DSP Microcomputer

ADSP-2191M

PERFORMANCE FEATURES

6.25 ns Instruction Cycle Time, for up to 160 MIPS Sustained Performance

ADSP-218x Family Code Compatible with the Same Easy to Use Algebraic Syntax

Single-Cycle Instruction Execution

Single-Cycle Context Switch between Two Sets of Computation and Memory Instructions

Instruction Cache Allows Dual Operand Fetches in Every Instruction Cycle

Multifunction Instructions

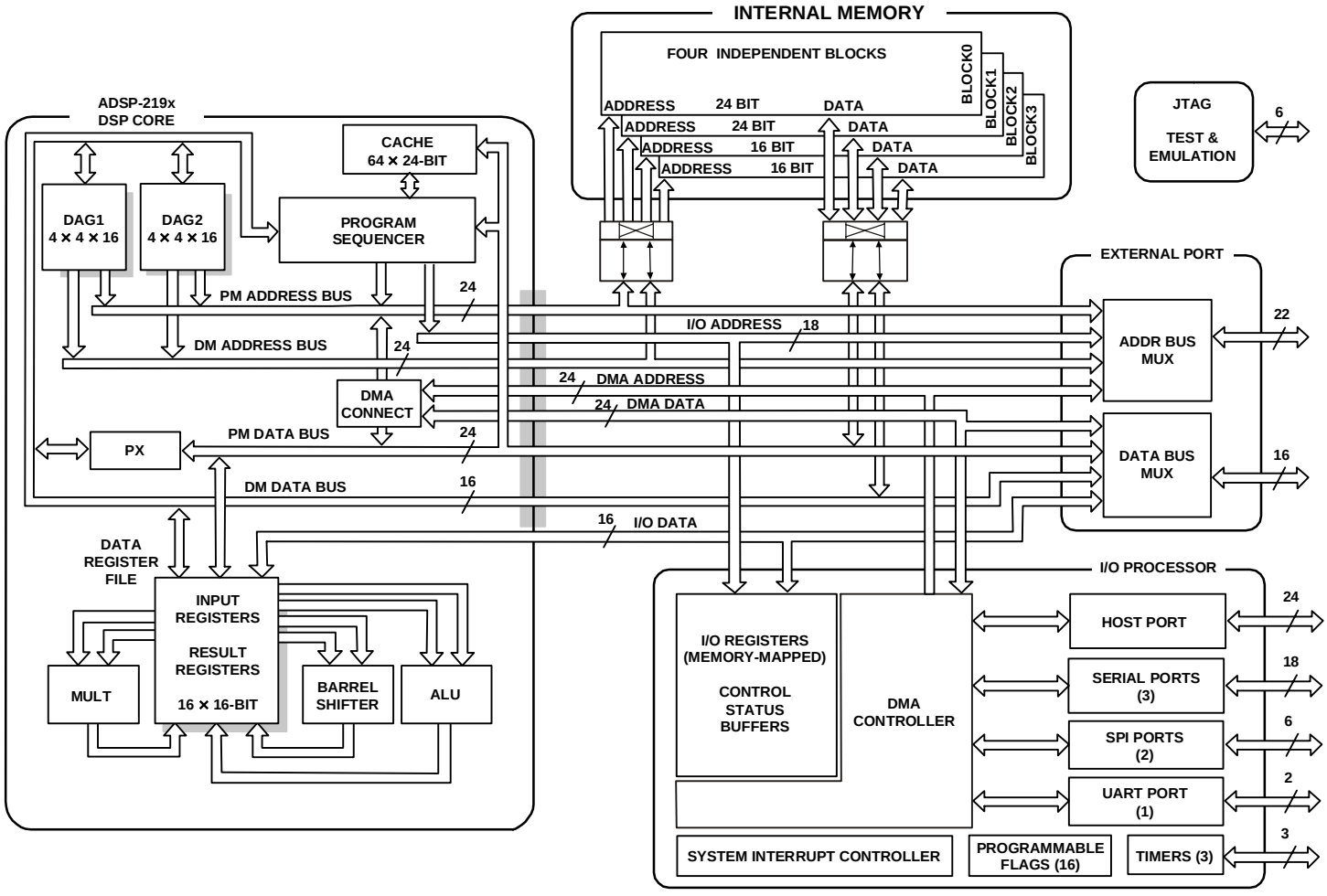
Pipelined Architecture Supports Efficient Code Execution

Architectural Enhancements for Compiled C and C++ Code Efficiency

Architectural Enhancements beyond ADSP-218x Family are Supported with Instruction Set Extensions for Added Registers, and Peripherals

Flexible Power Management with User-Selectable Power-Down and Idle Modes

FUNCTIONAL BLOCK DIAGRAM



REV. 0

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O.Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel:781/329-4700 www.analog.com
Fax:781/326-8703
© Analog Devices, Inc., 2002

ADSP-2191M

INTEGRATION FEATURES

160 K Bytes On-Chip RAM Configured as 32K Words 24-Bit Memory RAM and 32K Words 16-Bit Memory RAM
Dual-Purpose 24-Bit Memory for Both Instruction and Data Storage
Independent ALU, Multiplier/Accumulator, and Barrel Shifter Computational Units with Dual 40-bit Accumulators
Unified Memory Space Allows Flexible Address Generation, Using Two Independent DAG Units
Powerful Program Sequencer Provides Zero-Overhead Looping and Conditional Instruction Execution
Enhanced Interrupt Controller Enables Programming of Interrupt Priorities and Nesting Modes

SYSTEM INTERFACE FEATURES

Host Port with DMA Capability for Glueless 8- or 16-Bit Host Interface
16-Bit External Memory Interface for up to 16M Words of Addressable Memory Space
Three Full-Duplex Multichannel Serial Ports, with Support for H.100 and up to 128 TDM Channels with A-Law and μ-Law Companding Optimized for Telecommunications Systems
Two SPI-Compatible Ports with DMA Support
UART Port with DMA Support
16 General-Purpose I/O Pins with Integrated Interrupt Support
Three Programmable Interval Timers with PWM Generation, PWM Capture/Pulsewidth Measurement, and External Event Counter Capabilities
Up to 11 DMA Channels Can Be Active at Any Given Time for High I/O Throughput
On-Chip Boot ROM for Automatic Booting from External 8- or 16-Bit Host Device, SPI ROM, or UART with Autobaud Detection
Programmable PLL Supports 1× to 32× Input Frequency Multiplication and Can Be Altered during Runtime
IEEE JTAG Standard 1149.1 Test Access Port Supports On-Chip Emulation and System Debugging
2.5 V Internal Operation and 3.3 V I/O
144-Lead LQFP and 144-Ball Mini-BGA Packages

TABLE OF CONTENTS

GENERAL DESCRIPTION 3
DSP Core Architecture 3
DSP Peripherals Architecture 4
Memory Architecture 5
Interrupts 6
DMA Controller 7
Host Port 8
DSP Serial Ports (SPORTs) 9
Serial Peripheral Interface (SPI) Ports 9
UART Port 9
Programmable Flag (PFx) Pins 10
Low Power Operation 10
Clock Signals 11
Reset 11
Power Supplies 11
Booting Modes 11
Bus Request and Bus Grant 12
Instruction Set Description 13
Development Tools 13
Additional Information 15
PIN FUNCTION DESCRIPTIONS 15
SPECIFICATIONS 18
ABSOLUTE MAXIMUM RATINGS. 19
ESD SENSITIVITY 19
Power Dissipation 19
TIMING SPECIFICATIONS 20
Output Drive Currents 41
Power Dissipation 41
Test Conditions 41
Environmental Conditions 42
144-Lead LQFP Pinout 44
144-Lead Mini-BGA Pinout 46
OUTLINE DIMENSIONS 48
ORDERING GUIDE 49

GENERAL DESCRIPTION

The ADSP-2191M DSP is a single-chip microcomputer optimized for digital signal processing (DSP) and other high speed numeric processing applications.

The ADSP-2191M combines the ADSP-219x family base architecture (three computational units, two data address generators, and a program sequencer) with three serial ports, two SPI-compatible ports, one UART port, a DMA controller, three programmable timers, general-purpose Programmable Flag pins, extensive interrupt capabilities, and on-chip program and data memory spaces.

The ADSP-2191M architecture is code-compatible with DSPs of the ADSP-218x family. Although the architectures are compatible, the ADSP-2191M architecture has a number of enhancements over the ADSP-218x architecture. The enhancements to computational units, data address generators, and program sequencer make the ADSP-2191M more flexible and even easier to program.

Indirect addressing options provide addressing flexibility—premodify with no update, pre- and post-modify by an immediate 8-bit, two's-complement value and base address registers for easier implementation of circular buffering.

The ADSP-2191M integrates 64K words of on-chip memory configured as 32K words (24-bit) of program RAM, and 32K words (16-bit) of data RAM. Power-down circuitry is also provided to reduce power consumption. The ADSP-2191M is available in 144-lead LQFP and 144-ball mini-BGA packages.

Fabricated in a high-speed, low-power, CMOS process, the ADSP-2191M operates with a 6.25 ns instruction cycle time (160 MIPS). All instructions, except single-word instructions, execute in one processor.

The ADSP-2191M's flexible architecture and comprehensive instruction set support multiple operations in parallel. For example, in one processor cycle, the ADSP-2191M can:

- Generate an address for the next instruction fetch
- Fetch the next instruction
- Perform one or two data moves
- Update one or two data address pointers
- Perform a computational operation

These operations take place while the processor continues to:

- Receive and transmit data through two serial ports
- Receive and/or transmit data from a Host
- Receive or transmit data through the UART
- Receive or transmit data over two SPI ports
- Access external memory through the external memory interface
- Decrement the timers

DSP Core Architecture

The ADSP-2191M instruction set provides flexible data moves and multifunction (one or two data moves with a computation) instructions. Every single-word instruction can be executed in a single processor cycle. The ADSP-2191M assembly language uses an algebraic syntax for ease of coding and readability. A comprehensive set of development tools supports program development.

The functional block diagram on page 1 shows the architecture of the ADSP-219x core. It contains three independent computational units: the ALU, the multiplier/accumulator (MAC), and the shifter. The computational units process 16-bit data from the register file and have provisions to support multiprecision computations. The ALU performs a standard set of arithmetic and logic operations; division primitives are also supported. The MAC performs single-cycle multiply, multiply/add, and multiply/subtract operations. The MAC has two 40-bit accumulators, which help with overflow. The shifter performs logical and arithmetic shifts, normalization, denormalization, and derive exponent operations. The shifter can be used to efficiently implement numeric format control, including multiword and block floating-point representations.

Register-usage rules influence placement of input and results within the computational units. For most operations, the computational units' data registers act as a data register file, permitting any input or result register to provide input to any unit for a computation. For feedback operations, the computational units let the output (result) of any unit be input to any unit on the next cycle. For conditional or multifunction instructions, there are restrictions on which data registers may provide inputs or receive results from each computational unit. For more information, see the *ADSP-219x DSP Instruction Set Reference*.

A powerful program sequencer controls the flow of instruction execution. The sequencer supports conditional jumps, subroutine calls, and low interrupt overhead. With internal loop counters and loop stacks, the ADSP-2191M executes looped code with zero overhead; no explicit jump instructions are required to maintain loops.

Two data address generators (DAGs) provide addresses for simultaneous dual operand fetches (from data memory and program memory). Each DAG maintains and updates four 16-bit address pointers. Whenever the pointer is used to access data (indirect addressing), it is pre- or post-modified by the value of one of four possible modify registers. A length value and base address may be associated with each pointer to implement automatic modulo addressing for circular buffers. Page registers in the DAGs allow circular addressing within 64K word boundaries of each of the 256 memory pages, but these buffers may not cross page boundaries. Secondary registers duplicate all the primary registers in the DAGs; switching between primary and secondary registers provides a fast context switch.

Efficient data transfer in the core is achieved with the use of internal buses:

- Program Memory Address (PMA) Bus
- Program Memory Data (PMD) Bus
- Data Memory Address (DMA) Bus
- Data Memory Data (DMD) Bus
- DMA Address Bus
- DMA Data Bus

The two address buses (PMA and DMA) share a single external address bus, allowing memory to be expanded off-chip, and the two data buses (PMD and DMD) share a single external data bus. Boot memory space and I/O memory space also share the external buses.

Program memory can store both instructions and data, permitting the ADSP-2191M to fetch two operands in a single cycle, one from program memory and one from data memory. The DSP's dual memory buses also let the ADSP-219x core fetch an operand from data memory and the next instruction from program memory in a single cycle.

DSP Peripherals Architecture

The functional block diagram on page 1 shows the DSP's on-chip peripherals, which include the external memory interface, Host port, serial ports, SPI-compatible ports, UART port, JTAG test and emulation port, timers, flags, and interrupt controller. These on-chip peripherals can connect to off-chip devices as shown in Figure 1.

The ADSP-2191M has a 16-bit Host port with DMA capability that lets external Hosts access on-chip memory. This 24-pin parallel port consists of a 16-pin multiplexed data/address bus and provides a low-service overhead data move capability. Configurable for 8 or 16 bits, this port provides a glueless interface to a wide variety of 8- and 16-bit microcontrollers. Two chip-selects provide Hosts access to the DSP's entire memory map. The DSP is bootable through this port.

The ADSP-2191M also has an external memory interface that is shared by the DSP's core, the DMA controller, and DMA capable peripherals, which include the UART, SPORT0, SPORT1, SPORT2, SPI0, SPI1, and the Host port. The external port consists of a 16-bit data bus, a 22-bit address bus, and control signals. The data bus is configurable to provide an 8 or 16 bit interface to external memory. Support for word packing lets the DSP access 16- or 24-bit words from external memory regardless of the external data bus width. When configured for an 8-bit interface, the unused eight lines provide eight programmable, bidirectional general-purpose Programmable Flag lines, six of which can be mapped to software condition signals.

The memory DMA controller lets the ADSP-2191M move data and instructions from between memory spaces: internal-to-external, internal-to-internal, and external-to- external. On-chip peripherals can also use this controller for DMA transfers.

The ADSP-2191M can respond to up to seventeen interrupts at any given time: three internal (stack, emulator kernel, and power-down), two external (emulator and reset), and twelve user-defined (peripherals) interrupts. The programmer assigns a peripheral to one of the 12 user-defined interrupts. The priority of each peripheral for interrupt service is determined by these assignments.

There are three serial ports on the ADSP-2191M that provide a complete synchronous, full-duplex serial interface. This interface includes optional companding in hardware and a wide variety of framed or frameless data transmit and receive modes of operation. Each serial port can transmit or receive an internal or external, programmable serial clock and frame syncs. Each serial port supports 128-channel Time Division Multiplexing.

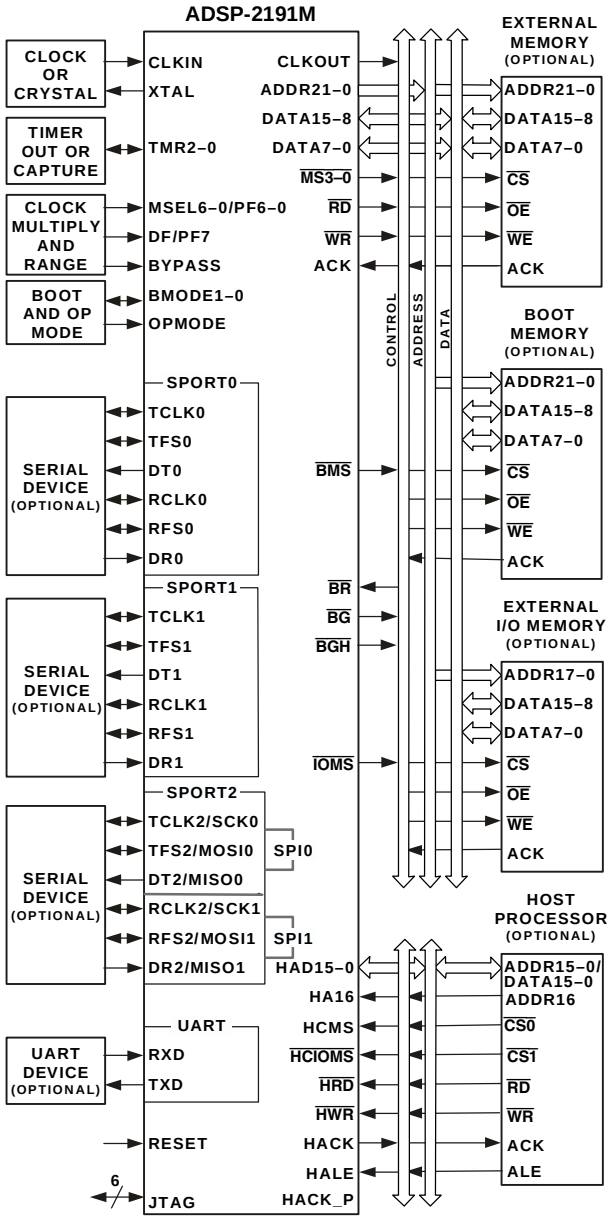


Figure 1. System Diagram

The ADSP-2191M provides up to sixteen general-purpose I/O pins, which are programmable as either inputs or outputs. Eight of these pins are dedicated-general purpose Programmable Flag pins. The other eight of them are multifunctional pins, acting as general-purpose I/O pins when the DSP connects to an 8-bit external data bus and acting as the upper eight data pins when the DSP connects to a 16-bit external data bus. These Programmable Flag pins can implement edge- or level-sensitive interrupts, some of which can be used to base the execution of conditional instructions.

Three programmable interval timers generate periodic interrupts. Each timer can be independently set to operate in one of three modes:

- Pulse Waveform Generation mode
- Pulsewidth Count/Capture mode
- External Event Watchdog mode

Each timer has one bidirectional pin and four registers that implement its mode of operation: A 7-bit configuration register, a 32-bit count register, a 32-bit period register, and a 32-bit pulsewidth register. A single status register supports all three timers. A bit in each timer's configuration register enables or disables the corresponding timer independently of the others.

Memory Architecture

The ADSP-2191M DSP provides 64K words of on-chip SRAM memory. This memory is divided into four 16K blocks located on memory Page 0 in the DSP's memory map. In addition to the internal and external memory space, the ADSP-2191M can address two additional and separate off-chip memory spaces: I/O space and boot space.

As shown in Figure 2, the DSP's two internal memory blocks populate all of Page 0. The entire DSP memory map consists of 256 pages (Pages 0–255), and each page is 64K words long. External memory space consists of four memory banks (banks 0–3) and supports a wide variety of SRAM memory devices. Each bank is selectable using the memory select pins ($\overline{MS3–0}$) and has configurable page boundaries, waitstates, and waitstate modes. The 1K word of on-chip boot-ROM populates the top of Page 255 while the remaining 254 pages are addressable off-chip. I/O memory pages differ from external memory pages in that I/O pages are 1K word long, and the external I/O pages have their own select pin ($\overline{IOMS}$). Pages 0–7 of I/O memory space reside on-chip and contain the configuration registers for the peripherals. Both the core and DMA-capable peripherals can access the DSP's entire memory map.

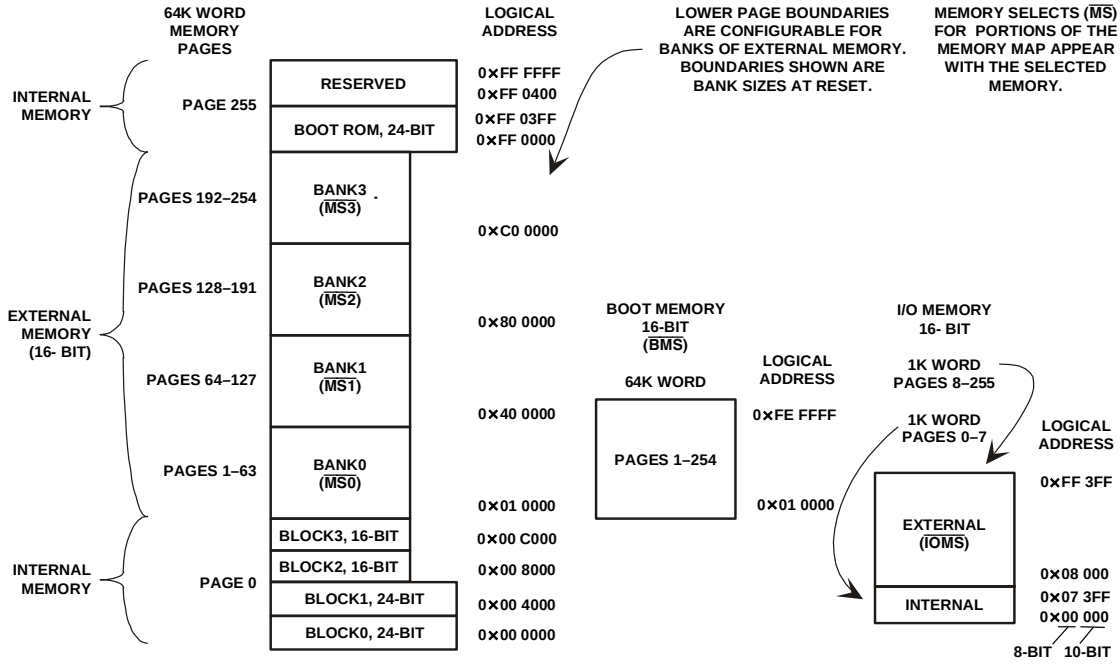


Figure 2. Memory Map

Internal (On-Chip) Memory

The ADSP-2191M's unified program and data memory space consists of 16M locations that are accessible through two 24-bit address buses, the PMA and DMA buses. The DSP uses slightly

different mechanisms to generate a 24-bit address for each bus. The DSP has three functions that support access to the full memory map.

- The DAGs generate 24-bit addresses for data fetches from the entire DSP memory address range. Because DAG index (address) registers are 16 bits wide and hold the lower 16 bits of the address, each of the DAGs has its own 8-bit page register (DMPGx) to hold the most significant eight address bits. Before a DAG generates an address, the program must set the DAG's DMPGx register to the appropriate memory page.
- The Program Sequencer generates the addresses for instruction fetches. For relative addressing instructions, the program sequencer bases addresses for relative jumps, calls, and loops on the 24-bit Program Counter (PC). In direct addressing instructions (two-word instructions), the instruction provides an immediate 24-bit address value. The PC allows linear addressing of the full 24-bit address range.
- For indirect jumps and calls that use a 16-bit DAG address register for part of the branch address, the Program Sequencer relies on an 8-bit Indirect Jump page (IJPG) register to supply the most significant eight address bits. Before a cross page jump or call, the program must set the program sequencer's IJPG register to the appropriate memory page.

The ADSP-2191M has 1K word of on-chip ROM that holds boot routines. If peripheral booting is selected, the DSP starts executing instructions from the on-chip boot ROM, which starts the boot process from the selected peripheral. For more information, see "Booting Modes" on page 11. The on-chip boot ROM is located on Page 255 in the DSP's memory space map.

External (Off-Chip) Memory

Each of the ADSP-2191M's off-chip memory spaces has a separate control register, so applications can configure unique access parameters for each space. The access parameters include read and write wait counts, waitstate completion mode, I/O clock divide ratio, write hold time extension, strobe polarity, and data bus width. The core clock and peripheral clock ratios influence the external memory access strobe widths. For more information, see "Clock Signals" on page 11. The off-chip memory spaces are:

- External memory space ($\overline{\text{MS3–0}}$ pins)
- I/O memory space ($\overline{\text{IOMS}}$ pin)
- Boot memory space ($\overline{\text{BMS}}$ pin)

All of these off-chip memory spaces are accessible through the External Port, which can be configured for data widths of 8 or 16 bits.

External Memory Space

External memory space consists of four memory banks. These banks can contain a configurable number of 64K word pages. At reset, the page boundaries for external memory have Bank0 containing pages 1–63, Bank1 containing pages 64–127, Bank2 containing pages 128–191, and Bank3 that contains pages 192–254. The $\overline{\text{MS3–0}}$ memory bank pins select Banks 3–0, respectively. The external memory interface is byte-addressable and decodes the 8 MSBs of the DSP program address to select one of the four banks. Both the ADSP-219x core and DMA-capable peripherals can access the DSP's external memory space.

I/O Memory Space

The ADSP-2191M supports an additional external memory called I/O memory space. This space is designed to support simple connections to peripherals (such as data converters and external registers) or to bus interface ASIC data registers. I/O space supports a total of 256K locations. The first 8K addresses are reserved for on-chip peripherals. The upper 248K addresses are available for external peripheral devices. The DSP's instruction set provides instructions for accessing I/O space. These instructions use an 18-bit address that is assembled from an 8-bit I/O page (IOPG) register and a 10-bit immediate value supplied in the instruction. Both the ADSP-219x core and a Host (through the Host Port Interface) can access I/O memory space.

Boot Memory Space

Boot memory space consists of one off-chip bank with 63 pages. The $\overline{\text{BMS}}$ memory bank pin selects boot memory space. Both the ADSP-219x core and DMA-capable peripherals can access the DSP's off-chip boot memory space. After reset, the DSP always starts executing instructions from the on-chip boot ROM. Depending on the boot configuration, the boot ROM code can start booting the DSP from boot memory. For more information, see "Booting Modes" on page 11.

Interrupts

The interrupt controller lets the DSP respond to 17 interrupts with minimum overhead. The controller implements an interrupt priority scheme as shown in Table 1. Applications can use the unassigned slots for software and peripheral interrupts.

Table 2 shows the ID and priority at reset of each of the peripheral interrupts. To assign the peripheral interrupts a different priority, applications write the new priority to their corresponding control bits (determined by their ID) in the Interrupt Priority Control register. The peripheral interrupt's position in the IMASK and IRPTL register and its vector address depend on its priority level, as shown in Table 1. Because the IMASK and IRPTL registers are limited to 16 bits, any peripheral interrupts

assigned a priority level of 11 are aliased to the lowest priority bit position (15) in these registers and share vector address 0x00 01E0.

Table 1. Interrupt Priorities/Addresses

Interrupt	IMASK/ IRPTL	Vector Address[1]
Emulator (NMI)— Highest Priority	NA	NA
Reset (NMI)	0	0x00 0000
Power-Down (NMI)	1	0x00 0020
Loop and PC Stack	2	0x00 0040
Emulation Kernel	3	0x00 0060
User Assigned Interrupt	4	0x00 0080
User Assigned Interrupt	5	0x00 00A0
User Assigned Interrupt	6	0x00 00C0
User Assigned Interrupt	7	0x00 00E0
User Assigned Interrupt	8	0x00 0100
User Assigned Interrupt	9	0x00 0120
User Assigned Interrupt	10	0x00 0140
User Assigned Interrupt	11	0x00 0160
User Assigned Interrupt	12	0x00 0180
User Assigned Interrupt	13	0x00 01A0
User Assigned Interrupt	14	0x00 01C0
User Assigned Interrupt— Lowest Priority	15	0x00 01E0

[1]These interrupt vectors start at address 0x10000 when the DSP is in "no-boot," run from external memory mode.

Table 2. Peripheral Interrupts and Priority at Reset

Interrupt	ID	Reset Priority
Slave DMA/Host Port Interface	0	0
SPORT0 Receive	1	1
SPORT0 Transmit	2	2
SPORT1 Receive	3	3
SPORT1 Transmit	4	4
SPORT2 Receive/SPI0	5	5
SPORT2 Transmit/SPI1	6	6
UART Receive	7	7
UART Transmit	8	8
Timer 0	9	9
Timer 1	10	10
Timer 2	11	11
Programmable Flag A (any PFx)	12	11
Programmable Flag B (any PFx)	13	11
Memory DMA port	14	11

Interrupt routines can either be nested with higher priority interrupts taking precedence or processed sequentially. Interrupts can be masked or unmasked with the IMASK register. Individual interrupt requests are logically ANDed with the bits in IMASK; the highest priority unmasked interrupt is then selected. The emulation, power-down, and reset interrupts are nonmaskable with the IMASK register, but software can use the DIS INT instruction to mask the power-down interrupt.

The Interrupt Control (ICNTL) register controls interrupt nesting and enables or disables interrupts globally.

The general-purpose Programmable Flag (PFx) pins can be configured as outputs, can implement software interrupts, and (as inputs) can implement hardware interrupts. Programmable Flag pin interrupts can be configured for level-sensitive, single edge-sensitive, or dual edge-sensitive operation.

Table 3. Interrupt Control (ICNTL) Register Bits

Bit	Description
0–3	Reserved
4	Interrupt Nesting Enable
5	Global Interrupt Enable
6	Reserved
7	MAC-Biased Rounding Enable
8–9	Reserved
10	PC Stack Interrupt Enable
11	Loop Stack Interrupt Enable
12–15	Reserved

The IRPTL register is used to force and clear interrupts. On-chip stacks preserve the processor status and are automatically maintained during interrupt handling. To support interrupt, loop, and subroutine nesting, the PC stack is 33 levels deep, the loop stack is eight levels deep, and the status stack is 16 levels deep. To prevent stack overflow, the PC stack can generate a stack-level interrupt if the PC stack falls below three locations full or rises above 28 locations full.

The following instructions globally enable or disable interrupt servicing, regardless of the state of IMASK.

```
ENA INT;
DIS INT;
```

At reset, interrupt servicing is disabled.

For quick servicing of interrupts, a secondary set of DAG and computational registers exist. Switching between the primary and secondary registers lets programs quickly service interrupts, while preserving the DSP's state.

DMA Controller

The ADSP-2191M has a DMA controller that supports automated data transfers with minimal overhead for the DSP core. Cycle stealing DMA transfers can occur between the ADSP-2191M's internal memory and any of its DMA-capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interface. DMA-capable peripherals include the Host port, SPORTs, SPI ports, and UART. Each individual DMA-capable peripheral has a dedicated DMA channel. To describe each DMA sequence, the DMA controller uses a set of parameters—called a DMA descriptor. When successive DMA sequences are needed, these DMA descriptors can be linked or chained together, so the completion of one DMA sequence auto-initiates and starts the next sequence. DMA sequences do not contend for bus access with the DSP core; instead DMAs "steal" cycles to access memory.

All DMA transfers use the DMA bus shown in the functional block diagram on page 1. Because all of the peripherals use the same bus, arbitration for DMA bus access is needed. The arbitration for DMA bus access appears in Table 4.

Table 4. I/O Bus Arbitration Priority

DMA Bus Master	Arbitration Priority
SPORT0 Receive DMA	0—Highest
SPORT1 Receive DMA	1
SPORT2 Receive DMA	2
SPORT0 Transmit DMA	3
SPORT1 Transmit DMA	4
SPORT2 Transmit DMA	5
SPI0 Receive/Transmit DMA	6
SPI1 Receive/Transmit DMA	7
UART Receive DMA	8
UART Transmit DMA	9
Host Port DMA	10
Memory DMA	11—Lowest

Host Port

The ADSP-2191M's Host port functions as a slave on the external bus of an external Host. The Host port interface lets a Host read from or write to the DSP's memory space, boot space, or internal I/O space. Examples of Hosts include external microcontrollers, microprocessors, or ASICs.

The Host port is a multiplexed address and data bus that provides both an 8-bit and a 16-bit data path and operates using an asynchronous transmission protocol. Through this port, an off-chip Host can directly access the DSP's entire memory space map, boot memory space, and internal I/O space. To access the DSP's internal memory space, a Host steals one cycle per access from the DSP. A Host access to the DSP's external memory uses the external port interface and does not stall (or steal cycles from) the DSP's core. Because a Host can access internal I/O memory space, a Host can control any of the DSP's I/O mapped peripherals.

The Host port is most efficient when using the DSP as a slave and uses DMA to automate the incrementing of addresses for these accesses. In this case, an address does not have to be transferred from the Host for every data transfer.

Host Port Acknowledge (HACK) Modes

The Host port supports a number of modes (or protocols) for generating a HACK output for the host. The host selects ACK or Ready Modes using the HACK_P and HACK pins. The Host port also supports two modes for address control: Address Latch Enable (ALE) and Address Cycle Control (ACC) modes. The DSP auto-detects ALE versus ACC Mode from the HALE and $\overline{\text{HWR}}$ inputs.

The Host port HACK signal polarity is selected (only at reset) as active high or active low, depending on the value driven on the HACK_P pin.The HACK polarity is stored into the Host port configuration register as a read only bit.

The DSP uses HACK to indicate to the Host when to complete an access. For a read transaction, a Host can proceed and complete an access when valid data is present in the read buffer and the Host port is not busy doing a write. For a write transactions, a Host can complete an access when the write buffer is not full and the Host port is not busy doing a write.

Two mode bits in the Host Port configuration register HPCR [7:6] define the functionality of the HACK line. HPCR6 is initialized at reset based on the values driven on HACK and HACK_P pins (shown in Table 5); HPCR7 is always cleared (0) at reset. HPCR [7:6] can be modified after reset by a write access to the Host port configuration register.

Table 5. Host Port Acknowledge Mode Selection

Values Driven At Reset		HPCR [7:6] Initial Values		Acknowledge Mode
HACK_P	HACK	Bit 7	Bit 6	
0	0	0	1	Ready Mode
0	1	0	0	ACK Mode
1	0	0	0	ACK Mode
1	1	0	1	Ready Mode

The functional modes selected by HPCR [7:6] are as follows (assuming active high signal):

- **ACK Mode**—Acknowledge is active on strobes; HACK goes high from the leading edge of the strobe to indicate when the access can complete. After the Host samples the HACK active, it can complete the access by removing the strobe.The Host port then removes the HACK.
- **Ready Mode**—Ready active on strobes, goes low to insert waitstate during the access.If the Host port cannot complete the access, it deasserts the HACK/READY line. In this case, the Host has to extend the access by keeping the strobe asserted. When the Host samples the HACK asserted, it can then proceed and complete the access by deasserting the strobe.

While in Address Cycle Control (ACC) mode and the ACK or Ready acknowledge modes, the HACK is returned active for any address cycle.

Host Port Chip Selects

There are two chip-select signals associated with the Host port: $\overline{\text{HCMS}}$ and $\overline{\text{HCIOMS}}$. The Host Chip Memory Select ($\overline{\text{HCMS}}$) lets the Host select the DSP and directly access the DSP's internal/external memory space or boot memory space. The Host Chip I/O Memory Select ($\overline{\text{HCIOMS}}$) lets the Host select the DSP and directly access the DSP's internal I/O memory space.

Before starting a direct access, the Host configures Host port interface registers, specifying the width of external data bus (8- or 16-bit) and the target address page (in the IJPG register). The DSP generates the needed memory select signals during the access, based on the target address. The Host port interface combines the data from one, two, or three consecutive Host accesses (up to one 24-bit value) into a single DMA bus access to prefetch Host direct reads or to post direct writes. During assembly of larger words, the Host port interface asserts ACK for

each byte access that does not start a read or complete a write. Otherwise, the Host port interface asserts ACK when it has completed the memory access successfully.

DSP Serial Ports (SPORTs)
The ADSP-2191M incorporates three complete synchronous serial ports (SPORT0, SPORT1, and SPORT2) for serial and multiprocessor communications. The SPORTs support the following features:

- Bidirectional operation—each SPORT has independent transmit and receive pins.
- Double-buffered transmit and receive ports—each port has a data register for transferring data words to and from memory and shift registers for shifting data in and out of the data registers.
- Clocking—each transmit and receive port can either use an external serial clock (40 MHz) or generate its own, in frequencies ranging from 19 Hz to 40 MHz.
- Word length—each SPORT supports serial data words from 3 to 16 bits in length transferred in Big Endian (MSB) or Little Endian (LSB) format.
- Framing—each transmit and receive port can run with or without frame sync signals for each data word. Frame sync signals can be generated internally or externally, active high or low, and with either of two pulsewidths and early or late frame sync.
- Companding in hardware—each SPORT can perform A-law or μ-law companding according to ITU recommendation G.711. Companding can be selected on the transmit and/or receive channel of the SPORT without additional latencies.
- DMA operations with single-cycle overhead—each SPORT can automatically receive and transmit multiple buffers of memory data, one data word each DSP cycle. Either the DSP's core or a Host processor can link or chain sequences of DMA transfers between a SPORT and memory. The chained DMA can be dynamically allocated and updated through the DMA descriptors (DMA transfer parameters) that set up the chain.
- Interrupts—each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer or buffers through DMA.
- Multichannel capability—each SPORT supports the H.100 standard.

Serial Peripheral Interface (SPI) Ports
The DSP has two SPI-compatible ports that enable the DSP to communicate with multiple SPI-compatible devices. These ports are multiplexed with SPORT2, so either SPORT2 or the SPI ports are active, depending on the state of the OPMODE pin during hardware reset.

The SPI interface uses three pins for transferring data: two data pins (Master Output-Slave Input, MOSIx, and Master Input-Slave Output, MISOx) and a clock pin (Serial Clock, SCKx). Two SPI chip select input pins ($\overline{\text{SPISSx}}$) let other SPI devices select the DSP, and fourteen SPI chip select output pins (SPIxSEL7–1) let the DSP select other SPI devices. The SPI select pins are reconfigured Programmable Flag pins. Using these pins, the SPI ports provide a full duplex, synchronous serial interface, which supports both master and slave modes and multimaster environments.

Each SPI port's baud rate and clock phase/polarities are programmable (see equation below for SPI clock rate calculation), and each has an integrated DMA controller, configurable to support both transmit and receive data streams. The SPI's DMA controller can only service unidirectional accesses at any given time.

$$SPI\ Clock\ Rate = \frac{HCLK}{2 \times SPIBAUD}$$

During transfers, the SPI ports simultaneously transmit and receive by serially shifting data in and out on their two serial data lines. The serial clock line synchronizes the shifting and sampling of data on the two serial data lines.

UART Port
The UART port provides a simplified UART interface to another peripheral or Host. It performs full duplex, asynchronous transfers of serial data. Options for the UART include support for 5–8 data bits; 1 or 2 stop bits; and none, even, or odd parity. The UART port supports two modes of operation:

- Programmed I/O

  The DSP's core sends or receives data by writing or reading I/O-mapped THR or RBR registers, respectively. The data is double-buffered on both transmit and receive.
- DMA (direct memory access)

  The DMA controller transfers both transmit and receive data. This reduces the number and frequency of interrupts required to transfer data to and from memory. The UART has two dedicated DMA channels. These DMA channels have lower priority than most DMA channels because of their relatively low service rates.

The UART's baud rate (see following equation for UART clock rate calculation), serial data format, error code generation and status, and interrupts are programmable:

- Supported bit rates range from 9.5 bits to 5M bits per second (80 MHz peripheral clock).
- Supported data formats are 7- to 12-bit frames.
- Transmit and receive status can be configured to generate maskable interrupts to the DSP's core.

The timers can be used to provide a hardware-assisted autobaud detection mechanism for the UART interface.

$$UART\ Clock\ Rate = \frac{HCLK}{16 \times D}$$

Where D is the programmable divisor = 1 to 65536.

Programmable Flag (PFx) Pins
The ADSP-2191M has 16 bidirectional, general-purpose I/O, Programmable Flag (PF15–0) pins. The PF7–0 pins are dedicated to general-purpose I/O. The PF15–8 pins serve either as general-purpose I/O pins (if the DSP is connected to an 8-bit external data bus) or serve as DATA15–8 lines (if the DSP is connected to a 16-bit external data bus). The Programmable Flag pins have special functions for clock multiplier selection and for SPI port operation. For more information, see Serial Peripheral Interface (SPI) Ports on page 9 and Clock Signals on page 11. Ten memory-mapped registers control operation of the Programmable Flag pins:

- Flag Direction register

  Specifies the direction of each individual PFx pin as input or output.

- Flag Control and Status registers

  Specify the value to drive on each individual PFx output pin. As input, software can predicate instruction execution on the value of individual PFx input pins captured in this register. One register sets bits, and one register clears bits.

- Flag Interrupt Mask registers

  Enable and disable each individual PFx pin to function as an interrupt to the DSP's core. One register sets bits to enable interrupt function, and one register clears bits to disable interrupt function. Input PFx pins function as hardware interrupts, and output PFx pins function as software interrupts—latching in the IMASK and IRPTL registers.

- Flag Interrupt Polarity register

  Specifies the polarity (active high or low) for interrupt sensitivity on each individual PFx pin.

- Flag Sensitivity registers

  Specify whether individual PFx pins are level- or edge-sensitive and specify—if edge-sensitive—whether just the rising edge or both the rising and falling edges of the signal are significant. One register selects the type of sensitivity, and one register selects which edges are significant for edge-sensitivity.

Low Power Operation
The ADSP-2191M has four low-power options that significantly reduce the power dissipation when the device operates under standby conditions. To enter any of these modes, the DSP executes an IDLE instruction. The ADSP-2191M uses configuration of the PDWN, STOPCK, and STOPALL bits in the PLLCTL register to select between the low-power modes as the DSP executes the IDLE. Depending on the mode, an IDLE shuts off clocks to different parts of the DSP in the different modes. The low power modes are:

- Idle
- Power-Down Core
- Power-Down Core/Peripherals
- Power-Down All

Idle Mode
When the ADSP-2191M is in Idle mode, the DSP core stops executing instructions, retains the contents of the instruction pipeline, and waits for an interrupt. The core clock and peripheral clock continue running.

To enter Idle mode, the DSP can execute the IDLE instruction anywhere in code. To exit Idle mode, the DSP responds to an interrupt and (after two cycles of latency) resumes executing instructions with the instruction after the IDLE.

Power-Down Core Mode
When the ADSP-2191M is in Power-Down Core mode, the DSP core clock is off, but the DSP retains the contents of the pipeline and keeps the PLL running. The peripheral bus keeps running, letting the peripherals receive data.

To enter Power-Down Core mode, the DSP executes an IDLE instruction after performing the following tasks:

- Enter a power-down interrupt service routine
- Check for pending interrupts and I/O service routines
- Clear (= 0) the PDWN bit in the PLLCTL register
- Clear (= 0) the STOPALL bit in the PLLCTL register
- Set (= 1) the STOPCK bit in the PLLCTL register

To exit Power-Down Core mode, the DSP responds to an interrupt and (after two cycles of latency) resumes executing instructions with the instruction after the IDLE.

Power-Down Core/Peripherals Mode
When the ADSP-2191M is in Power-Down Core/Peripherals mode, the DSP core clock and peripheral bus clock are off, but the DSP keeps the PLL running. The DSP does not retain the contents of the instruction pipeline.The peripheral bus is stopped, so the peripherals cannot receive data.

To enter Power-Down Core/Peripherals mode, the DSP executes an IDLE instruction after performing the following tasks:

- Enter a power-down interrupt service routine
- Check for pending interrupts and I/O service routines
- Clear (= 0) the PDWN bit in the PLLCTL register
- Set (= 1) the STOPALL bit in the PLLCTL register

To exit Power-Down Core/Peripherals mode, the DSP responds to a wake-up event and (after five to six cycles of latency) resumes executing instructions with the instruction after the IDLE.

Power-Down All Mode
When the ADSP-2191M is in Power-Down All mode, the DSP core clock, the peripheral clock, and the PLL are all stopped. The DSP does not retain the contents of the instruction pipeline. The peripheral bus is stopped, so the peripherals cannot receive data.

To enter Power-Down All mode, the DSP executes an IDLE instruction after performing the following tasks:

- Enter a power-down interrupt service routine
- Check for pending interrupts and I/O service routines
- Set (= 1) the PDWN bit in the PLLCTL register

To exit Power-Down Core/Peripherals mode, the DSP responds to an interrupt and (after 500 cycles to restabilize the PLL) resumes executing instructions with the instruction after the IDLE.

Clock Signals

The ADSP-2191M can be clocked by a crystal oscillator or a buffered, shaped clock derived from an external clock oscillator. If a crystal oscillator is used, the crystal should be connected across the CLKIN and XTAL pins, with two capacitors and a 1 MΩ shunt resistor connected as shown in Figure 3. Capacitor values are dependent on crystal type and should be specified by the crystal manufacturer. A parallel-resonant, fundamental frequency, microprocessor-grade crystal should be used for this configuration.

If a buffered, shaped clock is used, this external clock connects to the DSP's CLKIN pin. CLKIN input cannot be halted, changed, or operated below the specified frequency during normal operation. When an external clock is used, the XTAL input must be left unconnected.

The DSP provides a user-programmable 1× to 32× multiplication of the input clock, including some fractional values, to support 128 external to internal (DSP core) clock ratios. The MSEL6–0, BYPASS, and DF pins decide the PLL multiplication factor at reset. At runtime, the multiplication factor can be controlled in software. The combination of pullup and pull-down resistors in Figure sets up a core clock ratio of 6:1, which produces a 150 MHz core clock from the 25 MHz input. For other clock multiplier settings, see the *ADSP-219x/2191 DSP Hardware Reference*.

The peripheral clock is supplied to the CLKOUT pin.

All on-chip peripherals for the ADSP-2191M operate at the rate set by the peripheral clock. The peripheral clock is either equal to the core clock rate or one-half the DSP core clock rate. This selection is controlled by the IOSEL bit in the PLLCTL register. The maximum core clock is 160 MHz and the maximum peripheral clock is 80 MHz—the combination of the input clock and core/peripheral clock ratios may not exceed these limits.

Reset

The $\overline{\text{RESET}}$ signal initiates a master reset of the ADSP-2191M. The $\overline{\text{RESET}}$ signal must be asserted during the powerup sequence to assure proper initialization. $\overline{\text{RESET}}$ during initial powerup must be held long enough to allow the internal clock to stabilize.

The powerup sequence is defined as the total time required for the crystal oscillator circuit to stabilize after a valid V_{DD} is applied to the processor, and for the internal phase-locked loop (PLL) to lock onto the specific crystal frequency. A minimum of 100 µs ensures that the PLL has locked, but does not include the crystal oscillator start-up time. During this powerup sequence the $\overline{\text{RESET}}$ signal should be held low. On any subsequent resets, the $\overline{\text{RESET}}$ signal must meet the minimum pulsewidth specification, t_{WRST}.

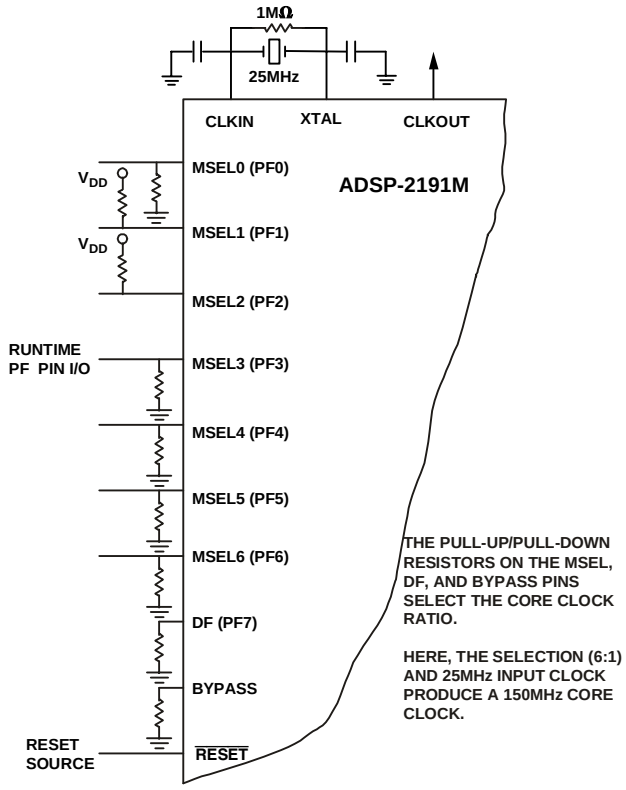


Figure 3. External Crystal Connections

The $\overline{\text{RESET}}$ input contains some hysteresis. If using an RC circuit to generate your $\overline{\text{RESET}}$ signal, the circuit should use an external Schmidt trigger.

The master reset sets all internal stack pointers to the empty stack condition, masks all interrupts, and resets all registers to their default values (where applicable). When $\overline{\text{RESET}}$ is released, if there is no pending bus request and the chip is configured for booting, the boot-loading sequence is performed. Program control jumps to the location of the on-chip boot ROM (0xFF 0000).

Power Supplies

The ADSP-2191M has separate power supply connections for the internal (V_{DDINT}) and external (V_{DDEXT}) power supplies. The internal supply must meet the 2.5 V requirement. The external supply must be connected to a 3.3 V supply. All external supply pins must be connected to the same supply.

Powerup Sequence

Power up together the two supplies V_{DDEXT} and V_{DDINT}. If they cannot be powered up together, power up the internal (core) supply first (powering up the core supply first reduces the risk of latchup events.

Booting Modes

The ADSP-2191M has five mechanisms (listed in Table 6) for automatically loading internal program memory after reset. Two No-boot modes are also supported.

Table 6. Select Boot Mode (OPMODE, BMODE1, and BMODE0)

OPMODE	BMODE1	BMODE0	Function
0	0	0	Execute from external memory 16 bits (No Boot)
0	0	1	Boot from EPROM
0	1	0	Boot from Host
0	1	1	Reserved
1	0	0	Execute from external memory 8 bits (No Boot)
1	0	1	Boot from UART
1	1	0	Boot from SPI, up to 4K bits
1	1	1	Boot from SPI, >4K bits up to 512K bits

The OPMODE, BMODE1, and BMODE0 pins, sampled during hardware reset, and three bits in the Reset Configuration Register implement these modes:

- Execute from memory external 16 bits—The memory boot routine located in boot ROM memory space executes a boot-stream-formatted program located at address 0x010000 of boot memory space, packing 16-bit external data into 24-bit internal data. The External Port Interface is configured for the default clock multiplier (128) and read waitstates (7).
- Boot from EPROM—The EPROM boot routine located in boot ROM memory space fetches a boot-stream-formatted program located at physical address 0x00 0000 of boot memory space, packing 8- or 16-bit external data into 24-bit internal data. The External Port Interface is configured for the default clock multiplier (32) and read waitstates (7).
- Boot from Host—The (8- or 16-bit) Host downloads a boot-stream-formatted program to internal or external memory. The Host's boot routine is located in internal ROM memory space and uses the top 16 locations of Page 0 program memory and the top 272 locations of Page 0 data memory.

  The internal boot ROM sets semaphore A (an IO register within the Host port) and then polls until the semaphore is reset. Once detected, the internal boot ROM will remap the interrupt vector table to Page 0 internal memory and jump to address 0x00 0000 internal memory. From the point of view of the host interface, an external host has full control of the DSP's memory map. The Host has the freedom to directly write internal memory, external memory, and internal I/O memory space. The DSP core execution is held off until the Host clears the semaphore register. This strategy allows the maximum flexibility for the Host to boot in the program and data code, by leaving it up to the programmer.
- Execute from memory external 8 bits (No Boot)—Execution starts from Page 1 of external memory space, packing either 8- or 16-bit external data into 24-bit internal data. The External Port Interface is configured for the default clock multiplier (128) and read waitstates (7).
- Boot from UART—The Host downloads boot-stream-formatted program using an autobaud handshake sequence. The Host agent selects a baud rate within the UART's clocking capabilities. After a hardware reset, the DSP's UART expects a 0xAA character (eight bits data, one start bit, one stop bit, no parity bit) on the RXD pin to determine the bit rate; and then replies with an OK string. Once the host receives this OK it downloads the boot stream without further handshake.The UART boot routine is located in internal ROM memory space and uses the top 16 locations of Page 0 program memory and the top 272 locations of Page 0 data memory.
- Boot from SPI, up to 4K bits—The SPI0 port uses the SPI0SEL1 (reconfigured PF2) output pin to select a single serial EEPROM device, submits a read command at address 0x00, and begins clocking consecutive data into internal or external memory. Use only SPI-compatible EEPROMs of ≤ 4K bit (12-bit address range). The SPI0 boot routine located in internal ROM memory space executes a boot-stream-formatted program, using the top 16 locations of Page 0 program memory and the top 272 locations of Page 0 data memory. The SPI boot configuration is SPIBAUD0=60 (decimal), CPHA=1, CPOL=1, 8-bit data, and MSB first.
- Boot from SPI, from >4K bits to 512K bits—The SPI0 port uses the SPI0SEL1 (re-configured PF2) output pin to select a single serial EEPROM device, submits a read command at address 0x00, and begins clocking consecutive data into internal or external memory. Use only SPI-compatible EEPROMs of ≥ 4K bit (16-bit address range). The SPI0 boot routine, located in internal ROM memory space, executes a boot-stream-formatted program, using the top 16 locations of Page 0 program memory and the top 272 locations of Page 0 data memory.

As indicated in Table 6, the OPMODE pin has a dual role, acting as a boot mode select during reset and determining SPORT or SPI operation at runtime. If the OPMODE pin at reset is the opposite of what is needed in an application during runtime, the application needs to set the OPMODE bit appropriately during runtime prior to using the corresponding peripheral.

Bus Request and Bus Grant

The ADSP-2191M can relinquish control of the data and address buses to an external device. When the external device requires access to the bus, it asserts the bus request ($\overline{BR}$) signal. The ($\overline{BR}$) signal is arbitrated with core and peripheral requests. External Bus requests have the lowest priority. If no other internal request is pending, the external bus request will be granted. Because of

synchronizer and arbitration delays, bus grants will be provided with a minimum of three peripheral clock delays. ADSP-2191M DSPs will respond to the bus grant by:

- Three-stating the data and address buses and the $\overline{MS3–0}$, $\overline{BMS}$, $\overline{IOMS}$, $\overline{RD}$, and $\overline{WR}$ output drivers.
- Asserting the bus grant ($\overline{BG}$) signal.

The ADSP-2191M will halt program execution if the bus is granted to an external device and an instruction fetch or data read/write request is made to external general-purpose or peripheral memory spaces. If an instruction requires two external memory read accesses, bus requests will not be granted between the two accesses. If an instruction requires an external memory read and an external memory write access, the bus may be granted between the two accesses. The external memory interface can be configured so that the core will have exclusive use of the interface. DMA and Bus Requests will be granted. When the external device releases $\overline{BR}$, the DSP releases $\overline{BG}$ and continues program execution from the point at which it stopped.

The bus request feature operates at all times, even while the DSP is booting and $\overline{RESET}$ is active.

The ADSP-2191M asserts the $\overline{BGH}$ pin when it is ready to start another external port access, but is held off because the bus was previously granted. This mechanism can be extended to define more complex arbitration protocols for implementing more elaborate multimaster systems.

Instruction Set Description

The ADSP-2191M assembly language instruction set has an algebraic syntax that was designed for ease of coding and readability. The assembly language, which takes full advantage of the processor's unique architecture, offers the following benefits:

- ADSP-219x assembly language syntax is a superset of and source-code-compatible (except for two data registers and DAG base address registers) with ADSP-218x family syntax. It may be necessary to restructure ADSP-218x programs to accommodate the ADSP-2191M's unified memory space and to conform to its interrupt vector map.
- The algebraic syntax eliminates the need to remember cryptic assembler mnemonics. For example, a typical arithmetic add instruction, such as AR = AX0 + AY0, resembles a simple equation.
- Every instruction, but two, assembles into a single, 24-bit word that can execute in a single instruction cycle. The exceptions are two dual word instructions. One writes 16- or 24-bit immediate data to memory, and the other is an absolute jump/call with the 24-bit address specified in the instruction.
- Multifunction instructions allow parallel execution of an arithmetic, MAC, or shift instruction with up to two fetches or one write to processor memory space during a single instruction cycle.
- Program flow instructions support a wider variety of conditional and unconditional jumps/calls and a larger set of conditions on which to base execution of conditional instructions.

Development Tools

The ADSP-2191M is supported with a complete set of software and hardware development tools, including Analog Devices emulators and VisualDSP++® development environment. The same emulator hardware that supports other ADSP-219x DSPs, also fully emulates the ADSP-2191M.

The VisualDSP++ project management environment lets programmers develop and debug an application. This environment includes an easy-to-use assembler that is based on an algebraic syntax; an archiver (librarian/library builder), a linker, a loader, a cycle-accurate instruction-level simulator, a C/C++ compiler, and a C/C++ run-time library that includes DSP and mathematical functions. Two key points for these tools are:

- Compiled ADSP-219x C/C++ code efficiency—the compiler has been developed for efficient translation of C/C++ code to ADSP-219x assembly. The DSP has architectural features that improve the efficiency of compiled C/C++ code.
- ADSP-218x family code compatibility—The assembler has legacy features to ease the conversion of existing ADSP-218x applications to the ADSP-219x.

Debugging both C/C++ and assembly programs with the VisualDSP++ debugger, programmers can:

- View mixed C/C++ and assembly code (interleaved source and object information)
- Insert break points
- Set conditional breakpoints on registers, memory, and stacks
- Trace instruction execution
- Perform linear or statistical profiling of program execution
- Fill, dump, and graphically plot the contents of memory
- Source level debugging
- Create custom debugger windows

The VisualDSP++ IDE lets programmers define and manage DSP software development. Its dialog boxes and property pages let programmers configure and manage all of the ADSP-219x development tools, including the syntax highlighting in the VisualDSP++ editor. This capability permits:

- Control how the development tools process inputs and generate outputs.
- Maintain a one-to-one correspondence with the tool's command line switches.

Analog Devices DSP emulators use the IEEE 1149.1 JTAG test access port of the ADSP-2191M processor to monitor and control the target board processor during emulation. The emulator provides full-speed emulation, allowing inspection and modification of memory, registers, and processor stacks. Nonintrusive in-circuit emulation is assured by the use of the processor's JTAG interface—the emulator does not affect target system loading or timing.

In addition to the software and hardware development tools available from Analog Devices, third parties provide a wide range of tools supporting the ADSP-219x processor family. Hardware tools include ADSP-219x PC plug-in cards. Third party software tools include DSP libraries, real-time operating systems, and block diagram design tools.

Designing an Emulator-Compatible DSP Board (Target)

The White Mountain DSP (Product Line of Analog Devices, Inc.) family of emulators are tools that every DSP developer needs to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG Test Access Port (TAP) on each JTAG DSP. The emulator uses the TAP to access the internal features of the DSP, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The DSP must be halted to send data and commands, but once an operation has been completed by the emulator, the DSP system is set running at full speed with no impact on system timing.

To use these emulators, the target's design must include the interface between an Analog Devices JTAG DSP and the emulation header on a custom DSP target board.

Target Board Header

The emulator interface to an Analog Devices JTAG DSP is a 14-pin header, as shown in Figure 4. The customer must supply this header on the target board in order to communicate with the emulator. The interface consists of a standard dual row 0.025" square post header, set on 0.1" × 0.1" spacing, with a minimum post length of 0.235". Pin 3 is the key position used to prevent the pod from being inserted backwards. This pin must be clipped on the target board.

Also, the clearance (length, width, and height) around the header must be considered. Leave a clearance of at least 0.15" and 0.10" around the length and width of the header, and reserve a height clearance to attach and detach the pod connector.

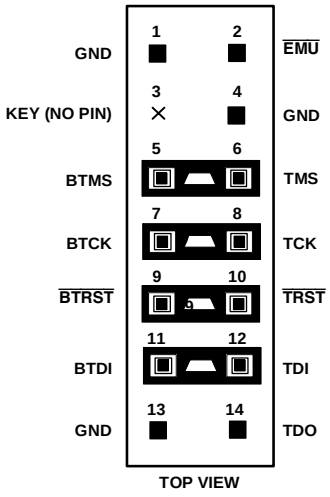


Figure 4. JTAG Target Board Connector for JTAG Equipped Analog Devices DSP (Jumpers in Place)

As can be seen in Figure 4, there are two sets of signals on the header. There are the standard JTAG signals TMS, TCK, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{EMU}}$ used for emulation purposes (via an emulator). There are also secondary JTAG signals BTMS, BTCK, BTDI, and $\overline{\text{BTRST}}$ that are optionally used for board-level (boundary scan) testing.

When the emulator is not connected to this header, place jumpers across BTMS, BTCK, $\overline{\text{BTRST}}$, and BTDI as shown in Figure 5. This holds the JTAG signals in the correct state to allow the DSP to run free. Remove all the jumpers when connecting the emulator to the JTAG header.

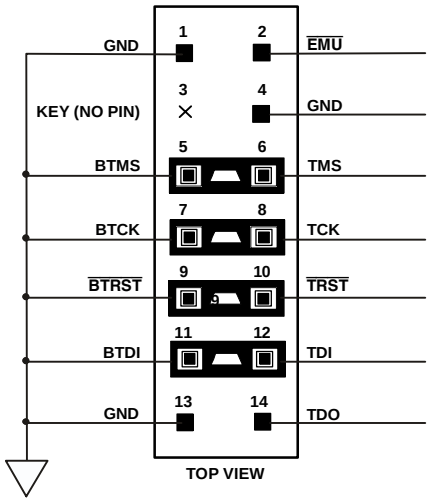


Figure 5. JTAG Target Board Connector with No Local Boundary Scan

JTAG Emulator Pod Connector

Figure 6 details the dimensions of the JTAG pod connector at the 14-pin target end. Figure 7 displays the keep-out area for a target board header. The keep-out area allows the pod connector to properly seat onto the target board header. This board area should contain no components (chips, resistors, capacitors, etc.). The dimensions are referenced to the center of the 0.25" square post pin.

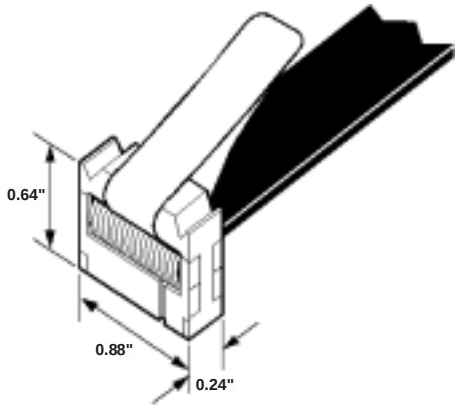


Figure 6. JTAG Pod Connector Dimensions

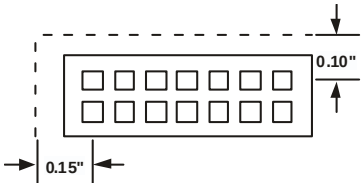


Figure 7. JTAG Pod Connector Keep-Out Area

Design-for-Emulation Circuit Information

For details on target board design issues including: single processor connections, multiprocessor scan chains, signal buffering, signal termination, and emulator pod logic, see the *EE-68: Analog Devices JTAG Emulation Technical Reference* on the Analog Devices website (www.analog.com)—use site search on "EE-68." This document is updated regularly to keep pace with improvements to emulator support.

Additional Information

This data sheet provides a general overview of the ADSP-2191M architecture and functionality. For detailed information on the core architecture of the ADSP-219x family, refer to the *ADSP-219x/2191 DSP Hardware Reference*. For details on the instruction set, refer to the *ADSP-219x Instruction Set Reference*.

PIN FUNCTION DESCRIPTIONS

ADSP-2191M pin definitions are listed in Table 7. All ADSP-2191M inputs are asynchronous and can be asserted asynchronously to CLKIN (or to TCK for $\overline{\text{TRST}}$).

Tie or pull unused inputs to V_{DDEXT} or GND, except for ADDR21–0, DATA15–0, PF7-0, and inputs that have internal pull-up or pull-down resistors ($\overline{\text{TRST}}$, BMODE0, BMODE1, OPMODE, BYPASS, TCK, TMS, TDI, and $\overline{\text{RESET}}$)—these pins can be left floating. These pins have a logic-level hold circuit that prevents input from floating internally.

The following symbols appear in the Type column of Table : G = Ground, I = Input, O = Output, P = Power Supply, and T = Three-State.

Table 7. Pin Function Descriptions

Pin	Type	Function
A21–0	O/T	External Port Address Bus
D7–0	I/O/T	External Port Data Bus, least significant 8 bits
D15	I/O/T	Data 15 (if 16-bit external bus)/Programmable Flags 15 (if 8-bit external bus)/SPI1 Slave
/PF15	I/O	Select output 7 (if 8-bit external bus, when SPI1 enabled)
/SPI1SEL7	I	
D14	I/O/T	Data 14 (if 16-bit external bus)/Programmable Flags 14 (if 8-bit external bus)/SPI0 Slave
/PF14	I/O	Select output 7 (if 8-bit external bus, when SPI0 enabled)
/SPI0SEL7	I	
D13	I/O/T	Data 13 (if 16-bit external bus)/Programmable Flags 13 (if 8-bit external bus)/SPI1 Slave
/PF12	I/O	Select output 6 (if 8-bit external bus, when SPI1 enabled)
/SPI1SEL6	I	
D12	I/O/T	Data 12 (if 16-bit external bus)/Programmable Flags 12 (if 8-bit external bus)/SPI0 Slave
/PF12	I/O	Select output 6 (if 8-bit external bus, when SPI0 enabled)
/SPI0SEL6	I	
D11	I/O/T	Data 11 (if 16-bit external bus)/Programmable Flags 11 (if 8-bit external bus)/SPI1 Slave
/PF11	I/O	Select output 5 (if 8-bit external bus, when SPI1 enabled)
/SPI1SEL5	I	
D10	I/O/T	Data 10 (if 16-bit external bus)/Programmable Flags 10 (if 8-bit external bus)/SPI0 Slave
/PF10	I/O	Select output 5 (if 8-bit external bus, when SPI0 enabled)
/SPI0SEL5	I	
D9	I/O/T	Data 9 (if 16-bit external bus)/Programmable Flags 9 (if 8-bit external bus)/SPI1 Slave Select
/PF9	I/O	output 4 (if 8-bit external bus, when SPI1 enabled)
/SPI1SEL4	I	
D8	I/O/T	Data 8 (if 16-bit external bus)/Programmable Flags 8 (if 8-bit external bus)/SPI0 Slave Select
/PF8	I/O	output 4 (if 8-bit external bus, when SPI0 enabled)
/SPI0SEL4	I	
PF7	I/O/T	Programmable Flags 7/SPI1 Slave Select output 3 (when SPI0 enabled)/Divisor Frequency
/SPI1SEL3	I	(divisor select for PLL input during boot)
/DF	I	
PF6	I/O/T	Programmable Flags 6/SPI0 Slave Select output 3 (when SPI0 enabled)/Multiplier Select 6
/SPI0SEL3	I	(during boot)
/MSEL6	I	

Table 7. Pin Function Descriptions (continued)

Pin	Type	Function
PF5	I/O/T	Programmable Flags 5/SPI1 Slave Select output 2 (when SPI0 enabled)/Multiplier Select 5
/SPI1SEL2	I	(during boot)
/MSEL5	I	
PF4	I/O/T	Programmable Flags 4/SPI0 Slave Select output 2 (when SPI0 enabled)/Multiplier Select 4
/SPI0SEL2	I	(during boot)
/MSEL4	I	
PF3	I/O/T	Programmable Flags 3/SPI1 Slave Select output 1 (when SPI0 enabled)/Multiplier Select 3
/SPI1SEL1	I	(during boot)
/MSEL3	I	
PF2	I/O/T	Programmable Flags 2/SPI0 Slave Select output 1 (when SPI0 enabled)/Multiplier Select 2
/SPI0SEL1	I	(during boot)
/MSEL2	I	
PF1	I/O/T	Programmable Flags 1/SPI1 Slave Select input (when SPI1 enabled)/Multiplier Select 1
/SPISS1	I	(during boot)
/MSEL1	I	
PF0	I/O/T	Programmable Flags 0/SPI0 Slave Select input (when SPI0 enabled)/Multiplier Select 0
/SPISS0	I	(during boot)
/MSEL0	I	
$\overline{RD}$	O/T	External Port Read Strobe
$\overline{WR}$	O/T	External Port Write Strobe
ACK	I	External Port Access Ready Acknowledge
$\overline{BMS}$	O/T	External Port Boot Space Select
$\overline{IOMS}$	O/T	External Port IO Space Select
$\overline{MS3–0}$	O/T	External Port Memory Space Selects
$\overline{BR}$	I	External Port Bus Request
$\overline{BG}$	O	External Port Bus Grant
$\overline{BGH}$	O	External Port Bus Grant Hang
HAD15–0	I/O/T	Host Port Multiplexed Address and Data Bus
HA16	I	Host Port MSB of Address Bus
HACK_P	I	Host Port ACK Polarity
$\overline{HRD}$	I	Host Port Read Strobe
$\overline{HWR}$	I	Host Port Write Strobe
HACK	O	Host Port Access Ready Acknowledge
HALE	I	Host Port Address Latch Strobe or Address Cycle Control
$\overline{HCMS}$	I	Host Port Internal Memory–Internal I/O Memory–Boot Memory Select
$\overline{HCIOMS}$	I	Host Port Internal I/O Memory Select
CLKIN	I	Clock Input/Oscillator input
XTAL	O	Oscillator output
BMODE1–0	I	Boot Mode 1–0. The BMODE1 and BMODE0 pins have 85 kΩ internal pull-up resistors.
OPMODE	I	Operating Mode. The OPMODE pin has a 85 kΩ internal pull-up resistor.
CLKOUT	O	Clock Output
BYPASS	I	Phase-Lock-Loop (PLL) Bypass mode. The BYPASS pin has a 85 kΩ internal pull-up resistor.
RCLK1–0	I/O/T	SPORT1–0 Receive Clock
RCLK2/SCK1	I/O/T	SPORT2 Receive Clock/SPI1 Serial Clock
RFS1–0	I/O/T	SPORT1–0 Receive Frame Sync
RFS2/MOSI1	I/O/T	SPORT2 Receive Frame Sync/SPI1 Master-Output, Slave-Input data
TCLK1–0	I/O/T	SPORT1–0 Transmit Clock
TCLK2/SCK0	I/O/T	SPORT2 Transmit Clock/SPI0 Serial Clock
TFS1–0	I/O/T	SPORT1–0 Transmit Frame Sync
TFS2/MOSI0	I/O/T	SPORT2 Transmit Frame Sync/SPI0 Master-Output, Slave-Input data
DR1–0	I/T	SPORT1–0 Serial Data Receive
DR2/MISO1	I/O/T	SPORT2 Serial Data Receive/SPI1 Master-Input, Slave-Output data
DT1–0	O/T	SPORT1–0 Serial Data Transmit
DT2/MISO0	I/O/T	SPORT2 Serial Data Transmit/SPI0 Master-Input, Slave-Output data
TMR2–0	I/O/T	Timer output or capture

Table 7. Pin Function Descriptions (continued)

Pin	Type	Function
RXD	I	UART Serial Receive Data
TXD	O	UART Serial Transmit Data
$\overline{\text{RESET}}$	I	Processor Reset. Resets the ADSP-2191M to a known state and begins execution at the program memory location specified by the hardware reset vector address. The $\overline{\text{RESET}}$ input must be asserted (low) at powerup. The RESET pin has an 85 kΩ internal pull-up resistor.
TCK	I	Test Clock (JTAG). Provides a clock for JTAG boundary scan. The TCK pin has an 85 kΩ internal pull-up resistor.
TMS	I	Test Mode Select (JTAG). Used to control the test state machine. The TMS pin has an 85 kΩ internal pull-up resistor.
TDI	I	Test Data Input (JTAG). Provides serial data for the boundary scan logic. The TDI pin has a 85 kΩ internal pull-up resistor.
TDO	O	Test Data Output (JTAG). Serial scan output of the boundary scan path.
$\overline{\text{TRST}}$	I	Test Reset (JTAG). Resets the test state machine. $\overline{\text{TRST}}$ must be asserted (pulsed low) after powerup or held low for proper operation of the ADSP-2191M. The $\overline{\text{TRST}}$ pin has a 65 kΩ internal pull-down resistor.
$\overline{\text{EMU}}$	O	Emulation Status (JTAG). Must be connected to the ADSP-2191M emulator target board connector only.
V_{DDINT}	P	Core Power Supply. Nominally 2.5 V dc and supplies the DSP's core processor. (four pins)
V_{DDEXT}	P	I/O Power Supply. Nominally 3.3 V dc. (nine pins)
GND	G	Power Supply Return. (twelve pins)
NC		Do Not Connect. Reserved pins that must be left open and unconnected.

ADSP-2191M
SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

Parameter[1]		Test Conditions	K Grade (Commercial) Min	Max	B Grade (Industrial) Min	Max	Unit
V_{DDINT}	Internal (Core) Supply Voltage		2.37	2.63	2.37	2.63	V
V_{DDEXT}	External (I/O) Supply Voltage		2.97	3.6	2.97	3.6	V
V_{IH}	High Level Input Voltage	@ V_{DDINT} = max, V_{DDEXT} = max	2.0	V_{DDEXT}+0.3	2.0	V_{DDEXT}+0.3	V
V_{IL}	Low Level Input Voltage	@ V_{DDINT} = min, V_{DDEXT} = min	–0.3	0.8	–0.3	0.8	V
T_{AMB}	Ambient Operating Temperature		0	70	–40	+85	°C

[1]Specifications subject to change without notice.

ELECTRICAL CHARACTERISTICS

Parameter[1]		Test Conditions	K and B Grades Min	Typ	Max	Unit
V_{OH}	High Level Output Voltage[2]	@ V_{DDEXT} = min, I_{OH} = –0.5 mA	2.4			V
V_{OL}	Low Level Output Voltage[2]	@ V_{DDEXT} = min, I_{OL} = 2.0 mA			0.4	V
I_{IH}	High Level Input Current[3, 4]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max			10	µA
I_{IL}	Low Level Input Current[3, 5]	@ V_{DDEXT} = max, V_{IN} = 0 V			10	µA
I_{IHP}	High Level Input Current[5]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max	30		100	µA
I_{ILP}	Low Level Input Current[4]	@ V_{DDEXT} = max, V_{IN} = 0 V	20		70	µA
I_{OZH}	Three-State Leakage Current[6]	@ V_{DDEXT} = max, V_{IN} = V_{DD} max			10	µA
I_{OZL}	Three-State Leakage Current[6]	@ V_{DDEXT} = max, V_{IN} = 0 V			10	µA
C_{IN}	Input Capacitance[7, 8]	f_{IN} = 1 MHz, T_{CASE} = 25°C, V_{IN} = 2.5 V			8	pF

[1]Specifications subject to change without notice.
[2]Applies to output and bidirectional pins: DATA15–0, ADDR21–0, HAD15–0, $\overline{MS3–0}$, $\overline{IOMS}$, $\overline{RD}$, $\overline{WR}$, CLKOUT, HACK, PF7–0, TMR2–0, $\overline{BGH}$, $\overline{BG}$, DT0, DT1, DT2/MISO0, TCLK0, TCLK1, TCLK2/SCK0, RCLK0, RCLK1, RCLK2/SCK1, TFS0, TFS1, TFS2/MOSI0, RFS0, RFS1, RFS2/MOSI1, $\overline{BMS}$, TDO, TXD, $\overline{EMU}$, DR2/MISO1.
[3]Applies to input pins: ACK, $\overline{BR}$, $\overline{HCMS}$, $\overline{HCIOMS}$, HA16, HALE, $\overline{HRD}$, $\overline{HWR}$, CLKIN, DR0, DR1, RXD, HACK_P.
[4]Applies to input pins with internal pull-ups: BMODE0, BMODE1, OPMODE, BYPASS, TCK, TMS, TDI, $\overline{RESET}$.
[5]Applies to input pin with internal pull-down: TRST.
[6]Applies to three-statable pins: DATA15–0, ADDR21–0, $\overline{MS3–0}$, $\overline{RD}$, $\overline{WR}$, PF7–0, $\overline{BMS}$, $\overline{IOMS}$, TFSx, RFSx, TDO, $\overline{EMU}$, TCLKx, RCLKx, DTx, HAD15–0, TMR2–0.
[7]Applies to all signal pins.
[8]Guaranteed, but not tested.

ABSOLUTE MAXIMUM RATINGS

V_{DDINT} Internal (Core) Supply Voltage[1,2]. . –0.3 V to 3.0 V
V_{DDEXT} External (I/O) Supply Voltage . . . . –0.3 V to 4.6 V
V_{IL}–V_{IH} Input Voltage . . . . . –0.5 V to V_{DDEXT}+0.5 V
V_{OL}–V_{OH} Output Voltage Swing. –0.5 V to V_{DDEXT}+0.5 V
T_{STORE}Storage Temperature Range . . . . . .–65°C to 150°C
T_{LEAD}Lead Temperature of ST-144 (5 seconds) . . . . 185°C

[1]Specifications subject to change without notice.
[2]Stresses greater than those listed above may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions greater than 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

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADSP-2191M features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high-energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.



Power Dissipation

Using the operation-versus-current information in Table 8, designers can estimate the ADSP-2191M's internal power supply (V_{DDINT}) input current for a specific application, according to the formula for I_{DDINT} calculation beneath Table 8. For calculation of external supply current and total supply current, see Power Dissipation on page 41.

Table 8. Operation Types Versus Input Current

	K-Grade I_{DDINT}(mA) CCLK = 160 MHz				B-Grade I_{DDINT}(mA)[1] CCLK = 140 MHz			
	Core		Peripheral		Core		Peripheral	
Activity	Typ[1]	Max[2]	Typ[1]	Max[2]	Typ[1]	Max[2]	Typ[1]	Max[2]
Power Down[3]	100µA	600µA	0	50µA	100µA	500µA	0	50µA
Idle 1[4]	1	2	5	8	1	2	4	7
Idle 2[5]	1	2	60	70	1	2	55	62
Typical[6]	184	210	60	70	165	185	55	62
Peak[7]	215	240	60	70	195	210	55	62

[1]Test conditions: V_{DDINT}= 2.50 V; HCLK (peripheral clock) frequency = CCLK/2 (core clock/2) frequency; T_{AMB} = 25°C.
[2]Test conditions: V_{DDINT}= 2.65 V; HCLK (peripheral clock) frequency = CCLK/2 (core clock/2) frequency; T_{AMB} = 25°C.
[3]PLL, Core, peripheral clocks, and CLKIN are disabled.
[4]PLL is enabled and Core and peripheral clocks are disabled.
[5]Core CLK is disabled and peripheral clock is enabled.
[6]All instructions execute from internal memory. 50% of the instructions are repeat MACs with dual operand addressing, with changing data fetched using a linear address sequence. 50% of the instructions are type 3 instructions.
[7]All instructions execute from internal memory. 100% of the instructions are MACs with dual operand addressing, with changing data fetched using a linear address sequence.

$$I_{DDINT} = (\%Typical \times I_{DDINT\text{-}TYPICAL}) + (\%Idle \times I_{DDINT\text{-}IDLE}) + (\%Power\ Down \times I_{DDINT\text{-}PWRDWN})$$

ADSP-2191M
TIMING SPECIFICATIONS

This section contains timing information for the DSP's external signals. Use the exact information given. Do not attempt to derive parameters from the addition or subtraction of other information. While addition or subtraction would yield meaningful results for an individual device, the values given in this datasheet reflect statistical variations and worst cases. Consequently, parameters cannot be added meaningfully to derive longer times.

Switching characteristics specify how the processor changes its signals. No control is possible over this timing; circuitry external to the processor must be designed for compatibility with these signal characteristics. Switching characteristics indicate what the processor will do in a given circumstance. Switching characteristics can also be used to ensure that any timing requirement of a device connected to the processor (such as memory) is satisfied.

Timing requirements apply to signals that are controlled by circuitry external to the processor, such as the data input for a read operation.Timing requirements guarantee that the processor operates correctly with other devices.

Clock In and Clock Out Cycle Timing

Table 9 and Figure 8 describe clock and reset operations. Combinations of CLKIN and clock multipliers must not select core/peripheral clocks in excess of 160/80 MHz for commercial grade and 140/70 MHz for industrial grade, when the peripheral clock rate is one-half the core clock rate. If the peripheral clock rate is equal to the core clock rate, the maximum peripheral clock rate is 80 MHz for both commercial and industrial grade parts. The peripheral clock is supplied to the CLKOUT pins.

When changing from bypass mode to PLL mode, allow 512 HCLK cycles for the PLL to stabilize.

Table 9. Clock In and Clock Out Cycle Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{CKOD}	CLKOUT Delay from CLKIN	0	5.8	ns
t_{CKO}	CLKOUT Period[1]	12.5		ns
Timing Requirements				
t_{CK}	CLKIN Period[2, 3]	10	200	ns
t_{CKL}	CLKIN Low Pulse	4.5		ns
t_{CKH}	CLKIN High Pulse	4.5		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulsewidth Low	$200t_{CLKOUT}$		ns
t_{MSS}	MSELx/BYPASS Stable Before $\overline{\text{RESET}}$ Deasserted Setup	40		µs
t_{MSH}	MSELx/BYPASS Stable After $\overline{\text{RESET}}$ Deasserted Hold	1000		ns
t_{MSD}	MSELx/BYPASS Stable After $\overline{\text{RESET}}$ Asserted		200	ns
t_{PFD}	Flag Output Disable Time After $\overline{\text{RESET}}$ Asserted		10	ns

[1]CLKOUT jitter can be as great as 8 ns when CLKOUT frequency is less than 20 MHz. For frequencies greater than 20 MHz, jitter is less than 1 ns.
[2]In clock multiplier mode and MSEL6–0 set for 1:1 (or CLKIN=CCLK), $t_{CK}=t_{CCLK}$.
[3]In bypass mode, $t_{CK}=t_{CCLK}$.

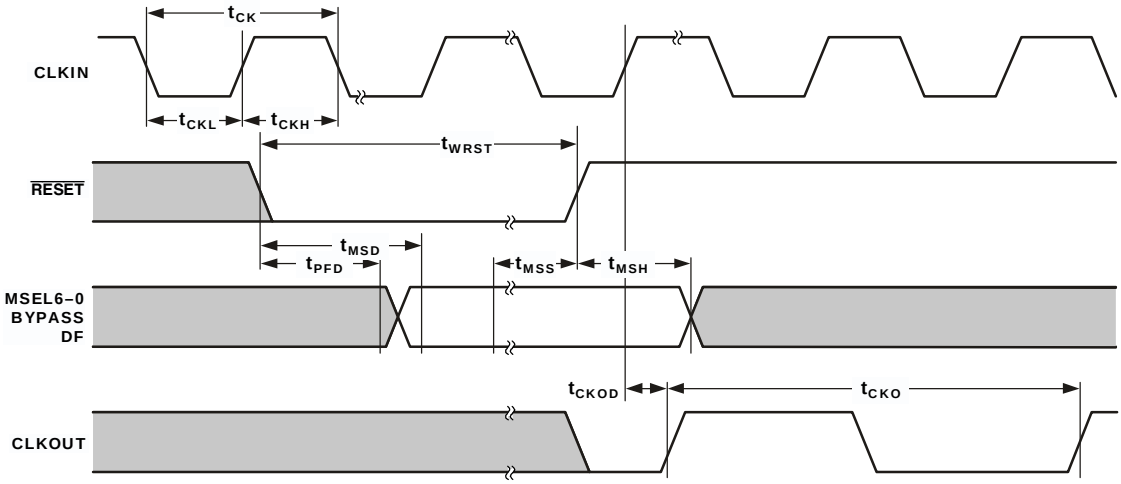


Figure 8. Clock In and Clock Out Cycle Timing

Programmable Flags Cycle Timing

Table 10 and Figure 9 describe Programmable Flag operations.

Table 10. Programmable Flags Cycle Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DFO}	Flag Output Delay with Respect to CLKOUT		7	ns
t_{HFO}	Flag Output Hold After CLKOUT High		6	ns
Timing Requirement				
t_{HFI}	Flag Input Hold is asynchronous	3		ns

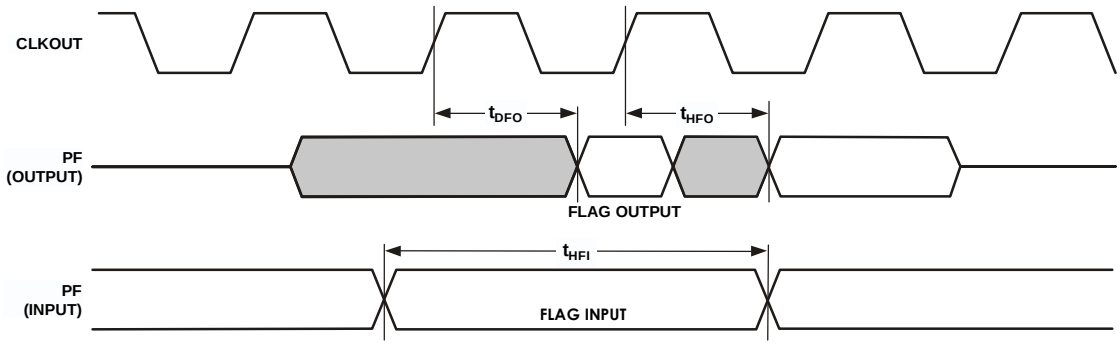


Figure 9. Programmable Flags Cycle Timing

Timer PWM_OUT Cycle Timing

Table 11 and Figure 10 describe timer expired operations. The input signal is asynchronous in "width capture mode" and has an absolute maximum input frequency of 40 MHz.

Table 11. Timer PWM_OUT Cycle Timing

Parameter		Min	Max	Unit
Switching Characteristic				
t_{HTO}	Timer Pulsewidth Output[1]	12.5	$(2^{32}-1)$ cycles	ns

[1]The minimum time for t_{HTO} is one cycle, and the maximum time for t_{HTO} equals $(2^{32}-1)$ cycles.

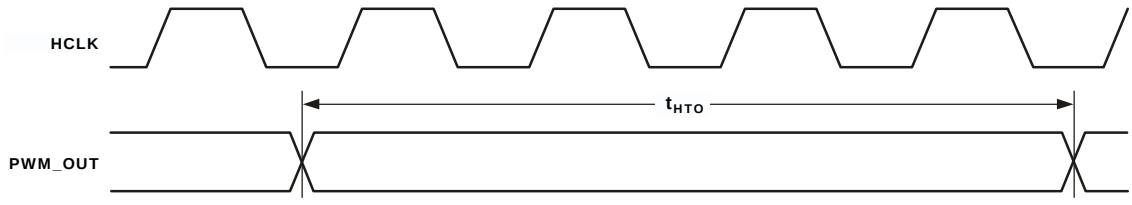


Figure 10. Timer PWM_OUT Cycle Timing

External Port Write Cycle Timing

Table 12 and Figure 11 describe external port write operations.

The external port lets systems extend read/write accesses in three ways: waitstates, ACK input, and combined waitstates and ACK. To add waits with ACK, the DSP must see ACK low at the rising edge of EMI clock. ACK low causes the DSP to wait, and the DSP requires two EMI clock cycles after ACK goes high to finish the access. For more information, see the External Port chapter in the *ADSP-219x/2191 DSP Hardware Reference*.

Table 12. External Port Write Cycle Timing

Parameter[1, 2]		Min	Max	Unit
Switching Characteristics				
t_{CSWS}	Chip Select Asserted to $\overline{WR}$ Asserted Delay	$0.5t_{HCLK}-4$		ns
t_{AWS}	Address Valid to $\overline{WR}$ Setup and Delay	$0.5t_{HCLK}-3$		ns
t_{WSCS}	$\overline{WR}$ Deasserted to Chip Select Deasserted	$0.5t_{HCLK}-4$		ns
t_{WSA}	$\overline{WR}$ Deasserted to Address Invalid	$0.5t_{HCLK}-3$		ns
t_{WW}	$\overline{WR}$ Strobe Pulsewidth	$t_{HCLK}-2+W^3$		ns
t_{CDA}	$\overline{WR}$ to Data Enable Access Delay		0	ns
t_{CDD}	$\overline{WR}$ to Data Disable Access Delay	$0.5t_{HCLK}-3$	$0.5t_{HCLK}+4$	ns
t_{DSW}	Data Valid to $\overline{WR}$ Deasserted Setup	$t_{HCLK}+1+W^3$	$t_{HCLK}+7+W^3$	ns
t_{DHW}	$\overline{WR}$ Deasserted to Data Invalid Hold Time; E_WHC[4]	3.4		ns
t_{DHW}	$\overline{WR}$ Deasserted to Data Invalid Hold Time; E_WHC[4]	$t_{HCLK}+3.4$		ns
t_{WWR}	$\overline{WR}$ Deasserted to $\overline{WR}$, $\overline{RD}$ Asserted	t_{HCLK}		
Timing Requirements				
t_{AKW}	ACK Strobe Pulsewidth	12.5		ns
t_{DWSAK}	ACK Delay from $\overline{WR}$ Low	0		ns

[1] t_{HCLK} is the peripheral clock period.
[2] These are timing parameters that are based on worst-case operating conditions.
[3] W = (number of waitstates specified in wait register) × t_{HCLK}.
[4] Write hold cycle–memory select control registers (MS × CTL).

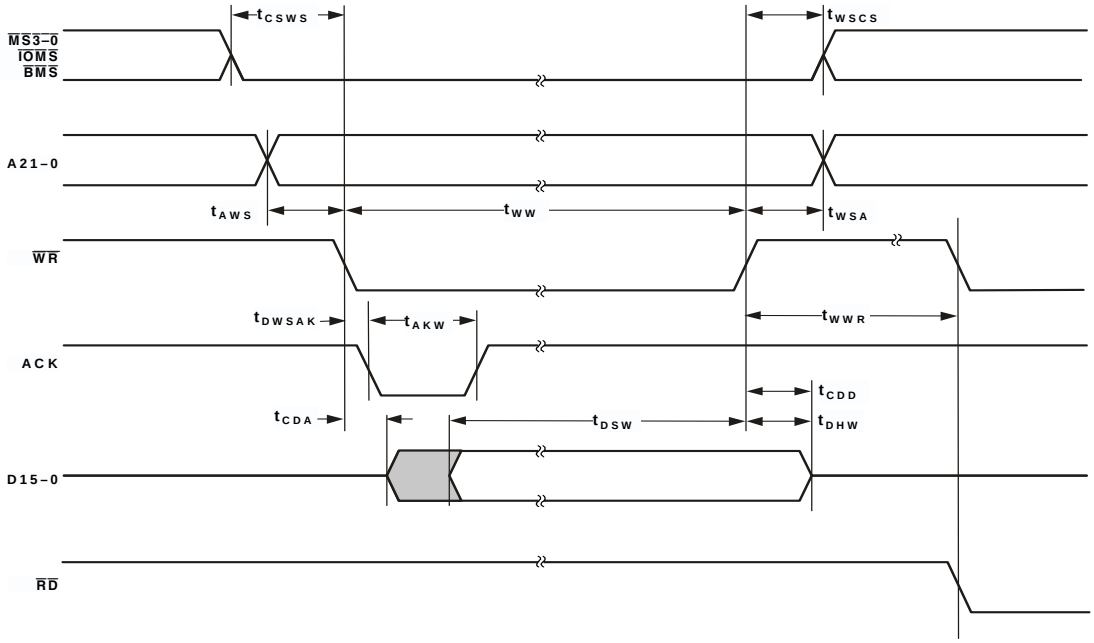


Figure 11. External Port Write Cycle Timing

External Port Read Cycle Timing

Table 13 and Figure 12 describe external port read operations. For additional information on the ACK signal, see the discussion on page 23.

Table 13. External Port Read Cycle Timing

Parameter[1, 2]		Min	Max	Unit
Switching Characteristics				
t_{CSRS}	Chip Select Asserted to $\overline{RD}$ Asserted Delay	$0.5t_{HCLK}-3$		ns
t_{ARS}	Address Valid to $\overline{RD}$ Setup and Delay	$0.5t_{HCLK}-3$		ns
t_{RSCS}	$\overline{RD}$ Deasserted to Chip Select Deasserted Setup	$0.5t_{HCLK}-2$		ns
t_{RW}	$\overline{RD}$ Strobe Pulsewidth	$t_{HCLK}-2+W^3$		ns
t_{RSA}	$\overline{RD}$ Deasserted to Address Invalid Setup	$0.5t_{HCLK}-2$		ns
t_{RWR}	$\overline{RD}$ Deasserted to $\overline{WR}$, $\overline{RD}$ Asserted	t_{HCLK}		
Timing Requirements				
t_{AKW}	ACK Strobe Pulsewidth	t_{HCLK}		ns
t_{RDA}	$\overline{RD}$ Asserted to Data Access Setup		$t_{HCLK}-4+W^3$	ns
t_{ADA}	Address Valid to Data Access Setup		$t_{HCLK}+W^3$	ns
t_{SDA}	Chip Select Asserted to Data Access Setup		$t_{HCLK}+W^3$	ns
t_{SD}	Data Valid to $\overline{RD}$ Deasserted Setup	7		ns
t_{HRD}	$\overline{RD}$ Deasserted to Data Invalid Hold	0		ns
t_{DRSAK}	ACK Delay from $\overline{RD}$ Low	0		ns

[1] t_{HCLK} is the peripheral clock period.
[2] These are timing parameters that are based on worst-case operating conditions.
[3] W = (number of waitstates specified in wait register) × t_{HCLK}.

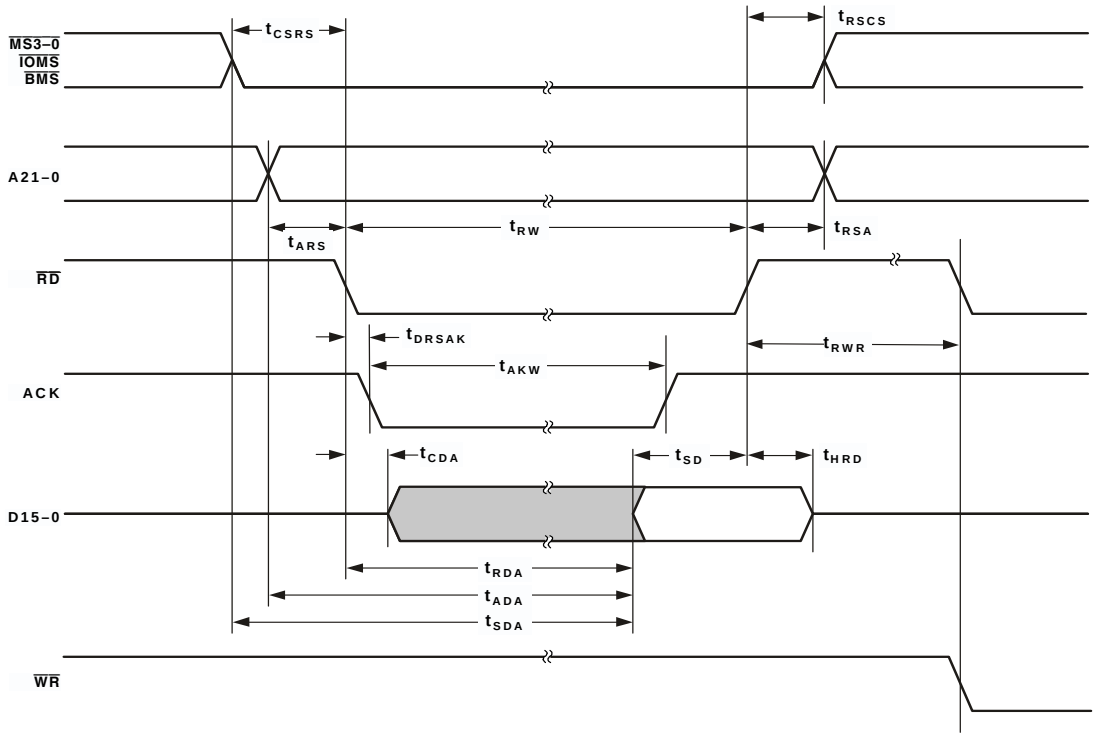


Figure 12. External Port Read Cycle Timing

External Port Bus Request and Grant Cycle Timing

Table 14 and Figure 13 describe external port bus request and bus grant operations.

Table 14. External Port Bus Request and Grant Cycle Timing

Parameter[1, 2]		Min	Max	Unit
Switching Characteristics				
t_{SD}	CLKOUT High to $\overline{xMS}$, Address, and $\overline{RD}/\overline{WR}$ Disable		$0.5t_{HCLK}+1$	ns
t_{SE}	CLKOUT Low to $\overline{xMS}$, Address, and $\overline{RD}/\overline{WR}$ Enable	0	4	ns
t_{DBG}	CLKOUT High to $\overline{BG}$ Asserted Setup	0	4	ns
t_{EBG}	CLKOUT High to $\overline{BG}$ Deasserted Hold Time	0	4	ns
t_{DBH}	CLKOUT High to $\overline{BGH}$ Asserted Setup	0	4	ns
t_{EBH}	CLKOUT High to $\overline{BGH}$ Deasserted Hold Time	0	4	ns
Timing Requirements				
t_{BS}	$\overline{BR}$ Asserted to CLKOUT High Setup	4.6		ns
t_{BH}	CLKOUT High to $\overline{BR}$ Deasserted Hold Time	0		ns

[1] t_{HCLK} is the peripheral clock period.
[2] These are timing parameters that are based on worst-case operating conditions.

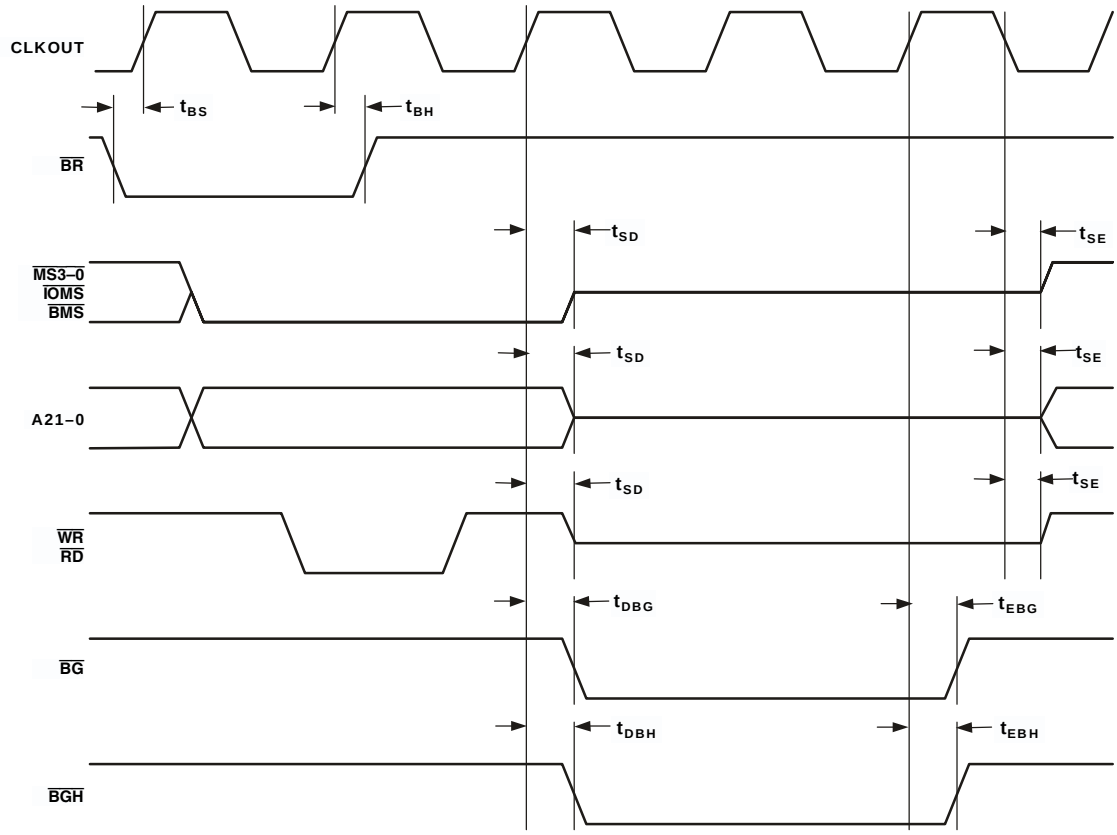


Figure 13. External Port Bus Request and Grant Cycle Timing

Host Port ALE Mode Write Cycle Timing

Table 15 and Figure 14 describe Host port write operations in Address Latch Enable (ALE) mode. For more information on ACK, Ready, ALE, and ACC mode selection, see the Host port modes description on page 8.

Table 15. Host Port ALE Mode Write Cycle Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{WHKS1}	$\overline{HWR}$ Asserted to HACK Asserted (Setup, ACK Mode) First Byte	10	$5t_{HCLK}+t_{NH}$[1]	ns
t_{WHKS2}	$\overline{HWR}$ Asserted to HACK Asserted (Setup, ACK Mode)[2]		10	ns
t_{WHKH}	$\overline{HWR}$ Deasserted to HACK Deasserted (Hold, ACK Mode)		10	ns
t_{WHS}	$\overline{HWR}$ Asserted to HACK Asserted (Setup, Ready Mode)		10	ns
t_{WHH}	$\overline{HWR}$ Asserted to HACK Deasserted (Hold, Ready Mode) First Byte	0	$5t_{HCLK}+t_{NH}$[1]	ns
Timing Requirements				
t_{CSAL}	$\overline{HCMS}$ or $\overline{HCIOMS}$ Asserted to HALE Asserted	0		ns
t_{ALPW}	HALE Asserted Pulsewidth	4		ns
t_{ALCSW}	HALE Deasserted to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	1		ns
t_{WCSW}	$\overline{HWR}$ Deasserted to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	0		ns
t_{ALW}	HALE Deasserted to $\overline{HWR}$ Asserted	1		ns
t_{WCS}	$\overline{HWR}$ Deasserted (After Last Byte) to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted (Ready for Next Write)	0		ns
t_{HKWD}	HACK Asserted to $\overline{HWR}$ Deasserted (Hold, ACK Mode)	1.5		ns
t_{AALS}	Address Valid to HALE Deasserted (Setup)	2		ns
t_{ALAH}	HALE Deasserted to Address Invalid (Hold)	4		ns
t_{DWS}	Data Valid to $\overline{HWR}$ Deasserted (Setup)	4		ns
t_{WDH}	$\overline{HWR}$ Deasserted to Data Invalid (Hold)	1		ns

[1] t_{NH} are peripheral bus latencies ($n \times t_{HCLK}$); these are internal DSP latencies related to the number of peripheral DMAs attempting to access DSP memory at the same time.

[2] Measurement is for the second, third, or fourth byte of a host write transaction. The quantity of bytes to complete a host write transaction is dependent on the data bus size (8 or 16 bits) and the data type (16 or 24 bits).

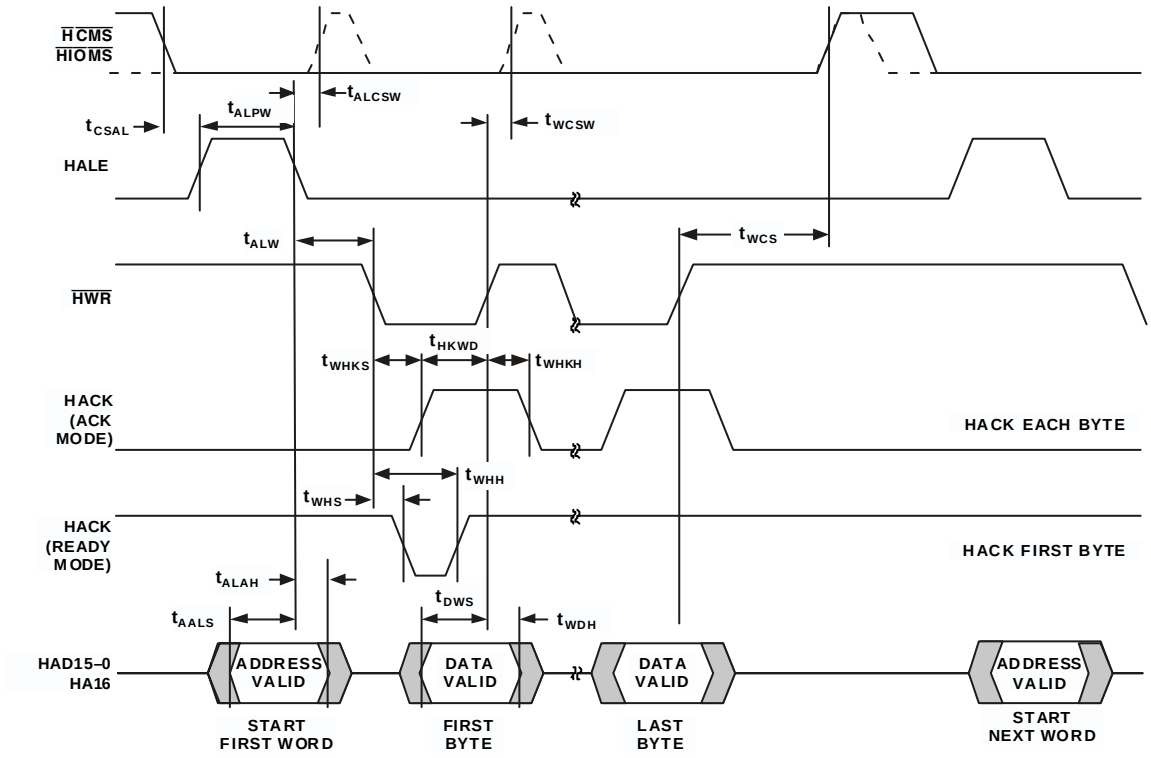


Figure 14. Host Port ALE Mode Write Cycle Timing

Host Port ACC Mode Write Cycle Timing

Table 16 and Figure 15 describe Host port write operations in Address Cycle Control (ACC) mode. For more information on ACK, Ready, ALE, and ACC mode selection, see the Host port modes description on page 8.

Table 16. Host Port ACC Mode Write Cycle Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{WHKS1}	$\overline{HWR}$ Asserted to HACK Asserted (ACK Mode) First Byte	10	$5t_{HCLK}+t_{NH}$[1]	ns
t_{WHKS2}	$\overline{HWR}$ Asserted to HACK Asserted (Setup, ACK Mode)[2]		12	ns
t_{WHKH}	$\overline{HWR}$ Deasserted to HACK Deasserted (Hold, ACK Mode)		10	ns
t_{WHS}	$\overline{HWR}$ Asserted to HACK Asserted (Setup, Ready Mode)		10	ns
t_{WHH}	$\overline{HWR}$ Asserted to HACK Deasserted (Hold, Ready Mode) First Byte	0	$5t_{HCLK}+t_{NH}$[1]	ns
t_{WSHKS}	$\overline{HWR}$ Asserted to HACK Asserted (Setup) During Address Latch		10	ns
t_{WHHKH}	$\overline{HWR}$ Deasserted to HACK Deasserted (Hold) During Address Latch		10	ns
Timing Requirements				
t_{WAL}	$\overline{HWR}$ Asserted to HALE Deasserted (Delay)	1.5		ns
t_{CSAL}	$\overline{HCMS}$ or $\overline{HCIOMS}$ Asserted to HALE Asserted (Delay)	0		ns
t_{ALCS}	HALE Deasserted to Optional $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	1		ns
t_{WCSW}	$\overline{HWR}$ Deasserted to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	0		ns
t_{ALW}	HALE Asserted to HWR Asserted	0.5		ns
t_{CSW}	$\overline{HCMS}$ or $\overline{HCIOMS}$ Asserted to $\overline{HWR}$ Asserted	0		ns
t_{WCS}	$\overline{HWR}$ Deasserted (After Last Byte) to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted (Ready for Next Write)	0		ns
t_{ALEW}	HALE Deasserted to HWR Asserted	1		ns
t_{HKWD}	HACK Asserted to $\overline{HWR}$ Deasserted (Hold, ACK Mode)	1.5		ns
t_{ADW}	Address Valid to $\overline{HWR}$ Asserted (Setup)	3		ns
t_{WAD}	$\overline{HWR}$ Deasserted to Address Invalid (Hold)	3		ns
t_{DWS}	Data Valid to $\overline{HWR}$ Deasserted (Setup)	2		ns
t_{WDH}	$\overline{HWR}$ Deasserted to Data Invalid (Hold)	2		ns
t_{HKWAL}	HACK Asserted to $\overline{HWR}$ Deasserted (Hold) During Address Latch[2]	2		ns

[1] t_{NH} are peripheral bus latencies ($n \times t_{HCLK}$); these are internal DSP latencies related to the number of peripheral DMAs attempting to access DSP memory at the same time.

[2] Measurement is for the second, third, or fourth byte of a host write transaction. The quantity of bytes to complete a host write transaction is dependent on the data bus size (8 or 16 bits) and the data type (16 or 24 bits).

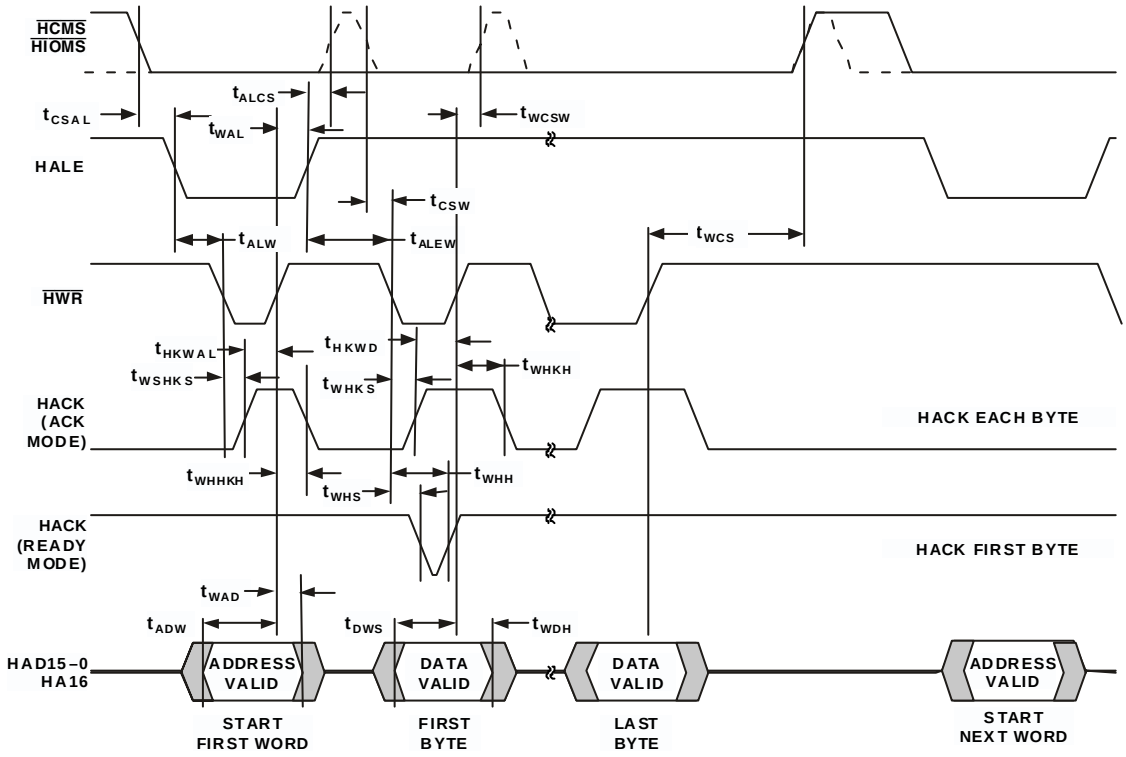


Figure 15. Host Port ACC Mode Write Cycle Timing

Host Port ALE Mode Read Cycle Timing

Table 17 and Figure 16 describe Host port read operations in Address Latch Enable (ALE) mode. For more information on ACK, Ready, ALE, and ACC mode selection, see the Host port modes description on page 8.

Table 17. Host Port ALE Mode Read Cycle Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{RHKS1}	$\overline{HRD}$ Asserted to HACK Asserted (ACK Mode) First Byte	$12t_{HCLK}$	$15t_{HCLK}+t_{NH}$[1]	ns
t_{RHKS2}	$\overline{HRD}$ Asserted to HACK Asserted (Setup, ACK Mode)[2]		12	ns
t_{RHKH}	$\overline{HRD}$ Deasserted to HACK Deasserted (Hold, ACK Mode)		10	ns
t_{RHS}	$\overline{HRD}$ Asserted to HACK Asserted (Setup, Ready Mode)		10	ns
t_{RHH}	$\overline{HRD}$ Asserted to HACK Deasserted (Hold, Ready Mode) First Byte	$12t_{HCLK}$	$15t_{HCLK}+t_{NH}$[1]	ns
t_{RDH}	$\overline{HRD}$ Deasserted to Data Invalid (Hold)	1		ns
t_{RDD}	$\overline{HRD}$ Deasserted to Data Disable		10	ns
Timing Requirements				
t_{CSAL}	$\overline{HCMS}$ or $\overline{HCIOMS}$ Asserted to HALE Asserted (Delay)	0		ns
t_{ALCS}	HALE Deasserted to Optional $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	1		ns
t_{RCSW}	$\overline{HRD}$ Deasserted to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	0		ns
t_{ALR}	HALE Deasserted to $\overline{HRD}$ Asserted	5		ns
t_{RCS}	$\overline{HRD}$ Deasserted (After Last Byte) to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted (Ready for Next Read)	0		ns
t_{ALPW}	HALE Asserted Pulsewidth	4		ns
t_{HKRD}	HACK Asserted to $\overline{HRD}$ Deasserted (Hold, ACK Mode)	1.5		ns
t_{AALS}	Address Valid to HALE Deasserted (Setup)	2		ns
t_{ALAH}	HALE Deasserted to Address Invalid (Hold)	4		ns

[1] t_{NH} are peripheral bus latencies ($n \times t_{HCLK}$); these are internal DSP latencies related to the number of peripherals attempting to access DSP memory at the same time.

[2] Measurement is for the second, third, or fourth byte of a host read transaction. The quantity of bytes to complete a host read transaction is dependent on the data bus size (8 or 16 bits) and the data type (16 or 24 bits).

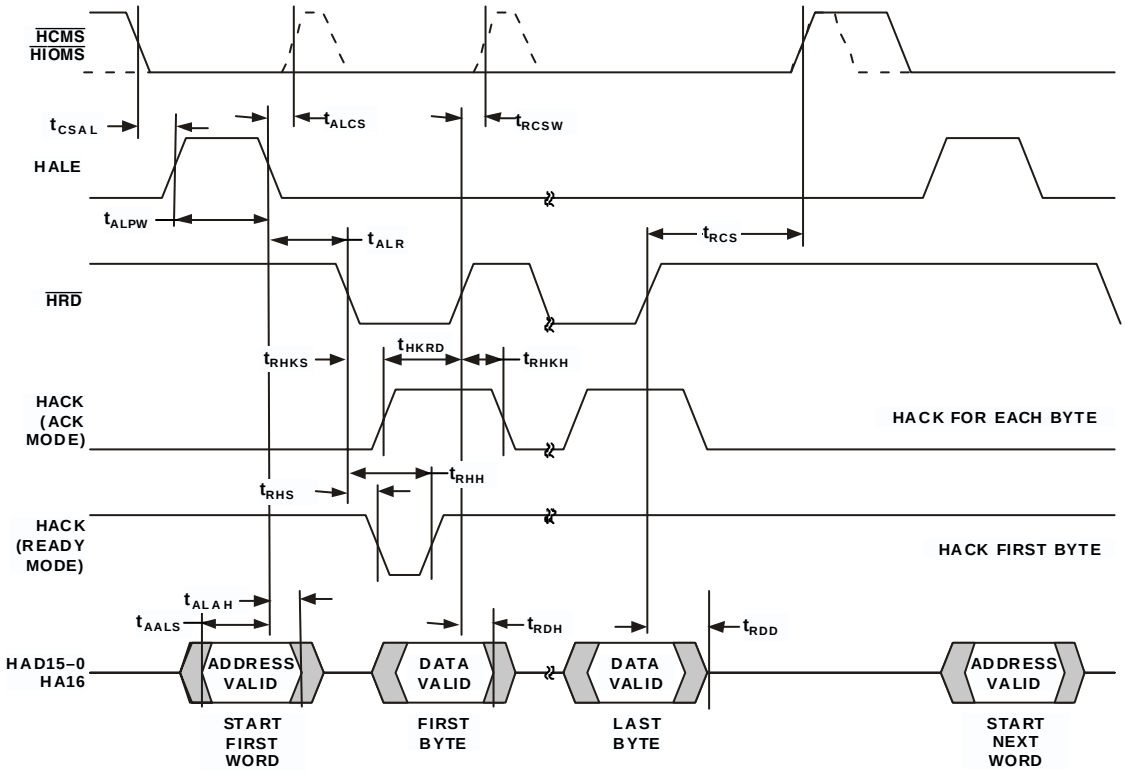


Figure 16. Host Port ALE Mode Read Cycle Timing

Host Port ACC Mode Read Cycle Timing

Table 18 and Figure 17 describe Host port read operations in Address Cycle Control (ACC) mode. For more information on ACK, Ready, ALE, and ACC mode selection, see the Host port modes description on page 8.

Table 18. Host Port ACC Mode Read Cycle Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{RHKS1}	$\overline{HRD}$ Asserted to HACK Asserted (ACK Mode) First Byte	$12t_{HCLK}$	$15t_{HCLK}+t_{NH}$[1]	ns
t_{RHKS2}	$\overline{HRD}$ Asserted to HACK Asserted (Setup, ACK Mode)[2]		10	ns
t_{RHKH}	$\overline{HRD}$ Deasserted to HACK Deasserted (Hold, ACK Mode)		10	ns
t_{RHS}	$\overline{HRD}$ Asserted to HACK Asserted (Setup, Ready Mode)		10	ns
t_{RHH}	$\overline{HRD}$ Asserted to HACK Deasserted (Hold, Ready Mode) First Byte	$12t_{HCLK}$	$15t_{HCLK}+t_{NH}$[1]	ns
t_{RDH}	$\overline{HRD}$ Deasserted to Data Invalid (Hold)	1		ns
t_{WSHKS}	$\overline{HWR}$ Asserted to HACK Asserted (Setup) During Address Latch		10	ns
t_{WHHKH}	$\overline{HWR}$ Deasserted to HACK Deasserted (Hold) During Address Latch		10	ns
t_{RDD}	$\overline{HRD}$ Deasserted to Data Disable		10	ns
Timing Requirements				
t_{CSAL}	$\overline{HCMS}$ or $\overline{HCIOMS}$ Asserted to HALE Asserted (Delay)	0		ns
t_{ALCS}	HALE Deasserted to Optional $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	1		ns
t_{RCSW}	$\overline{HRD}$ Deasserted to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted	0		ns
t_{ALW}	HALE Asserted to HWR Asserted	0.5		ns
t_{ALER}	HALE Deasserted to HWR Asserted	1		ns
t_{CSR}	$\overline{HCMS}$ or $\overline{HCIOMS}$ Asserted to $\overline{HRD}$ Asserted	0		ns
t_{RCS}	$\overline{HRD}$ Deasserted (After Last Byte) to $\overline{HCMS}$ or $\overline{HCIOMS}$ Deasserted (Ready for Next Read)	0		ns
t_{WAL}	HWR Deasserted to HALE Deasserted (Delay)	2.5		ns
t_{HKRD}	HACK Asserted to $\overline{HRD}$ Deasserted (Hold, ACK Mode)	1.5		ns
t_{ADW}	Address Valid to $\overline{HWR}$ Deasserted (Setup)	2		ns
t_{WAD}	$\overline{HWR}$ Deasserted to Address Invalid (Hold)	1		ns
t_{HKWAL}	HACK Asserted to $\overline{HWR}$ Deasserted (Hold) During Address Latch[2]	2		ns

[1] t_{NH} are peripheral bus latencies ($n \times t_{HCLK}$); these are internal DSP latencies related to the number of peripherals attempting to access DSP memory at the same time.

[2] Measurement is for the second, third, or fourth byte of a host read transaction. The quantity of bytes to complete a host read transaction is dependent on the data bus size (8 or 16 bits) and the data type (16 or 24 bits).

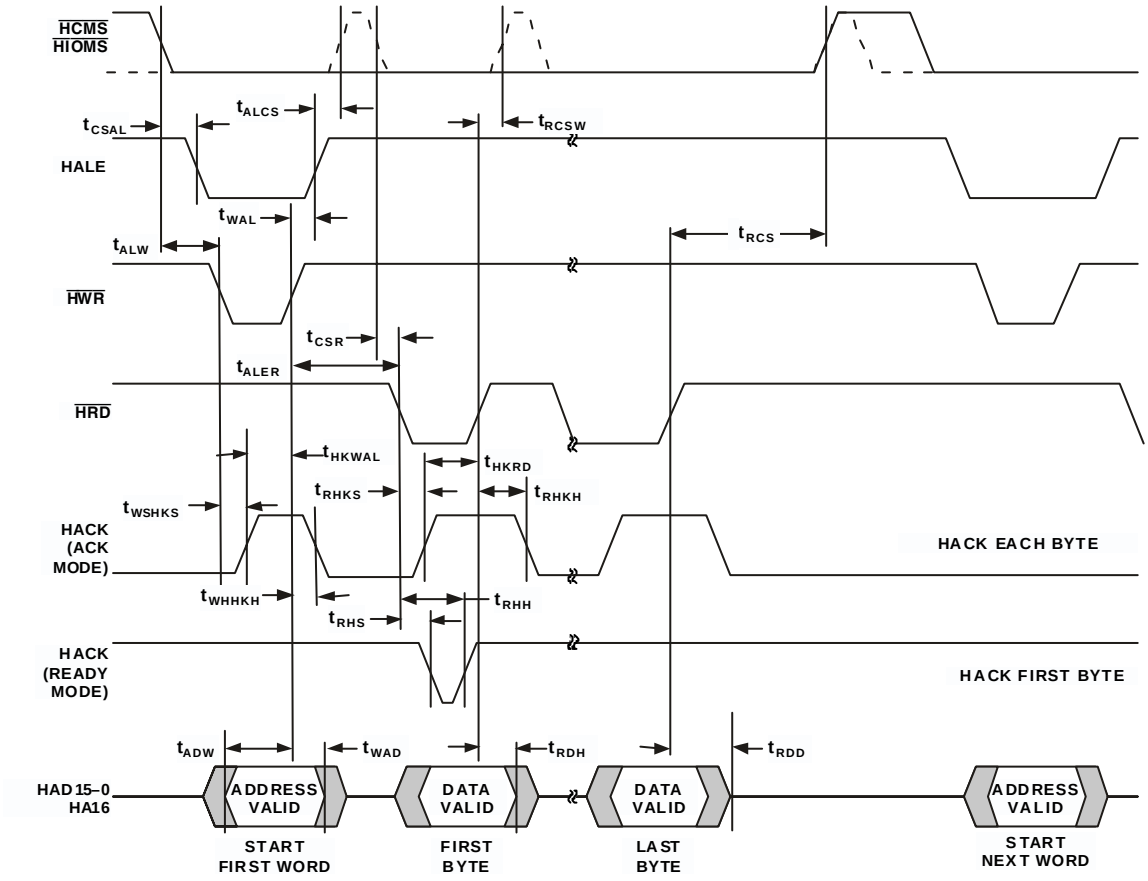


Figure 17. Host Port ACC Mode Read Cycle Timing

Serial Ports

Table 19 and Figure 18 describe SPORT transmit and receive operations, while Figure 19 and Figure 20 describe SPORT Frame Sync operations.

Table 19. Serial Ports[1, 2]

Parameter		Min	Max	Unit
External Clock				
Timing Requirements				
t_{SFSE}	TFS/RFS Setup Before TCLK/RCLK[3]	4		ns
t_{HFSE}	TFS/RFS Hold After TCLK/RCLK[3]	4		ns
t_{SDRE}	Receive Data Setup Before RCLK[3]	1.5		ns
t_{HDRE}	Receive Data Hold After RCLK[3]	4		ns
t_{SCLKW}	TCLK/RCLK Width	$0.5t_{HCLK}-1$		ns
t_{SCLK}	TCLK/RCLK Period	$2t_{HCLK}$		ns
Internal Clock				
Timing Requirements				
t_{SFSI}	TFS Setup Before TCLK[4]; RFS Setup Before RCLK[3]	4		ns
t_{HFSI}	TFS/RFS Hold After TCLK/RCLK[3]	3		ns
t_{SDRI}	Receive Data Setup Before RCLK[3]	2		ns
t_{HDRI}	Receive Data Hold After RCLK[3]	5		ns
External or Internal Clock				
Switching Characteristics				
t_{DFSE}	TFS/RFS Delay After TCLK/RCLK (Internally Generated FS)[4]		14	ns
t_{HOFSE}	TFS/RFS Hold After TCLK/RCLK (Internally Generated FS)[4]	3		ns
External Clock				
Switching Characteristics				
t_{DDTE}	Transmit Data Delay After TCLK[4]		13.4	ns
t_{HDTE}	Transmit Data Hold After TCLK[4]	4		ns
Internal Clock				
Switching Characteristics				
t_{DDTI}	Transmit Data Delay After TCLK[4]		13.4	ns
t_{HDTI}	Transmit Data Hold After TCLK[4]	4		ns
t_{SCLKIW}	TCLK/RCLK Width	$0.5t_{HCLK}-3.5$	$0.5t_{HCLK}+2.5$	ns
Enable and Three-State[5]				
Switching Characteristics				
t_{DTENE}	Data Enable from External TCLK[4]	0	12.1	ns
t_{DDTTE}	Data Disable from External TCLK[4]		13	ns
t_{DTENI}	Data Enable from Internal TCLK[4]	0	13	ns
t_{DDTTI}	Data Disable from External TCLK[4]		12	ns
External Late Frame Sync				
Switching Characteristics				
$t_{DDTLFSE}$	Data Delay from Late External TFS with MCE = 1, MFD = 0[6, 7]		10.5	ns
$t_{DTENLFSE}$	Data Enable from Late FS or MCE = 1, MFD = 0[6, 7]	3.5		ns

[1]To determine whether communication is possible between two devices at clock speed n, the following specifications must be confirmed: 1) frame sync delay and frame sync setup-and-hold, 2) data delay and data setup-and-hold, and 3) SCLK width.
[2]Word selected timing for I^2S mode is the same as TFS/RFS timing (normal framing only).
[3]Referenced to sample edge.
[4]Referenced to drive edge.
[5]Only applies to SPORT0/1.
[6]MCE = 1, TFS enable, and TFS valid follow $t_{DDTENFS}$ and $t_{DDTLFSE}$.
[7]If external RFSD/TFS setup to RCLK/TCLK > $0.5t_{LSCK}$, $t_{DDTLSCK}$ and $t_{DTENLSCK}$ apply; otherwise $t_{DDTLFSE}$ and $t_{DTENLFS}$ apply.

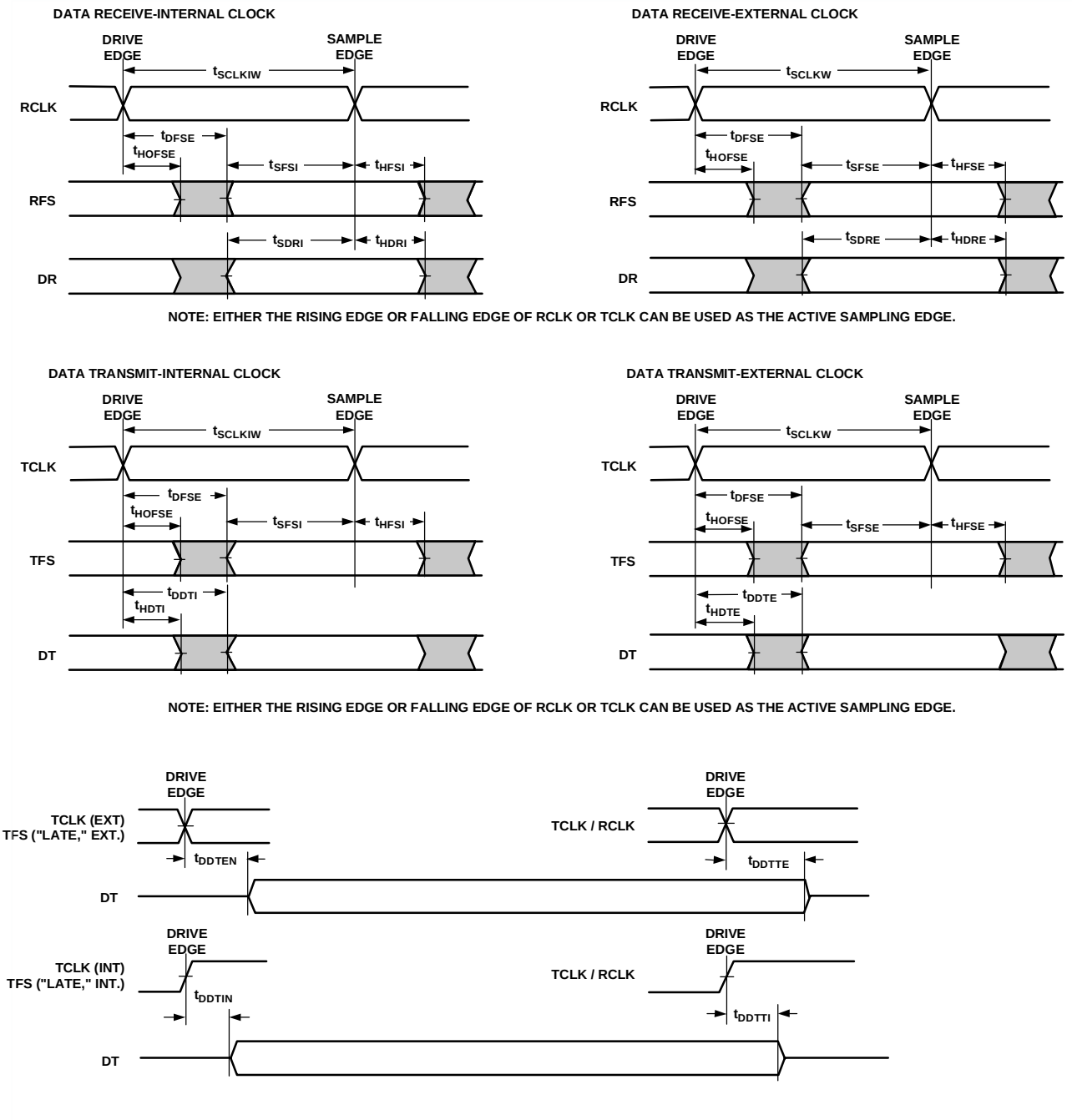


Figure 18. Serial Ports

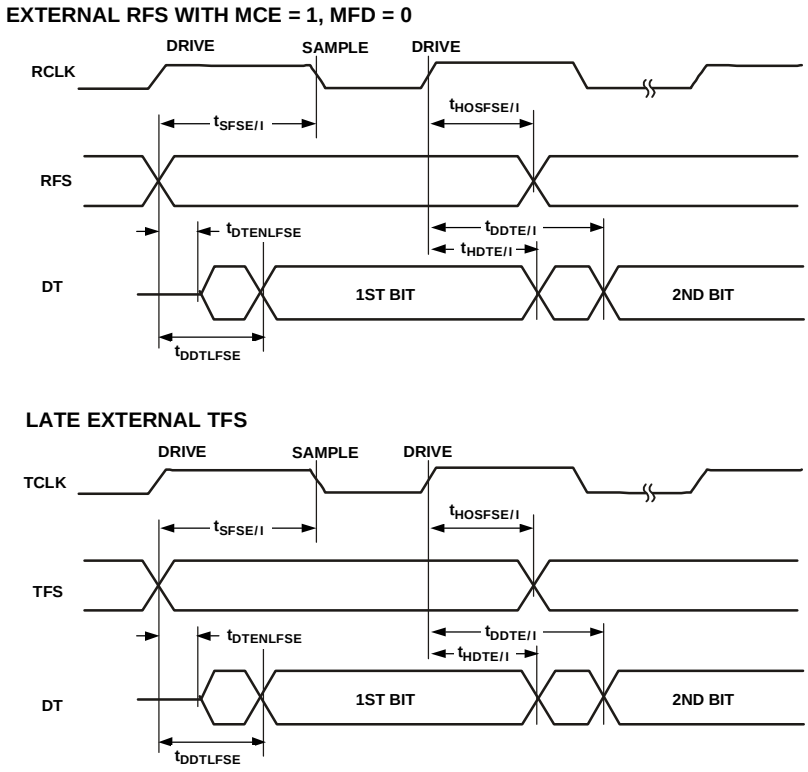


Figure 19. Serial Ports—External Late Frame Sync (Frame Sync Setup > 0.5t_{SCLK})

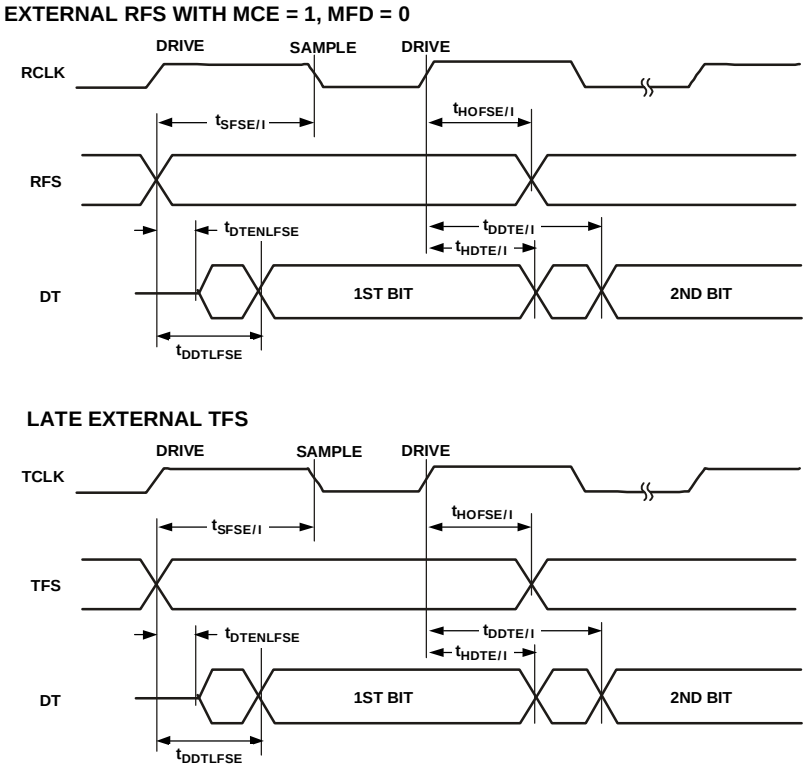


Figure 20. Serial Ports—External Late Frame Sync (Frame Sync Setup < 0.5t_{HCLK})

Serial Peripheral Interface (SPI) Port—Master Timing

Table 20 and Figure 21 describe SPI port master operations.

Table 20. Serial Peripheral Interface (SPI) Port—Master Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{SDSCIM}	$\overline{\text{SPIxSEL}}$ Low to First SCLK edge (x=0 or 1)	$2t_{HCLK}-3$		ns
t_{SPICHM}	Serial Clock High Period	$2t_{HCLK}-3$		ns
t_{SPICLM}	Serial Clock Low Period	$2t_{HCLK}-3$		ns
t_{SPICLK}	Serial Clock Period	$4t_{HCLK}-1$		ns
t_{HDSM}	Last SCLK Edge to $\overline{\text{SPIxSEL}}$ High (x=0 or 1)	$2t_{HCLK}-3$		ns
t_{SPITDM}	Sequential Transfer Delay	$2t_{HCLK}-2$		ns
t_{DDSPID}	SCLK Edge to Data Output Valid (Data Out Delay)	0	6	ns
t_{HDSPID}	SCLK Edge to Data Output Invalid (Data Out Hold)	0	5	ns
Timing Requirements				
t_{SSPID}	Data Input Valid to SCLK Edge (Data Input Setup)	8		ns
t_{HSPID}	SCLK Sampling Edge to Data Input Invalid (Data In Hold)	1		ns

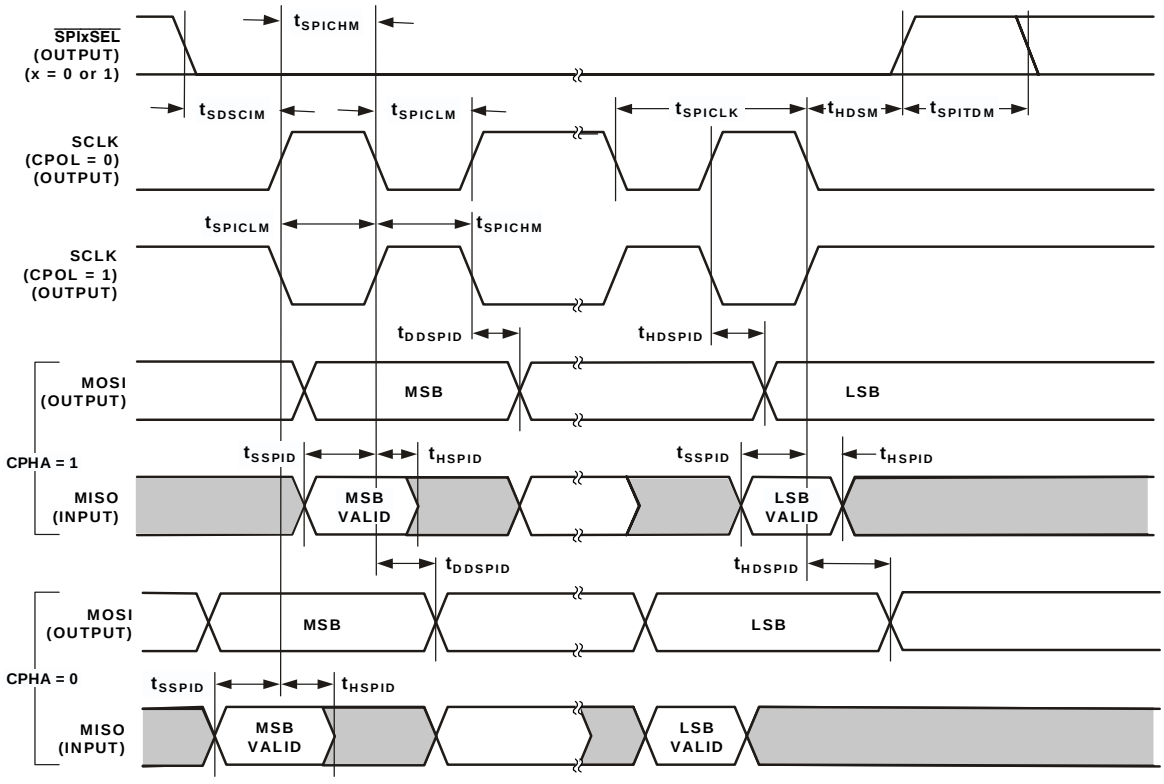


Figure 21. Serial Peripheral Interface (SPI) Port—Master Timing

Serial Peripheral Interface (SPI) Port—Slave Timing

Table 21 and Figure 22 describe SPI port slave operations.

Table 21. Serial Peripheral Interface (SPI) Port—Slave Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DSOE}	$\overline{SPISS}$ Assertion to Data Out Active	0	8	ns
t_{DSDHI}	$\overline{SPISS}$ Deassertion to Data High Impedance	0	10	ns
t_{DDSPID}	SCLK Edge to Data Out Valid (Data Out Delay)	0	10	ns
t_{HDSPID}	SCLK Edge to Data Out Invalid (Data Out Hold)	0	10	ns
Timing Requirements				
t_{SPICHS}	Serial Clock High Period	$2t_{HCLK}$		ns
t_{SPICLS}	Serial Clock Low Period	$2t_{HCLK}$		ns
t_{SPICLK}	Serial Clock Period	$4t_{HCLK}$		ns
t_{HDS}	Last SPICLK Edge to $\overline{SPISS}$ Not Asserted	$2t_{HCLK}$		ns
t_{SPITDS}	Sequential Transfer Delay	$2t_{HCLK}+4$		ns
t_{SDSCI}	$\overline{SPISS}$ Assertion to First SPICLK Edge	$2t_{HCLK}$		ns
t_{SSPID}	Data Input Valid to SCLK Edge (Data Input Setup)	1.6		ns
t_{HSPID}	SCLK Sampling Edge to Data Input Invalid (Data In Hold)	2.4		ns

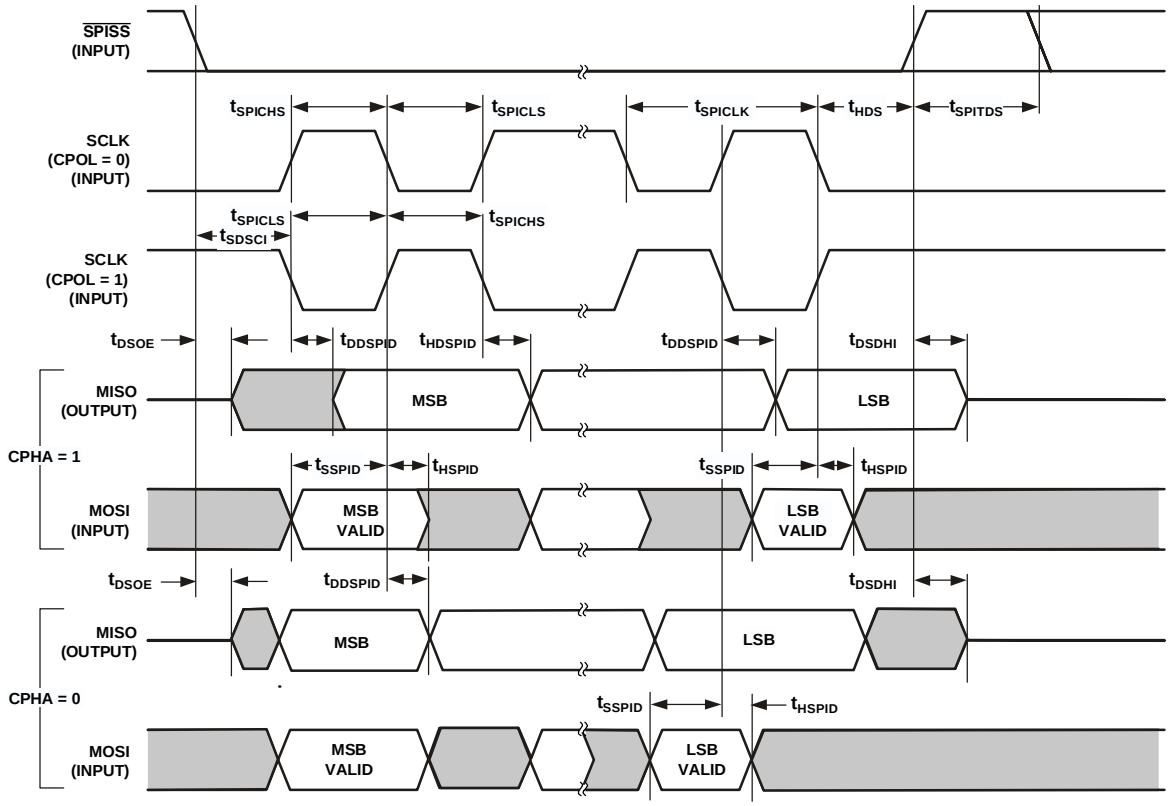


Figure 22. Serial Peripheral Interface (SPI) Port—Slave Timing

Universal Asynchronous Receiver-Transmitter (UART) Port—Receive and Transmit Timing

Figure 23 describes UART port receive and transmit operations. The maximum baud rate is HCLK/16. As shown in Figure 23 there is some latency between the generation internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART.

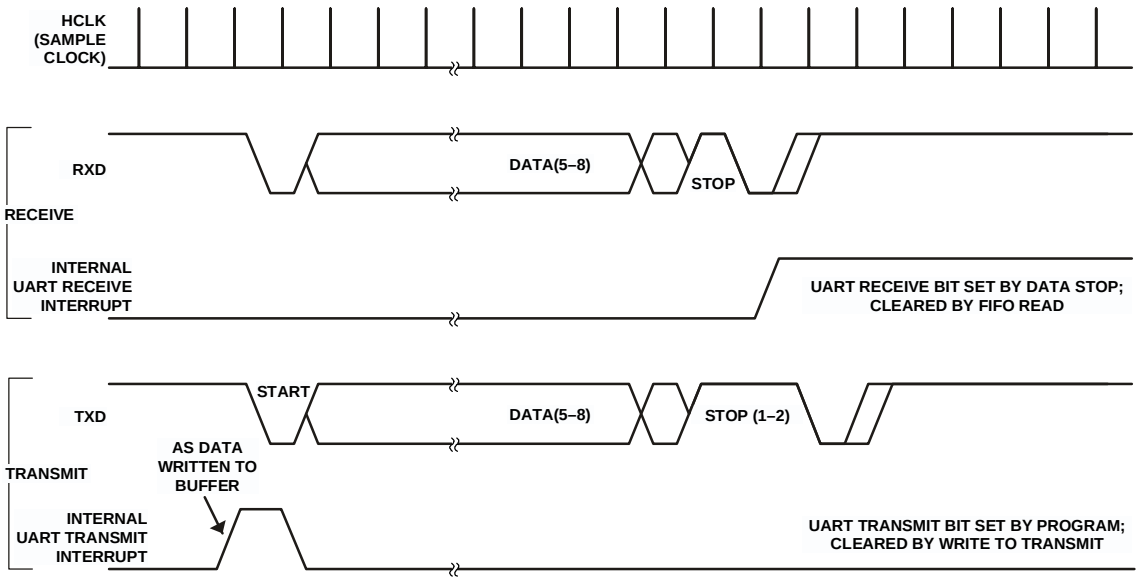


Figure 23. UART Port—Receive and Transmit Timing

JTAG Test And Emulation Port Timing

Table 22 and Figure 24 describe JTAG port operations.

Table 22. JTAG Port Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DTDO}	TDO Delay from TCK Low		8	ns
t_{DSYS}	System Outputs Delay After TCK Low[1]	0	22	ns
Timing Requirements				
t_{TCK}	TCK Period	20		ns
t_{STAP}	TDI, TMS Setup Before TCK High		4	ns
t_{HTAP}	TDI, TMS Hold After TCK High		4	ns
t_{SSYS}	System Inputs Setup Before TCK Low[2]		4	ns
t_{HSYS}	System Inputs Hold After TCK Low[2]		5	ns
t_{TRSTW}	$\overline{TRST}$ Pulsewidth[3]	$4t_{TCK}$		ns

[1]System Outputs = DATA15–0, ADDR21–0, $\overline{MS3–0}$, $\overline{RD}$, $\overline{WR}$, ACK, CLKOUT, $\overline{BG}$, PF7–0, TIMEXP, DT0, DT1, TCLK0, TCLK1, RCLK0, RCLK1, TFS0, TFS1, RFS0, RFS1, $\overline{BMS}$.
[2]System Inputs = DATA15–0, ADDR21–0, $\overline{RD}$, $\overline{WR}$, ACK, $\overline{BR}$, $\overline{BG}$, PF7–0, DR0, DR1, TCLK0, TCLK1, RCLK0, RCLK1, TFS0, TFS1, RFS0, RFS1, CLKIN, $\overline{RESET}$.
[3]50 MHz max.

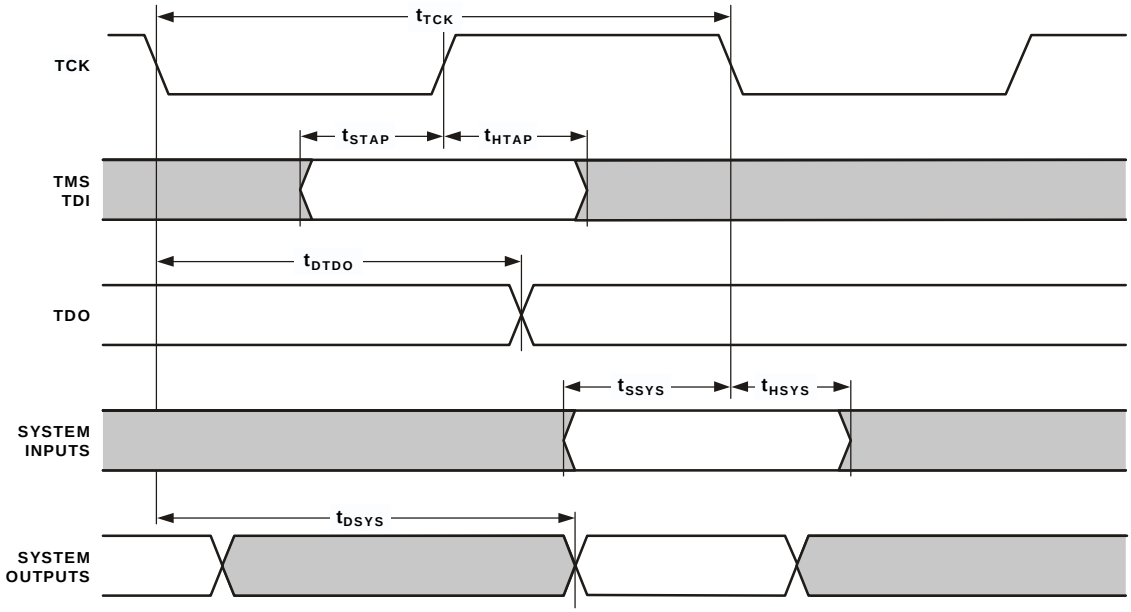


Figure 24. JTAG Port Timing

Output Drive Currents

Figure 25 shows typical I-V characteristics for the output drivers of the ADSP-2191M. The curves represent the current drive capability of the output drivers as a function of output voltage.

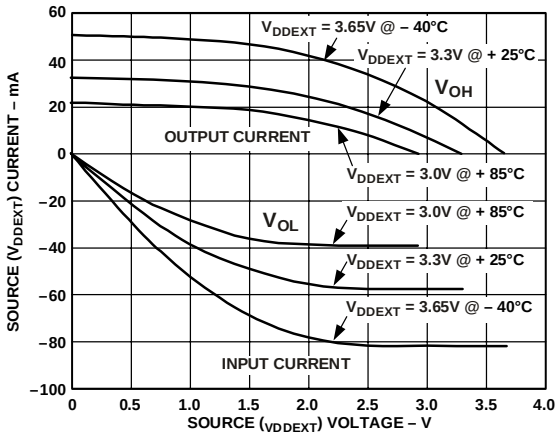


Figure 25. Typical Drive Currents

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 operands involved.

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

- Number of output pins that switch during each cycle (O)
- The maximum frequency at which they can switch (f)
- Their load capacitance (C)
- Their voltage swing (V_{DD})

and is calculated by the formula below.

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

The load capacitance includes 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 strobe can switch every cycle at a frequency of $1/t_{CK}$. Select pins switch at $1/(2t_{CK})$, but selects can switch on each cycle. For example, estimate P_{EXT} with the following assumptions:

- A system with one bank of external data memory—asynchronous RAM (16-bit)
- One 64K×16 RAM chip is used with a load of 10 pF
- Maximum peripheral speed CCLK = 80 MHz, HCLK = 80 MHz
- External data memory writes occur every other cycle, a rate of $1/(4t_{HCLK})$, with 50% of the pins switching
- The bus cycle time is 80 MHz (t_{HCLK} = 12.5 nsec)

The P_{EXT} equation is calculated for each class of pins that can drive as shown in Table 23.

Table 23. P_{EXT} Calculation Example

Pin Type	# of Pins	% Switching	× C	× f	× V_{DD}^2	= P_{EXT}
Address	15	50	10 pF	× 20 MHz	× 10.9 V	= .01635 W
$\overline{MSx}$	1	0	10 pF	× 20 MHz	× 10.9 V	= 0 W
$\overline{WR}$	1	—	10 pF	× 40 MHz	× 10.9 V	= .00436 W
Data	16	50	10 pF	× 20 MHz	× 10.9 V	= .01744 W
CLKOUT	1	—	10 pF	× 80 MHz	× 10.9 V	= .00872 W
						P_{EXT} = .04687 W

A typical power consumption can now be calculated for these conditions by adding a typical internal power dissipation with the following formula.

$$P_{TOTAL} = P_{EXT} + P_{INT}$$

Where:

- P_{EXT} is from Table 23
- P_{INT} is I_{DDINT} × 2.5 V, using the calculation I_{DDINT} listed in Power Dissipation on page 41.

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. Note also that it is not common for an application to have 100% or even 50% of the outputs switching simultaneously.

Test Conditions

The DSP is tested for output enable, disable, and hold 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 – V is dependent on the capacitive load, C_L and the load current, I_L. This decay time can be approximated by the equation below.

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

The output disable time t_{DIS} is the difference between $t_{MEASURED}$ and t_{DECAY} as shown in Figure 26. The time $t_{MEASURED}$ is the interval from when the reference signal switches to when the output voltage decays –V from the measured output high or output low voltage. The t_{DECAY} is calculated with test loads C_L and I_L, and with –V equal to 0.5 V.

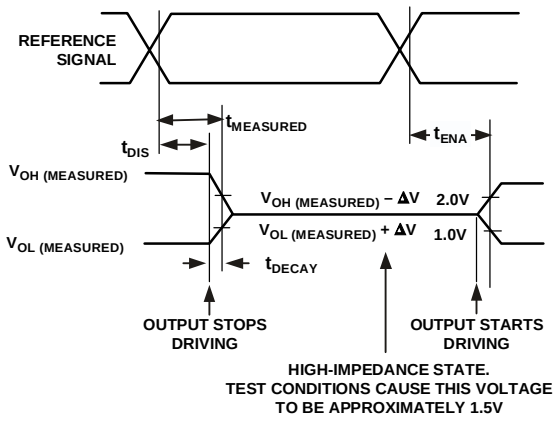


Figure 26. Output Enable/Disable

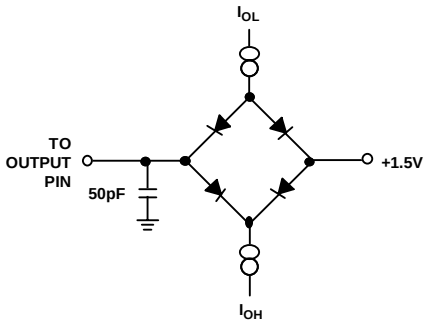


Figure 27. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

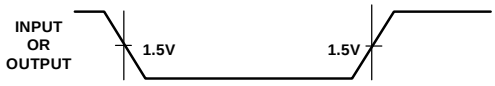


Figure 28. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

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 (Figure 26). 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 equation at Output Disable Time on page 41. Choose –V to be the difference between the ADSP-2191M's output voltage and the input threshold for the device requiring the hold time. A typical –V 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_{DATRWH} for the write cycle).

Capacitive Loading

Output delays and holds are based on standard capacitive loads: 50 pF on all pins (see Figure 30). The delay and hold specifications given should be derated by a factor of 1.5 ns/50 pF for loads other than the nominal value of 50 pF. Figure 28 and Figure 29 show how output rise time varies with capacitance. These figures also show graphically how output delays and holds vary with load capacitance. (Note that this graph or derating does not apply to output disable delays; see Output Disable Time on page 41.) The graphs in these figures may not be linear outside the ranges shown.

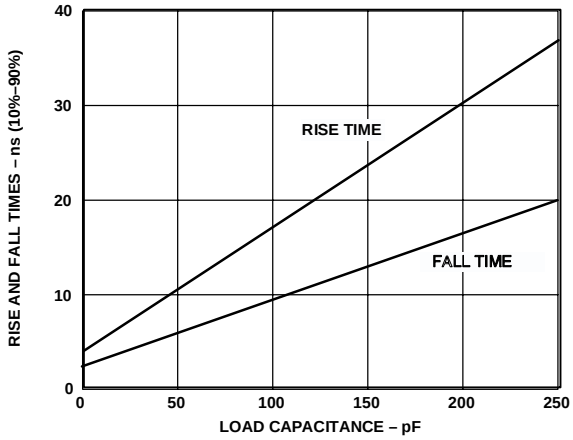


Figure 29. Typical Output Rise Time (10%-90%, V_{DDEXT} = Minimum at Maximum Ambient Operating Temperature) vs. Load Capacitance

Environmental Conditions

The thermal characteristics in which the DSP is operating influence performance.

Thermal Characteristics

The ADSP-2191M comes in a 144-lead LQFP or 144-lead Ball Grid Array (mini-BGA) package. The ADSP-2191M is specified for an ambient temperature (T_{AMB}) as calculated using the formula below.

To ensure that the T_{AMB} data sheet specification is not exceeded, a heatsink and/or an air flow source may be used. A heatsink should be attached to the ground plane (as close as possible to the thermal pathways) with a thermal adhesive.

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

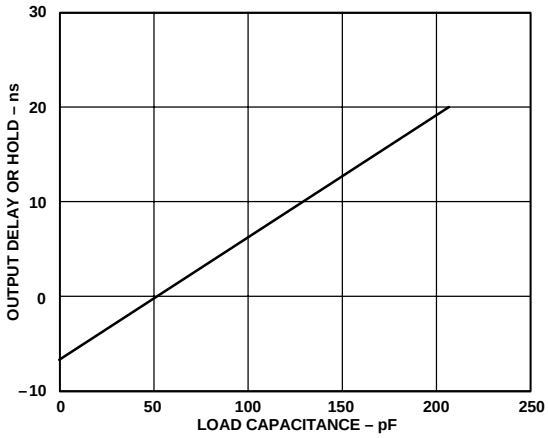


Figure 30. Typical Output Delay or Hold vs. Load Capacitance (at Maximum Case Temperature)

Where:

- T_{AMB} = Ambient temperature (measured near top surface of package)
- PD = Power dissipation in W (this value depends upon the specific application; a method for calculating PD is shown under Power Dissipation).
- θ_{CA} = Value from Table 24.
- For the LQFP package: θ_{JC} = 0.96°C/W
  For the mini-BGA package: θ_{JC} = 8.4°C/W

Table 24. θ_{CA} Values

Airflow (Linear Ft./Min.)	0	100	200	400	600
Airflow (Meters/Second)	0	0.5	1	2	3
LQFP: θ_{CA} (°C/W)	44.3	41.4	38.5	35.3	32.1
Mini-BGA: θ_{CA} (°C/W)	26	24	22	20.9	19.8

144-Lead LQFP Pinout

Table 25 lists the LQFP pinout by signal name. Table 26 lists the LQFP pinout by pin.

Table 25. 144-Lead LQFP Pins (Alphabetically by Signal)

Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.
A0	84	BYPASS	72	GND	33	$\overline{\text{HCMS}}$	27	TCLK1	65
A1	85	CLKIN	132	GND	54	$\overline{\text{HCIOMS}}$	28	TCLK2	47
A2	86	CLKOUT	130	GND	55	$\overline{\text{HRD}}$	31	TDI	75
A3	87	D0	123	GND	77	$\overline{\text{HWR}}$	32	TDO	74
A4	88	D1	124	GND	80	$\overline{\text{IOMS}}$	114	TFS0	59
A5	89	D2	125	GND	94	$\overline{\text{MS0}}$	115	TFS1	66
A6	91	D3	126	GND	105	$\overline{\text{MS1}}$	116	TFS2	48
A7	92	D4	128	GND	129	$\overline{\text{MS2}}$	117	TMR0	43
A8	93	D5	135	GND	134	$\overline{\text{MS3}}$	119	TMR1	44
A9	95	D6	136	HA16	23	OPMODE	83	TMR2	45
A10	96	D7	137	HACK	26	PF0	34	TMS	76
A11	97	D8	138	HACK_P	24	PF1	35	$\overline{\text{TRST}}$	79
A12	98	D9	139	HAD0	3	PF2	36	TXD	53
A13	99	D10	140	HAD1	4	PF3	37	V_{DDEXT}	13
A14	101	D11	141	HAD2	6	PF4	38	V_{DDEXT}	25
A15	102	D12	142	HAD3	7	PF5	39	V_{DDEXT}	40
A16	103	D13	144	HAD4	8	PF6	41	V_{DDEXT}	63
A17	104	D14	1	HAD5	9	PF7	42	V_{DDEXT}	90
A18	106	D15	2	HAD6	10	RCLK0	61	V_{DDEXT}	100
A19	107	DR0	60	HAD7	11	RCLK1	68	V_{DDEXT}	118
A20	108	DR1	67	HAD8	12	RCLK2	50	V_{DDEXT}	131
A21	109	DR2	49	HAD9	14	$\overline{\text{RD}}$	122	V_{DDEXT}	143
ACK	120	DT0	56	HAD10	15	$\overline{\text{RESET}}$	73	V_{DDINT}	19
$\overline{\text{BG}}$	111	DT1	64	HAD11	17	RFS0	62	V_{DDINT}	58
$\overline{\text{BGH}}$	110	DT2	46	HAD12	18	RFS1	69	V_{DDINT}	82
BMODE0	70	$\overline{\text{EMU}}$	81	HAD13	20	RFS2	51	V_{DDINT}	127
BMODE1	71	GND	5	HAD14	21	RXD	52	$\overline{\text{WR}}$	121
$\overline{\text{BMS}}$	113	GND	16	HAD15	22	TCK	78	XTAL	133
$\overline{\text{BR}}$	112	GND	29	HALE	30	TCLK0	57		

Table 26. 144-Lead LQFP Pins (Numerically by Pin Number)

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
1	D14	30	HALE	59	TFS0	88	A4	117	$\overline{MS2}$
2	D15	31	$\overline{HRD}$	60	DR0	89	A5	118	V_{DDEXT}
3	HAD0	32	$\overline{HWR}$	61	RCLK0	90	V_{DDEXT}	119	$\overline{MS3}$
4	HAD1	33	GND	62	RFS0	91	A6	120	ACK
5	GND	34	PF0	63	V_{DDEXT}	92	A7	121	$\overline{WR}$
6	HAD2	35	PF1	64	DT1	93	A8	122	$\overline{RD}$
7	HAD3	36	PF2	65	TCLK1	94	GND	123	D0
8	HAD4	37	PF3	66	TFS1	95	A9	124	D1
9	HAD5	38	PF4	67	DR1	96	A10	125	D2
10	HAD6	39	PF5	68	RCLK1	97	A11	126	D3
11	HAD7	40	V_{DDEXT}	69	RFS1	98	A12	127	V_{DDINT}
12	HAD8	41	PF6	70	BMODE0	99	A13	128	D4
13	V_{DDEXT}	42	PF7	71	BMODE1	100	V_{DDEXT}	129	GND
14	HAD9	43	TMR0	72	BYPASS	101	A14	130	CLKOUT
15	HAD10	44	TMR1	73	$\overline{RESET}$	102	A15	131	V_{DDEXT}
16	GND	45	TMR2	74	TDO	103	A16	132	CLKIN
17	HAD11	46	DT2	75	TDI	104	A17	133	XTAL
18	HAD12	47	TCLK2	76	TMS	105	GND	134	GND
19	V_{DDINT}	48	TFS2	77	GND	106	A18	135	D5
20	HAD13	49	DR2	78	TCK	107	A19	136	D6
21	HAD14	50	RCLK2	79	$\overline{TRST}$	108	A20	137	D7
22	HAD15	51	RFS2	80	GND	109	A21	138	D8
23	HA16	52	RXD	81	$\overline{EMU}$	110	$\overline{BGH}$	139	D9
24	HACK_P	53	TXD	82	V_{DDINT}	111	$\overline{BG}$	140	D10
25	V_{DDEXT}	54	GND	83	OPMODE	112	$\overline{BR}$	141	D11
26	HACK	55	GND	84	A0	113	$\overline{BMS}$	142	D12
27	$\overline{HCMS}$	56	DT0	85	A1	114	$\overline{IOMS}$	143	V_{DDEXT}
28	$\overline{HCIOMS}$	57	TCLK0	86	A2	115	$\overline{MS0}$	144	D13
29	GND	58	V_{DDINT}	87	A3	116	$\overline{MS1}$		

144-Lead Mini-BGA Pinout

Table 27 lists the mini-BGA pinout by signal name. Table 28 lists the mini-BGA pinout by ball number.

Table 27. 144-Lead Mini-BGA Pins (Alphabetically by Signal)

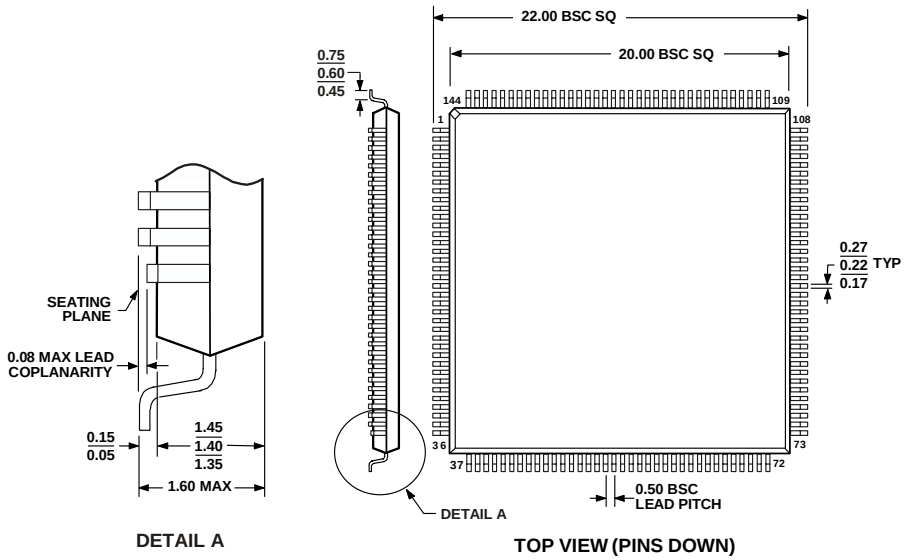
Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.
A0	J11	BYPASS	M11	GND	F7	HALE	J1	TCLK0	J6
A1	H9	CLKIN	A5	GND	F8	$\overline{\text{HCIOMS}}$	J3	TCLK1	M9
A2	H10	CLKOUT	C6	GND	F9	$\overline{\text{HCMS}}$	H1	TCLK2	K5
A3	G12	D0	D7	GND	G4	$\overline{\text{HRD}}$	J2	TDI	K12
A4	H11	D1	A7	GND	G5	$\overline{\text{HWR}}$	K2	TDO	L11
A5	G10	D2	C7	GND	G6	$\overline{\text{IOMS}}$	E8	TFS0	M8
A6	F12	D3	A6	GND	H5	$\overline{\text{MS0}}$	D9	TFS1	J8
A7	G11	D4	B7	GND	L6	$\overline{\text{MS1}}$	A9	TFS2	M5
A8	F10	D5	A4	GND	M1	$\overline{\text{MS2}}$	C9	TMR0	K4
A9	F11	D6	C5	GND	M12	$\overline{\text{MS3}}$	D8	TMR1	L4
A10	E12	D7	B5	HACK	H3	OPMODE	H12	TMR2	J4
A11	E11	D8	D5	HACK_P	G1	PF0	K1	TMS	K10
A12	E10	D9	A3	HAD0	C1	PF1	L1	$\overline{\text{TRST}}$	J12
A13	E9	D10	C4	HAD1	B3	PF2	M2	TXD	M7
A14	D11	D11	B4	HAD2	C2	PF3	L2	V_{DDEXT}	E5
A15	D10	D12	C3	HAD3	D1	PF4	M3	V_{DDEXT}	E6
A16	D12	D13	A2	HAD4	D4	PF5	L3	V_{DDEXT}	F5
A17	C11	D14	B1	HAD5	D3	PF6	K3	V_{DDEXT}	F6
A18	C12	D15	B2	HAD6	D2	PF7	M4	V_{DDEXT}	G7
A19	B12	DR0	L7	HAD7	E1	RCLK0	K7	V_{DDEXT}	G8
A20	B11	DR1	K9	HAD8	E4	RCLK1	J9	V_{DDEXT}	H7
A21	A11	DR2	L5	HAD9	E2	RCLK2	J5	V_{DDEXT}	H8
ACK	A8	DT0	H6	HAD10	F1	$\overline{\text{RD}}$	B8	V_{DDINT}	D6
$\overline{\text{BG}}$	C10	DT1	L8	HAD11	E3	$\overline{\text{RESET}}$	L12	V_{DDINT}	F4
$\overline{\text{BGH}}$	B10	DT2	H4	HAD12	F2	RFS0	K8	V_{DDINT}	G9
BMODE0	L10	$\overline{\text{EMU}}$	J10	HAD13	G2	RFS1	M10	V_{DDINT}	J7
BMODE1	L9	GND	A1	HAD14	F3	RFS2	M6	$\overline{\text{WR}}$	C8
$\overline{\text{BMS}}$	A10	GND	A12	HAD15	G3	RXD	K6	XTAL	B6
$\overline{\text{BR}}$	B9	GND	E7	HA16	H2	TCK	K11		

Table 28. 144-Lead Mini-BGA Pins (Numerically by Ball Number)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
A1	GND	C6	CLKOUT	E11	A11	H4	DT2	K9	DR1
A2	D13	C7	D2	E12	A10	H5	GND	K10	TMS
A3	D9	C8	$\overline{WR}$	F1	HAD10	H6	DT0	K11	TCK
A4	D5	C9	$\overline{MS2}$	F2	HAD12	H7	V_{DDEXT}	K12	TDI
A5	CLKIN	C10	$\overline{BG}$	F3	HAD14	H8	V_{DDEXT}	L1	PF1
A6	D3	C11	A17	F4	V_{DDINT}	H9	A1	L2	PF3
A7	D1	C12	A18	F5	V_{DDEXT}	H10	A2	L3	PF5
A8	ACK	D1	HAD3	F6	V_{DDEXT}	H11	A4	L4	TMR1
A9	$\overline{MS1}$	D2	HAD6	F7	GND	H12	OPMODE	L5	DR2
A10	$\overline{BMS}$	D3	HAD5	F8	GND	J1	$\overline{HALE}$	L6	GND
A11	A21	D4	HAD4	F9	GND	J2	$\overline{HRD}$	L7	DR0
A12	GND	D5	D8	F10	A8	J3	$\overline{HCIOMS}$	L8	DT1
B1	D14	D6	V_{DDINT}	F11	A9	J4	TMR2	L9	BMODE1
B2	D15	D7	D0	F12	A6	J5	RCLK2	L10	BMODE0
B3	HAD1	D8	$\overline{MS3}$	G1	HACK_P	J6	TCLK0	L11	TDO
B4	D11	D9	$\overline{MS0}$	G2	HAD13	J7	V_{DDINT}	L12	$\overline{RESET}$
B5	D7	D10	A15	G3	HAD15	J8	TFS1	M1	GND
B6	XTAL	D11	A14	G4	GND	J9	RCLK1	M2	PF2
B7	D4	D12	A16	G5	GND	J10	$\overline{EMU}$	M3	PF4
B8	$\overline{RD}$	E1	HAD7	G6	GND	J11	A0	M4	PF7
B9	$\overline{BR}$	E2	HAD9	G7	V_{DDEXT}	J12	$\overline{TRST}$	M5	TFS2
B10	$\overline{BGH}$	E3	HAD11	G8	V_{DDEXT}	K1	PF0	M6	RFS2
B11	A20	E4	HAD8	G9	V_{DDINT}	K2	$\overline{HWR}$	M7	TXD
B12	A19	E5	V_{DDEXT}	G10	A5	K3	PF6	M8	TFS0
C1	HAD0	E6	V_{DDEXT}	G11	A7	K4	TMR0	M9	TCLK1
C2	HAD2	E7	GND	G12	A3	K5	TCLK2	M10	RFS1
C3	D12	E8	$\overline{IOMS}$	H1	$\overline{HCMS}$	K6	RXD	M11	BYPASS
C4	D10	E9	A13	H2	HA16	K7	RCLK0	M12	GND
C5	D6	E10	A12	H3	HACK	K8	RFS0		

OUTLINE DIMENSIONS

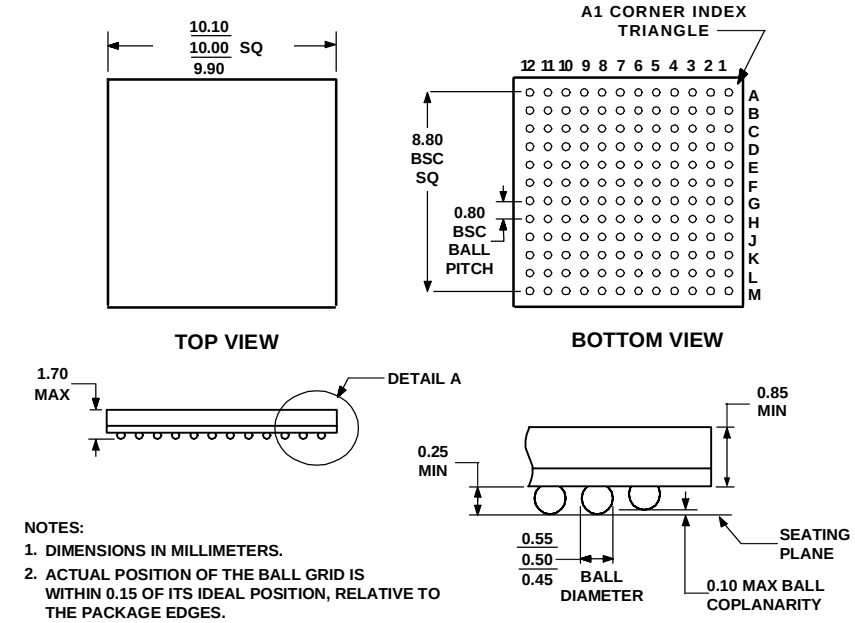
144-LEAD METRIC THIN PLASTIC QUAD FLATPACK (LQFP) (ST-144)



NOTES:

1. DIMENSIONS IN MILLIMETERS.
2. ACTUAL POSITION OF EACH LEAD IS WITHIN 0.08 OF ITS IDEAL POSITION, WHEN MEASURED IN THE LATERAL DIRECTION.
3. CENTER DIMENSIONS ARE NOMINAL.

144-BALL MINI-BGA (CA-144A)



NOTES:

1. DIMENSIONS IN MILLIMETERS.
2. ACTUAL POSITION OF THE BALL GRID IS WITHIN 0.15 OF ITS IDEAL POSITION, RELATIVE TO THE PACKAGE EDGES.
3. ACTUAL POSITION OF EACH BALL IS WITHIN 0.08 OF ITS IDEAL POSITION, RELATIVE TO THE BALL GRID.
4. CENTER DIMENSIONS ARE NOMINAL.

ORDERING GUIDE

Part Number[1, 2]	Ambient Temperature Range	Instruction Rate (MHz)	Package Description	Operating Voltage
ADSP-2191MKST-160	0°C to 70°C	160	144-Lead LQFP	2.5 Int./3.3 Ext. V
ADSP-2191MBST-140	–40°C to +85°C	140	144-Lead LQFP	2.5 Int./3.3 Ext. V
ADSP-2191MKCA-160	0°C to 70°C	160	144-Ball Mini-BGA	2.5 Int./3.3 Ext. V
ADSP-2191MBCA-140	–40°C to +85°C	140	144-Ball Mini-BGA	2.5 Int./3.3 Ext. V

[1]ST = Plastic Thin Quad Flatpack (LQFP).
[2]CA = Mini Ball Grid Array

C02936-0-4/02(0)

PRINTED IN U.S.A.