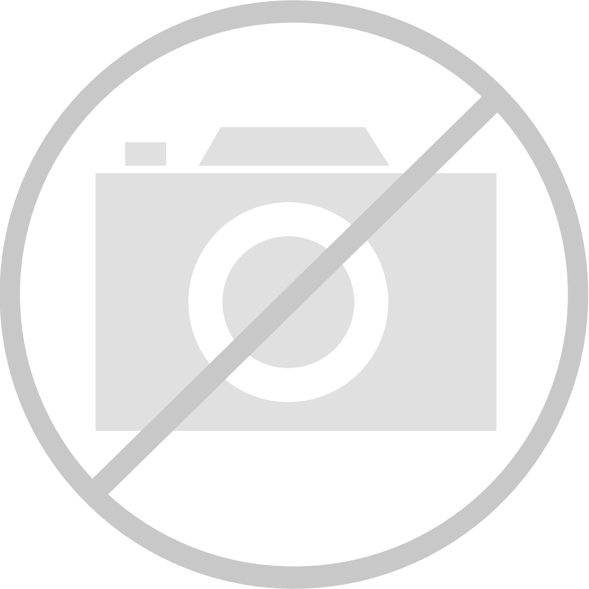
Linear Technology - ADSP-2191MBCA-140 Datasheet



Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Obsolete
Type	Fixed Point
Interface	Host Interface, SPI, SSP, UART
Clock Rate	140MHz
Non-Volatile Memory	External
On-Chip RAM	160kB
Voltage - I/O	3.00V, 3.30V
Voltage - Core	2.5V
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LFBGA, CSPBGA
Supplier Device Package	144-CSPBGA (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=adsp-2191mbca-140

ADSP-2191M* PRODUCT PAGE QUICK LINKS

Last Content Update: 02/23/2017

COMPARABLE PARTS

View a parametric search of comparable parts.

EVALUATION KITS

- USB-Based Emulator and High Performance USB-Based Emulator

DOCUMENTATION

Application Notes

- AN-227: Digital Control System Design with the ADSP-2100 Family
- AN-572: Overlay Linking on the ADSP-219x
- EE-06: ADSP-21xx Serial Port Startup Issues
- EE-100: ADSP-218x External Overlay Memory
- EE-102: Mode D and ADSP-218x Pin Compatibility - the FAQs
- EE-104: Setting Up Streams with the VisualDSP Debugger
- EE-11: ADSP-2181 Priority Chain & IDMA Holdoffs
- EE-115: ADSP-2189 IDMA Interface to Motorola MC68300 Family of Microprocessors
- EE-12: Interrupts and Programmable Flags on the ADSP-2185/2186
- EE-121: Porting Code from ADSP-21xx to ADSP-219x
- EE-122: Coding for Performance on the ADSP-219x
- EE-123: An Overview of the ADSP-219x Pipeline
- EE-124: Booting up the ADSP-2192
- EE-125: ADSP-218x Embedded System Software Management and In-System-Programming (ISP)
- EE-128: DSP in C++: Calling Assembly Class Member Functions From C++
- EE-129: ADSP-2192 Interprocessor Communication
- EE-130: Making Fast Transition from ADSP-21xx to ADSP-219x
- EE-131: Booting the ADSP-2191/95/96 DSPs
- EE-133: Converting From Legacy Architecture Files To Linker Description Files for the ADSP-218x
- EE-139: Interfacing the ADSP-2191 to an AD7476 via the SPI Port
- EE-142: Autobuffering, C and FFTs on the ADSP-218x
- EE-144: Creating a Master-Slave SPI Interface Between Two ADSP-2191 DSPs
- EE-145: SPI Booting of the ADSP-2191 using the Atmel AD25020N on an EZ-KIT Lite Evaluation Board
- EE-146: Implementing a Boot Manager for ADSP-218x Family DSPs
- EE-152: Using Software Overlays with the ADSP-219x and VisualDSP 2.0++
- EE-153: ADSP-2191 Programmable PLL
- EE-154: ADSP-2191 Host Port Interface

- EE-156: Support for the H.100 protocol on the ADSP-2191
- EE-158: ADSP-2181 EZ-Kit Lite IDMA to PC Printer Port Interface
- EE-159: Initializing DSP System & Control Registers From C and C++
- EE-164: Advanced EPROM Boot and No-boot Scenarios with ADSP-219x DSPs
- EE-168: Using Third Overtone Crystals with the ADSP-218x DSP
- EE-17: ADSP-2187L Memory Organization
- EE-18: Choosing and Using FFTs for ADSP-21xx
- EE-188: Using C To Implement Interrupt-Driven Systems On ADSP-219x DSPs
- EE-2: Using ADSP-218x I/O Space
- EE-226: ADSP-2191 DSP Host Port Booting
- EE-227: CAN Configuration Procedure for ADSP-21992 DSPs
- EE-249: Implementing Software Overlays on ADSP-218x DSPs with VisualDSP++®
- EE-32: Language Extensions: Memory Storage Types, ASM & Inline Constructs
- EE-35: Troubleshooting your ADSP-218x EZ-ICE
- EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users
- EE-38: ADSP-2181 IDMA Port - Cycle Steal Timing
- EE-39: Interfacing 5V Flash Memory to an ADSP-218x (Byte Programming Algorithm)
- EE-5: ADSP-218x Full Memory Mode vs. Host Memory Mode
- EE-60: Simulating an RS-232 UART Using the Synchronous Serial Ports on the ADSP-21xx Family DSPs
- EE-64: Setting Mode Pins on Reset
- EE-68: Analog Devices JTAG Emulation Technical Reference
- EE-71: Minimum Rise Time Specs for Critical Interrupt and Clock Signals on the ADSP-21x1/21x5
- EE-74: Analog Devices Serial Port Development and Troubleshooting Guide
- EE-78: BDMA Usage on 100 pin ADSP-218x DSPs Configured for IDMA Use
- EE-79: EPROM Booting In Host Mode with 100 Pin 218x Processors
- EE-82: Using an ADSP-2181 DSP's IO Space to IDMA Boot Another ADSP-2181
- EE-89: Implementing A Software UART on the ADSP-2181 EZ-Kit-Lite
- EE-96: Interfacing Two AD73311 Codecs to the ADSP-218x

Data Sheet

- ADSP-2191M: DSP Microcomputer Data Sheet

Evaluation Kit Manuals

- ADSP-2191 EZ-KIT Lite® Manual

Integrated Circuit Anomalies

- ADSP-2191/95/96 Anomaly List for Revision 1.0

Processor Manuals

- ADSP-219x/2191 DSP Hardware Reference
- ADSP 21xx Processors: Manuals
- ADSP-219x DSP Instruction Set Reference
- ADSP-219x/2192 DSP Hardware Reference (Rev 1.1, April 2004)
- Using the ADSP-2100 Family Volume 1
- Using the ADSP-2100 Family Volume 2

Product Highlight

- ADSP-2191 16-Bit Fixed Point DSP Product Brief

Software Manuals

- VisualDSP++ 3.5 Assembler and Preprocessor Manual for ADSP-218x and ADSP-219x DSPs
- VisualDSP++ 3.5 C Compiler and Library Manual for ADSP-218x DSPs
- VisualDSP++ 3.5 C/C++ Compiler and Library Manual for ADSP-219x Processors
- VisualDSP++ 3.5 Component Software Engineering User's Guide for 16-Bit Processors
- VisualDSP++ 3.5 Getting Started Guide for 16-Bit Processors
- VisualDSP++ 3.5 Kernel VDK User's Guide for 16-Bit Processors
- VisualDSP++ 3.5 Linker and Utilities Manual for 16-Bit Processors
- VisualDSP++ 3.5 Loader Manual for 16-Bit Processors
- VisualDSP++ 3.5 Quick Installation Reference Card
- VisualDSP++ 3.5 User's Guide for 16-Bit Processors

SOFTWARE AND SYSTEMS REQUIREMENTS

- Software and Tools Anomalies Search

TOOLS AND SIMULATIONS

- ADSP-2191M: LQFP package
- ADSP-21xx Processors: Software and Tools
- ADSP-2191M IBIS Datafile (BGA Package), Version 2.1
- ADSP-2191M IBIS Datafile (LQFP Package)
- ADSP-2191M IBIS Datafile (LQFP Package), Version 2.1

REFERENCE MATERIALS

Product Selection Guide

- ADI Complementary Parts Guide - Supervisory Devices and DSP Processors

DESIGN RESOURCES

- ADSP-2191M Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

DISCUSSIONS

View all ADSP-2191M EngineerZone Discussions.

SAMPLE AND BUY

Visit the product page to see pricing options.

TECHNICAL SUPPORT

Submit a technical question or find your regional support number.

DOCUMENT FEEDBACK

Submit feedback for this data sheet.

INTEGRATION FEATURES

160K 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

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.

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 3 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/ADSP-2191 DSP Hardware Reference*.

The peripheral clock is supplied to the CLKOUT pin.

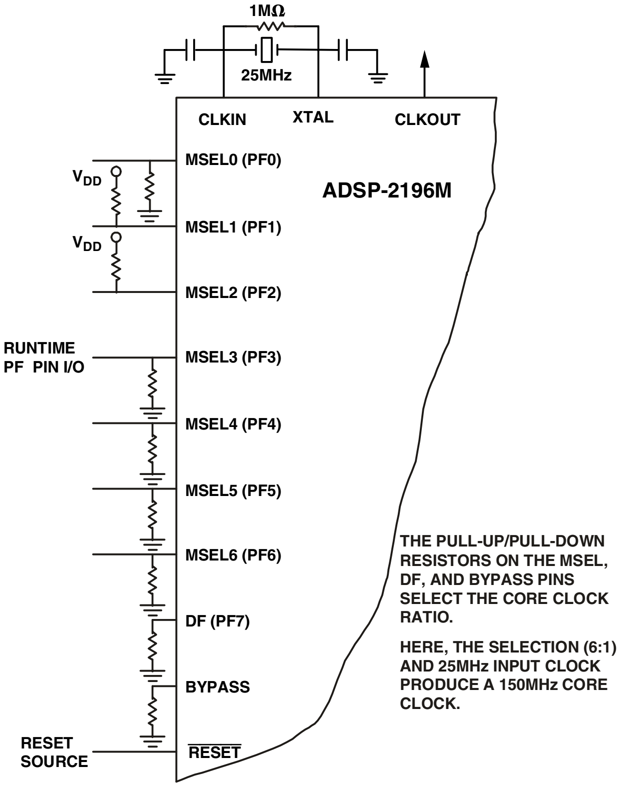
All on-chip peripherals for the ADSP-2191M operate at the rate set by the peripheral clock. The peripheral clock is either equal to the core clock rate or one-half the DSP core clock rate. This selection is controlled by the IOSEL bit in the PLLCTL register. The maximum core clock is 160 MHz and the 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}.

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 PULL-UP/PULL-DOWN RESISTORS ON THE MSEL, DF, AND BYPASS PINS SELECT THE CORE CLOCK RATIO.

HERE, THE SELECTION (6:1) AND 25MHz INPUT CLOCK PRODUCE A 150MHz CORE CLOCK.

Figure 3. External Crystal Connections

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.

Power-Up 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—Using an autobaud handshake sequence, a boot-stream-formatted program is downloaded by the Host. 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 ($\overline{\text{BR}}$) signal. The ($\overline{\text{BR}}$) 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.

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.

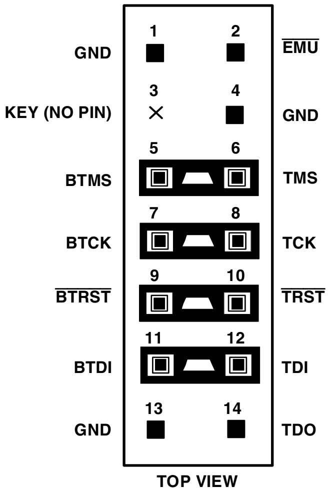


Figure 4. JTAG Target Board Connector for JTAG Equipped Analog Devices DSP (Jumpers in Place)

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.

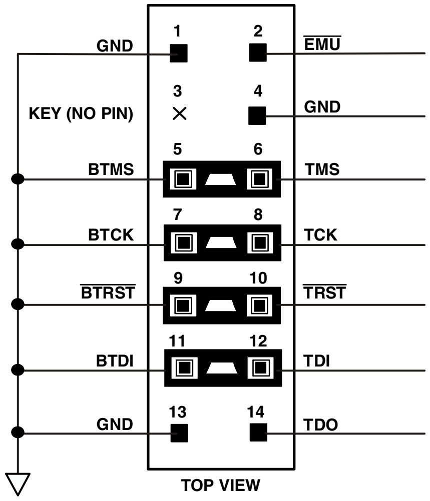


Figure 5. JTAG Target Board Connector with No Local Boundary Scan

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.

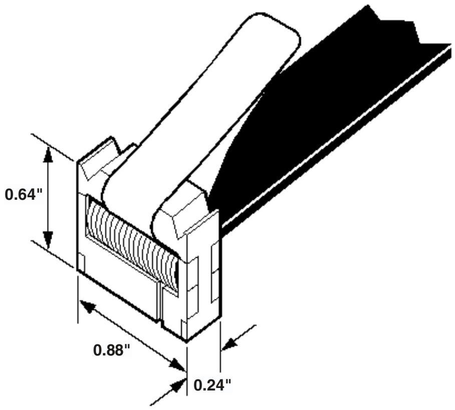


Figure 6. JTAG Pod Connector Dimensions

ABSOLUTE MAXIMUM RATINGS

V_{DDINT} Internal (Core) Supply Voltage[1] . . . –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] 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 4000 V 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 40.

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]	100 µA	600 µA	0	50 µA	100 µA	500 µA	0	50 µA
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.

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

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 22.

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}.

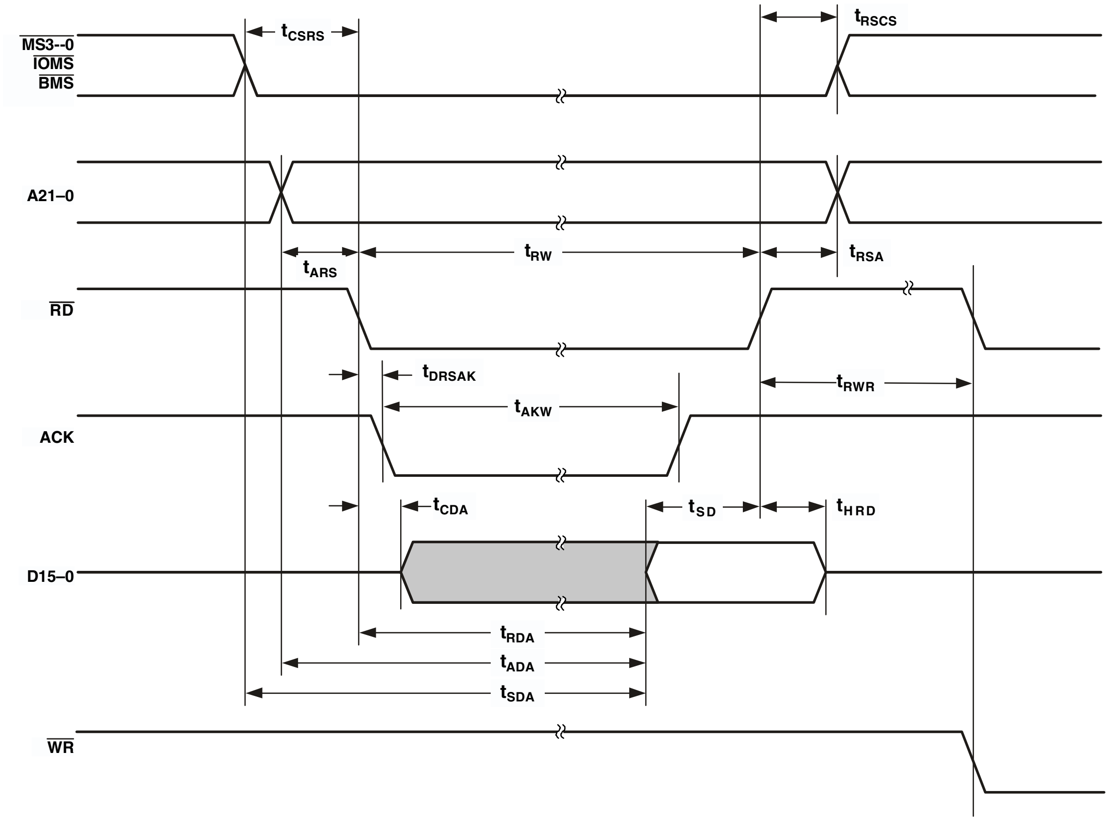


Figure 12. External Port Read Cycle Timing

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}	$\overline{\text{HWR}}$ Asserted to HACK Asserted (Setup, ACK Mode) First Byte	10	$5t_{HCLK}+t_{NH}$[1]	ns
t_{WHKS2}	$\overline{\text{HWR}}$ Asserted to HACK Asserted (Setup, ACK Mode)[2]		10	ns
t_{WHKH}	$\overline{\text{HWR}}$ Deasserted to HACK Deasserted (Hold, ACK Mode)		10	ns
t_{WHS}	$\overline{\text{HWR}}$ Asserted to HACK Asserted (Setup, Ready Mode)		10	ns
t_{WHH}	$\overline{\text{HWR}}$ Asserted to HACK Deasserted (Hold, Ready Mode) First Byte	0	$5t_{HCLK}+t_{NH}$[1]	ns
Timing Requirements				
t_{CSAL}	$\overline{\text{HCMS}}$ or $\overline{\text{HCIOMS}}$ Asserted to HALE Asserted	0		ns
t_{ALPW}	HALE Asserted Pulsewidth	4		ns
t_{ALCSW}	HALE Deasserted to $\overline{\text{HCMS}}$ or $\overline{\text{HCIOMS}}$ Deasserted	1		ns
t_{WCSW}	$\overline{\text{HWR}}$ Deasserted to $\overline{\text{HCMS}}$ or $\overline{\text{HCIOMS}}$ Deasserted	0		ns
t_{ALW}	HALE Deasserted to $\overline{\text{HWR}}$ Asserted	1		ns
t_{WCS}	$\overline{\text{HWR}}$ Deasserted (After Last Byte) to $\overline{\text{HCMS}}$ or $\overline{\text{HCIOMS}}$ Deasserted (Ready for Next Write)	0		ns
t_{HKWD}	HACK Asserted to $\overline{\text{HWR}}$ 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 $\overline{\text{HWR}}$ Deasserted (Setup)	4		ns
t_{WDH}	$\overline{\text{HWR}}$ 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).

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

[1] t_{NH} are peripheral bus latencies (n × 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).

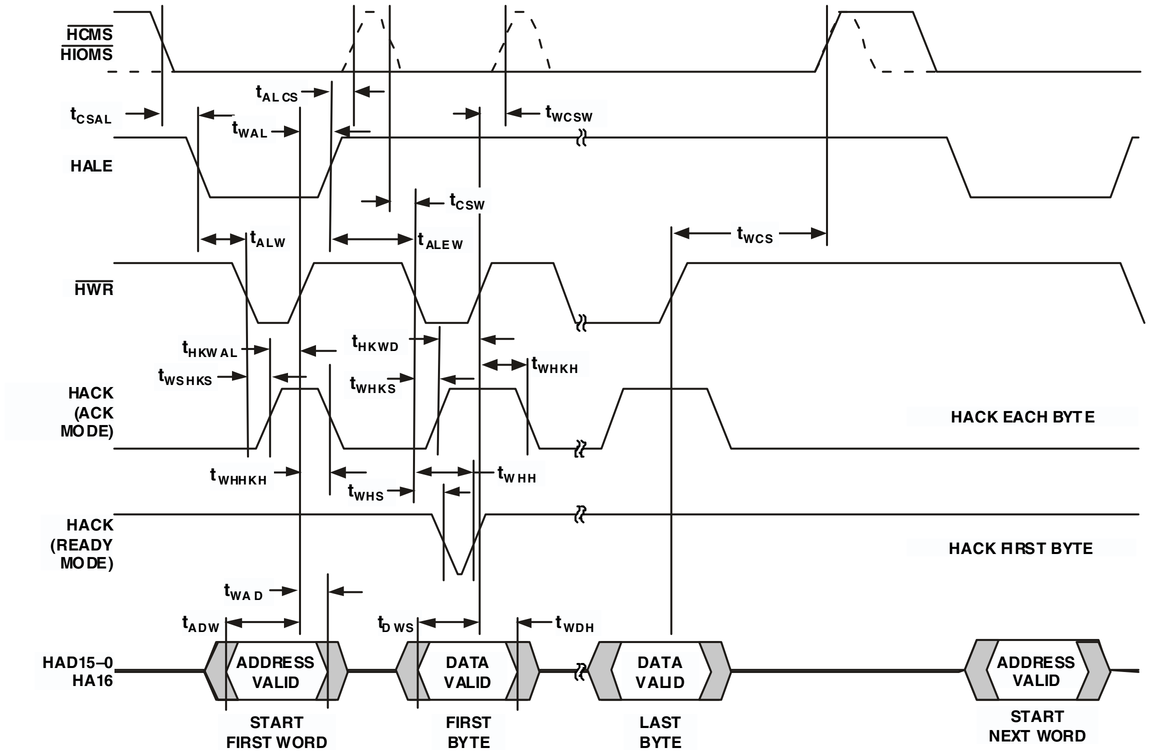


Figure 15. Host Port ACC Mode Write Cycle Timing

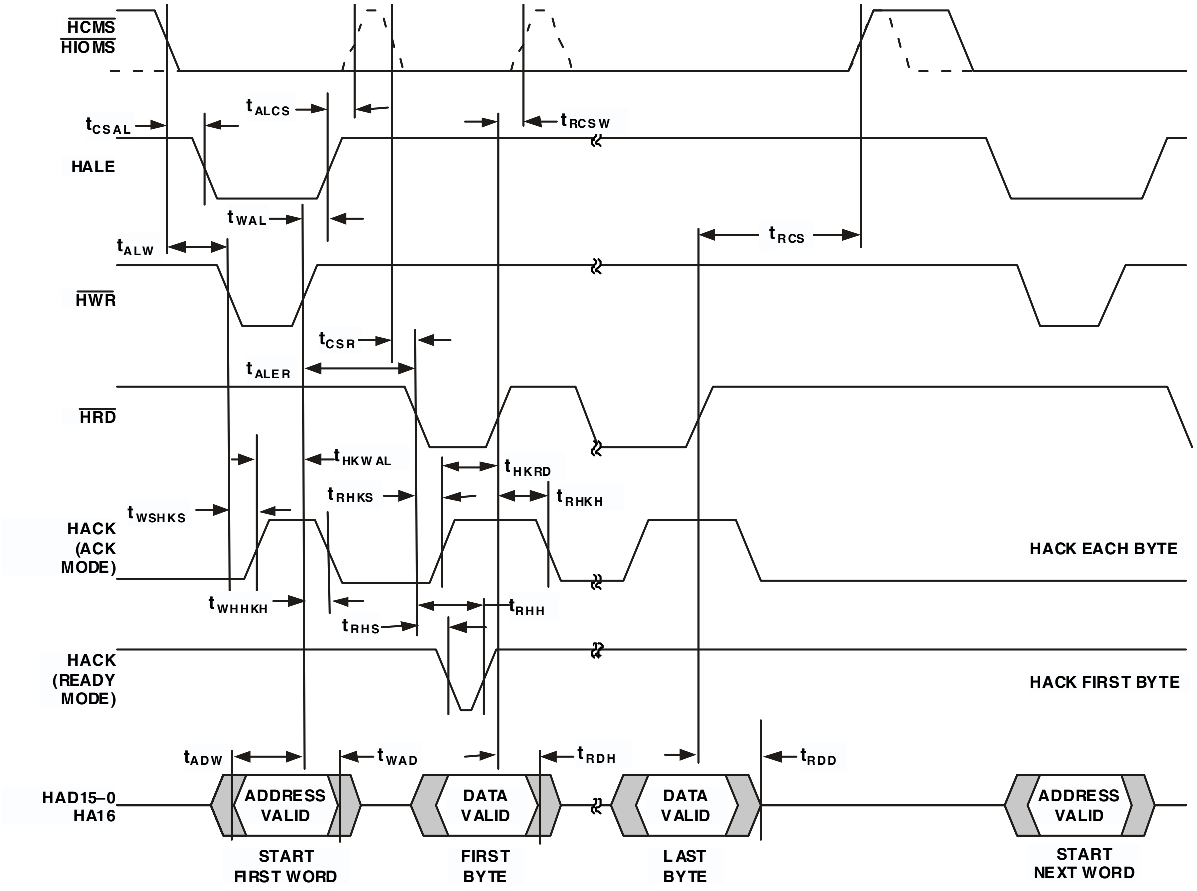
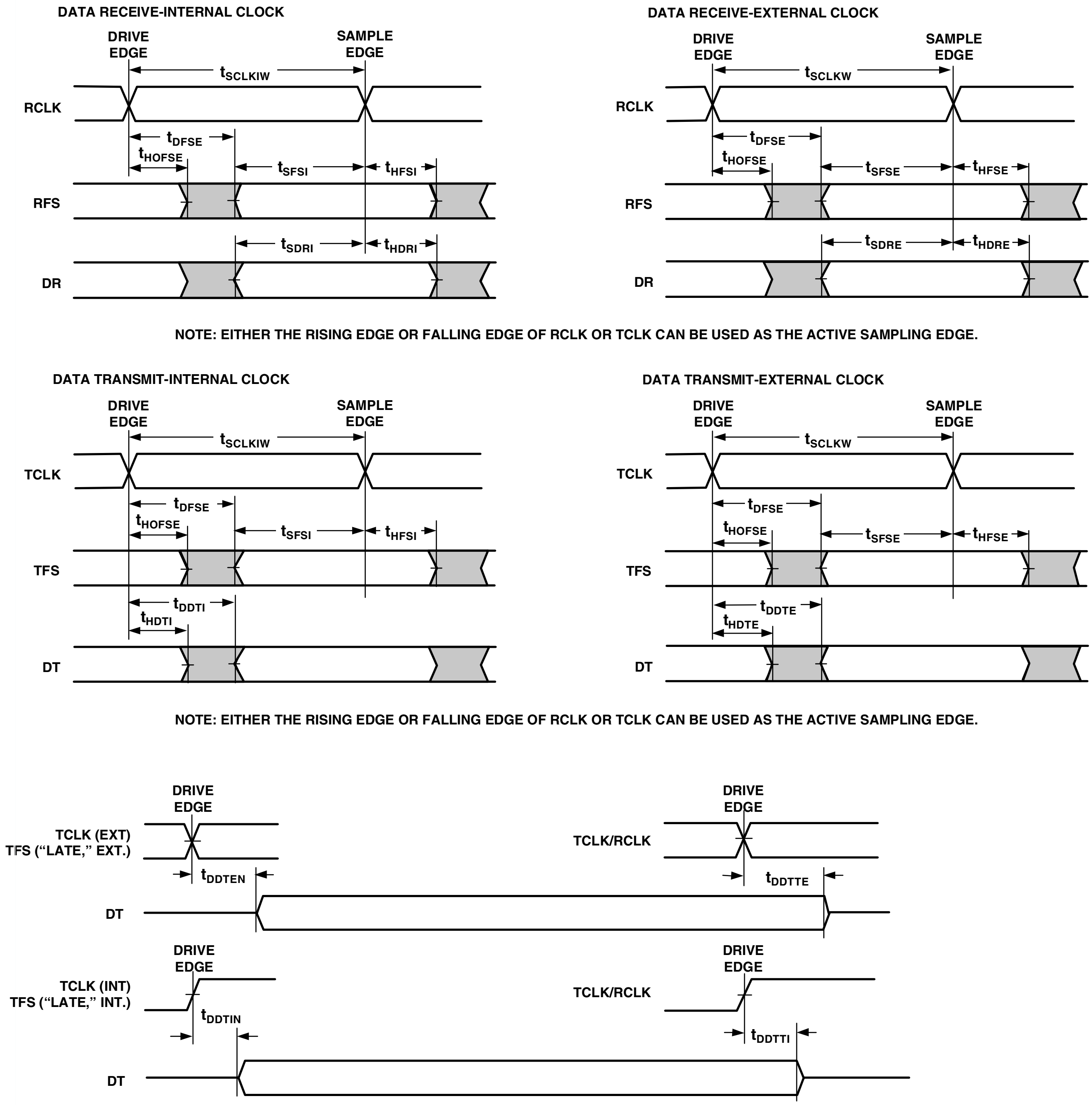


Figure 17. Host Port ACC Mode Read Cycle Timing



NOTE: EITHER THE RISING EDGE OR FALLING EDGE OF RCLK OR TCLK CAN BE USED AS THE ACTIVE SAMPLING EDGE.

NOTE: EITHER THE RISING EDGE OR FALLING EDGE OF RCLK OR TCLK CAN BE USED AS THE ACTIVE SAMPLING EDGE.

Figure 18. Serial Ports

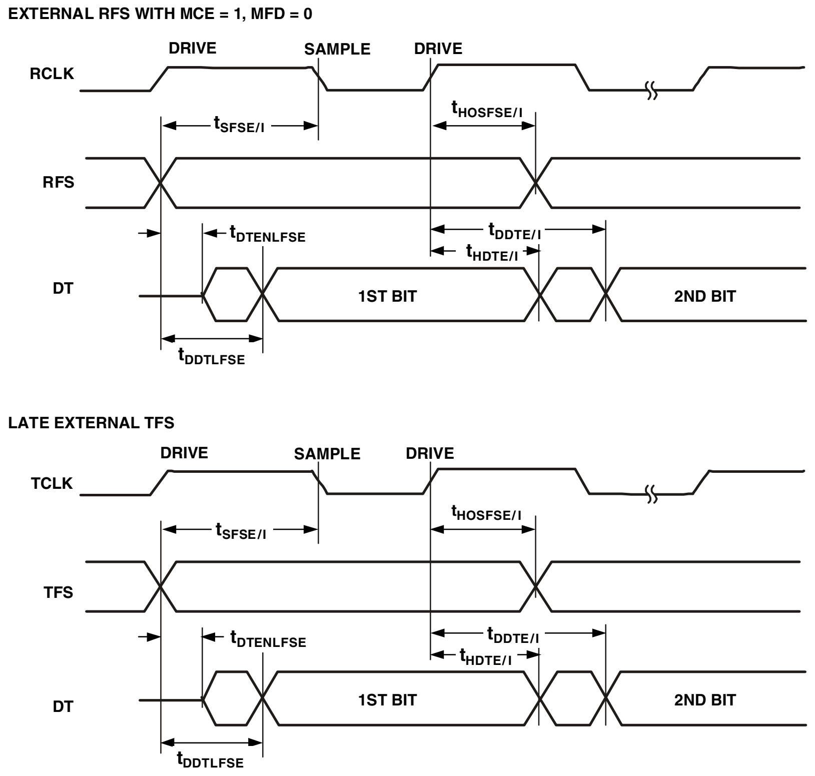


Figure 19. Serial Ports—External Late Frame Sync (Frame Sync Setup > 0.5t_{SCLK})

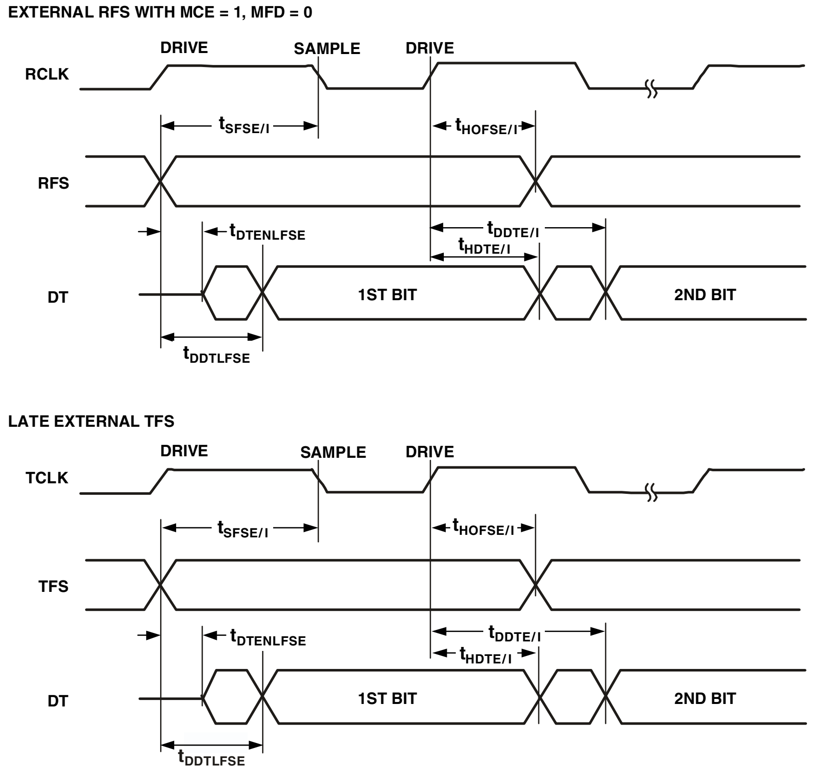


Figure 20. Serial Ports—External Late Frame Sync (Frame Sync Setup < 0.5t_{HCLK})

Serial Peripheral Interface (SPI) Port—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}	$\overline{\text{SPISS}}$ Assertion to Data Out Active	0	8	ns
t_{DSDHI}	$\overline{\text{SPISS}}$ 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 $\overline{\text{SPISS}}$ Not Asserted	$2t_{HCLK}$		ns
t_{SPITDS}	Sequential Transfer Delay	$2t_{HCLK}+4$		ns
t_{SDSCI}	$\overline{\text{SPISS}}$ 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

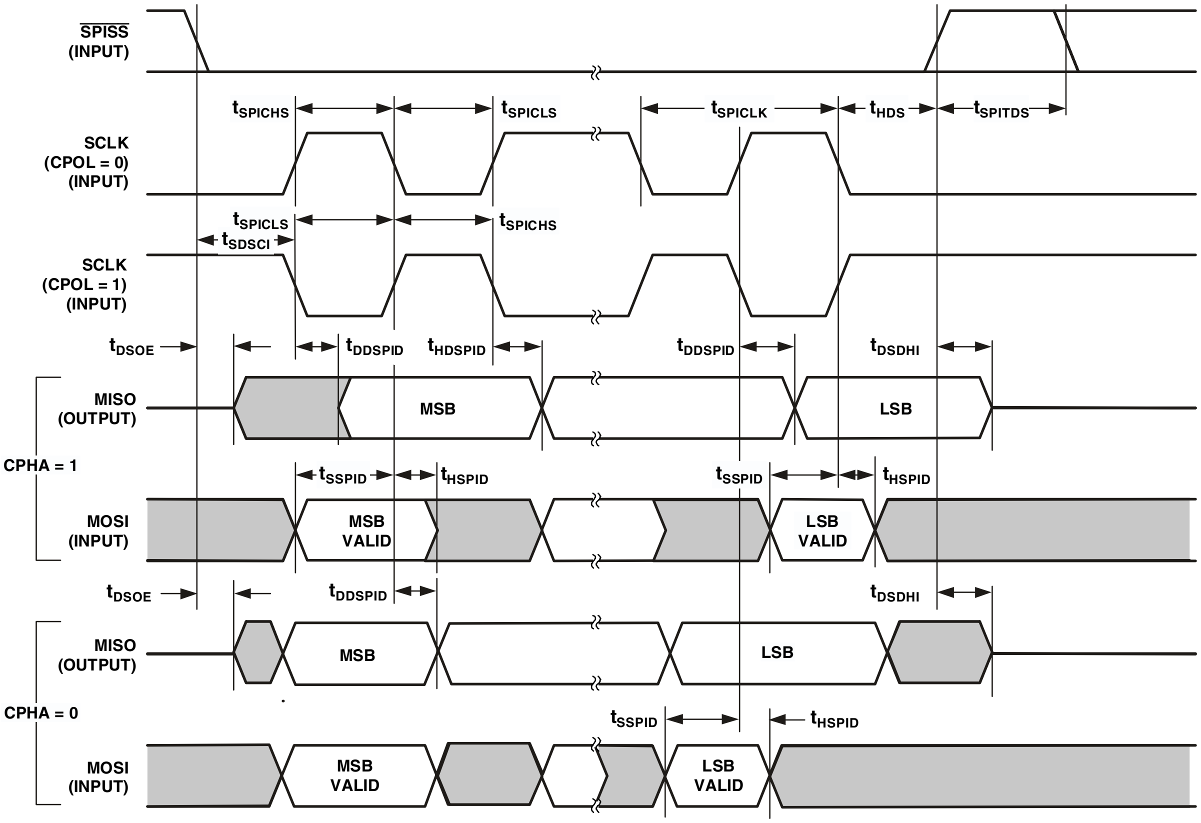


Figure 22. Serial Peripheral Interface (SPI) Port—Slave Timing

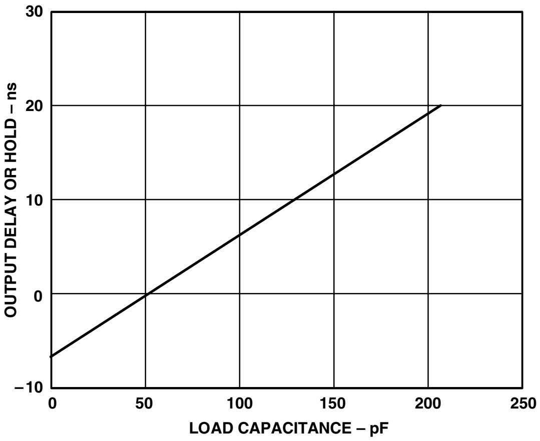


Figure 30. Typical Output Delay or Hold vs. Load Capacitance (at Maximum Case Temperature)

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.

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

Where:

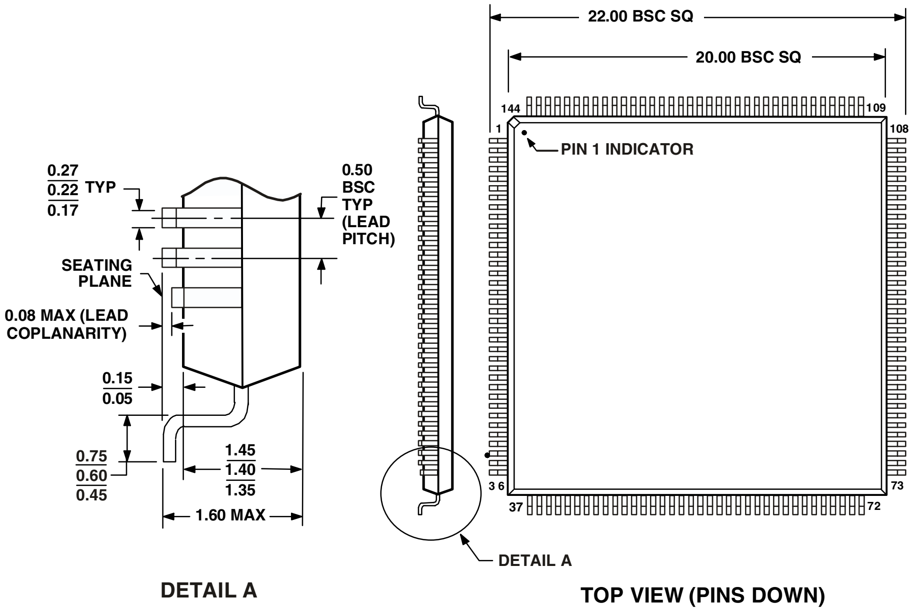
- T_{AMB} = Ambient temperature (measured near top surface of package)
- PD = Power dissipation in W (this value depends upon the specific application; a method for calculating PD is shown under Power Dissipation).
- θ_{CA} = Value from Table 24.
- For the LQFP package: θ_{JC} = 0.96°C/W
  For the mini-BGA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

OUTLINE DIMENSIONS

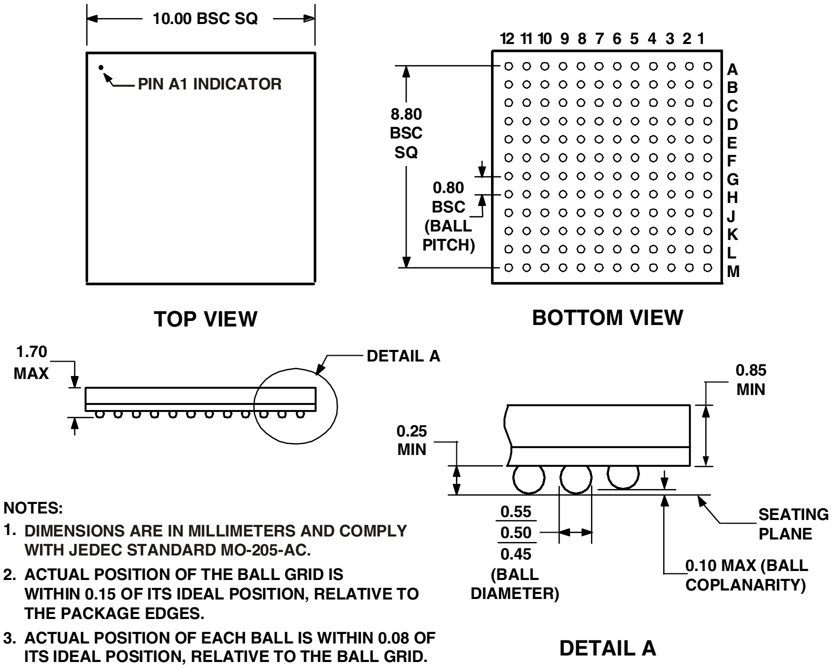
144-Lead Metric Thin Plastic Quad Flatpack [LQFP] (ST-144)



NOTES:

1. DIMENSIONS ARE IN MILLIMETERS AND COMPLY WITH JEDEC STANDARD MS-026-BFB.
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 [PBGA] (CA-144-2)



NOTES:

1. DIMENSIONS ARE IN MILLIMETERS AND COMPLY WITH JEDEC STANDARD MO-205-AC.
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.