

The ADSP-21010 implements the boundary scan testing provisions specified by IEEE Standard 1149.1 of the Joint Testing Action Group (JTAG). The ADSP-21010's test access port and on-chip JTAG circuitry is fully compliant with the IEEE 1149.1 specification. The test access port enables boundary scan testing of circuitry connected to the ADSP-21010's I/O pins.

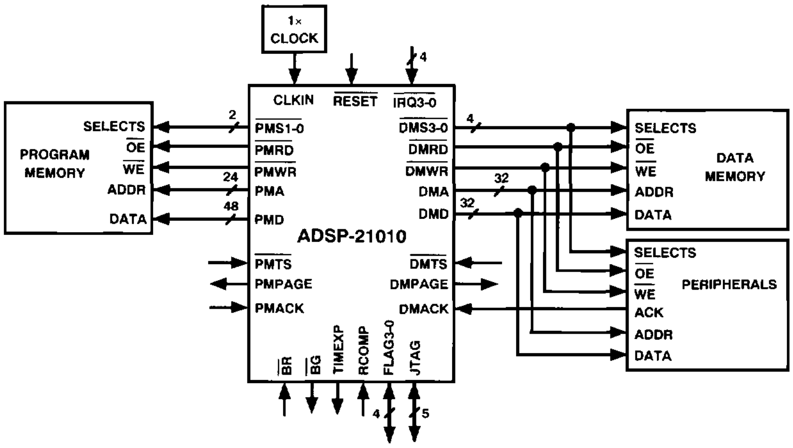


Figure 2. Basic System Configuration

The ADSP-21010 also implements on-chip emulation through the JTAG test access port. The processor's eight sets of breakpoint range registers enable program execution at full speed until reaching a desired breakpoint address range. The processor can then halt and allow reading/writing of all the processor's internal registers and external memories through the JTAG port.

PIN DESCRIPTIONS

This section describes the pins of the ADSP-21010. When groups of pins are identified with subscripts, e.g., PMD_{47-0}, the highest numbered pin is the MSB (in this case, PMD_{47}). Inputs identified as synchronous (S) must meet timing requirements with respect to CLKIN; those that are asynchronous (A) can be asserted asynchronously to CLKIN.

O = Output; I = Input; S = Synchronous; A = Asynchronous;
P = Power Supply; G = Ground.

Pin Name	Type	Function
PMA_{23-0}	O	Program Memory Address. The ADSP-21010 outputs an address in program memory on these pins.
PMD_{47-0}	I/O	Program Memory Data. The ADSP-21010 inputs and outputs data and instructions on these pins. 32-bit fixed-point data and 32-bit single-precision floating-point data is transferred over bits 47–16 of the PMD bus.
$\overline{PMS}_{1-0}$	O	Program Memory Select lines. These pins are asserted as chip selects for the corresponding banks of program memory. Memory banks must be defined in the memory control registers. These pins are decoded program memory address lines and provide an early indication of a possible bus cycle.
$\overline{PMRD}$	O	Program Memory Read strobe. This pin is asserted when the ADSP-21010 reads from program memory.
$\overline{PMWR}$	O	Program Memory Write strobe. This pin is asserted when the ADSP-21010 writes to program memory.
PMACK	I/S	Program Memory Acknowledge. An external device deasserts this input to add wait states to a memory access.
PMPAGE	O	Program Memory Page Boundary. The ADSP-21010 asserts this pin to signal that a program memory page boundary has been crossed. Memory pages must be defined in the memory control registers.
$\overline{PMTS}$	I/S	Program Memory Three-State Control. $\overline{PMTS}$ places the program memory address, data, selects, and strobes in a high-impedance state. If $\overline{PMTS}$ is asserted while a PM access is occurring, the processor will halt and the memory access will not be completed. PMACK must be asserted for at least one cycle when $\overline{PMTS}$ is deasserted to allow any pending memory access to complete properly. $\overline{xTS}$ should only be asserted (low) during an active memory access cycle.
DMA_{31-0}	O	Data Memory Address. The ADSP-21010 outputs an address in data memory on these pins.
DMD_{31-0}	I/O	Data Memory Data. The ADSP-21010 inputs and outputs data on these pins. 32-bit fixed-point data and 32-bit floating-point data is transferred over the DMD bus. (DMD_{31-0} on the ADSP-21010 corresponds to DMD_{39-8} on the ADSP-21020. This should be taken into account if upgrading is planned.)
$\overline{DMS}_{3-0}$	O	Data Memory Select lines. These pins are asserted as chip selects for the corresponding banks of data memory. Memory banks must be defined in the memory control registers. These pins are decoded data memory address lines and provide an early indication of a possible bus cycle.
$\overline{DMRD}$	O	Data Memory Read strobe. This pin is asserted when the ADSP-21010 reads from data memory.
$\overline{DMWR}$	O	Data Memory Write strobe. This pin is asserted when the ADSP-21010 writes to data memory.
DMACK	I/S	Data Memory Acknowledge. An external device deasserts this input to add wait states to a memory access.

Pin Name	Type	Function
DMPAGE	O	Data Memory Page Boundary. The ADSP-21010 asserts this pin to signal that a data memory page boundary has been crossed. Memory pages must be defined in the memory control registers.
$\overline{DMTS}$	I/S	Data Memory Three-State Control. $\overline{DMTS}$ places the data memory address, data, selects, and strobes in a high-impedance state. If $\overline{DMTS}$ is asserted while a DM access is occurring, the processor will halt and the memory access will not be completed. DMACK must be asserted for at least one cycle when $\overline{DMTS}$ is deasserted to allow any pending memory access to complete properly. $\overline{xTS}$ should only be asserted (low) during an active memory access cycle.
CLKIN	I	External clock input to the ADSP-21010. The instruction cycle rate is equal to CLKIN. CLKIN may not be halted, changed, or operated below the specified frequency.
$\overline{RESET}$	I/A	Sets the ADSP-21010 to a known state and begins execution at the program memory location specified by the hardware reset vector (address). This input must be asserted (low) at powerup.
$\overline{IRQ}_{3-0}$	I/A	Interrupt request lines; may be either edge-triggered or level-sensitive.
$FLAG_{3-0}$	I/O/A	External Flags. Each is configured via control bits as either an input or output. As an input, it can be tested as a condition. As an output, it can be used to signal external peripherals.
$\overline{BR}$	I/A	Bus Request. Used by an external device to request control of the memory interface. When $\overline{BR}$ is asserted, the processor halts execution after completion of the current cycle, places all memory data, addresses, selects, and strobes in a high-impedance state, and asserts $\overline{BG}$. The processor continues normal operation when $\overline{BR}$ is released.
$\overline{BG}$	O	Bus Grant. Acknowledges a bus request ($\overline{BR}$), indicating that the external device may take control of the memory interface. $\overline{BG}$ is asserted (held low) until $\overline{BR}$ is released.
TIMEXP	O	Timer Expired. Asserted for four cycles when the value of TCOUNT is decremented to zero.
RCOMP		Compensation Resistor input. Controls compensated output buffers. Connect RCOMP through a 1.8 kΩ ± 15% resistor to EVDD. Use of a capacitor (approximately 100 pF), placed in parallel with the 1.8 kΩ resistor, is recommended.
EVDD	P	Power supply (for output drivers), nominally +5 VDC (20 pins).
EGND	G	Power supply return (for output drivers); (21 pins).
IVDD	P	Power supply (for internal circuitry), nominally +5 VDC (8 pins).
IGND	G	Power supply return (for internal circuitry); (60 pins).
TCK	I	Test Clock. Provides an asynchronous clock for JTAG boundary scan.
TMS	I/S	Test Mode Select. Used to control the test state machine. TMS has a 20 kΩ internal pullup resistor.
TDI	I/S	Test Data Input. Provides serial data for the boundary scan logic. TDI has a 20 kΩ internal pullup resistor.
TDO	O	Test Data Output. Serial scan output of the boundary scan path.
$\overline{TRST}$	I/A	Test Reset. Resets the test state machine. $\overline{TRST}$ must be asserted (pulsed low) after powerup or held low for proper operation of the ADSP-21010. $\overline{TRST}$ has a 20 kΩ internal pullup resistor.
NC		No Connect. No Connects are reserved pins that must be left open and unconnected.

INSTRUCTION SET SUMMARY

The ADSP-21010 instruction set provides a wide variety of programming capabilities. Every instruction assembles into a single word and can execute in a single processor cycle. Multifunction instructions enable simultaneous multiplier and ALU operations, as well as computations executed in parallel with data transfers. The addressing power of the ADSP-21010 gives you flexibility in moving data both internally and externally. The ADSP-21000 Family assembly language uses an algebraic syntax for ease of coding and readability.

The instruction types are grouped into four categories: Compute and Move or Modify, Program Flow Control, Immediate Move, and Miscellaneous.

The instruction types are numbered; there are 22 types. Some instructions have more than one syntactical form; for example, Instruction 4 has four distinct forms. The instruction number itself has no bearing on programming, but corresponds to the op code recognized by the ADSP-21010 device.

Because of the width and orthogonality of the instruction word, there are many possible instructions. For example, the ALU supports 21 fixed-point operations and 24 floating-point operations; each of these can be the compute portion of an instruction.

The following pages provide an overview and summary of the ADSP-21010 instruction set. For complete information, see the *ADSP-21020 User's Manual* and *ADSP-21020 Programmer's Quick Reference*.

This section also contains several reference tables for using the instruction set.

- Table I describes the notation and abbreviations used.
- Table II lists all condition and termination code mnemonics.
- Table III lists all register mnemonics.
- Tables IV through VII list the syntax for all compute (ALU, multiplier, shifter or multifunction) operations.
- Table VIII lists interrupts and their vector addresses.

COMPUTE AND MOVE OR MODIFY INSTRUCTIONS

1.		*compute,*	[DM(Ia, Mb) = dreg1 / dreg1 = DM(Ia, Mb)] , [PM(Ic, Md) = dreg2 / dreg2 = PM(Ic, Md)] ;
2.	*IF condition*	compute ;	
3a.	*IF condition*	*compute,*	[DM(Ia, Mb) / PM(Ic, Md)] = ureg ;
3b.	*IF condition*	*compute,*	[DM(Mb, Ia) / PM(Md, Ic)] = ureg ;
3c.	*IF condition*	*compute,*	ureg = [DM(Ia, Mb) / PM(Ic, Md)] ;
3d.	*IF condition*	*compute,*	ureg = [DM(Mb, Ia) / PM(Md, Ic)] ;
4a.	*IF condition*	*compute,*	[DM(Ia, <data6>) / PM(Ic, <data6>)] = dreg ;
4b.	*IF condition*	*compute,*	[DM(<data6>, Ia) / PM(<data6>, Ic)] = dreg ;
4c.	*IF condition*	*compute,*	dreg = [DM(Ia, <data6>) / PM(Ic, <data6>)] ;
4d.	*IF condition*	*compute,*	dreg = [DM(<data6>, Ia) / PM(<data6>, Ic)] ;
5.	*IF condition*	*compute,*	ureg1 = ureg2 ;
6a.	*IF condition*	shiftimm,	[*DM(Ia, Mb)* / *PM(Ic, Md)*] = *dreg* ;
6b.	*IF condition*	shiftimm,	*dreg* = [*DM(Ia, Mb)* / *PM(Ic, Md)*] ;
7.	*IF condition*	*compute,*	MODIFY [(Ia, Mb) / (Ic, Md)] ;

PROGRAM FLOW CONTROL INSTRUCTIONS

8.	*IF condition*	[JUMP / CALL]	[<addr24> / (PC, <reladdr24>)] ([*DB* / *LA* / *DB, LA*]) ;
9.	*IF condition*	[JUMP / CALL]	[(Md, Ic) / (PC, <reladdr6>)] ([*DB* / *LA* / *DB, LA*]) , *compute* ;
11.	*IF condition*	[RTS / RTI]	([*DB* / *LA* / *DB, LA*]) , *compute* ;
12.	LCNTR =	[<data16> / ureg] , DO	[<addr24> / (<PC, <reladdr24>)] UNTIL LCE ;
13.		DO	[<addr24> / (PC, <reladdr24>)] UNTIL termination ;

(DB) Delayed branch
(LA) Loop abort (pop loop PC stacks on branch)

IMMEDIATE MOVE INSTRUCTIONS

```
14a.  | DM(<addr32>) |  = ureg  ;
      | PM(<addr24>) |
14b.  ureg =         | DM(<addr32>) |  ;
                     | PM(<addr24>) |
15a.  | DM(<data32>, Ia) |  = ureg  ;
      | PM(<data24>, Ic) |
15b.  ureg =    | DM(<data32>, Ia) |  ;
                | PM(<data24>, Ic) |
16.   | DM(Ia, Mb) |  = <data32>;
      | PM(Ic, Md) |
17.   ureg = <data32>;
```

MISCELLANEOUS INSTRUCTIONS

```
18.   BIT        | SET |      sreg <data32>;
                 | CLR |
                 | TGL |
                 | TST |
                 | XOR |
19a.  MODIFY     | (Ia, <data32>) |  ;
                 | (Ic, <data32>) |
19b.  BITREV     (Ia, <data32>)  ;
20.   | PUSH |   LOOP   ,    | PUSH |   STS  ;
      | POP  |               | POP  |
21.   NOP ;
22.   IDLE ;
```

(In item 20, PUSH/POP before LOOP, LOOP and STS are printed in italics.)

Table I. Syntax Notation Conventions

<table>
<tr><th>Notation</th><th>Meaning</th></tr>
<tr><td>UPPERCASE</td><td>Explicit syntax—assembler keyword (notation only; assembler is not case-sensitive and lowercase is the preferred programming convention)</td></tr>
<tr><td>;</td><td>Instruction terminator</td></tr>
<tr><td>,</td><td>Separates parallel operations in an instruction</td></tr>
<tr><td>italics</td><td>Optional part of instruction</td></tr>
<tr><td>| between lines |</td><td>List of options (choose one)</td></tr>
<tr><td><data<i>n</i>></td><td><i>n</i>-bit immediate data value</td></tr>
<tr><td><addr<i>n</i>></td><td><i>n</i>-bit immediate address value</td></tr>
<tr><td><reladdr<i>n</i>></td><td><i>n</i>-bit immediate PC-relative address value</td></tr>
<tr><td>compute</td><td>ALU, multiplier, shifter or multifunction operation (from Tables IV–VII)</td></tr>
<tr><td>shiftimm</td><td>Shifter immediate operation (from Table VI)</td></tr>
<tr><td>condition</td><td>Status condition (from Table II)</td></tr>
<tr><td>termination</td><td>Termination condition (from Table II)</td></tr>
<tr><td>ureg</td><td>Universal register (from Table III)</td></tr>
<tr><td>sreg</td><td>System register (from Table III)</td></tr>
<tr><td>dreg</td><td>R15-R0, F15-F0; register file location</td></tr>
<tr><td>Ia</td><td>I7-I0; DAG1 index register</td></tr>
<tr><td>Mb</td><td>M7-M0; DAG1 modify register</td></tr>
<tr><td>Ic</td><td>I15-I8; DAG2 index register</td></tr>
<tr><td>Md</td><td>M15-M8; DAG2 modify register</td></tr>
</table>

Table II. Condition and Termination Codes

Name	Description
eq	ALU equal to zero
ne	ALU not equal to zero
ge	ALU greater than or equal to zero
lt	ALU less than zero
le	ALU less than or equal to zero
gt	ALU greater than zero
ac	ALU carry
not ac	Not ALU carry
av	ALU overflow
not av	Not ALU overflow
mv	Multiplier overflow
not mv	Not multiplier overflow
ms	Multiplier sign
not ms	Not multiplier sign
sv	Shifter overflow
not sv	Not shifter overflow
sz	Shifter zero
not sz	Not shifter zero
flag0_in	Flag 0
not flag0_in	Not Flag 0
flag1_in	Flag 1
not flag1_in	Not Flag 1
flag2_in	Flag 2
not flag2_in	Not Flag 2
flag3_in	Flag 3
not flag3_in	Not Flag 3
tf	Bit test flag
not tf	Not bit test flag
lce	Loop counter expired (DO UNTIL)
not lce	Loop counter not expired (IF)
forever	Always False (DO UNTIL)
true	Always True (IF)

In a conditional instruction, the execution of the entire instruction is based on the specified condition.

Table III. Universal Registers

Name	Function
Register File	
R15–R0	Register file locations
Program Sequencer	
PC*	Program counter; address of instruction currently executing
PCSTK	Top of PC stack
PCSTKP	PC stack pointer
FADDR*	Fetch address
DADDR*	Decode address
LADDR	Loop termination address, code; top of loop address stack
CURLCNTR	Current loop counter; top of loop count stack
LCNTR	Loop count for next nested counter-controlled loop
Data Address Generators	
I7–I0	DAG1 index registers
M7–M0	DAG1 modify registers
L7–L0	DAG1 length registers
B7–B0	DAG1 base registers
I15–I8	DAG2 index registers
M15–M8	DAG2 modify registers
L15–L8	DAG2 length registers
B15–B8	DAG2 base registers
Bus Exchange	
PX1	PMD-DMD bus exchange 1 (16 bits)
PX2	PMD-DMD bus exchange 2 (32 bits)
PX	48-bit PX1 and PX2 combination
Timer	
TPERIOD	Timer period
TCOUNT	Timer counter
Memory Interface	
DMWAIT	Wait state and page size control for data memory
DMBANK1	Data memory bank 1 upper boundary
DMBANK2	Data memory bank 2 upper boundary
DMBANK3	Data memory bank 3 upper boundary
DMADR*	Copy of last data memory address
PMWAIT	Wait state and page size control for program memory
PMBANK1	Program memory bank 1 upper boundary
PMADR*	Copy of last program memory address
System Registers	
MODE1	Mode control bits for bit-reverse, alternate registers, interrupt nesting and enable, ALU saturation, floating-point rounding mode and boundary
MODE2	Mode control bits for interrupt sensitivity, cache disable and freeze, timer enable, and I/O flag configuration
IRPTL	Interrupt latch
IMASK	Interrupt mask
IMASKP	Interrupt mask pointer (for nesting)
ASTAT	Arithmetic status flags, bit test, I/O flag values, and compare accumulator
STKY	Sticky arithmetic status flags, circular buffer overflow flags, stack status flags (not sticky)
USTAT1	User status register 1
USTAT2	User status register 2

*read-only
Refer to *ADSP-21020 User's Manual* for bit-level definitions of each register.

Table IV. ALU Compute Operations

Fixed-Point	Floating-Point
Rn = Rx + Ry	Fn = Fx + Fy
Rn = Rx − Ry	Fn = Fx − Fy
Rn = Rx + Ry, Rm = Rx − Ry	Fn = Fx + Fy, Fm = Fx − Fy
Rn = Rx + Ry + CI	Fn = ABS (Fx + Fy)
Rn = Rx − Ry + CI − 1	Fn = ABS (Fx − Fy)
Rn = (Rx + Ry)/2	Fn = (Fx + Fy)/2
COMP(Rx, Ry)	COMP(Fx, Fy)
Rn = −Rx	Fn = −Fx
Rn = ABS Rx	Fn = ABS Fx
Rn = PASS Rx	Fn = PASS Fx
Rn = MIN(Rx, Ry)	Fn = MIN(Fx, Fy)
Rn = MAX(Rx, Ry)	Fn = MAX(Fx, Fy)
Rn = CLIP Rx BY Ry	Fn = CLIP Fx BY Fy
Rn = Rx + CI	Fn = RND Fx
Rn = Rx + CI − 1	Fn = SCALB Fx BY Ry
Rn = Rx + 1	Rn = MANT Fx
Rn = Rx − 1	Rn = LOGB Fx
Rn = Rx AND Ry	Rn = FIX Fx BY Ry
Rn = Rx OR Ry	Rn = FIX Fx
Rn = Rx XOR Ry	Fn = FLOAT Rx BY Ry
Rn = NOT Rx	Fn = FLOAT Rx
	Fn = RECIPS Fx
	Fn = RSQRTS Fx
	Fn = Fx COPYSIGN Fy

Rn, Rx, Ry R15–R0; register file location, fixed-point
Fn, Fx, Fy F15–F0; register file location, floating point

Table V. Multiplier Compute Operations

|Rn, MRF, MRB| = Rx * Ry (|S, U| |S, U| |F, I, FR|)

Fn = Fx * Fy

|Rn = MRF, Rn = MRB, MRF = MRF, MRB = MRB| + Rx * Ry (|S, U| |S, U| |F, I, FR|)

|Rn = MRF, Rn = MRB, MRF = MRF, MRB = MRB| − Rx * Ry (|S, U| |S, U| |F, I, FR|)

|Rn = SAT MRF, Rn = SAT MRB, MRF = SAT MRF, MRB = SAT MRB| |(SI), (UI), (SF), (UF)|

|Rn = RND MRF, Rn = RND MRB, MRF = RND MRF, MRB = RND MRB| |(SF), (UF)|

|MRF, MRB| = 0

|MRxF, MRxB| = Rn

Rn = |MRxF, MRxB|

Rn, Rx, Ry	R15–R0; register file location, fixed-point
Fn, Fx, Fy	F15–F0; register file location, floating-point
MRxF	MR2F, MR1F; MR0F; multiplier result accumulators, foreground
MRxB	MR2B, MR1B, MR0B; multiplier result accumulators, background

(|x-input| |y-input| |data format, rounding|)

S	Signed input
U	Unsigned input
I	Integer input(s)
F	Fractional input(s)
FR	Fractional inputs, Rounded output
(SF)	Default format for 1-input operations
(SSF)	Default format for 2-input operations

Table VI. Shifter and Shifter Immediate Compute Operations

Shifter	Shifter Immediate
Rn = LSHIFT Rx BY Ry	Rn = LSHIFT Rx BY<data8>
Rn = Rn OR LSHIFT Rx BY Ry	Rn = Rn OR LSHIFT Rx BY<data8>
Rn = ASHIFT Rx BY Ry	Rn = ASHIFT Rx BY<data8>
Rn = Rn OR ASHIFT Rx BY Ry	Rn = Rn OR ASHIFT Rx BY<data8>
Rn = ROT Rx BY RY	Rn = ROT Rx BY<data8>
Rn = BCLR Rx BY Ry	Rn = BCLR Rx BY<data8>
Rn = BSET Rx BY Ry	Rn = BSET Rx BY<data8>
Rn = BTGL Rx BY Ry	Rn = BTGL Rx BY<data8>
BTST Rx BY Ry	BTST Rx BY<data8>
Rn = FDEP Rx BY Ry	Rn = FDEP Rx BY <bit6>: <len6>
Rn = Rn OR FDEP Rx BY Ry	Rn = Rn OR FDEP Rx BY <bit6>:<len6>
Rn = FDEP Rx BY Ry (SE)	Rn = FDEP Rx BY <bit6>:<len6> (SE)
Rn = Rn OR FDEP Rx BY Ry (SE)	Rn = Rn OR FDEP Rx BY <bit6>:<len6> (SE)
Rn = FEXT Rx BY Ry	Rn = FEXT Rx BY <bit6>:<len6>
Rn = FEXT Rx BY Ry (SE)	Rn = FEXT Rx BY <bit6>:<len6> (SE)
Rn = EXP Rx	
Rn = EXP Rx (EX)	
Rn = LEFTZ Rx	
Rn = LEFTO Rx	

Rn, Rx, Ry	R15–R0; register file location, fixed-point
<bit6>:<len6>	6-bit immediate bit position and length values (for shifter immediate operations)

Table VII. Multifunction Compute Operations

Fixed-Point

Rm=R3-0 * R7-4 (SSFR), Ra=R11-8 + R15-12
Rm=R3-0 * R7-4 (SSFR), Ra=R11-8 − R15-12
Rm=R3-0 * R7-4 (SSFR), Ra=(R11-8 + R15-12)/2
MRF=MRF + R3-0 * R7-4 (SSF), Ra=R11-8 + R15-12
MRF=MRF + R3-0 * R7-4 (SSF), Ra=R11-8 − R15-12
MRF=MRF + R3-0 * R7-4 (SSF), Ra=(R11-8 + R15-12)/2
Rm=MRF + R3-0 * R7-4 (SSFR), Ra=R11-8 + R15-12
Rm=MRF + R3-0 * R7-4 (SSFR), Ra=R11-8 − R15-12
Rm=MRF + R3-0 * R7-4 (SSFR), Ra=(R11-8 + R15-12)/2
MRF=MRF − R3-0 * R7-4 (SSF), Ra=R11-8 + R15-12
MRF=MRF − R3-0 * R7-4 (SSF), Ra=R11-8 − R15-12
MRF=MRF − R3-0 * R7-4 (SSF), Ra=(R11-8 + R15-12)/2
Rm=MRF − R3-0 * R7-4 (SSFR), Ra=R11-8 + R15-12
Rm=MRF − R3-0 * R7-4 (SSFR), Ra=R11-8 − R15-12
Rm=MRF − R3-0 * R7-4 (SSFR), Ra=(R11-8 + R15-12)/2
Rm=R3-0 * R7-4 (SSFR), Ra=R11-8 + R15-12,
Rs=R11-8 − R15-12

Floating-Point

Fm=F3-0 * F7-4, Fa=F11-8 + F15-12
Fm=F3-0 * F7-4, Fa=F11-8 − F15-12
Fm=F3-0 * F7-4, Fa=FLOAT R11-8 by R15-12
Fm=F3-0 * F7-4, Fa=FIX R11-8 by R15-12
Fm=F3-0 * F7-4, Fa=(F11-8 + F15-12)/2
Fm=F3-0 * F7-4, Fa=ABS F11-8
Fm=F3-0 * F7-4, Fa=MAX (F11-8, F15-12)
Fm=F3-0 * F7-4, Fa=MIN (F11-8, F15-12)
Fm=F3-0 * F7-4, Fa=F11-8 + F15-12,
Fs=F11-8 − F15-12

Ra, Rm	Any register file location (fixed-point)
R3-0	R3, R2, R1, R0
R7-4	R7, R6, R5, R4
R11-8	R11, R10, R9, R8
R15-12	R15, R14, R13, R12
Fa, Fm	Any register file location (floating-point)
F3-0	F3, F2, F1, F0
F7-4	F7, F6, F5, F4
F11-8	F11, F10, F9, F8
F15-12	F15, F14, F13, F12
(SSF)	X-input signed, Y-input signed, fractional inputs
(SSFR)	X-input signed, Y-input signed, fractional inputs, rounded output

Table VIII. Interrupt Vector Addresses and Priorities

No.	Vector Address (Hex)	Function
0	0x00	Reserved
1*	0x08	Reset
2	0x10	Reserved
3	0x18	Status stack or loop stack overflow or PC stack full
4	0x20	Timer=0 (high priority option)
5	0x28	$\overline{IRQ3}$ asserted
6	0x30	$\overline{IRQ2}$ asserted
7	0x38	$\overline{IRQ1}$ asserted
8	0x40	$\overline{IRQ0}$ asserted
9	0x48	Reserved
10	0x50	Reserved
11	0x58	DAG 1 circular buffer 7 overflow
12	0x60	DAG 2 circular buffer 15 overflow
13	0x68	Reserved
14	0x70	Timer=0 (low priority option)
15	0x78	Fixed-point overflow
16	0x80	Floating-point overflow
17	0x88	Floating-point underflow
18	0x90	Floating-point invalid operation
19–23	0x98-0xB8	Reserved
24–31	0xC0–0xF8	User software interrupts

*Nonmaskable

ADSP-21010—SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

Parameter		K Grade Min	K Grade Max	B Grade Min	B Grade Max	Unit
V_{DD}	Supply Voltage	4.50	5.50	4.50	5.50	V
T_{AMB}	Ambient Operating Temperature	0	+70	−40	+85	°C

Refer to Environmental Conditions for information on thermal specifications.

ELECTRICAL CHARACTERISTICS

Parameter		Test Conditions	Min	Max	Unit
V_{IH}	Hi-Level Input Voltage[1]	V_{DD} = max	2.0		V
V_{IHCR}	Hi-Level Input Voltage[2, 12]	V_{DD} = max	3.0		V
V_{IL}	Lo-Level Input Voltage[1, 12]	V_{DD} = min		0.8	V
V_{ILC}	Lo-Level Input Voltage[2]	V_{DD} = max		0.6	V
V_{OH}	Hi-Level Output Voltage[3, 11]	V_{DD} = min, I_{OH} = −1.0 mA	2.4		V
V_{OL}	Lo-Level Output Voltage[3, 11]	V_{DD} = min, I_{OL} = 4.0 mA		0.4	V
I_{IH}	Hi-Level Input Current[4, 5]	V_{DD} = max, V_{IN} = V_{DD} max		10	μA
I_{IL}	Lo-Level Input Current[4]	V_{DD} = max, V_{IN} = 0 V		10	μA
I_{ILT}	Lo-Level input Current[5]	V_{DD} = max, V_{IN} = 0 V		350	μA
I_{OZH}	Tristate Leakage Current[6]	V_{DD} = max, V_{IN} = V_{DD} max		10	μA
I_{OZL}	Tristate Leakage Current[6]	V_{DD} = max, V_{IN} = 0 V		10	μA
I_{DDIN}	Supply Current (Internal)[7]	t_{CK} = min, V_{DD} = max, V_{IHCR} = 3.0 V, V_{IH} = 2.4 V, V_{IL} = V_{ILC} = 0.4 V		390	mA
I_{DDIDLE}	Supply Current (Idle)[8]	V_{DD} = max, V_{IN} = 0 V or V_{DD} max		110	mA
C_{IN}	Input Capacitance[9, 10]	f_{IN} = 1 MHz, T_{CASE} = 25°C, V_{IN} = 2.5 V		8	pF

NOTES
[1]Applies to: PMD47–0, PMACK, PMTS, DMD31–0, DMACK, DMTS, IRQ3–0, FLAG3–0, BR, TMS, TDI.
[2]Applies to: CLKIN, TCK.
[3]Applies to: PMA23–0, PMD47–0, PMS1–0, PMRD, PMWR, PMPAGE, DMA31–0, DMD31–0, DMS3–0, DMRD, DMWR, DMPAGE, FLAG3–0, TIMEXP, BG.
[4]Applies to: PMACK, PMTS, DMACK, DMTS, IRQ3–0, BR, CLKIN, RESET, TCK.
[5]Applies to: TMS, TDI, TRST.
[6]Applies to: PMA23–0, PMD47–0, PMS1–0, PMRD, PMWR, PMPAGE, DMA31–0, DMD31–0, DMS3–0, DMRD, DMWR, DMPAGE, FLAG3–0, TDO.
[7]Applies to IVDD pins. I_{DDIN} typically equals 183 mA. See "Power Dissipation" for calculation of external (EVDD) supply current for total supply current.
[8]Applies to IVDD pins. Idle refers to ADSP-21010 state of operation during execution of the IDLE instruction.
[9]Guaranteed but not tested.
[10]Applies to all signal pins.
[11]Although specified for TTL outputs, all ADSP-21010 outputs are CMOS-compatible and will drive to V_{DD} and GND assuming no dc loads.
[12]Applies to RESET, TRST.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage . . . . −0.3 V to +7 V
Input Voltage . . . . −0.3 V to V_{DD} + 0.3 V
Output Voltage Swing . . . . −0.3 V to V_{DD} + 0.3 V
Load Capacitance . . . . 200 pF
Operating Temperature Range (Ambient) . . . −55°C to +125°C
Storage Temperature Range . . . . −65°C to +150°C
Lead Temperature (5 seconds) . . . . 280°C

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings 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.

ESD SENSITIVITY

The ADSP-21010 features proprietary input protection circuitry to dissipate high-energy discharges (Human Body Model). Per method 3015 of MIL-STD-883, the ADSP-21010 has been classified as a Class 3 device, with the ability to withstand up to 4000 V ESD.

Proper ESD precautions are recommended to avoid functional damage or performance degradation. Charges 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*.

TIMING PARAMETERS

General Notes

See Figure 15 on page 20 for voltage reference levels. Use the exact timing information given. Do not attempt to derive parameters from the addition or subtraction of others. While addition or subtraction would yield meaningful results for an individual device, the values given in this data sheet reflect statistical variations and worst cases. Consequently, you cannot meaningfully add parameters to derive other specifications.

Clock Signal

Parameter		12.5 MHz Min	Max	Unit
Timing Requirement:				
t_{CK}	CLKIN Period	80	200	ns
t_{CKH}	CLKIN Width High	15		ns
t_{CKL}	CLKIN Width Low	15		ns

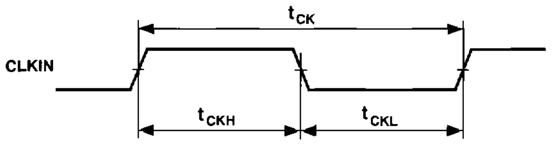


Figure 3. Clock

Reset

Parameter		12.5 MHz Min	Max	Frequency Dependency* Min	Max	Unit
Timing Requirement:						
t_{WRST}[1]	RESET Width Low	320		$4t_{CK}$		ns
t_{SRST}[2]	RESET Setup Before CLKIN High	44	80	29 + DT/2	t_{CK}	ns

*DT = t_{CK} − 50 ns

NOTES

[1]Applies after the powerup sequence is complete. At powerup, the Internal Phase Locked Loop requires no more than 1000 CLKIN cycles while RESET is low, assuming stable V_{DD} and CLKIN (not including clock oscillator start-up time).

[2]Specification only applies in cases where multiple ADSP-21010 processors are required to execute in program counter lock-step (all processors start execution at location 8 in the same cycle). See the Hardware Configuration chapter of the *ADSP-21020 User's Manual* for reset sequence information.

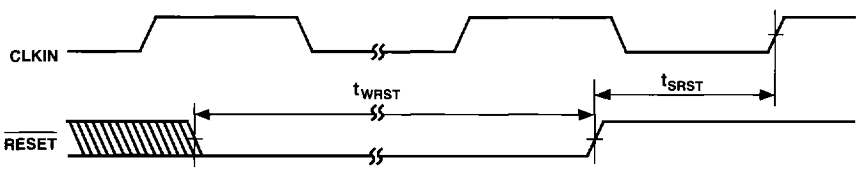


Figure 4. Reset

Interrupts

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Timing Requirement:						
t_{SIR}	$\overline{IRQ3-0}$ Setup Before CLKIN High	63		40 + 3DT/4		ns
t_{HIR}	$\overline{IRQ3-0}$ Hold After CLKIN High	0				ns
t_{IPW}	$\overline{IRQ3-0}$ Pulse Width	85		t_{CK} + 5		ns

*DT = t_{CK} − 50 ns

NOTE

Meeting setup and hold guarantees interrupts will be latched in that cycle. Meeting the pulse width is not necessary if the setup and hold is met. Likewise, meeting the setup and hold is not necessary if the pulse width is met. See the Hardware Configuration chapter of the *ADSP-21020 User's Manual* for interrupt servicing information.

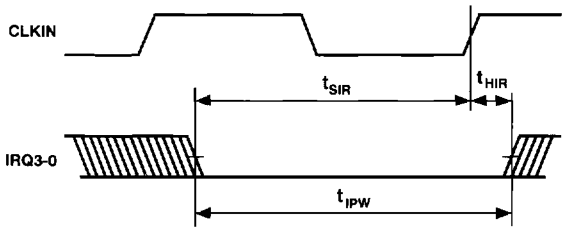


Figure 5. Interrupts

Flags

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Timing Requirement:[1]						
t_{SFI}	FLAG3-0_{IN} Setup before CLKIN High	28		19 + 5DT/16		ns
t_{HFI}	FLAG3-0_{IN} Hold after CLKIN High	0				ns
t_{DWRFI}	FLAG3-0_{IN} Delay from $\overline{xRD}$, $\overline{xWR}$ Low		23		10 + 7DT/16	ns
t_{HFIWR}	FLAG3-0_{IN} Hold after $\overline{xRD}$, $\overline{xWR}$ Deasserted	0				ns
Switching Characteristic:						
t_{DFO}	FLAG3-0_{OUT} Delay from CLKIN High		31			ns
t_{HFO}	FLAG3-0_{OUT} Hold after CLKIN High	5				ns
t_{DFOE}	CLKIN High to FLAG3-0_{OUT} Enable	1				ns
t_{DFOD}	CLKIN High to FLAG3-0_{OUT} Disable		31			ns

*DT = t_{CK} − 50 ns

NOTES

[1]Flag inputs meeting these setup and hold times will affect conditional operations in the next instruction cycle. See the Hardware Configuration chapter of the *ADSP-21020 User's Manual* for additional flag servicing information.

x = PM or DM.

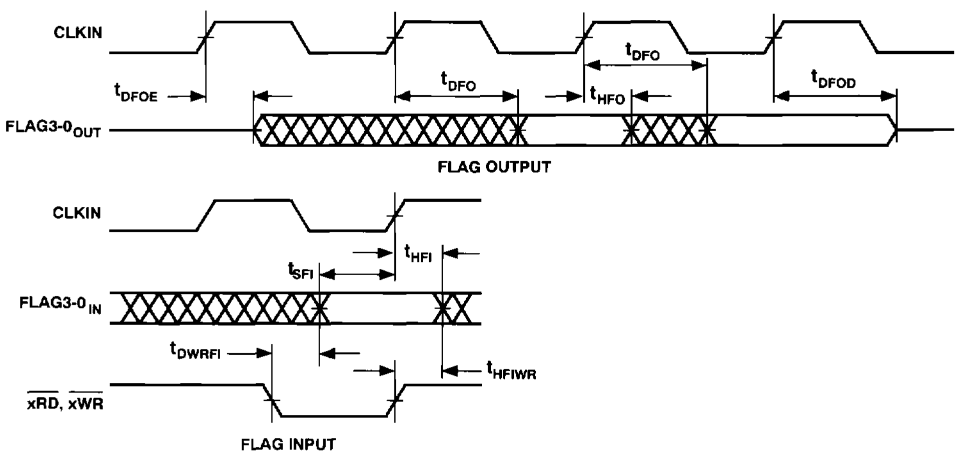


Figure 6. Flags

Timer

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Switching Characteristic:						
t_{DTEX}	CLKIN High to TIMEXP		60		45 + DT/2	ns

*DT = t_{CK} − 50 ns

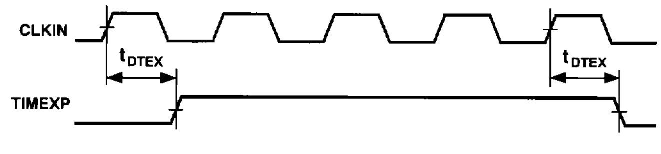


Figure 7. TIMEXP

Bus Request/Bus Grant

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Timing Requirement:						
t_{HBR}	$\overline{BR}$ Hold after CLKIN High	0				ns
t_{SBR}	$\overline{BR}$ Setup before CLKIN High	27		18 + 5DT/16		ns
Switching Characteristic:						
t_{DMDBGL}	Memory Interface Disable to $\overline{BG}$ Low	−5				ns
t_{DME}	CLKIN High to Memory Interface Enable	40		25 + DT/2		ns
t_{DBGL}	CLKIN High to $\overline{BG}$ Low		31			ns
t_{DBGH}	CLKIN High to $\overline{BG}$ High		60		45 + DT/2	ns

*DT = t_{CK} − 50 ns

NOTES
Memory Interface = PMA23-0, PMD47-0, $\overline{PMS1\text{-}0}$, $\overline{PMRD}$, $\overline{PMWR}$, PMPAGE, DMA31-0, DMD31-0, $\overline{DMS3\text{-}0}$, $\overline{DMRD}$, $\overline{DMWR}$, DMPAGE.
Buses are not granted until completion of current memory access.
See the Memory Interface chapter of the *ADSP-21020 User's Manual* for $\overline{BG}$, $\overline{BR}$ cycle relationships.

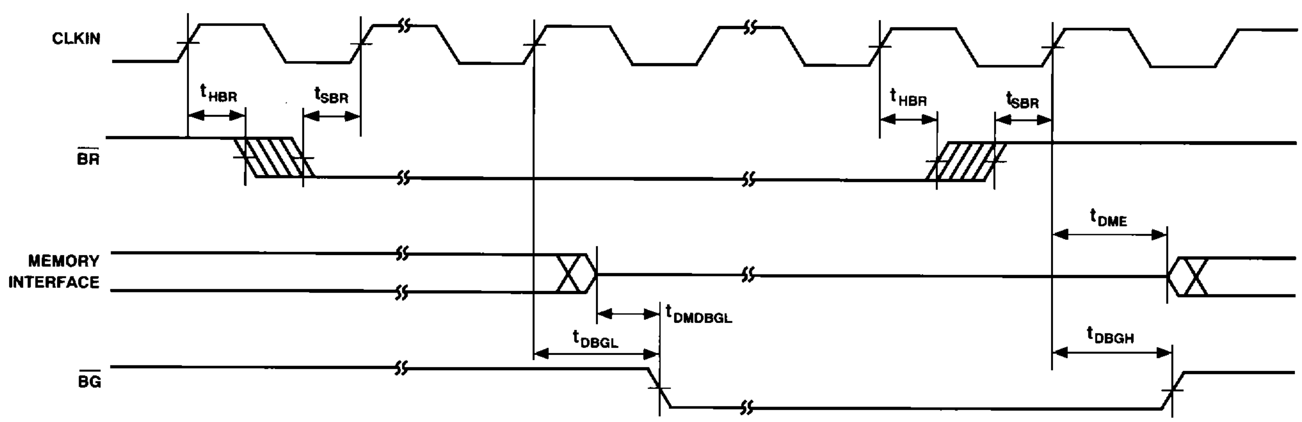


Figure 8. Bus Request/Bus Grant

External Memory Three-State Control

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Timing Requirement:						
t_{STS}	$\overline{xTS}$, Setup before CLKIN High	22	80	14 + DT/4	t_{CK}	ns
t_{DADTS}	$\overline{xTS}$ Delay after Address, Select		51		25 + 7DT/8	ns
t_{DSTS}	xTS Delay after $\overline{XRD}$, $\overline{XWR}$ Low	0	30		15 + DT/2	ns
Switching Characteristic:						
t_{DTSD}	Memory Interface Disable before CLKIN High	2		−5 + DT/4		ns
t_{DTSAE}	$\overline{xTS}$ High to Address, Select Enable	0				ns

*DT = t_{CK} − 50 ns

NOTES

$\overline{xTS}$ should only be asserted (low) during an active memory access cycle.

Memory Interface = PMA23-0, PMD47-0, $\overline{PMS1-0}$, $\overline{PMRD}$, $\overline{PMWR}$, PMPAGE, DMA31-0, DMD31-0, $\overline{DMS3-0}$, $\overline{DMRD}$, $\overline{DMWR}$, DMPAGE.

Address = PMA23-0, DMA31-0. Select = $\overline{PMS1-0}$, $\overline{DMS3-0}$.

x = PM or DM.

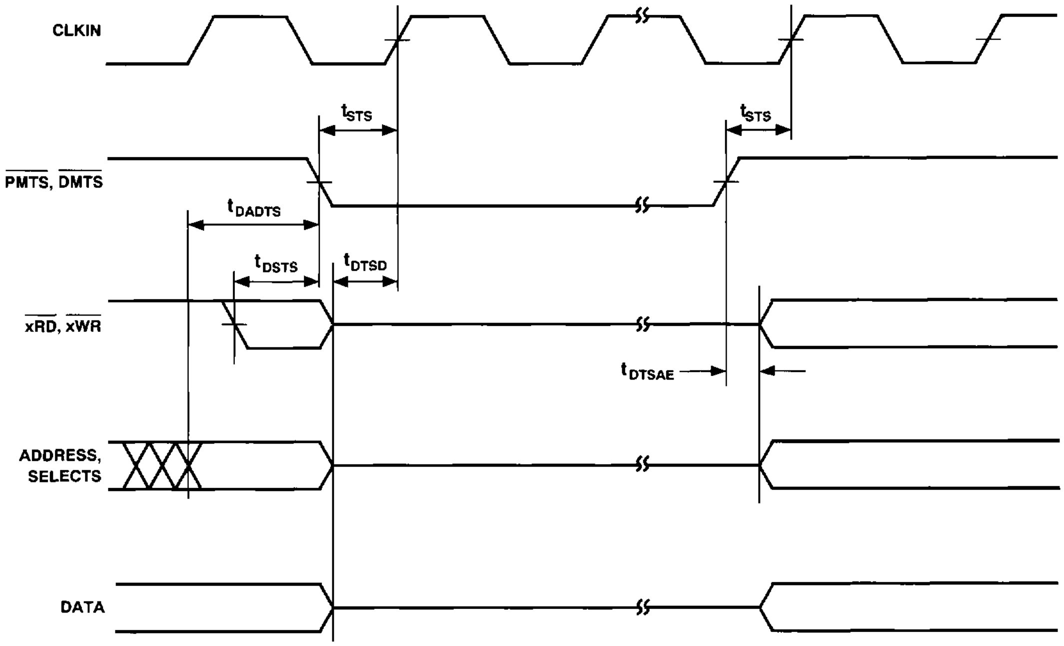


Figure 9. External Memory Three-State Control

Memory Read

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Timing Requirement:						
t_{DAD}	Address, Select to Data Valid		67		37 + DT	ns
t_{DRLD}	$\overline{xRD}$ Low to Data Valid		42		23 + 5DT/8	ns
t_{HDA}	Data Hold from Address, Select	0				ns
t_{HDRH}	Data Hold from $\overline{xRD}$ High	1				ns
t_{DAAK}	xACK Delay from Address		47		21 + 7DT/8	ns
t_{DRAK}	xACK Delay from $\overline{xRD}$ Low		25		10 + DT/2	ns
t_{SAK}	xACK Setup before CLKIN High	24		16 + DT/4		ns
t_{HAK}	xACK Hold after CLKIN High	0				ns
Switching Characteristic:						
t_{DARL}	Address, Select to $\overline{xRD}$ Low	16		5 + 3DT/8		ns
t_{DAP}	xPAGE Delay from Address, Select		5			ns
t_{DCKRL}	CLKIN High to $\overline{xRD}$ Low	21	37	14 + DT/4	29+DT/4	ns
t_{RW}	$\overline{xRD}$ Pulse Width	42		23 + 5DT/8		ns
t_{RWR}	$\overline{xRD}$ High to $\overline{xRD}$, $\overline{xWR}$ Low	21		10 + 3DT/8		ns

*DT = t_{CK} − 50 ns

NOTE
x = PM or DM; Address = PMA23-0, DMA31-0; Data = PMD47-0, DMD31-0; Select = $\overline{PMS1-0}$, $\overline{DMS3-0}$.

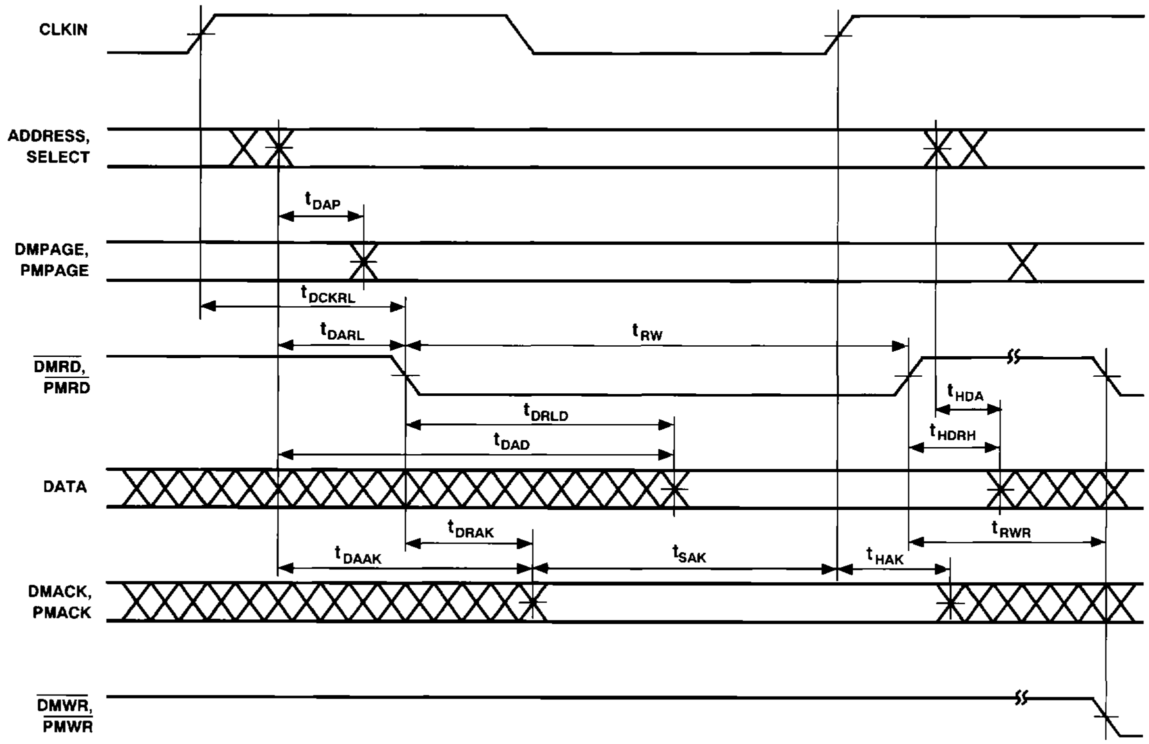


Figure 10. Memory Read

Memory Write

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Timing Requirement:						
t_{DAAK}	xACK Delay from Address, Select		47		21 + 7DT/8	ns
t_{DWAK}	xACK Delay from $\overline{xWR}$ Low		25		10 + DT/2	ns
t_{SAK}	xACK Setup before CLKIN High	22		14 + DT/4		ns
t_{HAK}	xACK Hold after CLKIN High	0				ns
Switching Characteristic:						
t_{DAWH}	Address, Select to $\overline{xWR}$ Deasserted	61		33 + 15DT/16		ns
t_{DAWL}	Address, Select to $\overline{xWR}$ Low	17		6 + 3DT/8		ns
t_{WW}	$\overline{xWR}$ Pulse Width	38		21 + 9DT/16		ns
t_{DDWH}	Data Setup before $\overline{xWR}$ High	32		17 + DT/2		ns
t_{DWHA}	Address, Select Hold after $\overline{xWR}$ Deasserted	2		DT/16		ns
t_{HDWH}	Data Hold after $\overline{xWR}$ Deasserted[1]	−2				ns
t_{DAP}	xPAGE Delay from Address, Select		5			ns
t_{DCKWL}	CLKIN High to $\overline{xWR}$ Low	21	37	14 + DT/4	29 + DT/4	ns
t_{WWR}	$\overline{xWR}$ High to $\overline{xWR}$ or $\overline{xRD}$ Low	23		10 + 7DT/16		ns
t_{DDWR}	Data Disable before $\overline{xWR}$ or $\overline{xRD}$ Low	23		10 + 7DT/16		ns
t_{WDE}	$\overline{xWR}$ Low to Data Enabled	0		−2 + DT/16		ns

*DT = t_C − 50 ns

NOTE
[1]See "System Hold Time Calculation" in "Test Conditions" section for calculating hold times given capacitive and DC loads.
x = PM or DM; Address = PMA23-0, DMA31-0; Data = PMD47-0, DMD31-0; Select = $\overline{PMS1-0}$, $\overline{DMS3-0}$.

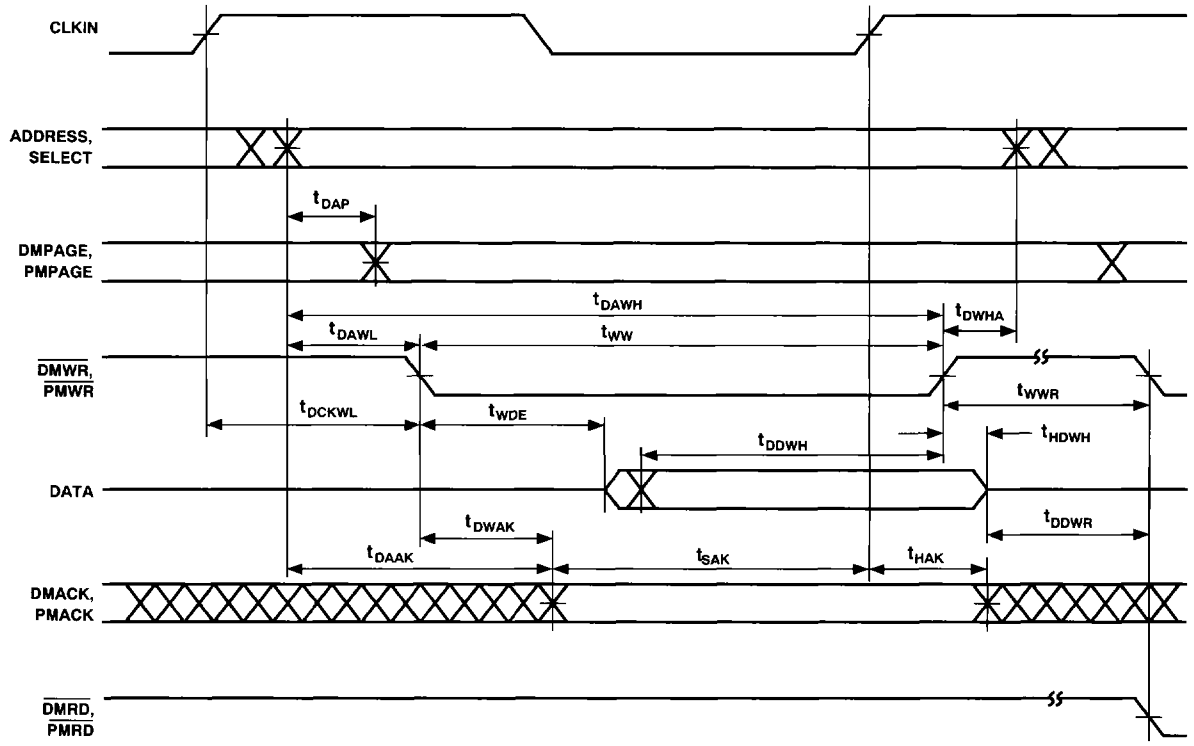


Figure 11. Memory Write

IEEE 1149.1 Test Access Port

Parameter		12.5 MHz Min	12.5 MHz Max	Frequency Dependency* Min	Frequency Dependency* Max	Unit
Timing Requirement:						
t_{TCK}	TCK Period	80		t_{CK}		ns
t_{STAP}	TDI, TMS Setup before TCK High	7				ns
t_{HTAP}	TDI, TMS Hold after TCK High	6				ns
t_{SSYS}	System Inputs Setup before TCK High	45		5 + $t_{CK}/2$		ns
t_{HSYS}	System Inputs Hold after TCK High	14				ns
t_{TRSTW}	$\overline{TRST}$ Pulse Width	200				ns
Switching Characteristic:						
t_{DTDO}	TDO Delay from TCK Low		18			ns
t_{DSYS}	System Outputs Delay from TCK Low		30			ns

*DT = t_{CK} − 50 ns

NOTES
System Inputs = PMD47-0, PMACK, $\overline{PMTS}$, DMD31-0, DMACK, $\overline{DMTS}$, CLKIN, $\overline{IRQ3\text{-}0}$, $\overline{RESET}$, FLAG3-0, $\overline{BR}$.
System Outputs = PMA23-0, $\overline{PMS1\text{-}0}$, $\overline{PMRD}$, $\overline{PMWR}$, PMD47-0, PMPAGE, DMA31-0, $\overline{DMS1\text{-}0}$, $\overline{DMRD}$, $\overline{DMWR}$, DMD31-0, DMPAGE, FLAG3-0, $\overline{BG}$, TIMEXP.
See the IEEE 1149.1 Test Access Port chapter of the *ADSP-21020 User's Manual* for further detail.

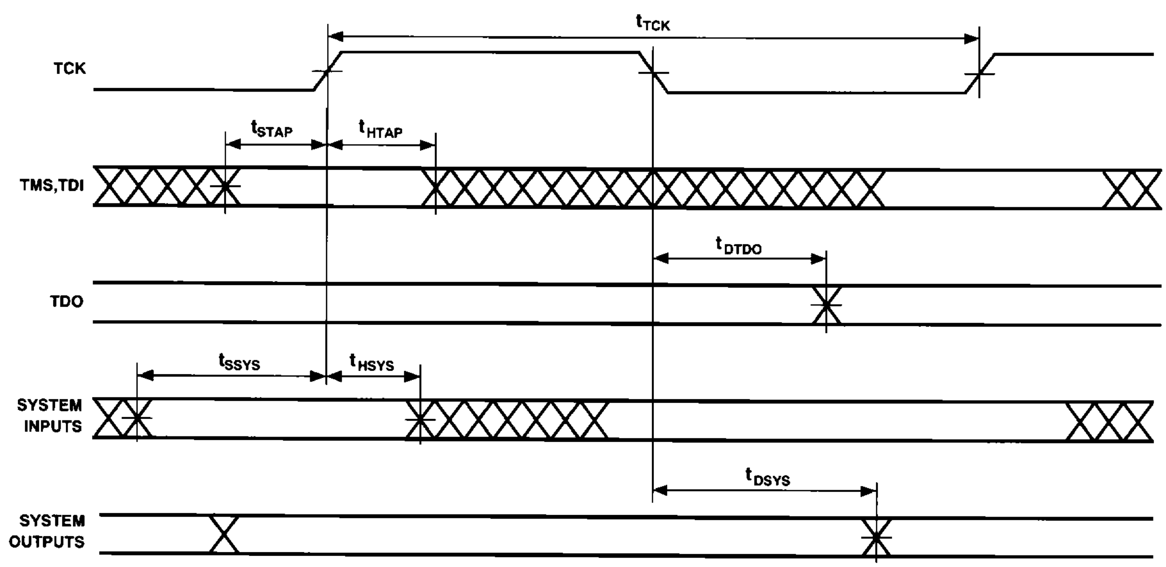


Figure 12. IEEE 1149.1 Test Access Port

TEST CONDITIONS

Output Disable Time

Output pins are considered to be disabled when they stop driving, go into a high-impedance state, and start to decay from their output high or low voltage. The time for the voltage on the bus to decay by ΔV is dependent on the capacitive load, C_L, and the load current, I_L. It can be approximated by the following equation:

$$t_{DECAY} = \frac{C_L \, \Delta V}{I_L}$$

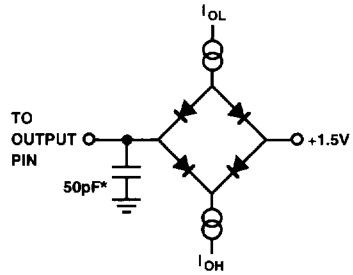
The output disable time (t_{DIS}) is the difference between $t_{MEASURED}$ and t_{DECAY} as shown in Figure 13. The time $t_{MEASURED}$ is the interval from when the reference signal switches to when the output voltage decays ΔV from the measured output high or output low voltage. t_{DECAY} is calculated with ΔV equal to 0.5 V, and test loads C_L and I_L.

Output Enable Time

Output pins are considered to be enabled when they have made a transition from a high-impedance state to when they start driving. The output enable time (t_{ENA}) is the interval from when a reference signal reaches a high or low voltage level to when the output has reached a specified high or low trip point, as shown in the Output Enable/Disable diagram. If multiple pins (such as the data bus) are enabled, the measurement value is that of the first pin to start driving.

Example System Hold Time Calculation

To determine the data output hold time in a particular system, first calculate t_{DECAY} using the above equation. Choose ΔV to be the difference between the ADSP-21010's output voltage and the input threshold for the device requiring the hold time. A typical ΔV will be 0.4 V. C_L is the total bus capacitance (per data line), and I_L is the total leakage or three-state current (per data line). The hold time will be t_{DECAY} plus the minimum disable time (i.e., t_{HDWD} for the write cycle).



*AC TIMING SPECIFICATIONS ARE CALCULATED FOR 100pF DERATING ON THE FOLLOWING PINS: PMA23–0, PMS1–0, PMRD, PMWR, PMPAGE, DMA31–0, DMS3–0, DMRD, DMWR, DMPAGE

Figure 14. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

Figure 15. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

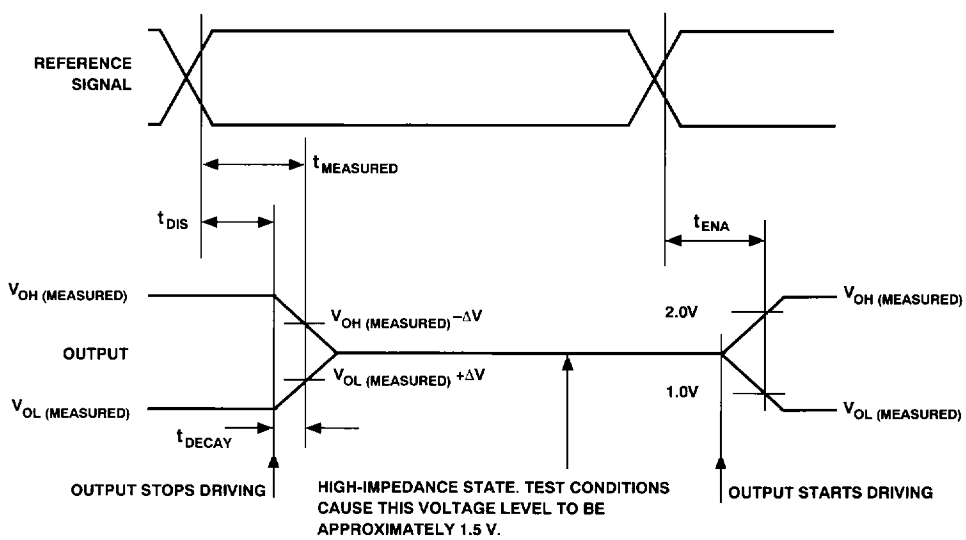


Figure 13. Output Enable/Disable

Capacitive Loading

Output delays are based on standard capacitive loads: 100 pF on address, select, page and strobe pins, and 50 pF on all others (see Figure 14). For different loads, these timing parameters should be derated. See the Hardware Configuration chapter of the *ADSP-21020 User's Manual* for further information on derating of timing specifications.

Figures 16 and 17 show how the output rise time varies with capacitance. Figures 18 and 19 show how output delays vary with capacitance. Note that the graphs may not be linear outside the ranges shown.

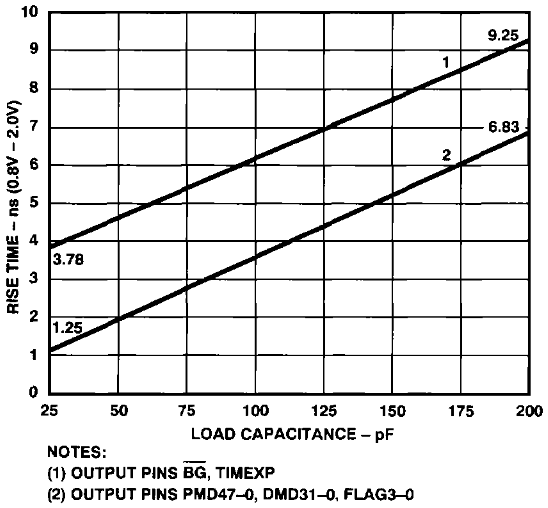


Figure 16. Typical Output Rise Time vs. Load Capacitance (at Maximum Case Temperature)

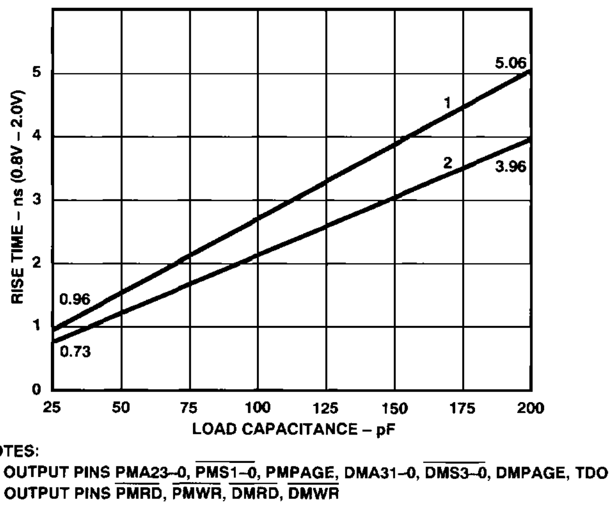


Figure 17. Typical Output Rise Time vs. Load Capacitance (at Maximum Case Temperature)

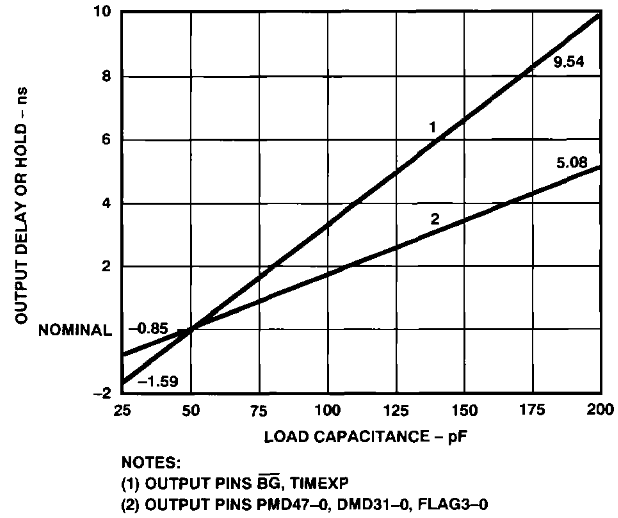


Figure 18. Typical Output Delay or Hold vs. Load Capacitance (at Maximum Case Temperature)

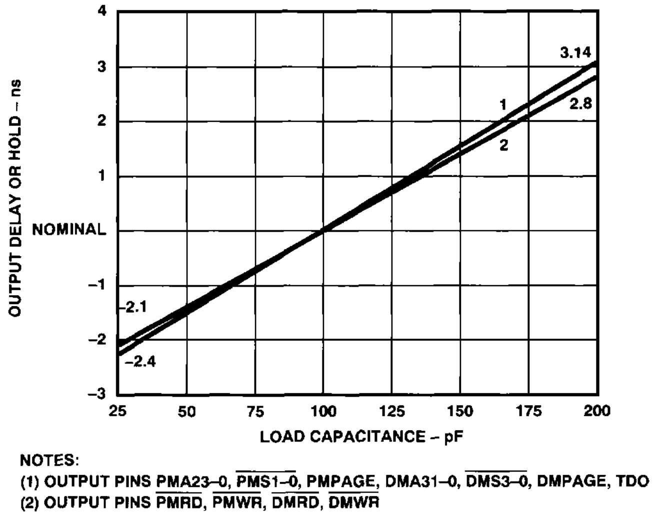


Figure 19. Typical Output Delay or Hold vs. Load Capacitance (at Maximum Case Temperature)

ENVIRONMENTAL CONDITIONS

The ADSP-21010 is packaged in a Plastic Quad Flatpack (PQFP). It is a commercial grade (K grade) device, specified for operation at T_{AMB} of 0°C to 70°C.

Maximum T_{CASE} (case temperature) can be calculated from the following equation:

$$T_{CASE} = T_{AMB} + (PD \times \theta_{CA})$$

where PD is power dissipation and θ_{CA} is the case-to-ambient thermal resistance. The value of PD depends on your application; the method for calculating PD is shown under "Power Dissipation" below. θ_{CA} varies with airflow. Table IX shows a range of θ_{CA} values.

Table IX. Maximum θ_{CA} for Various Airflow Values

Airflow (Linear ft./min.)	0	100	200	300
PQFP	13.9 °C/W	11.3 °C/W	9.0 °C/W	7.6 °C/W

NOTE:
As per method 1012 MIL-STD-883. Ambient temperature: 25°C.
Power: 2.0 W.

Power Dissipation

Total power dissipation has two components: one due to internal circuitry and one due to the switching of external output drivers. Internal power dissipation is dependent on the instruction execution sequence and the data values involved. Internal power dissipation is calculated in the following way:

$$P_{INT} = I_{DDIN} \times V_{DD}$$

The external component of total power dissipation is caused by the switching of output pins. Its magnitude depends on:

1) the number of output pins that switch during each cycle (O),
2) the maximum frequency at which they can switch (f),
3) their load capacitance (C), and
4) their voltage swing (V_{DD}).

It is calculated by:

$$P_{EXT} = O \times C \times V_{DD}^2 \times f$$

The load capacitance should include the processor's package capacitance (C_{IN}). The switching frequency includes driving the load high and then back low. Address and data pins can drive high and low at a maximum rate of $1/(2t_{CK})$. The write strobes can switch every cycle at a frequency of $1/t_{CK}$. Select pins switch at $1/(2t_{CK})$, but 2 DM and 2 PM selects can switch on each cycle. If only one bank is accessed, no select line will switch.

Example:

Estimate P_{EXT} with the following assumptions:

- A system with one RAM bank each of PM (48 bits) and DM (32 bits).
- 32K × 8 RAM chips are used, each with a load of 10 pF.
- Single-precision mode is enabled so that only 32 data pins can switch at once.
- PM and DM writes occur every other cycle, with 50% of the pins switching.
- The instruction cycle rate is 12.5 MHz (t_{CK} = 80 ns) and V_{DD} = 5.0 V.

The P_{EXT} equation is calculated for each class of pins that can drive:

Pin Type	# Pins	% Switch	× f	$\times V_{DD}^2$	P_{EXT}
PMA	15	50	3.15 MHz	25 V	0.040 W
$\overline{PMS}$	2	0	3.15 MHz	25 V	0.000 W
$\overline{PMWR}$	1	–	6.25 MHz	25 V	0.0105 W
PMD	32	50	3.15 MHz	25 V	0.0225 W
DMA	15	50	3.15 MHz	25 V	0.0285 W
$\overline{DMS}$	2	0	3.15 MHz	25 V	0.000 W
$\overline{DMWR}$	1	–	6.25 MHz	25 V	0.0075 W
DMD	32	50	3.15 MHz	25 V	0.0225 W

P_{EXT} = 0.132 W

A typical power consumption can now be calculated for this situation by adding a typical internal power dissipation:

$$P_{TOTAL} = P_{EXT} + (5\ V \times I_{DDIN}\ (typ)) = 0.132 + 0.915\ W$$

$$P_{TOTAL} = 1.05\ W$$

Note that the conditions causing a worst case P_{EXT} are different from those causing a worst case P_{INT}. Maximum P_{INT} cannot occur while 100% of the output pins are switching from all ones to all zeros. Also note that it is not common for a program to have 100% or even 50% of the outputs switching simultaneously.

Power and Ground Guidelines

To achieve its fast cycle time, including instruction fetch, data access, and execution, the ADSP-21010 is designed with high speed drivers on all output pins. Large peak currents may pass through a circuit board's ground and power lines, especially when many output drivers are simultaneously charging or discharging their load capacitances. These transient currents can cause disturbances on the power and ground lines. To minimize these effects, the ADSP-21010 provides separate supply pins for its internal logic (IGND and IVDD) and for its external drivers (EGND and EVDD).

To reduce system noise at low temperatures when transistors switch fastest, the ADSP-21010 employs compensated output drivers. These drivers equalize slew rate over temperature extremes and process variations. A 1.8 kΩ resistor placed between the RCOMP pin and EVDD (+5 V) provides a reference for the compensated drivers. Use of a capacitor (approximately 100 pF), placed in parallel with the 1.8 kΩ resistor, is recommended.

All GND pins should have a low impedance path to ground. A ground plane is required in ADSP-21010 systems to reduce this impedance, minimizing noise.

The EVDD and IVDD pins should be bypassed to the ground plane using (approximately) 12 high-frequency capacitors (0.1 µF ceramic). Keep each capacitor's lead and trace length to the pins as short as possible. This low inductive path provides the ADSP-21010 with the peak currents required when its output drivers switch. The capacitors' ground leads should also be short and connect directly to the ground plane. This provides a low impedance return path for the load capacitance of the ADSP-21010's output drivers.

If a V_{DD} plane is not used, the following recommendations apply. Traces from the +5 V supply to the 20 EVDD pins should be designed to satisfy the minimum V_{DD} specification while carrying average dc currents of [I_{DDEX}/20 × (number of EVDD pins per trace)]. I_{DDEX} is the calculated external supply current. A similar calculation should be made for the eight IVDD pins using the I_{DDIN} specification. The traces connecting +5 V to the IVDD pins should be separate from those connecting to the EVDD pins.

A low frequency bypass capacitor (20 μF tantalum) located near the junction of the IVDD and EVDD traces is also recommended.

Target System Requirements for Use of EZ-ICE Emulator
The ADSP-21020/21010 EZ-ICE uses the IEEE 1149.1 JTAG test access port of the ADSP-21010 to monitor and control the target board processor during emulation. The EZ-ICE probe requires that CLKIN, TMS, TCK, $\overline{TRST}$, TDI, TDO, and GND be made accessible on the target system via a 12-pin connector (pin strip header) such as that shown in Figure 20. The EZ-ICE probe plugs directly onto this connector for chip-on-board emulation; you must add this connector to your ADSP-21010 target board design if you intend to use the EZ-ICE. Figure 21 shows the dimensions of the EZ-ICE probe; be sure to allow space in your system to fit the probe onto the 12-pin connector.

The 12-pin, 2-row pin strip header is keyed at the Pin 1 location—you must clip Pin 1 off of the header. The pins must be 0.025 inch square and at least 0.20 inch in length. Pin spacing is 0.1 × 0.1 inches.

The tip of the pins must be at least 0.10 inch higher than the tallest component under the probe to allow clearance for the bottom of the probe. Pin strip headers are available from vendors such as 3M, McKenzie, and Samtec.

The length of the traces between the EZ-ICE probe connector and the ADSP-21010's test access port pins should be less than 1 inch. Note that the EZ-ICE probe adds two TTL loads to the CLKIN pin of the ADSP-21010.

The BMTS, BTCK, $\overline{BTRST}$, and BTDI signals are provided so that the test access port can also be used for board-level testing. When the connector is not being used for emulation, place jumpers between the BXXX pins and the XXX pins as shown in Figure 20. If you are not going to use the test access port for board test, tie $\overline{BTRST}$ to GND and tie or pull up BTCK to V_{DD}. The $\overline{TRST}$ pin must be asserted (pulsed low) after power up (through BTRST on the connector) or held low for proper operation of the ADSP-21010 device.

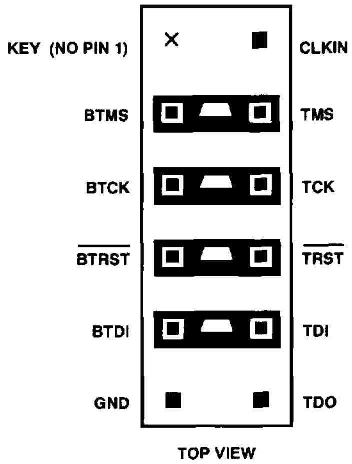


Figure 20. Target Board Connector for EZ-ICE Emulator (Jumpers in Place)

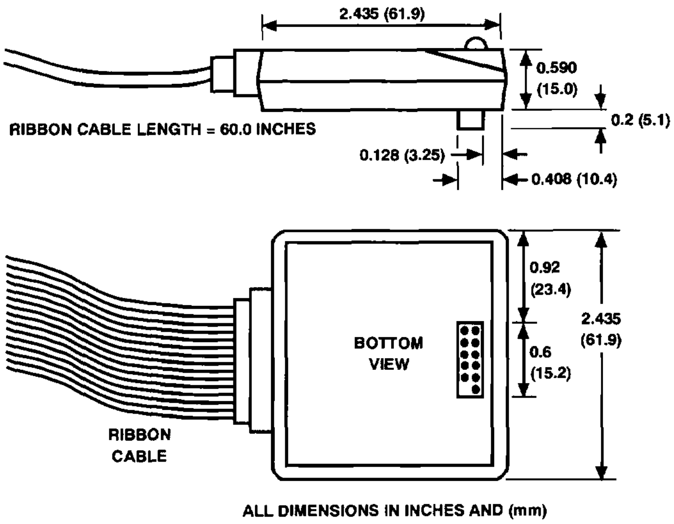
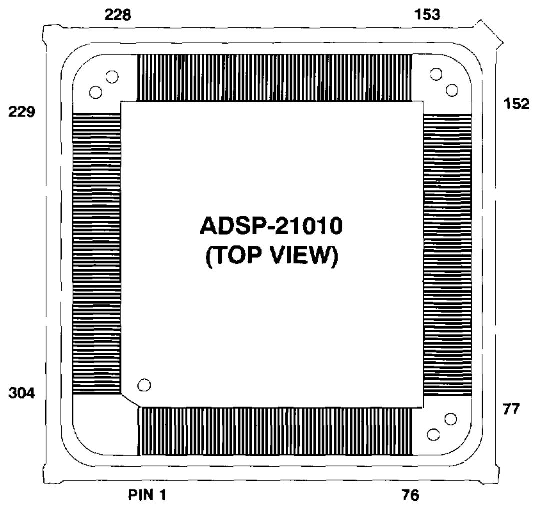


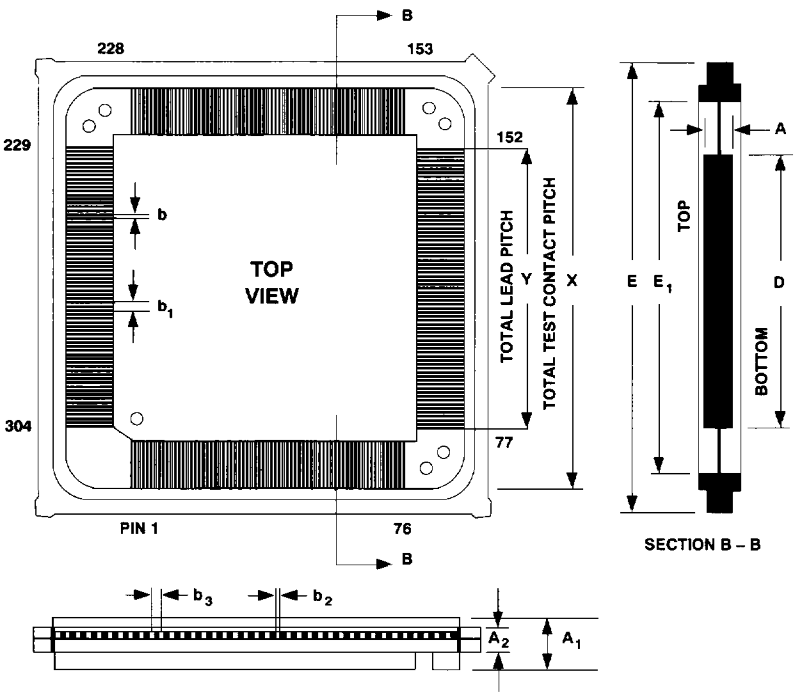
Figure 21. EZ-ICE Probe

PIN NUMBER	PIN NAME	PIN NUMBER	PIN NAME	PIN NUMBER	PIN NAME	PIN NUMBER	PIN NAME	PIN NUMBER	PIN NAME	PIN NUMBER	PIN NAME
1	IGND	52	PMD16	103	EVDD	154	PMA16	205	DMA0	256	DMPAGE
2	DMD11	53	PMD17	104	PMTS	155	EGND	206	DMA1	257	BR
3	DMD10	54	PMD18	105	PMWR	156	PMA15	207	DMA2	258	BG
4	DMD9	55	PMD19	106	PMACK	157	PMA14	208	DMA3	259	DMS0
5	DMD8	56	EVDD	107	PMRD	158	PMA13	209	EVDD	260	DMS1
6	DMD7	57	PMD20	108	RCOMP	159	PMA12	210	DMA4	261	EGND
7	EGND	58	PMD21	109	NC	160	EVDD	211	DMA5	262	DMS2
8	DMD6	59	PMD22	110	NC	161	PMA11	212	DMA6	263	DMS3
9	DMD5	60	IGND	111	NC	162	PMA10	213	DMA7	264	NC
10	DMD4	61	IVDD	112	RESET	163	PMA9	214	EGND	265	IGND
11	DMD3	62	PMD23	113	IGND	164	PMA8	215	DMA8	266	IGND
12	DMD2	63	PMD24	114	IGND	165	EGND	216	DMA9	267	IGND
13	EVDD	64	EGND	115	IGND	166	PMA7	217	DMA10	268	IGND
14	DMD1	65	PMD25	116	IGND	167	PMA6	218	DMA11	269	NC
15	DMD0	66	PMD26	117	EGND	168	PMA5	219	EVDD	270	EVDD
16	NC	67	PMD27	118	CLKIN	169	PMA4	220	DMA12	271	DMD31
17	NC	68	PMD28	119	DMRD	170	EVDD	221	DMA13	272	DMD30
18	NC	69	PMD29	120	DMACK	171	PMA3	222	DMA14	273	DMD29
19	EGND	70	EVDD	121	DMWR	172	PMA2	223	DMA15	274	DMD28
20	NC	71	PMD30	122	DMTS	173	IGND	224	EGND	275	DMD27
21	NC	72	PMD31	123	TCK	174	IVDD	225	DMA16	276	EGND
22	IGND	73	PMD32	124	TMS	175	PMA1	226	DMA17	277	DMD26
23	IVDD	74	PMD33	125	TDI	176	PMA0	227	DMA18	278	DMD25
24	NC	75	PMD34	126	TDO	177	EGND	228	IGND	279	DMD24
25	NC	76	IGND	127	EVDD	178	TIMEXP	229	IGND	280	DMD23
26	NC	77	IGND	128	TRST	179	NC	230	IGND	281	DMD22
27	NC	78	IGND	129	NC	180	NC	231	IGND	282	EVDD
28	EVDD	79	IGND	130	NC	181	NC	232	IGND	283	DMD21
29	PMD0	80	IGND	131	PMPAGE	182	EVDD	233	IGND	284	DMD20
30	PMD1	81	IGND	132	PMS0	183	NC	234	IGND	285	IVDD
31	PMD2	82	IGND	133	PMS1	184	NC	235	IGND	286	IGND
32	PMD3	83	IGND	134	EGND	185	NC	236	IGND	287	DMD19
33	PMD4	84	IGND	135	PMA23	186	NC	237	DMA19	288	DMD18
34	EGND	85	EGND	136	PMA22	187	EGND	238	EVDD	289	DMD17
35	PMD5	86	PMD35	137	IGND	188	NC	239	DMA20	290	EGND
36	PMD6	87	PMD36	138	IVDD	189	IGND	240	DMA21	291	DMD16
37	IGND	88	PMD37	139	PMA21	190	IGND	241	DMA22	292	DMD15
38	IGND	89	IGND	140	PMA20	191	IGND	242	DMA23	293	DMD14
39	IGND	90	IVDD	141	EVDD	192	IGND	243	EGND	294	DMD13
40	IGND	91	PMD38	142	PMA19	193	IRQ3	244	DMA24	295	DMD12
41	PMD7	92	PMD39	143	PMA18	194	IRQ2	245	DMA25	296	EVDD
42	PMD8	93	EVDD	144	PMA17	195	IRQ1	246	DMA26	297	IGND
43	PMD9	94	PMD40	145	IGND	196	IRQ0	247	DMA27	298	IGND
44	EVDD	95	PMD41	146	IGND	197	EVDD	248	EVDD	299	IGND
45	PMD10	96	PMD42	147	IGND	198	FLAG0	249	DMA28	300	IGND
46	PMD11	97	PMD43	148	IGND	199	FLAG1	250	DMA29	301	IGND
47	PMD12	98	PMD44	149	IGND	200	IGND	251	IGND	302	IGND
48	PMD13	99	EGND	150	IGND	201	IVDD	252	IVDD	303	IGND
49	PMD14	100	PMD45	151	IGND	202	FLAG2	253	DMA30	304	IGND
50	EGND	101	PMD46	152	IGND	203	FLAG3	254	DMA31		
51	PMD15	102	PMD47	153	IGND	204	EGND	255	EGND		

NC = NO CONNECT


228
153
229
152
ADSP-21010
(TOP VIEW)
304
77
PIN 1
76

304-Lead Plastic Quad Flatpack (PQFP) with Carrier Ring



		MILLIMETERS	TOLERANCE
BODY DIMENSION	D	40.00	±0.050
	A	3.80	±0.050
	Y	37.50	±0.050
RING DIMENSION	E	56.00	±0.050
	E_1	48.00	±0.050
	X	51.50	±0.050
	A_1	4.80	±0.10
	A_2	2.00	±0.10
LEAD WIDTH	b	0.23	±0.050
LEAD PITCH	b_1	0.50	±0.050
TEST CONTACT WIDTH	b_2	0.45	±0.050
TEST CONTACT PITCH	b_3	0.65	±0.050

ORDERING GUIDE

Part Number*	Ambient Temperature Range	Instruction Rate (MHz)	Cycle Time (ns)	Package**
ADSP-21010KS-50	0°C to +70°C	12.5	80	304-Lead Plastic Quad Flatpack
ADSP-21010BS-50	−40°C to +85°C	12.5	80	304-Lead Plastic Quad Flatpack

*S = Plastic Quad Flatpack.

**Because the ADSP-21010 comes in a PQFP package with carrier ring attached, carrier ring excise and lead trim/form is necessary. Contact Analog Devices DSP Marketing at (617) 461-3881 for information on contract manufacturing firms that provide these services.