January 1991

ADSP-21020
IEEE Floating-Point DSP Microprocessor

FEATURES

Off-Chip Harvard Architecture Maximizes Signal Processing Performance

Performance:
- 60 MFLOPS Peak, 40 MFLOPS Sustained, Superscalar IEEE Floating-Point Processor
- 50 ns, 20 MIPS Instruction Rate, Single-Cycle Execution
- 1024-point Complex FFT Benchmarks Under 1 ms
- Divide (y/x): 400 ns
- Inverse Square Root Computation ($1/\sqrt{x}$): 650 ns

Flexible Data Formats & Extended Precision:
- 32-Bit and 40-Bit IEEE Floating-Point Formats
- 32-Bit Fixed-Point Formats, Integer and Fractional, with 80-Bit Accumulation

Parallel Arithmetic Instruction Set:
- Off-Chip Dual Memory Reads & Writes in Parallel with Single-Cycle Multiply and ALU Operations and Instruction Fetch
- Three Independent Computation Units: Multiplier, ALU, and Shifter
- Multiply with Add & Subtract for FFT Butterfly Computation
- MAX, MIN, Average, Compare, for Spectral Peak Extraction
- Bit Manipulation and Count Leading 1's and 0's for Control

IEEE Exception Handling:
- Interrupt on Arithmetic Exception, Single-Cycle Interrupt Response
- Sticky (Latched) Arithmetic Status

Efficient Program Sequencing:
- Zero Overhead Looping: Single-Cycle Loop Setup & Exit
- Loops are Interruptable and Nestable
- Single-Cycle Context Switch of Data Register File and Address Registers

Flexible Data Address Generation:
- Two Independent Address Generators
- Addressing Modes Include: Indirect (pre- and post-modify), Immediate, Modulo and Bit-Reversed Addressing both with Unconstrained Buffer Placement

External Interface Features:
- 35 ns External RAM Access Time for Zero-Wait-State, 50 ns Instruction Execution
- Programmable Wait-States, External Acknowledge Memory Control
- Page-Mode DRAM Addressing Support
- JTAG Test and Emulation Support
- 223-Pin PGA Package (Plastic and Ceramic)

ANALOG DEVICES

This information applies to a product under development. Its characteristics and specifications are subject to change without notice. Analog Devices assumes no obligation regarding future manufacturing unless otherwise agreed to in writing.

GENERAL DESCRIPTION

The ADSP-21020 is the first member of Analog Devices' family of single-chip, programmable, IEEE floating-point processors optimized for digital signal processing applications. Its architecture is similar to that of Analog Devices' ADSP-2100 family of fixed-point DSP processsors.

The ADSP-21020 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 single-precision (32-bit) floating-point, extended 40-bit floating-point and 32-bit fixed point data formats.

- Single-cycle fetch of instruction and two operands

  The ADSP-21020 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 its high-performance instruction cache, the processor can fetch an operand from data memory, an operand from program memory, and an instruction from the cache simultaneously.

- Hardware circular buffers

  The ADSP-21020 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-21020's 48-bit instruction word accommodates a variety of parallel operations, for concise programming. For example, the ADSP-21020 can conditionally execute a computation, a data memory access and a branch in a single instruction.

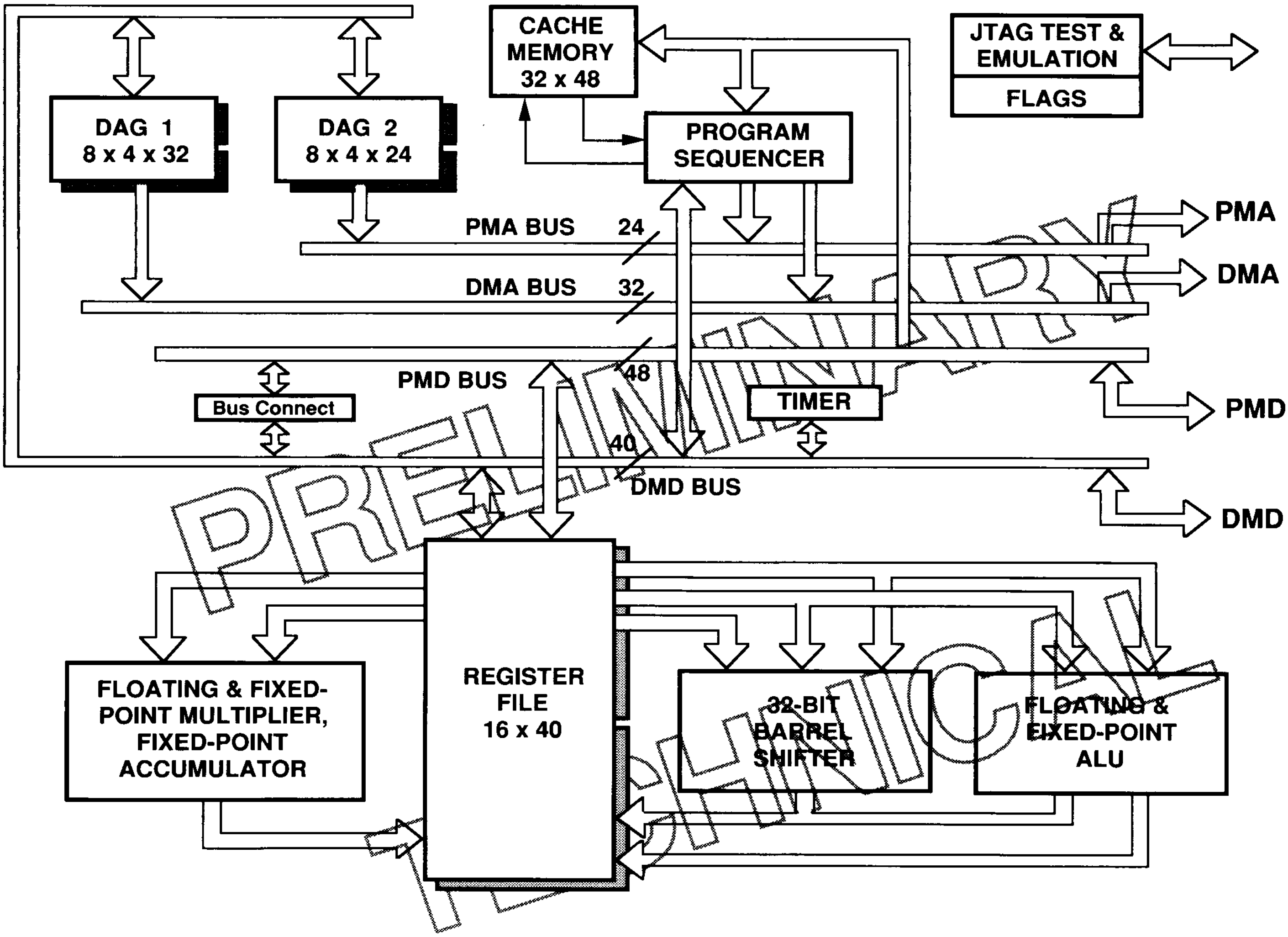


Figure 1 ADSP-21020 Block Diagram

ARCHITECTURE OVERVIEW

Figure 1 is a block diagram of the ADSP-21020. The processor features:

- Three computation units—ALU, multiplier and shifter—with a shared data register file
- Two address generators
- Program sequencer with instruction cache
- Timer
- Memory Buses and Interface
- JTAG Test and Emulation Support

Computation Units

The ADSP-21020 contains three independent computation units: an ALU, a multiplier with fixed-point accumulator, and a shifter. For meeting a wide variety of processing needs, the computation units process data in three formats: 32-bit fixed-point, 32-bit floating-point and 40-bit floating-point. The floating-point operations are single-precision IEEE-compatible (IEEE Standard 754/854). The 32-bit floating-point format is the standard IEEE format, whereas the 40-bit IEEE extended-precision format has eight more LSBs of mantissa for additional accuracy.

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 operations, including logical and arithmetic shifts, bit manipulation, field deposit, and extract and derive exponent operations, on 32-bit operands.

The computation units perform single-cycle operations; there is *no* computation pipeline. The units are connected in parallel rather than serially. The output of any unit may be the input of any unit on the next cycle.

In a *multifunction* computation, the ALU and multiplier perform independent simultaneous operations. A 10-port 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 40-bit registers each, for fast context switching. The primary or alternate set of each half of the register file (top eight or bottom eight registers) is selected independently.

Address Generators and Program Sequencer

Two dedicated address generators and a program sequencer supply addresses for memory accesses. Thus the computation units never need to be used to calculate addresses. Because of its instruction cache, the ADSP-21020 can simultaneously fetch an instruction and access data in both off-chip program memory and off-chip data memory in a single cycle. If the instruction is in the cache, there is no need to halt or wait for data.

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. DAG1 supplies 32-bit addresses to data memory. DAG2 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 length values and eight base values. A pointer used for indirect addressing can be modified by a value in a specified register, either before (pre-modify) or after (post-modify) the access. To implement automatic modulo addressing for circular buffers, the ADSP-21020 provides length values that can be associated with each pointer. Base values for pointers allow relocatable data storage 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 the program memory. It controls loop iterations and evaluates conditional instructions. To execute looped code with zero overhead, the ADSP-21020 maintains an internal loop counter and loop stack. No explicit jump instructions are required to loop or to decrement and test the counter.

The ADSP-21020 derives its high clock rate from pipelined *fetch, decode* and *execute* cycles. External memories have more time to complete an access than if there were no decode cycle; consequently, ADSP-21020 systems can be built using slower and therefore less expensive memories.

The program sequencer includes a high-performance instruction cache. This 2-way, set associative cache holds 32 instructions. Only the instructions whose fetches conflict with program memory data accesses are cached, so the ADSP-21020 can perform a program memory data access and execute the corresponding instruction in the same cycle. The program sequencer fetches the instruction from the cache instead of program memory, and the ADSP-21020 can simultaneously access data in program memory.

Interrupts

The ADSP-21020 has four external hardware interrupts, nine internally generated interrupts and eight software interrupts. For the external user interrupts and the internal timer interrupt, the ADSP-21020 automatically stacks the arithmetic status and mode (MODE1) registers in parallel with servicing the interrupt, allowing four nesting levels of very fast service for these interrupts.

An interrupt can occur at any time while the ADSP-21020 is executing a program. Internal events that generate interrupts include arithmetic exceptions, allowing 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-21020 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, on page 13, shows a basic system configuration with the ADSP-21020.

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 program memory and data memory addressing support page mode addressing of page-mode DRAMs.

The internal components are supported by four internal buses: the program memory address (PMA) and data memory address (DMA) buses are used for the addresses associated with program and data memory. The program memory data (PMD) and data memory data (DMD) buses are used for the data associated with the memory spaces. 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.

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 40-bit DMD bus or between the 40-bit register file and the PMD bus.

The program memory address (PMA) bus is 24 bits wide allowing direct access of up to 16M words of mixed instruction code and data. The program memory data (PMD) is 48 bits wide to accommodate the 48-bit instruction width.

The data memory address (DMA) bus is 32 bits wide allowing direct access of up to 4 Gwords of data. The data memory data (DMD) bus is 40 bits wide (for 32-bit data, the lower 8 bits are unused). 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 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 memory buses from the ADSP-21020 with bus request/grant signals ($\overline{\text{BR}}$ and $\overline{\text{BG}}$). To grant its buses in response to a bus request, the ADSP-21020 halts internal operations and places its program and data memory interfaces in a high-impedance state. In addition, three-state controls ($\overline{\text{DMTS}}$ and $\overline{\text{PMTS}}$) allow an external device to place either program or data memory interface in a high-impedance state without affecting the other interface and without halting the ADSP-21020 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-21020 off the bus while it updates an external cache memory.

Context Switching

Many of the ADSP-21020'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, and the others are *alternate* registers. Bits in a mode control register determine the registers that are active at any particular time.

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

Instruction Set

The ADSP-21020 instruction set provides a wide variety of programming capabilities. *Multifunction* instructions enable simultaneous multiplier and ALU operation, as well as computations in parallel with data transfers. The addressing power of the ADSP-21020 gives you flexibility in moving data both internally and externally. Every instruction can be executed in a single processor cycle. The ADSP-21020 assembly language uses an algebraic syntax for ease of coding and readability. A comprehensive set of development tools supports program development.

JTAG Test and Emulation Support

The ADSP-21020 implements the boundary scan testing provisions fully compliant to IEEE JTAG Standard 1149.1. This JTAG (Joint Testing Action Group) interface enables boundary scan testing of ICs mounted on a printed circuit board.

The ADSP-21020 also implements on-chip emulation through the JTAG 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 all the processor's internal registers and external memories through the JTAG port.

DEVELOPMENT SYSTEM

The ADSP-21020 is supported with a complete set of software and hardware development tools. The ADSP-21020 Development System includes Development Software for software design and Emulators for hardware debugging.

The Development Software includes:

- Assembler/Library

The assembler assembles the source code and data modules. In addition to supporting a full range of system diagnostics, the assembler provides flexible macro processing and modular code development. The library contains routines callable from assembly language programs; these routines implement standard DSP operations.

- Code Compactor

The code compactor searches an assembly language source code file looking for ways to combine sequential instructions into multi-function parallel instructions.

- Linker

The linker links separately assembled modules. It reads the user-defined architecture file, which specifies the allocations of program and data memory, the locations of memory-mapped I/O ports and the amounts of RAM and ROM, and maps the linked code and data output to the target system hardware.

- Simulator

The simulator performs an interactive, instruction-level simulation of ADSP-21020 code within the hardware configuration described by the system architecture file. It flags illegal operations and supports full symbolic assembly and disassembly.

- ANSI C Compiler/DSP/C™ Compiler/Library

The C Compiler supports ANSI Standard C and the ANSI Standard (X3J11.1) Numeric C Extensions as being defined by the Numeric C Extensions Group ("DSP/C™"). It inputs C language source and outputs ADSP-21020 source code ready to be assembled. It also supports inline assembler code. The library contains C callable routines for standard functions.

- PROM Splitter

This module reads the linker output and generates PROM-programmer compatible files.

- In-circuit Emulators

Two hardware in-circuit emulators for the ADSP-21020 provide a choice of emulation capabilities. The Full-Function Emulator allows full-speed in-

This information applies to a product under development. Its characteristics and specifications are subject to change without notice. Analog Devices assumes no obligation regarding future manufacturing unless otherwise agreed to in writing.

circuit emulation through a probe connected to a host platform. The Full-Function Emulator provides 256 channels of 256K-deep trace capability, enabling the user to trace instructions, data I/O, and peripheral interfaces at full speed. The Full-Function Emulator also implements single-chassis multi-processor emulation and a high-speed Ethernet interface.

The JTAG-Interface Emulator is a low-cost emulator that provides full-speed emulation through the 21020's JTAG port. The JTAG Interface Emulator allows non-intrusive in-circuit emulation (no additional load capacitance), with eight sets of breakpoint ranges supplied to the processor. The user can emulate at full-speed and read/write internal registers and external memories.

PIN DESCRIPTION

This section describes the pins of the ADSP-21020. 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 (Sync) must meet timing requirements with respect to CLKIN; those that are asynchronous (Async) can be asserted asynchronously to CLKIN.

Pin Name	*Type*	*Function*
PMA_{23-0}	Output	Program Memory Address. The ADSP-21020 outputs an address in program memory on these pins.
PMD_{47-0}	Bidirectional	Program Memory Data. The ADSP-21020 inputs and outputs data and instructions on these pins.
$\overline{PMS0}$	Output	Program Memory Select 0. This pin is asserted to select bank 0 of program memory. Memory banks are user-defined in memory control registers.
$\overline{PMS1}$	Output	Program Memory Select 1. This pin is asserted to select bank 1 of program memory. Memory banks are user-defined in memory control registers.
$\overline{PMRD}$	Output	Program Memory Read strobe. This pin is asserted when the ADSP-21020 reads program memory.
$\overline{PMWR}$	Output	Program Memory Write strobe. This pin is asserted when the ADSP-21020 writes program memory.

Pin Name	*Type*	*Function*
PMACK	Input/Sync	Program Memory Acknowledge. An external device asserts this ADSP-21020 input to terminate a memory access. Not asserting this pin is one method of creating wait states.
PMPAGE	Output	Program Memory Page Boundary. The ADSP-21020 asserts this pin to signal that a program memory page boundary has been crossed. Memory pages are user-defined in memory control registers.
$\overline{\text{PMTS}}$	Input/Sync	Program Memory Three-state Enable. Places program memory address, data and control signals in a high-impedance state.
DMA_{31-0}	Output	Data Memory Address. The ADSP-21020 outputs an address in data memory on these pins.
DMD_{39-0}	Bidirectional	Data Memory Data. The ADSP-21020 inputs and outputs data on these pins.
$\overline{\text{DMS0}}$	Output	Data Memory Select 0. This pin is asserted to select bank 0 of data memory. Memory banks are user-defined in memory control registers.
$\overline{\text{DMS1}}$	Output	Data Memory Select 1. This pin is asserted to select bank 1 of data memory. Memory banks are user-defined in memory control registers.
$\overline{\text{DMS2}}$	Output	Data Memory Select 2. This pin is asserted to select bank 2 of data memory. Memory banks are user-defined in memory control registers.
$\overline{\text{DMS3}}$	Output	Data Memory Select 3. This pin is asserted to select bank 3 of data memory. Memory banks are user-defined in memory control registers.
$\overline{\text{DMRD}}$	Output	Data Memory Read strobe. This pin is asserted when the ADSP-21020 reads data memory.

Pin Name	*Type*	*Function*
$\overline{\text{DMWR}}$	Output	Data Memory Write strobe. This pin is asserted when the ADSP-21020 writes data memory.
DMACK	Input/Sync	Data Memory Acknowledge. An external device asserts this ADSP-21020 input to terminate a memory access. This is one method of creating wait states.
DMPAGE	Output	Data Memory Page Boundary. The ADSP-21020 asserts this pin to signal that a data memory page boundary has been crossed. Memory pages are user-defined in memory control registers.
$\overline{\text{DMTS}}$	Input/Sync	Data Memory Three-state Enable. Places data memory address, data and control signals in a high-impedance state, without halting the processor.
CLKIN	Input	External clock to the ADSP-21020. The internal system clock has the same frequency as CLKIN.
$\overline{\text{IRQ3-0}}$	Input/Async	Interrupt request lines (4). May be either edge-triggered or level-sensitive.
$\overline{\text{RESET}}$	Input/Async	Sets chip to a known state and begins execution at the program memory location specified by the hardware reset vector. This input must be asserted (low) at powerup. Schmitt trigger input.
FLAG3-0	Bidirectional	External Flags (4). Each is configured via control bits as either an input or an output. As an input, it can be tested as a condition. As an output, it can be used to signal external peripherals.
$\overline{\text{BR}}$	Input/Async	Bus Request. Used by an external device to request control of the memory interface. When $\overline{\text{BR}}$ is asserted, the processor halts execution at the completion of the current cycle, tristates all memory data, addresses, selects, and strobes, and asserts $\overline{\text{BG}}$. The processor continues normal operation when $\overline{\text{BR}}$ is released.

Pin Name	*Type*	*Function*
$\overline{BG}$	Output	Bus Grant. Acknowledges a bus request ($\overline{BR}$), indicating that the external device may take control of the memory interface. $\overline{BG}$ is held asserted until $\overline{BR}$ is released.
TIMEXP	Output	Timer Expired. Asserted for 4 cycles when the value of TCOUNT is decremented to zero.
RCOMP	Input	Compensation Resistor input. Controls compensated output buffers. Connect RCOMP through a 2.2 kΩ resistor to +5 V.
EVDD	Supply	Power supply (for output drivers), nominally +5 VDC; 10 pins.
EGND	Ground	Power supply return (for output drivers); 16 pins.
IVDD	Supply	Power supply (for internal circuitry), nominally +5 VDC; 4 pins.
IGND	Ground	Power supply return (for internal circuitry); 7 pins.
TCK	Input	Test Clock provides an asynchronous clock for the boundary scan.
TMS	Input/Sync	Test Mode Select is used to control the test state machine. TMS has a 20 kΩ internal pullup resistor.
TDI	Input/Sync	Test Data Input provides serial data for the boundary scan logic. TDI has a 20 kΩ internal pullup resistor.
TDO	Output	Test Data Output acts as the serial scan output of the boundary scan path.
$\overline{TRST}$	Input/Async	Test Reset resets the test state machine. $\overline{TRST}$ has a 20 kΩ internal pullup resistor.

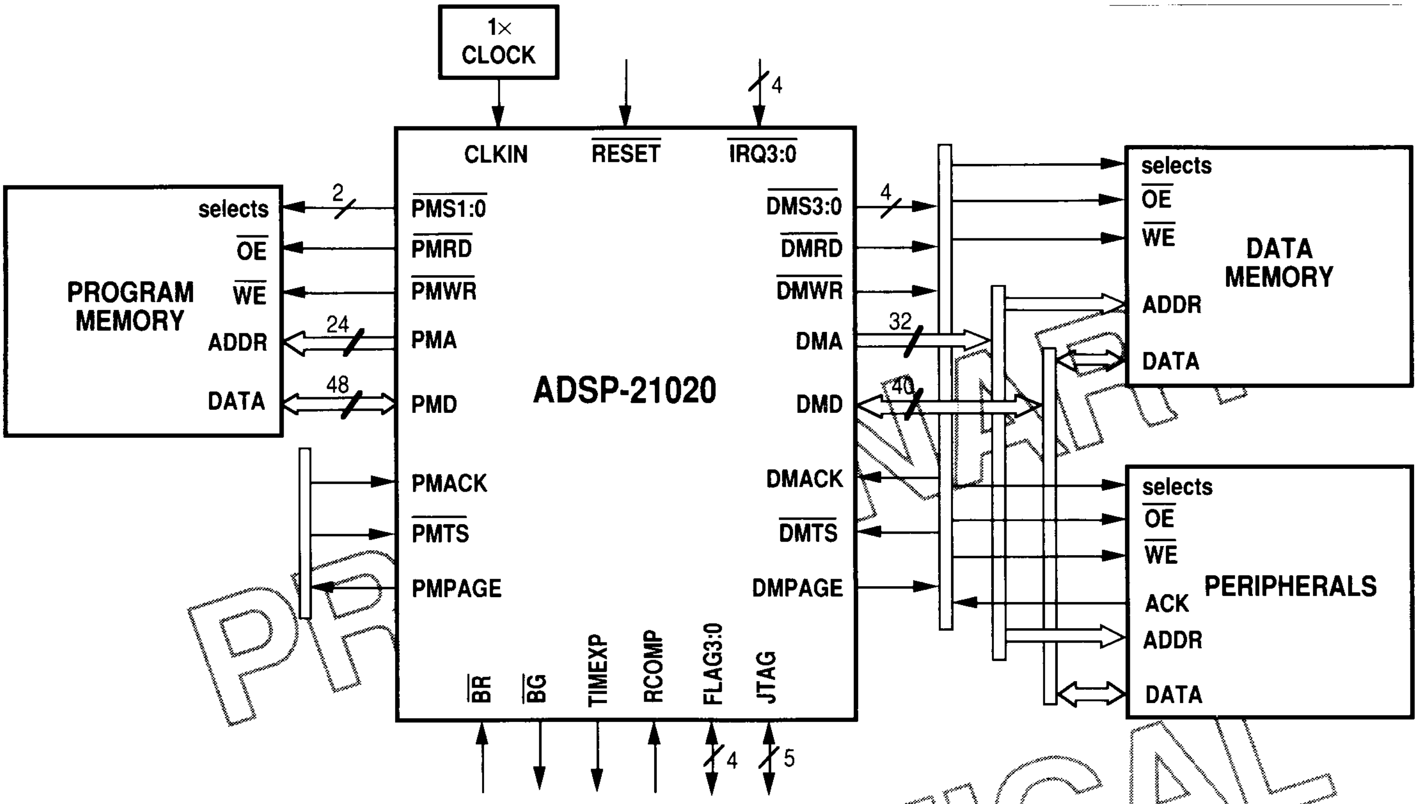


Figure 2 ADSP-21020 Basic System Configuration

INSTRUCTION SET SUMMARY

This section overviews the ADSP-21020 instruction set. Tables I to VIII below provide a reference for using the instruction set. For more information, see the *ADSP-21020 User's Manual.*

The instruction types are grouped into four categories:

I. Compute and Move or Modify
II. Program Flow Control
III. Immediate Move
IV. 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 has no bearing on programming, but corresponds to the opcode recognized by the ADSP-21020 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 operations can be the compute portion of an instruction.

I. Compute and Move or Modify

1. *compute,* | DM(Ia, Mb) = dreg1 / dreg1 = DM(Ia, Mb) | , | PM(Ic, Md) = dreg2 / dreg2 = PM(Ic, Md) | ;

2. *IF condition* compute ;

3.
   a. *IF condition* *compute,* | DM(Ia, Mb) / PM(Ic, Md) | = ureg ;
   b. *IF condition* *compute,* | DM(Mb, Ia) / PM(Md, Ic) | = ureg ;
   c. *IF condition* *compute,* ureg = | DM(Ia, Mb) / PM(Ic, Md) | ;
   d. *IF condition* *compute,* ureg = | DM(Mb, Ia) / PM(Md, Ic) | ;

4.
   a. *IF condition* *compute,* | DM(Ia, <data6>) / PM(Ic, <data6>) | = dreg ;
   b. *IF condition* *compute,* | DM(<data6>, Ia) / PM(<data6>, Ic) | = dreg ;
   c. *IF condition* *compute,* dreg = | DM(Ia, <data6>) / PM(Ic, <data6>) | ;
   d. *IF condition* *compute,* dreg = | DM(<data6>, Ia) / PM(<data6>, Ic) | ;

5. *IF condition* *compute,* ureg1 = ureg2 ;

6.
   a. *IF condition* shiftimm , | *DM(Ia, Mb) / PM(Ic, Md)* | *= dreg ;*
   b. *IF condition* shiftimm , *dreg =* | *DM(Ia, Mb) / PM(Ic, Md)* | *;*

7. *IF condition* *compute,* MODIFY | (Ia, Mb) / (Ic, Md) | ;

II. Program Flow Control

8.	*IF condition*	JUMP	<addr24>	*(*	*DB*	*) ;*	
		CALL	(PC, <reladdr24>)		*LA*		
					DB, LA		

9.	*IF condition*	JUMP	(Md, Ic)	*(*	*DB*	*)*	*, compute ;*
		CALL	(PC, <reladdr6>)		*LA*		
					DB, LA		

10.	*IF condition*	JUMP	(Md, Ic)	*, compute*	, DM(Ia, Mb) = dreg	;
			(PC, <reladdr6>)		dreg = DM(Ia, Mb)	

11.	*IF condition*	RTS	*(*	*DB*	*)*	*, compute ;*
		RTI		*LA*		
				DB, LA		

12.	LCNTR =	<data16>	, DO	<addr24>	UNTIL LCE ;
		ureg		(<PC, reladdr24>)	

13.	DO	<addr24>	UNTIL termination ;
		(PC, <reladdr24>)	

III. Immediate Move

14. a.
DM(<addr32>) = ureg ;
PM(<addr24>)

b.
ureg = DM(<addr32>) ;
PM(<addr24>)

15. a.
DM(<data32>, Ia) = ureg ;
PM(<data24>, Ic)

b.
ureg = DM(<data32>, Ia) ;
PM(<data24>, Ic)

16.
DM(Ia, Mb) = <data32> ;
PM(Ic, Md)

17.
ureg = <data32> ;

IV. Miscellaneous

18.
BIT SET sreg <data32> ;
CLR
TGL
TST
XOR

19. a.
MODIFY (Ia, <data32>) ;
(Ic, <data32>)

b.
BITREV (Ia, <data32>) ;

20.
PUSH LOOP , *PUSH STS* ;
POP *POP*

21.
NOP ;

22.
IDLE ;

Notation	*Meaning*
UPPERCASE	explicit syntax; assembler keyword
;	instruction terminator
,	separates parallel operations in an instruction
italics	optional part of instruction
| between lines |	list of options (choose one)
<data*n*>	*n*-bit immediate data value
<addr*n*>	*n*-bit immediate address value
<reladdr*n*>	*n*-bit immediate PC-relative address value
<bit6>:<len6>	6-bit immediate bit position and length values (for shifter immediate operations)
compute	ALU, multiplier, shifter or multifunction operation (from Tables 4-7)
shiftimm	shifter immediate operation (from Table 6)
condition	status condition (from Table 2)
termination	termination condition (from Table 2)
ureg	universal register (from Table 3)
sreg	system register (from Table 3)
dreg	R15-R0, F15-F0; register file location
Rn, Rx, Ry, Ra, Rm, Rs	R15-R0; register file location, fixed-point
Fn, Fx, Fy, Fa, Fm, Fs	F15-F0; register file location, floating-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
F3-0	F3, F2, F1, F0
F7-4	F7, F6, F5, F4
F11-8	F11, F10, F9, F8
F15-12	F15, F14, F13, F12
Ia	I7-I0; DAG1 index register
Mb	M7-M0; DAG1 modify register
Ic	I15-I8; DAG2 index register
Md	M15-M8; DAG2 modify register
(DB)	Delayed branch
(LA)	Loop abort (pop loop, PC stacks on branch)
MR0F	Multiplier result accumulator 0, foreground
MR1F	Multiplier result accumulator 1, foreground
MR2F	Multiplier result accumulator 2, foreground
MR0B	Multiplier result accumulator 0, background
MR1B	Multiplier result accumulator 1, background
MR2B	Multiplier result accumulator 2, background

Table 1 Syntax Notation Conventions

Name	*Description*
EQ	ALU equal zero
NE	ALU not equal to zero
GE	ALU greater than or equal zero
LT	ALU less than zero
LE	ALU less than or equal 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 2 Condition and Termination Codes

Mnemonic	*Contents*
PC*	program counter
PCSTK	top of PC stack
PCSTKP	PC stack pointer
FADDR*	fetch address
DADDR*	decode address
LADDR	top of loop address stack
CURLCNTR	top of loop count stack
LCNTR	loop count for next loop
R15 - R0	register file locations (fixed-point data)
F15 - F0	register file locations (floating-point data)
I15 - I8	DAG2 index registers
I7 - I0	DAG1 index registers
M15 - M8	DAG2 modify registers
M7 - M0	DAG1 modify registers
L15 - L8	DAG2 length registers
L7 - L0	DAG1 length registers
B15 - B8	DAG2 base registers
B7 - B0	DAG1 base registers
DMWAIT	wait state and page size control for data memory
DMBANK1	data memory bank 1 lower boundary
DMBANK2	data memory bank 2 lower boundary
DMBANK3	data memory bank 3 lower boundary
DMADR	copy of last data memory address
PMWAIT	wait state and page size control for program memory
PMBANK1	program memory bank 1 lower boundary
PMADR	copy of last program memory address
PX	48-bit PX1 and PX2 combination
PX1	bus exchange 1 (16 bits)
PX2	bus exchange 2 (32 bits)
TPERIOD	timer period
TCOUNT	timer counter
System Registers (these are also universal registers):	
MODE1	mode control 1
MODE2	mode control 2
IRPTL	interrupt latch
IMASK	interrupt mask
IMASKP	interrupt mask pointer
ASTAT	arithmetic status
STKY	sticky status
USTAT1	user status reg 1
USTAT2	user status reg 2

* read-only

Table 3 Universal Registers and System Registers

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

Table 4 ALU Instructions

Rn MRF MRB	= Rx * Ry *(mod2**)*	Fn = Fx * Fy

Rn = MRF Rn = MRB MRF = MRF MRB = MRB	+ Rx * Ry *(mod2**)*		Rn = MRF Rn = MRB MRF = MRF MRB = MRB	– Rx * Ry *(mod2**)*	

Rn = SAT MRF Rn = SAT MRB MRF = SAT MRF MRB = SAT MRB	*(mod1*)*	Rn = RND MRF Rn = RND MRB MRF = RND MRF MRB = RND MRB	*(mod1*)*

MRF MRB	= 0

MRxF MRxB	= Rn	Rn =	MRxF MRxB

MRxF = MR2F, MR1F, MR0F
MRxB = MR2B, MR1B, MR0B

** optional modifier mod1:*

SI	Signed, Integer input (for SAT only)
UI	Unsigned, Integer input (for SAT only)
SF	Signed, Fractional input (DEFAULT)
UF	Unsigned, Fractional input

*** optional modifier mod2:*

SSI	X-input signed, Y-input signed, Integer inputs
SUI	X-input signed, Y-input unsigned, Integer inputs
USI	X-input unsigned, Y-input signed, Integer inputs
UUI	X-input unsigned, Y-input unsigned, Integer inputs
SSF	X-input signed, Y-input signed, Fractional inputs (DEFAULT)
SUF	X-input signed, Y-input unsigned, Fractional inputs
USF	X-input unsigned, Y-input signed, Fractional inputs
UUF	X-input unsigned, Y-input unsigned, Fractional inputs
SSFR	X-input signed, Y-input signed, Fractional inputs, Rounded output
SUFR	X-input signed, Y-input unsigned, Fractional inputs, Rounded output
USFR	X-input unsigned, Y-input signed, Fractional inputs, Rounded output
UUFR	X-input unsigned, Y-input unsigned, Fractional inputs, Rounded output

Table 5 Multiplier Instructions

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	

Table 6 Shifter and Shifter Immediate Instructions

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, Fs	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 7 Multifunction Instructions

No.	*Vector*	*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	Circular buffer 7 overflow
12	0x60	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	Reserved	
24-31	0xC0-F1	User software interrupts

* Nonmaskable

Table 8 Interrupt Vectors and Priority

SPECIFICATIONS

Recommended Operating Conditions

Parameter		K Grade (Commercial) Min	Max	Unit	
V_{DD}	Supply Voltage	4.5	5.5	V	
$T_{AMBIENT}$	Ambient Operating Temperature	0	+70	°C	(Plastic Pin Grid Array)

Refer to Package Thermal Specifications for information on case temperature and thermal specifications.

Electrical Characteristics

Parameter		Test Conditions	Min	Typ	Max	Unit
V_{IH}	Hi-Level Input Voltage	V_{DD}=max	2.0			V
V_{IHC}	Hi-Level CLKIN Voltage	V_{DD}=max	3.0			V
V_{IL}	Lo-Level Input Voltage	V_{DD}=min			0.8	V
V_{OH}	Hi-Level Output Voltage	V_{DD}=min, I_{OH}=–.5 mA	2.4			V
V_{OL}	Lo-Level Output Voltage	V_{DD}=min, I_{OL}=2 mA			0.4	V
I_{IH}	Hi-Level Input Current	V_{DD}=max, V_{IN}=max			10	µA
I_{IL}	Lo-Level Input Current	V_{DD}=max, V_{IN}=0 V			10	µA
I_{OH}	Hi-Level Output Current	$V_{OUT}=V_{DD}$ – 0.8 V			–4.0	mA
I_{OL}	Lo-Level Output Current	V_{OUT}=0.4 V			4.0	mA
I_{OZH}	Tristate Leakage Current	V_{DD}=max, V_{IN}=max			10	µA
I_{OZL}	Tristate Leakage Current	V_{DD}=max, V_{IN}=0 V			10	µA
I_{DDTYP}	Typical Current	V_{DD}=5 V, capacitive loads, t_{CK}(max)		400		mA
I_{DDIN*}	Supply Current, Internal	t_{CK}(max), V_{IHC}=3.0 V, all other V_{IH}=2.0 V, V_{IL}=0.8 V			TBD	mA
I_{DDEX*}	Supply Current, External	t_{CK}(max), V_{IHC}=3.0 V, all other V_{IH}=2.0 V, V_{IL}=0.8 V			TBD	µA/(pF•MHz)

* Current consumption is specified in two parts: I_{DDIN} is the internal current under no load at worst case frequency, V_{DD}, T_{AMB} and internal logic state (vectors). I_{DDEX} is the normalized external drive current.

I_{DDEX} = (pF/pin) (# pins switching/cycle) (# cycles/sec) (TBD µA/(pF•MHz)) = (TBD) mA

Total $I_{DD} = I_{DDIN} + I_{DDEX}$

I_{DDEX} is a maximum when every output switches on every cycle. Typical I_{DDEX} depends on the data output and duty cycle of the device in your application.

ESD Sensitivity

The ADSP-21020 features proprietary input protection circuitry. Per Method 3015.6 of MIL-STD-883, the ADSP-21020 has been classified as a Class (TBD) 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.*

Absolute Maximum Ratings*

Supply Voltage	–0.3V to +7V
Input Voltage	–0.3V to V_{DD}
Output Voltage Swing	–0.3V to V_{DD}
Load Capacitance	200pF
Operating Temperature Range (Ambient)	–55°C to +125°C
Storage Temperature Range	–65°C to +150°C
Lead Temperature (10 seconds)	+300°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.

Timing Parameters

Clock Signals

Parameter		20 MHz (50 ns) Min	Max	15 MHz (66 ns) Min	Max	Unit
Timing Requirement						
t_{CK}	CLKIN period	50		66		ns
t_{CKH}	CLKIN high period	20		20		ns
t_{CKL}	CLKIN low period	20		20		ns
t_{CKR}	CLKIN rise and fall time (measured from 0.8 V to 2.0 V)		6		8	ns
t_{CKON}	CLKIN active after powerup		100		100	ms

x = PM or DM

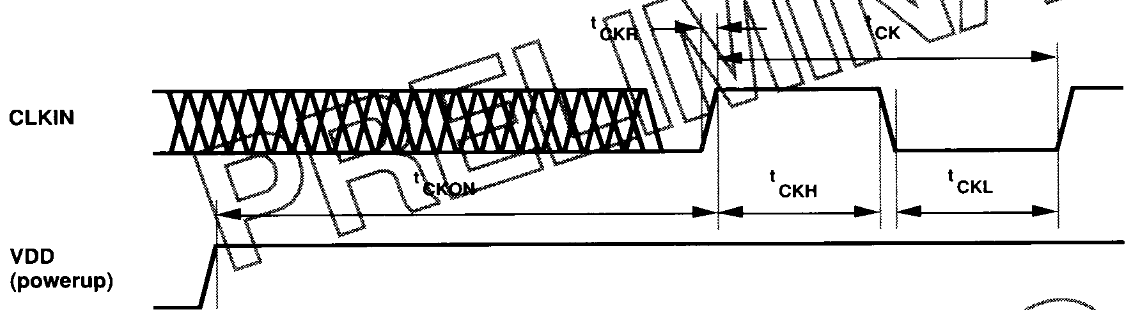


Figure 3 Clock Timing

Timer

Parameter		20 MHz (50 ns) Min	Max	15 MHz (66 ns) Min	Max	Frequency Dependency t_{CK} = T ns; DT=T–50 ns Min	Max	Unit
Switching Characteristic								
t_{DTEX}	CLKIN rising edge to TIMEXP	31		31		31		ns

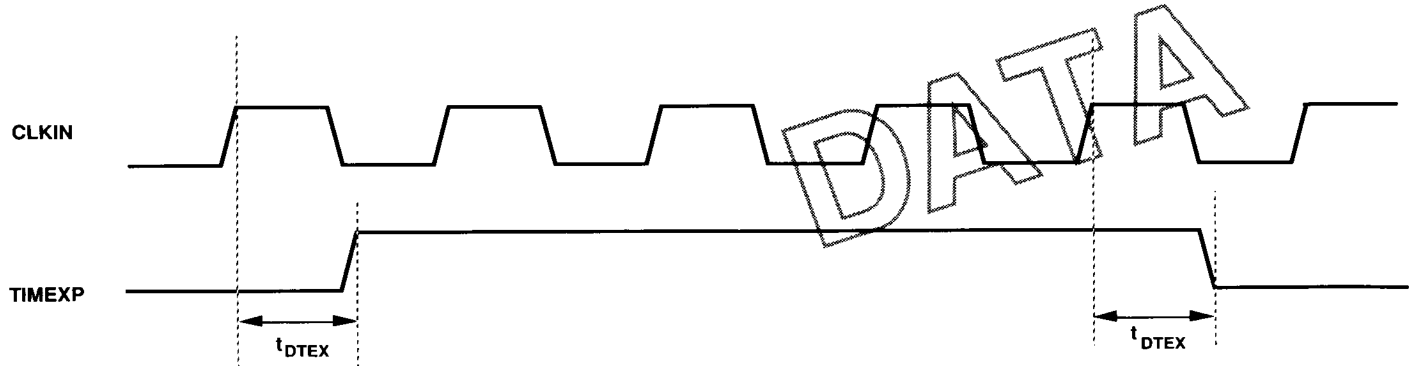
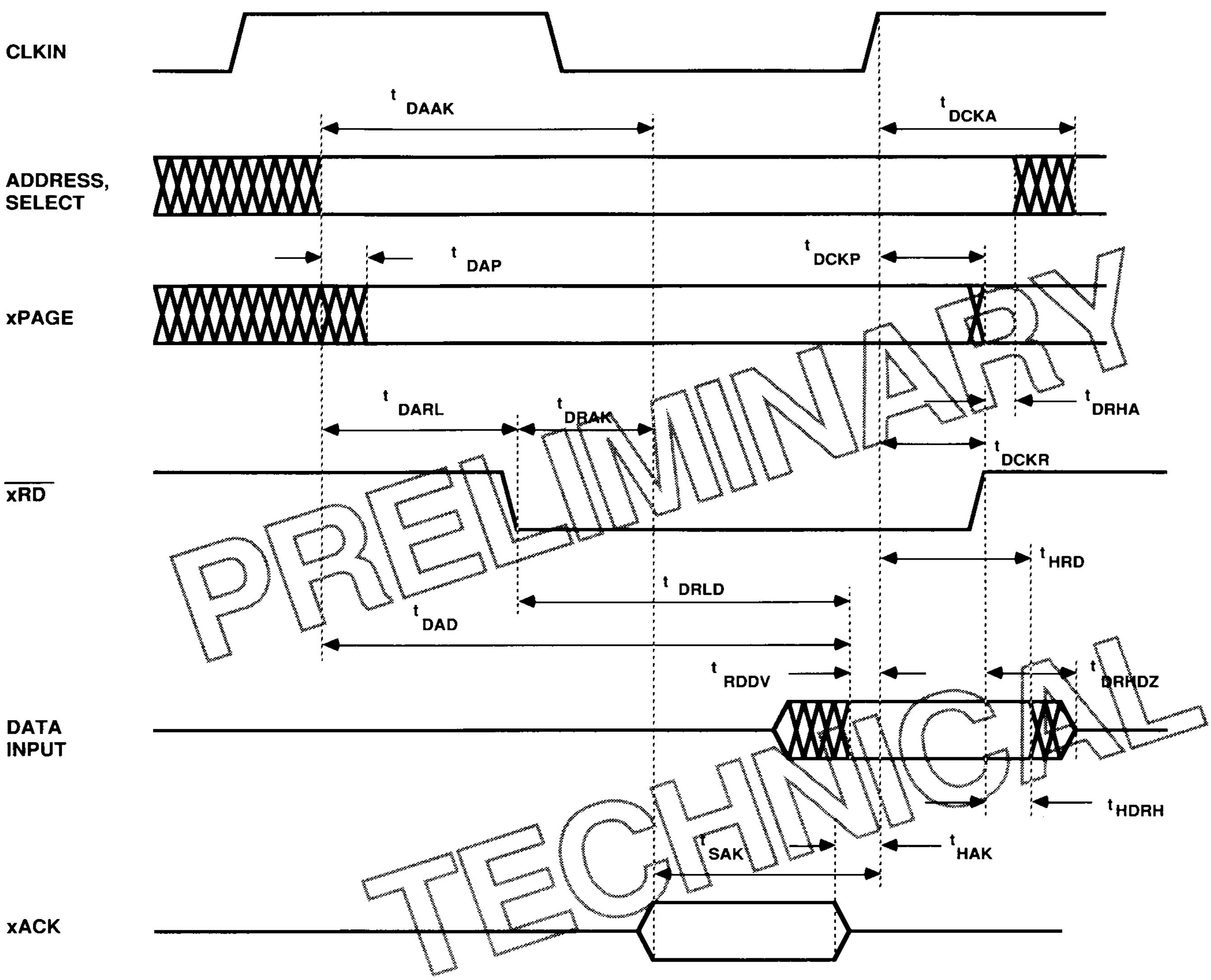


Figure 4 Timer Timing

Memory Read

Parameter		20 MHz (50 ns) Min	20 MHz (50 ns) Max	15 MHz (66 ns) Min	15 MHz (66 ns) Max	Frequency Dependency (t_{CK} = T ns; DT=T–50 ns) Min	Frequency Dependency Max	Unit
Switching Characteristic								
t_{DARL}	Address, Select valid to $\overline{xRD}$	11		17		11+3DT/8		ns
t_{DRHA}	$\overline{xRD}$ deasserted to Address, Select invalid	0		0		0		ns
t_{DCKA}	CLKIN rising edge to Address valid		18		16		18–DT/8	ns
t_{DAP}	Address valid to xPAGE valid		4		4		4	ns
t_{DCKP}	CLKIN rising edge to xPAGE valid		21		19		21–DT/8	ns
t_{DCKR}	CLKIN rising edge to $\overline{xRD}$ deasserted		11		9		11–DT/8	ns
Timing Requirement								
t_{RDDV}	Data valid to CLKIN rising edge	4		6		4+DT/8		ns
t_{HRD}	CLKIN rising edge to Data invalid	8		6		8–DT/8		ns
t_{DAD}	Address, Select valid to external Data valid		36		53		36+DT	ns
t_{DRLD}	$\overline{xRD}$ asserted to external Data valid		25		27		25+5DT/8	ns
t_{DRHDZ}	$\overline{xRD}$ deasserted to Data high impedance	0	15	0	22	0	15+7DT/16	ns
t_{HDRH}	$\overline{xRD}$ deasserted to Data invalid	0		0		0		ns
t_{DAAK}	Address valid to xACK valid		21		36		21+7DT/8	ns
t_{DRAK}	$\overline{xRD}$ valid to xACK valid		8		16		8+DT/2	ns
t_{SAK}	xACK enabled to CLKIN rising edge	11		7		11–DT/4		ns
t_{HAK}	xACK disabled to CLKIN rising edge		2		–2		2–DT/4	ns

x = PM or DM
Select = $\overline{PMS1\text{-}0}$, $\overline{DMS3\text{-}0}$



SELECT = PMS1-0, DMS3-0
x = PM or DM

Figure 5 Memory Read Timing

Memory Write

Parameter		20 MHz (50 ns) Min	20 MHz (50 ns) Max	15 MHz (66 ns) Min	15 MHz (66 ns) Max	Frequency Dependency (t_{CK} = T ns; DT=T–50 ns) Min	Frequency Dependency Max	Unit
Switching Characteristic								
t_{DAWH}	Address, Select valid to x$\overline{\text{WR}}$ deasserted	36		52		36+15DT/16		ns
t_{DAWL}	Address, Select valid to x$\overline{\text{WR}}$ asserted	10		16		10+3DT/8		ns
t_{WW}	x$\overline{\text{WR}}$ pulse width	25		34		25+9DT/16		ns
t_{DDWH}	Data valid to x$\overline{\text{WR}}$ deasserted	17		25		17+DT/2		ns
t_{WDDV}	CLKIN rising edge to Data valid		38		43		38+5DT/16	ns
t_{DDZL}	CLKIN rising edge to Data low impedance	18		23		18+5DT/16		ns
t_{DWHA}	x$\overline{\text{WR}}$ deasserted to Address, Select invalid	2		3		2+DT/16		ns
t_{DWHDZH}*	x$\overline{\text{WR}}$ deasserted to Data high impedance	4	15	5	16	4+DT/16	15+DT/16	ns
t_{DCKA}	CLKIN rising edge to Address valid		18		16		18–DT/8	ns
t_{DAP}	Address valid to xPAGE valid		4		4		4	ns
t_{DCKP}	CLKIN rising edge to xPAGE valid		21		19		21–DT/8	ns
t_{DCKW}	CLKIN rising edge to x$\overline{\text{WR}}$ deasserted		7		4		7–3DT/16	ns
Timing Requirement								
t_{DAAK}	Address valid to xACK valid		21		35		21+7DT/8	ns
t_{DWAK}	x$\overline{\text{WR}}$ valid to xACK valid		8		16		8+DT/2	ns
t_{SAK}	xACK enabled to CLKIN rising edge	11		15		11+DT/4		ns
t_{HAK}	xACK disabled to CLKIN rising edge		2		6		2+DT/4	ns

x = PM or DM
Select = $\overline{\text{PMS1-0}}$, $\overline{\text{DMS3-0}}$

* t_{DWHDZH} is the time from the rising edge of x$\overline{\text{WR}}$ to when the ADSP-21020 stops driving the data bus, until the next write or read cycle. In between the two memory accesses, the data output remains valid on the output bus for a time determined by the system's total bus capacitance and the total leakage current (see page 37 for this calculation).

This information applies to a product under development. Its characteristics and specifications are subject to change without notice. Analog Devices assumes no obligation regarding future manufacturing unless otherwise agreed to in writing.

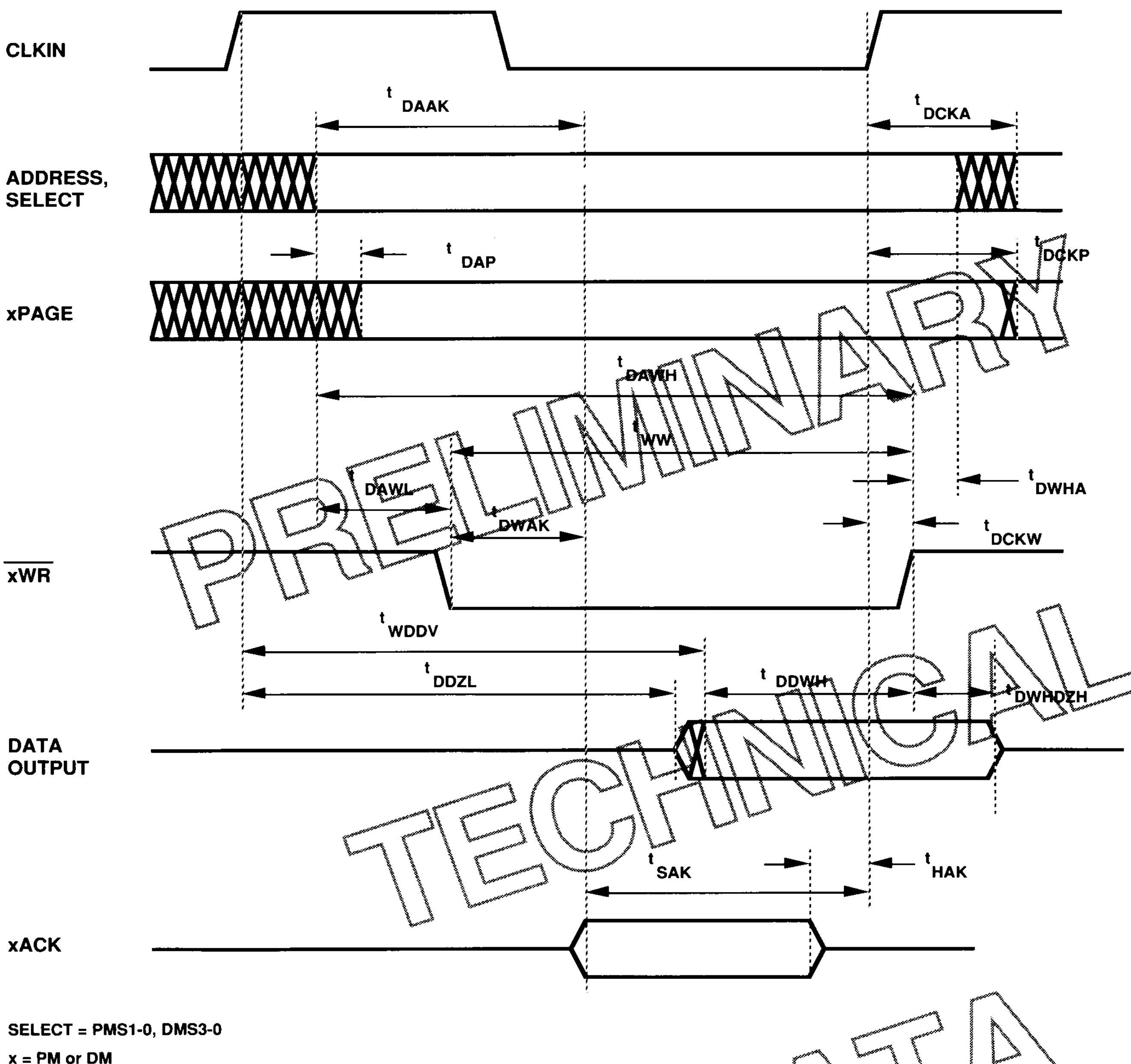


Figure 6 Memory Write Timing

Bus Request/Bus Grant

		20 MHz (50 ns)		15 MHz (66 ns)		Frequency Dependency t_{CK} = T ns; DT=T–50 ns		
Parameter		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
Switching Characteristic								
t_{DMZH}	CLKIN rising edge to memoryI/F HI-Z		29		32		29+3DT/16	ns
t_{DMZL}	Memory I/F LOW- to CLKIN rising edge		5		13		5+DT/2	ns
t_{DBGL}	CLKIN rising edge to $\overline{BG}$ asserted		31		31		31	ns
t_{DBGH}	CLKIN rising edge to $\overline{BG}$ deasserted		31		31		31	ns
Timing Requirement								
t_{HBR}	$\overline{BR}$ hold to CLKIN rising edge		4		9		4+5DT/16	ns
t_{SBR}	$\overline{BR}$ asserted or deasserted to CLKIN rising edge	13		18		13+5DT/16		ns

Bus request is not granted until completion of current external memory access.

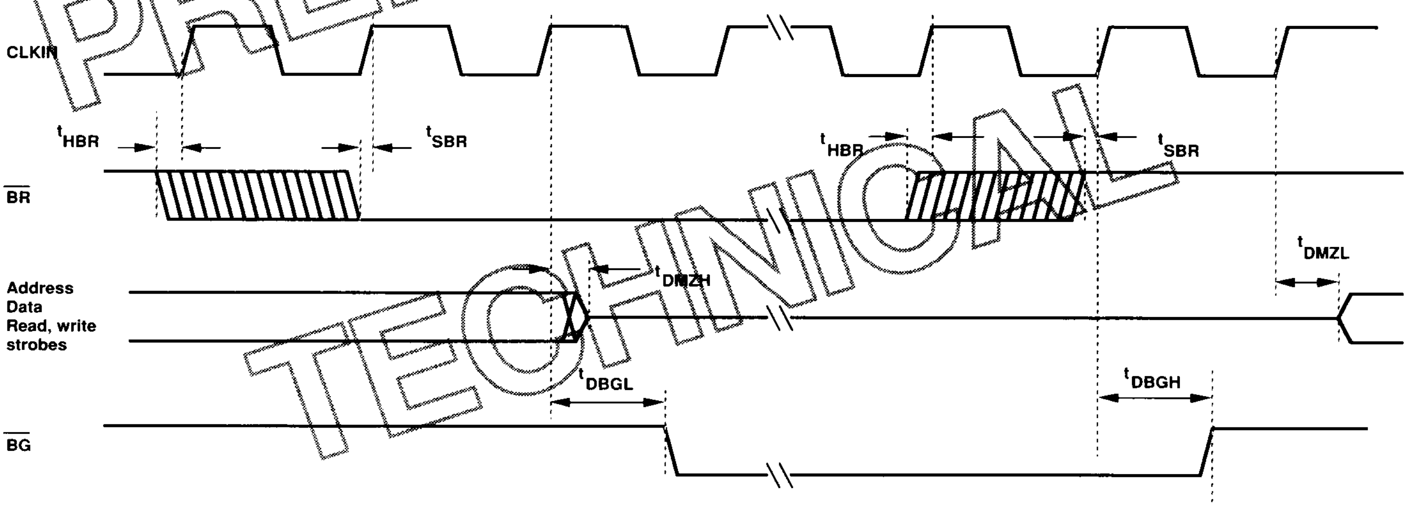


Figure 7 Bus Request/Bus Grant Timing

External Memory Three-State Control

Parameter		20 MHz (50 ns) Min	20 MHz (50 ns) Max	15 MHz (66 ns) Min	15 MHz (66 ns) Max	Frequency Dependency (t_{CK} = T ns; DT=T–50 ns) Min	Frequency Dependency Max	Unit
Switching Characteristic								
t_{DTSZH}	Memory I/F HI-Z before CLKIN rising edge	0		4		0+DT/4		ns
t_{DTSZL}	Address enabled to CLKIN rising edge	0		4		0+DT/4		ns
Timing Requirement								
t_{STS}	xTS asserted or deasserted to CLKIN rising edge	18		22		18+DT/4		ns

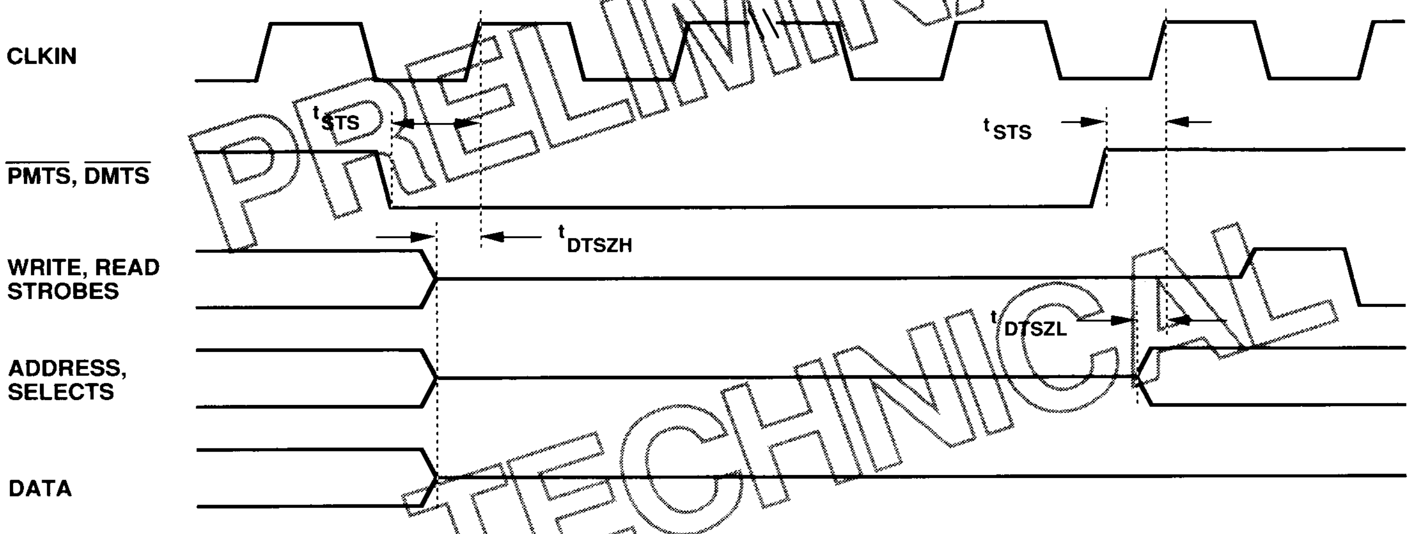


Figure 8 External Memory Three-State Control Timing

Reset

Parameter		20 MHz (50 ns) Min	20 MHz (50 ns) Max	15 MHz (66 ns) Min	15 MHz (66 ns) Max	Frequency Dependency (t_{CK} = T ns; DT=T–50 ns) Min	Frequency Dependency Max	Unit
Timing Requirement								
t_{WRST}	$\overline{\text{RESET}}$ pulse width	53		69		T+3		ns
t_{SRST}	$\overline{\text{RESET}}$ deasserted to CLKIN rising edge (for multiple ADSP-21020 synchronization only)	22		30		22+DT/2		ns
t_{DRST}	$\overline{\text{RESET}}$ asserted to $\overline{\text{PMRD}}$ deasserted	15		15		15		ns
t_{SCKRH}	CLKIN stable to $\overline{\text{RESET}}$ deasserted (on power up)	100000		134000		2000T		ns

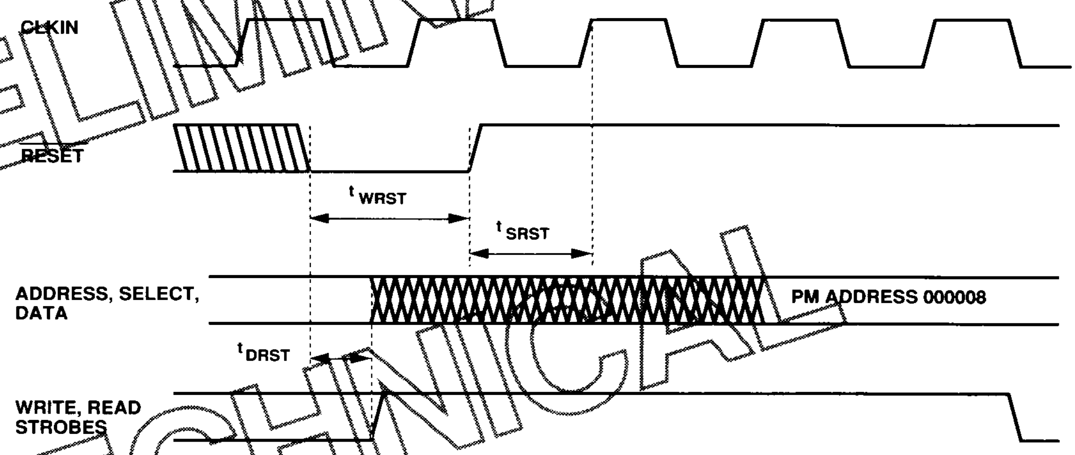


Figure 9 Reset Timing

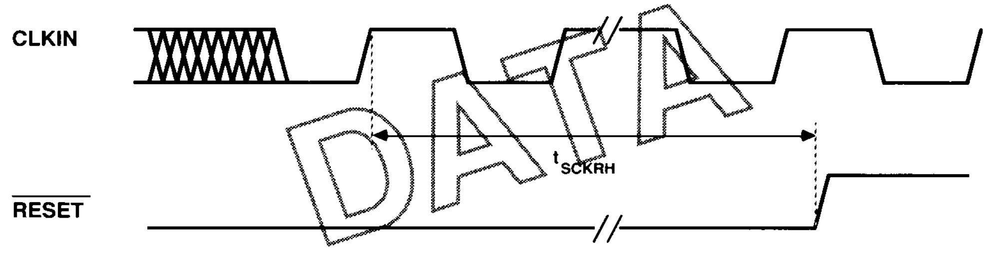


Figure 10 Reset at Power Up Timing

Interrupts

Parameter		20 MHz (50 ns) Min	20 MHz (50 ns) Max	Frequency Dependency (t_{CK} = T ns; DT=T–50 ns) Min	Frequency Dependency Max	Unit
Timing Requirement (Synchronous to CLKIN)						
t_{SIR}	$\overline{IRQ3\text{-}0}$ asserted to CLKIN rising edge	36		36+3DT/4		ns
t_{HIR}	$\overline{IRQ3\text{-}0}$ deasserted to CLKIN rising edge		26		26+3DT/4	ns

Meeting setup and hold for a particular cycle guarantees recognition in that cycle.

Parameter		20 MHz (50 ns) Min	20 MHz (50 ns) Max	Frequency Dependency Min	Frequency Dependency Max	Unit
Timing Requirement (Asynchronous to CLKIN)						
t_{IPW}	$\overline{IRQ3\text{-}0}$ pulse width	53		T+3		ns

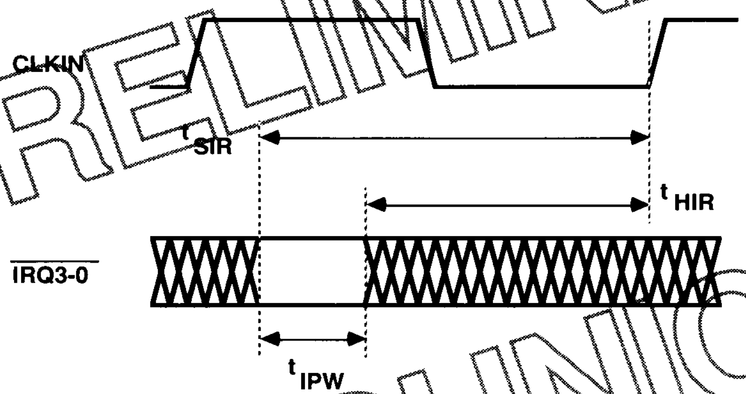


Figure 11. Interrupt Timing

Flags

Parameter		20 MHz (50 ns) Min	20 MHz (50 ns) Max	Frequency Dependency t_{CK} = T ns; DT=T–50 ns Min	Frequency Dependency Max	Unit
Timing Requirement						
t_{SFI}	FLAG3-0 (in) asserted to CLKIN rising edge	14		14+5DT/16		ns
t_{HFI}	FLAG3-0 (in) deasserted to CLKIN rising edge		4		4+5DT/16	ns
Switching Characteristic						
t_{DFO}	CLKIN rising edge to FLAG3-0 (out) asserted		31			ns
t_{HFO}	CLKIN rising edge to FLAG3-0 (out) deasserted	15				ns

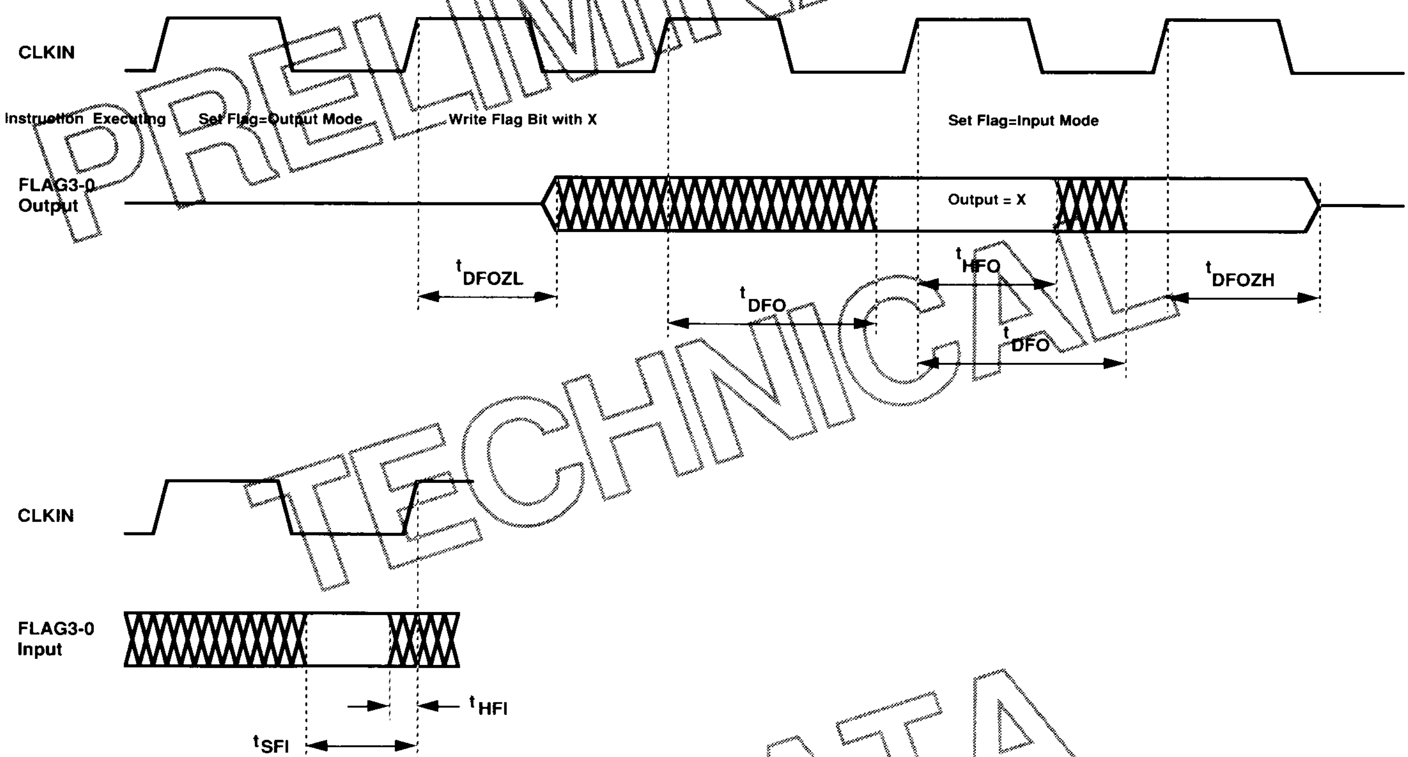


Figure 12 Flag Timing

Test Conditions

Below we show how we measure Output Disable Time and Output Enable Time.

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 Δ V is dependent on the capacitive load, C_L, and the current load, I_L. It can be approximated by the following equation:

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

The t_{xZL} specification is derived from the difference between $t_{measured}$ and t_{DECAY} as shown in the figure below. $t_{measured}$ is the interval from when the reference signal switches to when the output voltage decays 0.5 V from the measured output high or output low voltage. t_{DECAY} is calculated with Δ V equal to 0.5 V, and test loads C_L and I_L. If multiple pins (such as the data bus) are disabled, the min measurement value is that of the first pin to stop driving and the max measurement value is that of the last pin to stop driving.

Example System Hold Time Calculation

To determine the data hold time in a particular system, first calculate t_{DECAY} using the above equation. Choose Δ V to be the difference between the ADSP-21020's output voltage and the input threshold for the device requiring the hold time. A typical Δ V will be 0.4 V. C_L is the total bus capacitance and I_L is the total leakage or 3-state current of all devices on the bus. The hold time will be t_{DECAY} plus the minimum disable time (t_{xZL}).

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_{xZL}) 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.

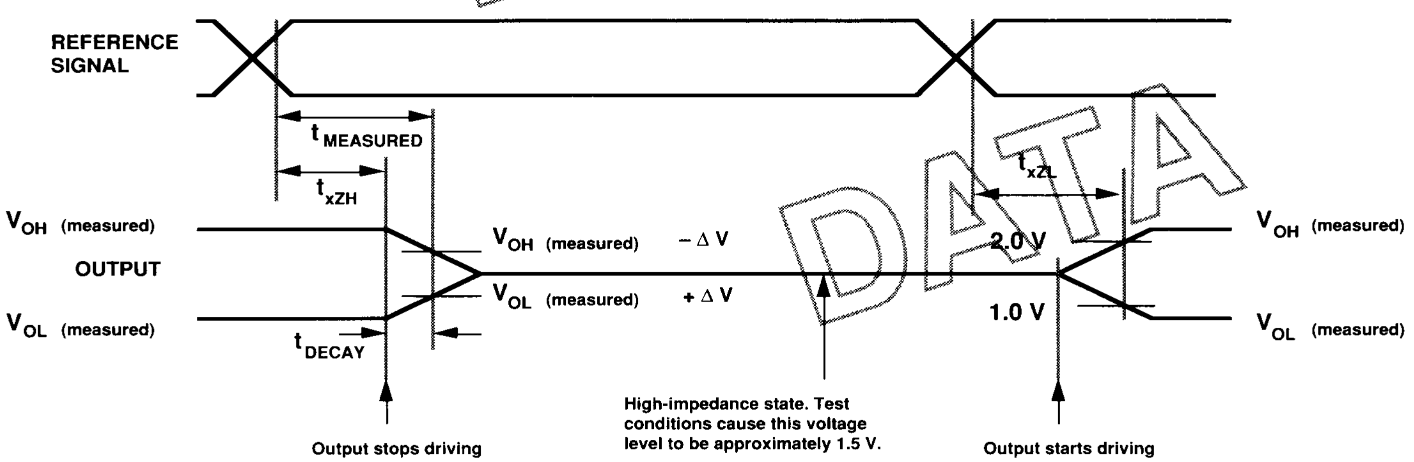
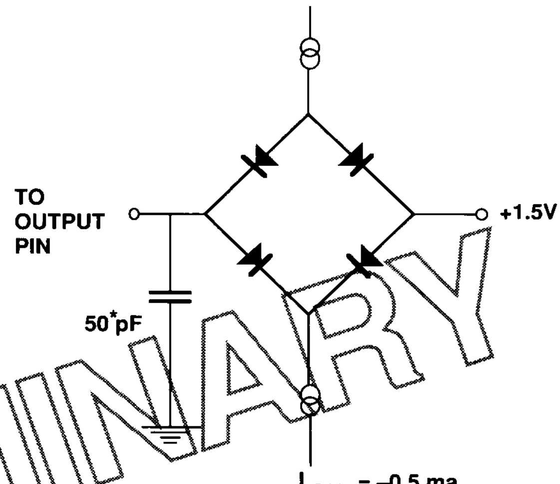


Figure 13 Output Enable/Disable Timing



* AC timing specifications are calculated for 100pF derating on Address, Select, Page and Strobe pins.

Figure 14 Equivalent Device Loading for AC Measurements (includes all fixtures)

Input Pulse Levels	GND to 3.0 V
Input Rise/Fall Times	5 ns
Input Timing Reference Levels	1.5 V
Output Reference Levels	1.5 V
Output Load	See Figure 14

Figure 15 AC Test Conditions

Typical Capacitance Derating

Output delays are based on standard capacitive loads (100 pF on Address, Select, Page and Strobe pins, 50 pF on others). For different loads, these timing parameters should be derated. These graphs show how output delays vary with capacitance. The graphs may not be linear outside the ranges shown.

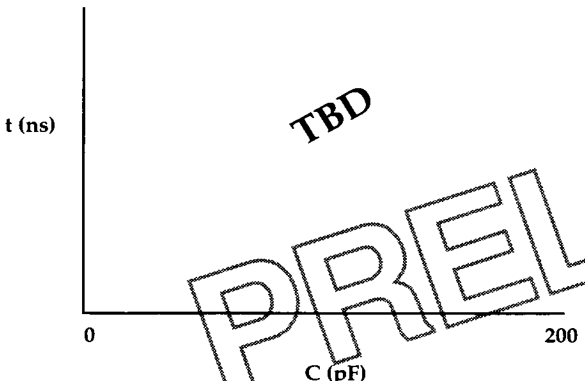


Figure 16a Address, Select and Page Pins

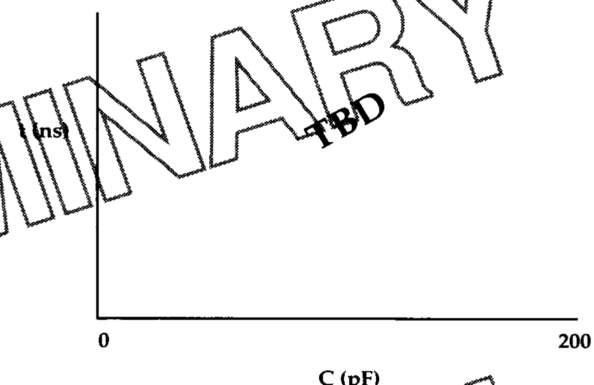


Figure 16b Read Strobes and Write Strobes

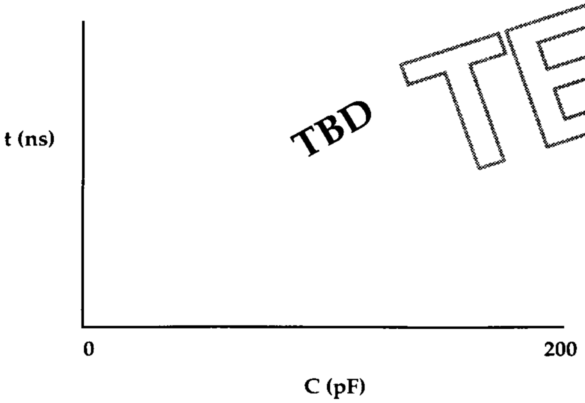


Figure 16c Data and Flag Pins

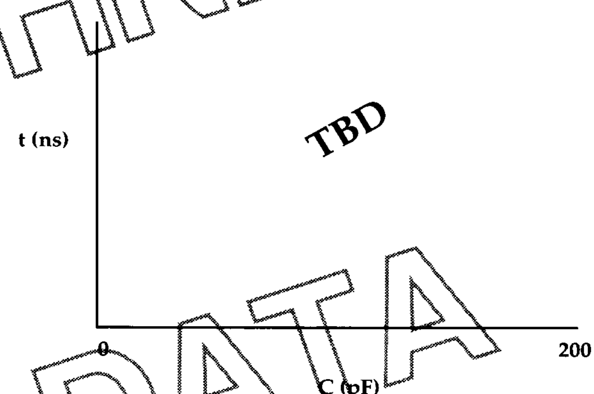


Figure 16d Bus Grant and Timer Expired Pins

Power and Ground Requirements

To achieve fast instruction execution and data accesses, the ADSP-21020 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 capacitance. These transient currents can cause disturbances on the power and ground lines. To minimize these effects, the ADSP-21020 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-21020 employs compensated output drivers. These drivers equalize slew rate over temperature extremes and process variations. A 2.2 kΩ resistor between the RCOMP pin and +5 V provides a reference for the compensated drivers.

All GND pins should have a low-impedance path to ground. A ground plane is required in ADSP-21020 systems to reduce this impedance, minimizing noise.

The EVDD pins should be bypassed to the ground plane using nine or ten high frequency capacitors (0.1µF ceramic). Keep each capacitor's lead and trace length to the EVDD pins as short as possible. This low-inductive path provides the ADSP-21020 with the peak currents required when its output drivers switch. The capacitor's 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-21020's output drivers.

The IVDD pins should be bypassed to the IGND pins using four high frequency capacitors (≥0.01µF ceramic). These can be located either under the ADSP-21020 or peripherally with short traces from the IVDD pins and to the ground plane.

If a V_{DD} plane is not used the following recommendations apply. Traces from the +5 V supply to the 10 EVDD pins should be designed to carry average DC currents of [$I_{DDEX}/10 \times$ number of EVDD pins per trace] while satisfying the minimum V_{DD} specification. A similar calculation should be made for the four 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.

Package Thermal Specifications

The ADSP-21020 is specified for operation for $T_{AMBIENT}$ of 0°C to 70°C. T_{CASE} can be measured under any environmental conditions to confirm that the device is operating in the allowable temperature range. Measure T_{CASE} on the top of the device.

Maximum T_{AMB} (ambient temperature) can be calculated from the following equation:

$$T_{AMB} = T_{CASE} - PD \times \Theta_{CA}$$

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 Electrical Characteristics. Θ_{CA} varies with airflow and the presence or absence of a heat sink. Table 9 shows a range of Θ_{CA} values. Table 10 indicates the corresponding maximum values for T_{AMB}, assuming nominal power dissipation.

	Airflow (ft./min.) 0	100	200	300
High Profile* Heat Sink	10.7	2.2	2.1	1.7
Low Profile* Heat Sink	13.9	3.3	2.2	1.9
No Heat Sink	16.5	4.3	3.3	2.6

All results are in °C/W. Theta JC (Θ_{JC}) is approximately 1 °C/W.

Table 9 Maximum Θ_{CA} for Various Airflow/Heat Sink Combinations (Plastic Pin Grid Array)

* Omnidirectional pin fin heat sink. High profile = 0.69" tall. Low profile = 0.15" tall.
All heat sinks were attached using thermal epoxy. The pin fin heat sinks were attached on top of the flat plate heat sinks.

Note: Test conditions – the method used follows MIL 1012.1 procedure. The ambient temperature was 25°C, and the power was .7 W.

	Airflow (ft./min.) 0	100	200	300
High Profile* Heat Sink	TBD	TBD	TBD	TBD
Low Profile* Heat Sink	TBD	TBD	TBD	TBD
No Heat Sink	TBD	TBD	TBD	TBD

Table 10 Maximum T_{AMB} for Various Airflow/Heat Sink Combinations (Plastic Pin Grid Array)

Note: Assumes nominal power dissipation ($PD_{max}/2$).
* Omnidirectional pin fin heat sink. High profile = 0.69" tall. Low profile = 0.15" tall.

JTAG Connector For Chip-On-Board Emulation

The IEEE JTAG specification (Joint Testing Action Group) designates a standard interface protocol for serial scan testing of components mounted on a PC board. The ADSP-21020 implements the JTAG interface (IEEE Standard 1149.1, Draft 6) for both boundary scan testing and low-cost chip-on-board emulation. For more information on the JTAG Interface ADSP-21020 Emulator, see the data sheet on ADSP-21020 Hardware Development Tools.

To use the J-ICE probe, the signals shown in Figure 17 must be provided on the target system via a 12 pin, 2 row, pin strip header (jumper header), which is keyed with pin 1 removed. The pin spacing is 0.1"x0.1" and the contact pins are 0.025" square and nominally 0.318" long. The J-ICE Emulator adds 2 TTL loads to the CLKIN line.

The Bxxx signals are provided so that the JTAG port can be used for board testing. When the connector is not being used for emulation, place jumpers between the Bxxx and xxx pins (as shown). If the JTAG connector is not to be used for board test, tie $\overline{BTRST}$ to GND and tie or pullup $\overline{BTCLK}$ to VCC.

Some example part numbers for the pin strip header are:

3M	929715xx06xx
Samtec	TSW-1614-x-D
McKenzie	PH2-318/110-12-x

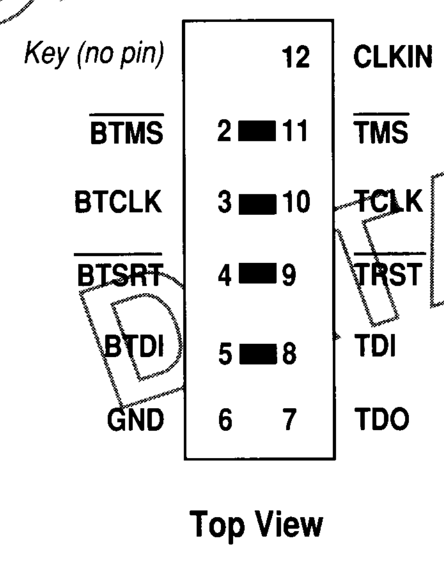


Figure 17

This information applies to a product under development. Its characteristics and specifications are subject to change without notice. Analog Devices assumes no obligation regarding future manufacturing unless otherwise agreed to in writing.

	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
U	PMA17	PMA20	TMS	EGND	TCK	EVDD	RCOMP	EGND	PMACK	EVDD	$\overline{PMWR}$	EGND	PMD44	EGND	PMD40	PMD39	PMD35	PMD31	U
T	EGND	PMA19	PMA23	$\overline{PMS1}$	$\overline{TRST}$	$\overline{DMWR}$	DMACK	CLKIN	NC	NC	$\overline{PMTS}$	PMD45	PMD42	NC	PMD37	PMD32	PMD30	PMD27	T
S	PMA11	PMA14	PMA18	PMA22	PMPAGE	TDI	$\overline{DMTS}$	$\overline{DMRD}$	NC	$\overline{PMRD}$	PMD47	PMD43	PMD41	PMD36	PMD34	PMD28	PMD26	PMD21	S
R	EGND	PMA10	PMA15	PMA16	PMA21	$\overline{PMS0}$	TDO	IGND	$\overline{RESET}$	IVDD	PMD46	IGND	PMD38	PMD33	PMD29	PMD25	PMD23	EGND	R
P	PMA8	PMA9	PMA13	PMA12											PMD24	PMD22	PMD19	PMD18	P
N	EVDD	PMA5	PMA6	PMA7											PMD20	PMD17	PMD16	EVDD	N
M	PMA1	PMA4	PMA3	PMA2											PMD15	PMD14	PMD13	PMD12	M
L	EGND	PMA0	TIMEXP	IGND											IGND	PMD10	PMD11	EGND	L
K	EVDD	NC	$\overline{IRQ2}$	$\overline{IRQ3}$					TOP VIEW						PMD6	PMD7	PMD8	PMD9	K
J	EVDD	$\overline{IRQ0}$	$\overline{IRQ1}$	IVDD											IVDD	PMD2	PMD5	EVDD	J
H	EGND	FLAG2	FLAG0	FLAG1											DMD1	DMD0	PMD3	PMD4	H
G	FLAG3	DMA1	DMA0	IGND											IGND	DMD3	NC	EGND	G
F	DMA2	DMA3	DMA4	DMA5											DMD9	DMD6	PMD0	PMD1	F
E	DMA6	DMA7	DMA8	DMA10											DMD13	DMD10	DMD2	EGND	E
D	DMA9	DMA11	DMA12	DMA15	DMA19	DMA23	DMA27	IGND	$\overline{DMS0}$	IVDD	DMD36	DMD31	DMD27	DMD22	DMD17	DMD11	DMD5	DMD4	D
C	DMA13	DMA14	DMA18	DMA20	DMA24	DMA28	DMA31	$\overline{DMS1}$	NC	DMD38	DMD35	DMD30	DMD28	DMD24	DMD20	DMD15	DMD8	DMD7	C
B	DMA16	DMA17	DMA21	DMA25	DMA26	DMA30	DMPAGE	$\overline{DMS3}$	DMD39	DMD37	DMD33	DMD32	DMD26	DMD25	DMD21	DMD18	DMD14	DMD12	B
A	$\overline{BR}$	$\overline{BG}$	DMA22	EGND	DMA29	EVDD	$\overline{DMS2}$	EGND	DMD34	EVDD	DMD29	EGND	DMD23	EVDD	DMD19	EGND	DMD16		A
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

NC = No Connect

Figure 18 Pin Configuration (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
U	PMD31	PMD35	PMD39	PMD40	EGND	PMD44	EGND	$\overline{PMWR}$	EVDD	PMACK	EGND	RCOMP	EVDD	TCK	EGND	TMS	PMA20	PMA17	**U**
T	PMD27	PMD30	PMD32	PMD37	NC	PMD42	PMD45	$\overline{PMTS}$	NC	NC	CLKIN	DMACK	$\overline{DMWR}$	$\overline{TRST}$	$\overline{PMS1}$	PMA23	PMA19	EGND	**T**
S	PMD21	PMD26	PMD28	PMD34	PMD36	PMD41	PMD43	PMD47	$\overline{PMRD}$	NC	$\overline{DMRD}$	$\overline{DMTS}$	TDI	PMPAGE	PMA22	PMA18	PMA14	PMA11	**S**
R	EGND	PMD23	PMD25	PMD29	PMD33	PMD38	IGND	PMD46	IVDD	$\overline{RESET}$	IGND	TDO	$\overline{PMS0}$	PMA21	PMA16	PMA15	PMA10	EGND	**R**
P	PMD18	PMD19	PMD22	PMD24											PMA12	PMA13	PMA9	PMA8	**P**
N	EVDD	PMD16	PMD17	PMD20											PMA7	PMA6	PMA5	EVDD	**N**
M	PMD12	PMD13	PMD14	PMD15											PMA2	PMA3	PMA4	PMA1	**M**
L	EGND	PMD11	PMD10	IGND											IGND	TIMEXP	PMA0	EGND	**L**
K	PMD9	PMD8	PMD7	PMD6					**BOTTOM VIEW**						$\overline{IRQ3}$	$\overline{IRQ2}$	NC	EVDD	**K**
J	EVDD	PMD5	PMD2	IVDD											IVDD	$\overline{IRQ1}$	$\overline{IRQ0}$	EVDD	**J**
H	PMD4	PMD3	DMD0	DMD1											FLAG1	FLAG0	FLAG2	EGND	**H**
G	EGND	NC	DMD3	IGND											IGND	DMA0	DMA1	FLAG3	**G**
F	PMD1	PMD0	DMD6	DMD9											DMA5	DMA4	DMA3	DMA2	**F**
E	EGND	DMD2	DMD10	DMD13											DMA10	DMA8	DMA7	DMA6	**E**
D	DMD4	DMD5	DMD11	DMD17	DMD22	DMD27	DMD31	DMD36	IVDD	$\overline{DMS0}$	IGND	DMA27	DMA23	DMA19	DMA15	DMA12	DMA11	DMA9	**D**
C	DMD7	DMD8	DMD15	DMD20	DMD24	DMD28	DMD30	DMD35	DMD38	NC	$\overline{DMS1}$	DMA31	DMA28	DMA24	DMA20	DMA18	DMA14	DMA13	**C**
B	DMD12	DMD14	DMD18	DMD21	DMD25	DMD26	DMD32	DMD33	DMD37	DMD39	$\overline{DMS3}$	DMPAGE	DMA30	DMA26	DMA25	DMA21	DMA17	DMA16	**B**
A		DMD16	EGND	DMD19	EVDD	DMD23	EGND	DMD29	EVDD	DMD34	EGND	$\overline{DMS2}$	EVDD	DMA29	EGND	DMA22	$\overline{BG}$	$\overline{BR}$	**A**
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

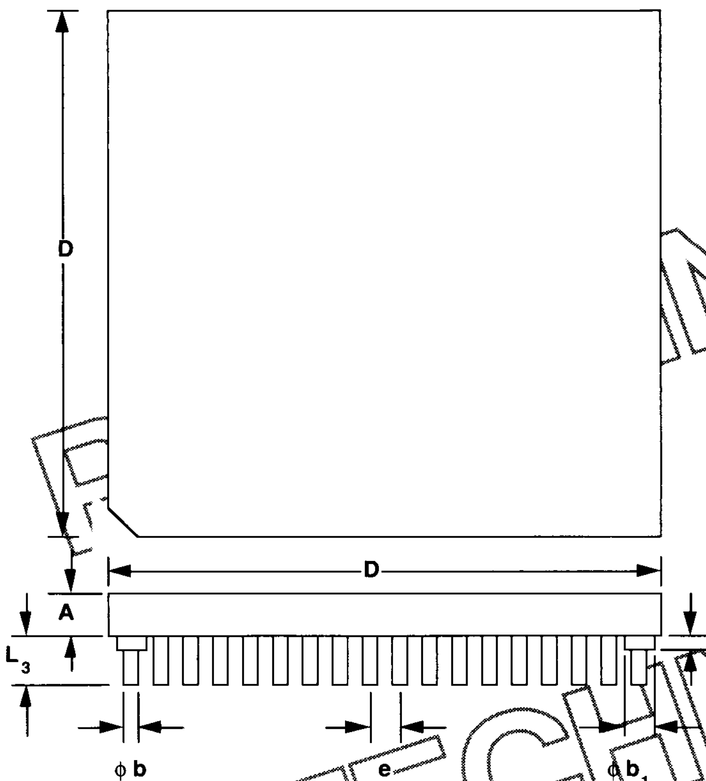
NC = No Connect

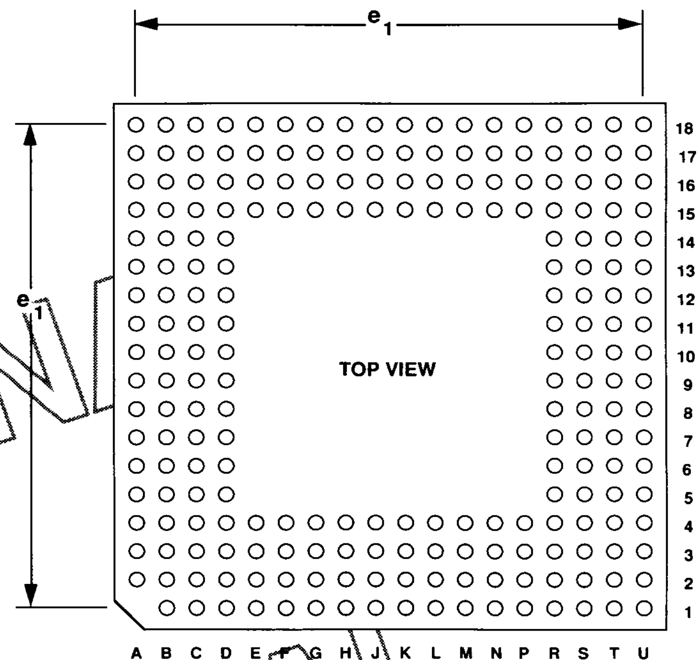
Figure 19 Pin Configuration (Bottom View)

FUNCTION	LOCATION	FUNCTION	LOCATION	FUNCTION	LOCATION	FUNCTION	LOCATION
DMA0	G16	DMD24	C5	PMD7	K3	IRQ2	K16
DMA1	G17	DMD25	B5	PMD8	K2	IRQ3	K15
DMA2	F18	DMD26	B6	PMD9	K1	RESET	R10
DMA3	F17	DMD27	D6	PMD10	L3	TIMEXP	L16
DMA4	F16	DMD28	C6	PMD11	L2	RCOMP	U12
DMA5	F15	DMD29	A8	PMD12	M1	CLKIN	T11
DMA6	E18	DMD30	C7	PMD13	M2	TRST	T14
DMA7	E17	DMD31	D7	PMD14	M3	TD0	R12
DMA8	E16	DMD32	B7	PMD15	M4	TDI	S13
DMA9	D18	DMD33	B8	PMD16	N2	TMS	U16
DMA10	E15	DMD34	A10	PMD17	N3	TCK	U14
DMA11	D17	DMD35	C8	PMD18	P1	EGND	H18
DMA12	D16	DMD36	D8	PMD19	P2	EGND	A3
DMA13	C18	DMD37	B9	PMD20	N4	EGND	A7
DMA14	C17	DMD38	C9	PMD21	S1	EGND	A11
DMA15	D15	DMD39	B10	PMD22	P3	EGND	A15
DMA16	B18	DMS0	D10	PMD23	R2	EGND	E1
DMA17	B17	DMS1	C11	PMD24	P4	EGND	G1
DMA18	C16	DMS2	A12	PMD25	R3	EGND	L1
DMA19	D14	DMS3	B11	PMD26	S2	EGND	L18
DMA20	C15	DMWR	T13	PMD27	T1	EGND	R1
DMA21	B16	DMRD	S11	PMD28	S3	EGND	R18
DMA22	A16	DMPAGE	B12	PMD29	R4	EGND	T18
DMA23	D13	DMTS	S12	PMD30	T2	EGND	U5
DMA24	C14	DMACK	T12	PMD31	U1	EGND	U7
DMA25	B15	PMA0	L17	PMD32	T3	EGND	U11
DMA26	B14	PMA1	M18	PMD33	R5	EGND	U15
DMA27	D12	PMA2	M15	PMD34	S4	IGND	D11
DMA28	C13	PMA3	M16	PMD35	U2	IGND	G4
DMA29	A14	PMA4	M17	PMD36	S5	IGND	G15
DMA30	B13	PMA5	N17	PMD37	T4	IGND	L4
DMA31	C12	PMA6	N16	PMD38	R6	IGND	L15
DMD0	H3	PMA7	N15	PMD39	U3	IGND	R7
DMD1	H4	PMA8	P18	PMD40	U4	IGND	R11
DMD2	E2	PMA9	P17	PMD41	S6	EVDD	A5
DMD3	G3	PMA10	R17	PMD42	T6	EVDD	A9
DMD4	D1	PMA11	S18	PMD43	S7	EVDD	A13
DMD5	D2	PMA12	P15	PMD44	U6	EVDD	J1
DMD6	F3	PMA13	P16	PMD45	T7	EVDD	J18
DMD7	C1	PMA14	S17	PMD46	R8	EVDD	N1
DMD8	C2	PMA15	R16	PMD47	S8	EVDD	N18
DMD9	F4	PMA16	R15	PMS0	R13	EVDD	U9
DMD10	E3	PMA17	U18	PMS1	T15	EVDD	U13
DMD11	D3	PMA18	S16	PMWR	U8	EVDD	K18
DMD12	B1	PMA19	T17	PMRD	S9	IVDD	D9
DMD13	E4	PMA20	U17	PMPAGE	S14	IVDD	J4
DMD14	B2	PMA21	R14	PMTS	T8	IVDD	J15
DMD15	C3	PMA22	S15	PMACK	U10	IVDD	R9
DMD16	A2	PMA23	T16	BG	A17	NC	C10
DMD17	D4	PMD0	F2	BR	A18	NC	S10
DMD18	B3	PMD1	F1	FLAG0	H16	NC	T10
DMD19	A4	PMD2	J3	FLAG1	H15	NC	T9
DMD20	C4	PMD3	H2	FLAG2	H17	NC	K17
DMD21	B4	PMD4	H1	FLAG3	G18	NC	T5
DMD22	D5	PMD5	J2	IRQ0	J17	NC	G2
DMD23	A6	PMD6	K4	IRQ1	J16		

NC = No Connect

223-Pin Plastic Pin Grid Array





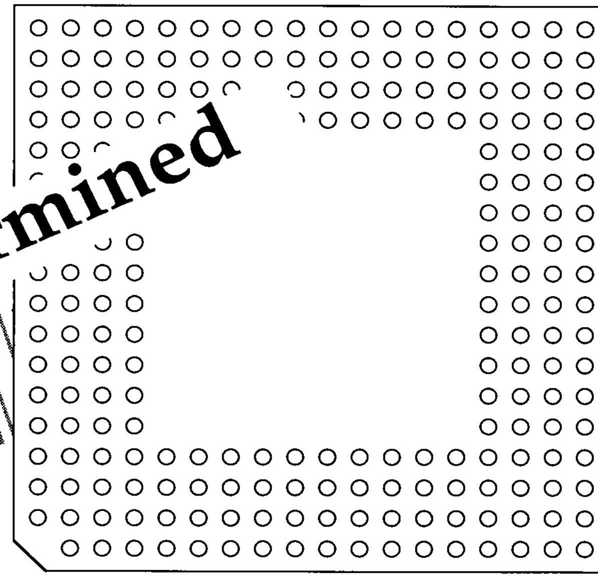
SYMBOL	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.190	0.214	4.826	5.436
A_1	0.065	0.076	1.65	1.95
φ b	0.017	0.019	0.42	0.50
φ b_1	0.046	0.054	1.17	1.37
D	1.856	1.864	47.14	47.34
e_1	1.688	1.712	42.88	43.48
e	0.988	0.112	2.24	2.84
L_3	0.185	0.209	4.7	5.3

Note: When socketing with the PPGA package, Analog Devices recommends using a low-insertion-force socket.

This information applies to a product under development. Its characteristics and specifications are subject to change without notice. Analog Devices assumes no obligation regarding future manufacturing unless otherwise agreed to in writing.

223-Pin Ceramic Pin Grid Array

To Be Determined



Note: When socketing with the CPGA package, Analog Devices recommends using a low-insertion-force socket.

Surface Mount Package

To Be Determined

Ordering Information*

Part Number	Speed (MHz)	Ambient Temperature Range	Package
ADSP-21020KU-80	20	0 to 70°C	223-lead Plastic Pin Grid Array
ADSP-21020KU-60	15	0 to 70°C	223-lead Plastic Pin Grid Array

* Military versions are being planned.

Note: The part numbering system for the ADSP-21020 reflects historical part numbering for DSP Processors, indicating the processor's effective internal clock rate. Most DSP processors operate with a CLKIN that is 2x-4x the processor's instruction rate.

The 21020 accepts a 1x CLKIN signal, which equals the 21020's instruction rate. From this, the 21020 generates internal clocks all equal to the CLKIN frequency, but representing separate phases relative to CLKIN. The ADSP-21020 offers users the advantage of a low-noise 1x CLKIN signal on their printed circuit board, with the performance advantage of effectively a 4x internal clocking scheme for maximum throughput. The separate internal clocks do not generate any appreciable additional RF noise.

The numbering scheme is shown below:

	CLKIN Signal (1x Clock) Frequency	*Instruction Cycle*	*Effective Internal Clock (4x CLKIN)*
ADSP-21020Kx-80	20 MHz	50 nsec	80 MHz
ADSP-21020Kx-60	15 MHz	66 nsec	60 MHz

P1492–8–1/91

PRINTED IN U.S.A.

One Technology Way; P.O. Box 9106; Norwood, MA 02062–9106 U.S.A. Tel: 617/329–4700
Twx: 710/394–6577 Telex: 174059 Cables: ANALOG NORWOODMASS

024295 ✓ _ R