Welcome to E-XFL.COM

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	Active
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	2.50V
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-2191mbstz-140

- 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

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

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.

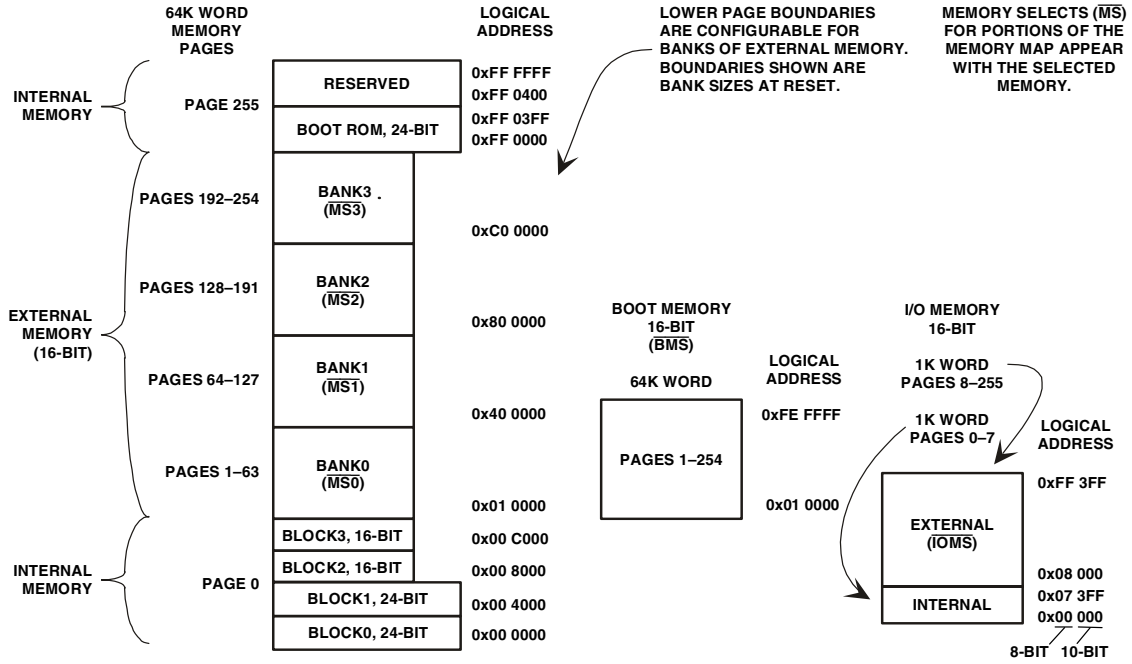


Figure 2. Memory Map

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.

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),

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.

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 ($\overline{BR}$) 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.

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

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).

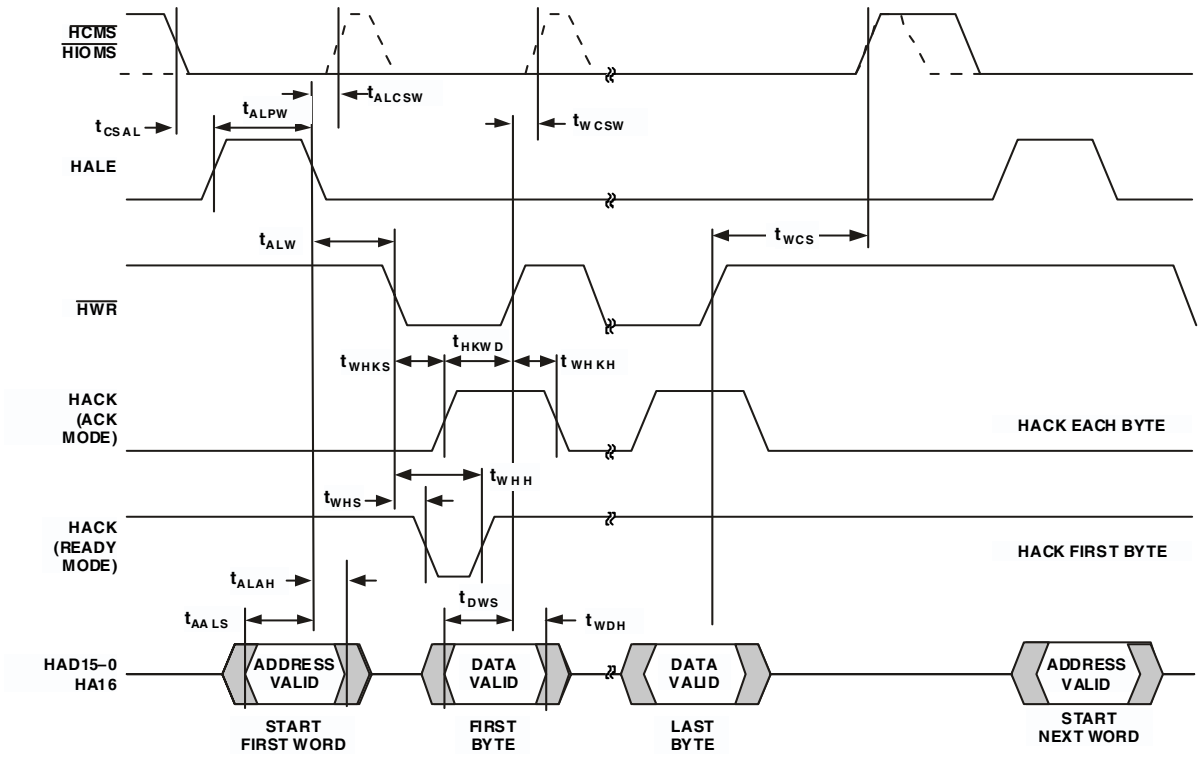


Figure 14. Host Port ALE Mode Write Cycle Timing

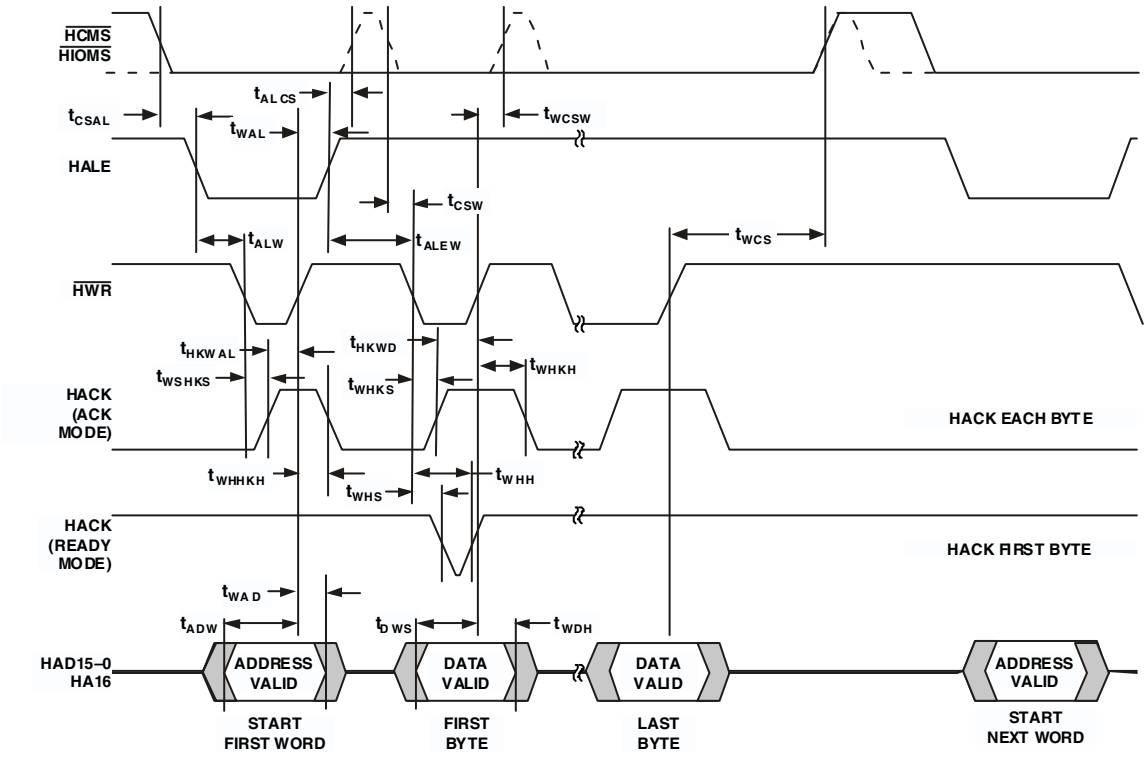


Figure 15. Host Port ACC Mode Write Cycle Timing

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).

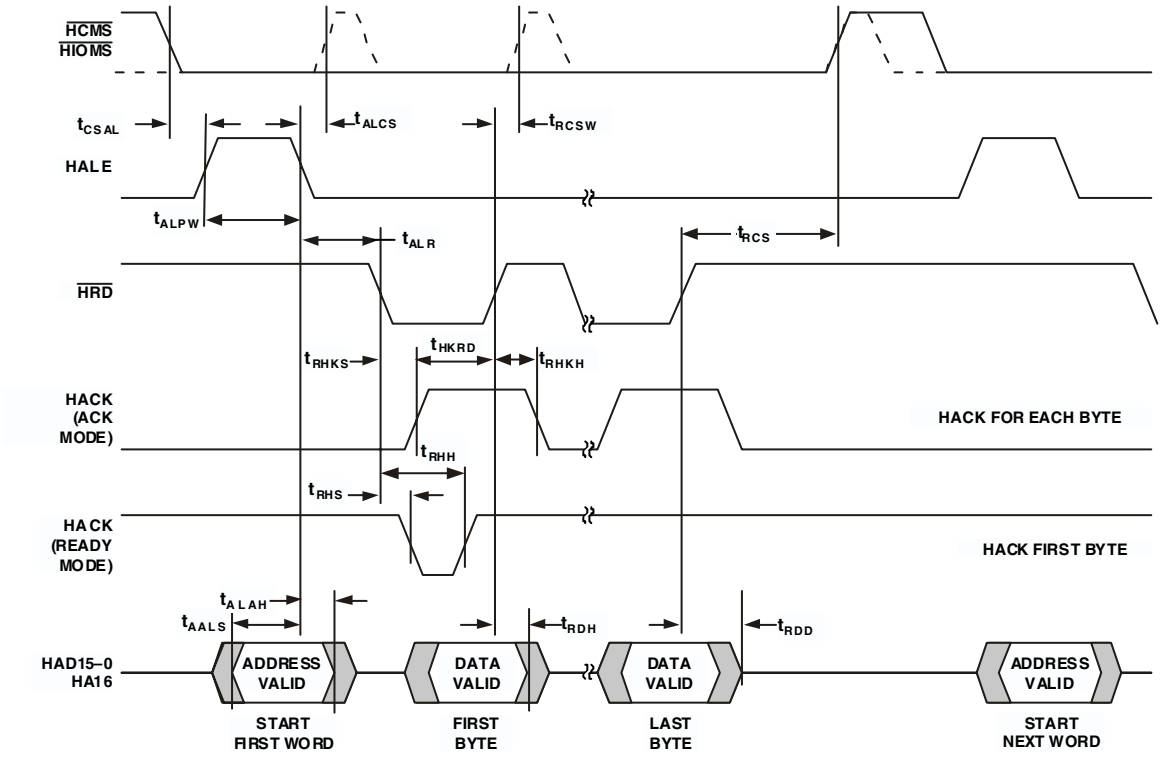


Figure 16. Host Port ALE Mode Read Cycle Timing

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).

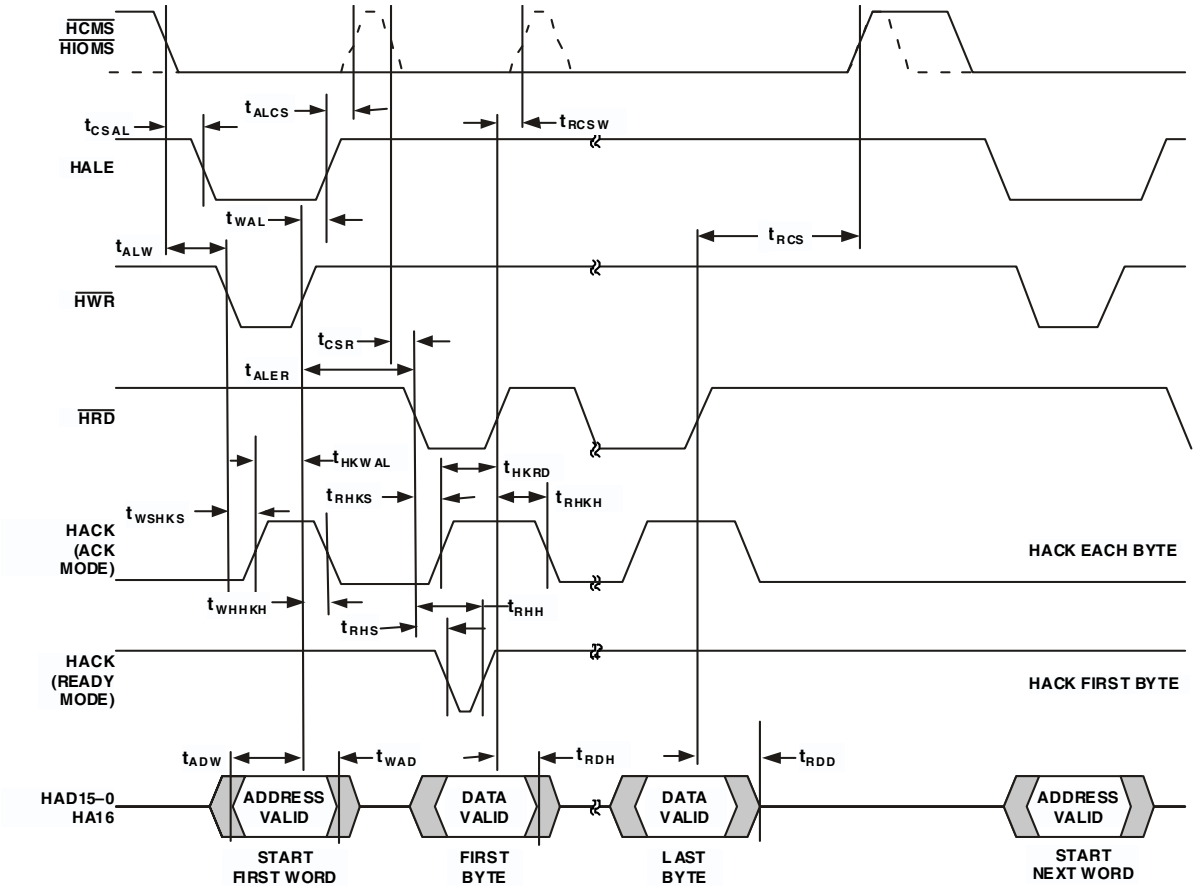


Figure 17. Host Port ACC Mode Read Cycle Timing

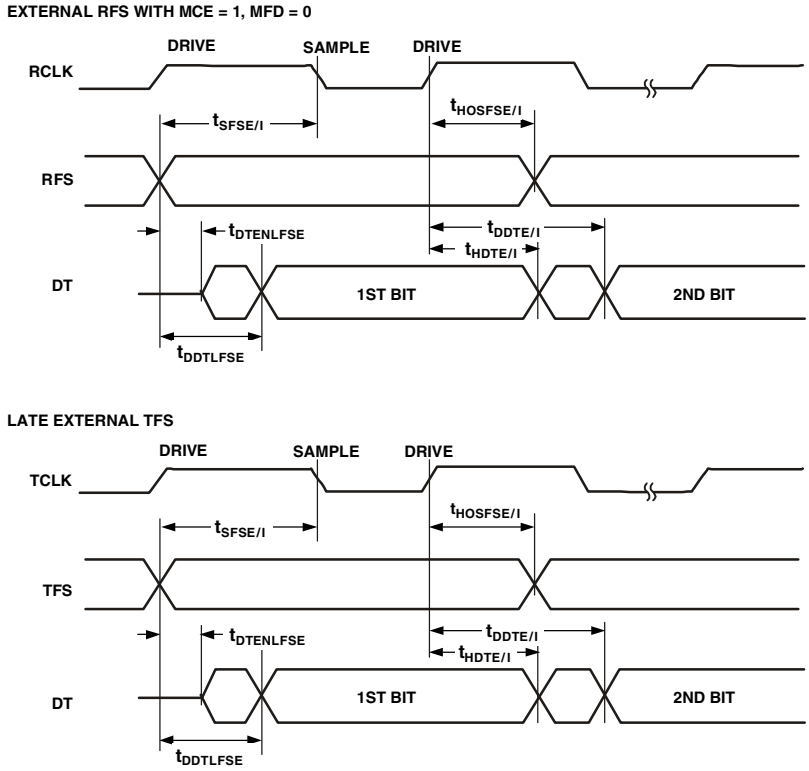


Figure 19. Serial Ports—External Late Frame Sync (Frame Sync Setup > 0.5t_{SCLK})

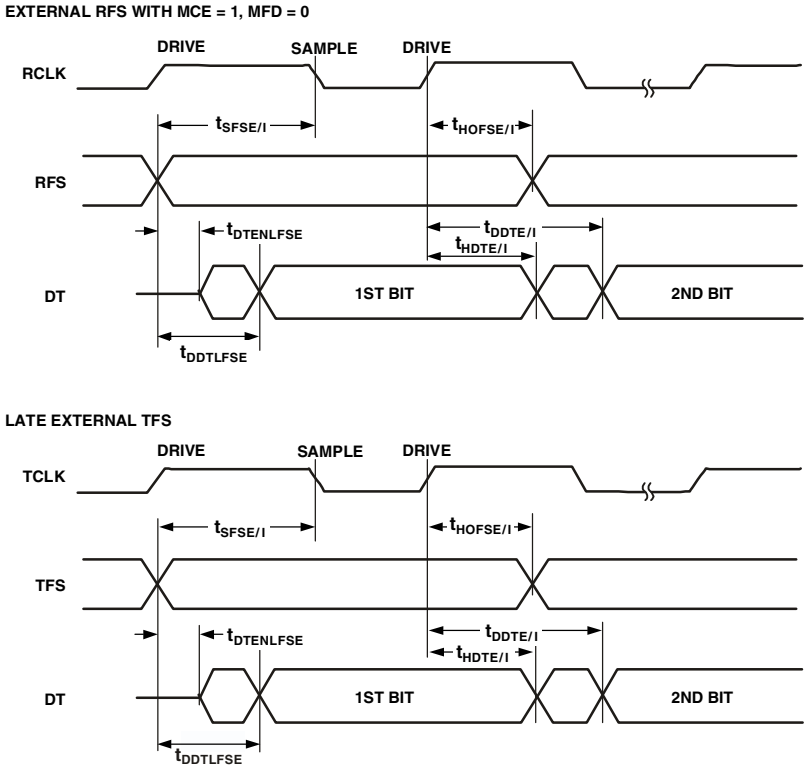


Figure 20. Serial Ports—External Late Frame Sync (Frame Sync Setup < 0.5t_{HCLK})

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}	$\overline{SPIxSEL}$ 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 $\overline{SPIxSEL}$ 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

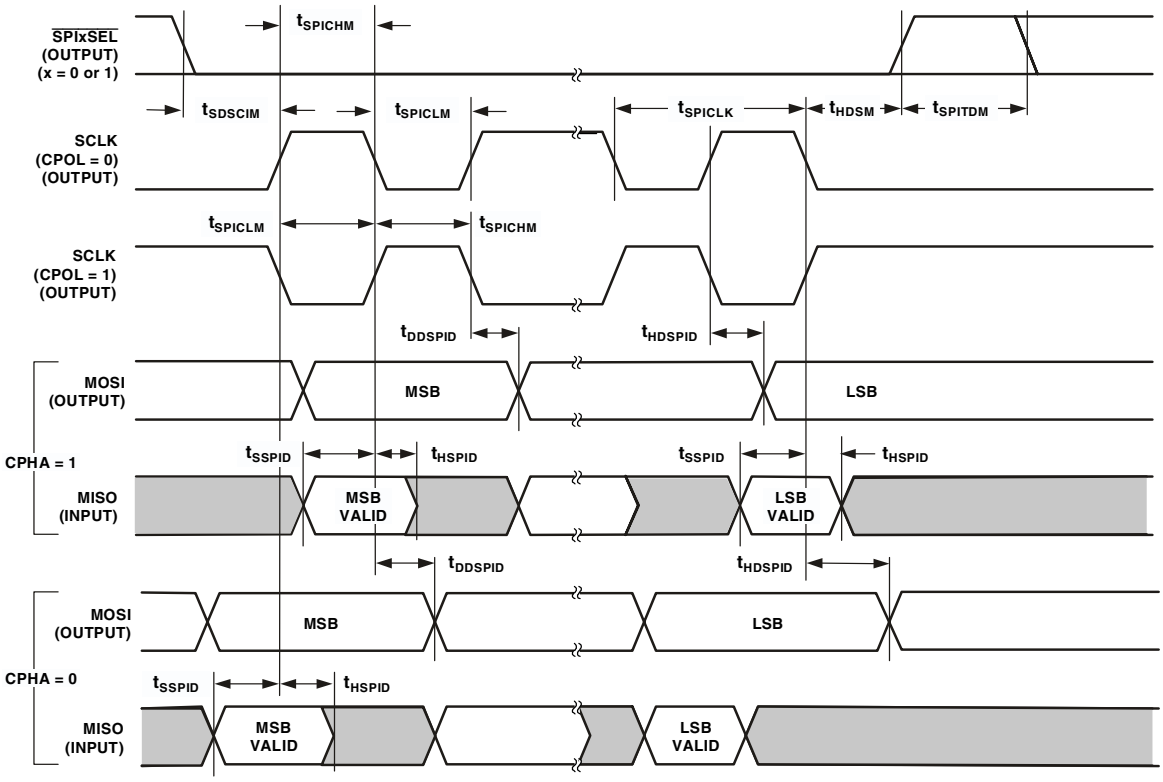


Figure 21. Serial Peripheral Interface (SPI) Port—Master Timing

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

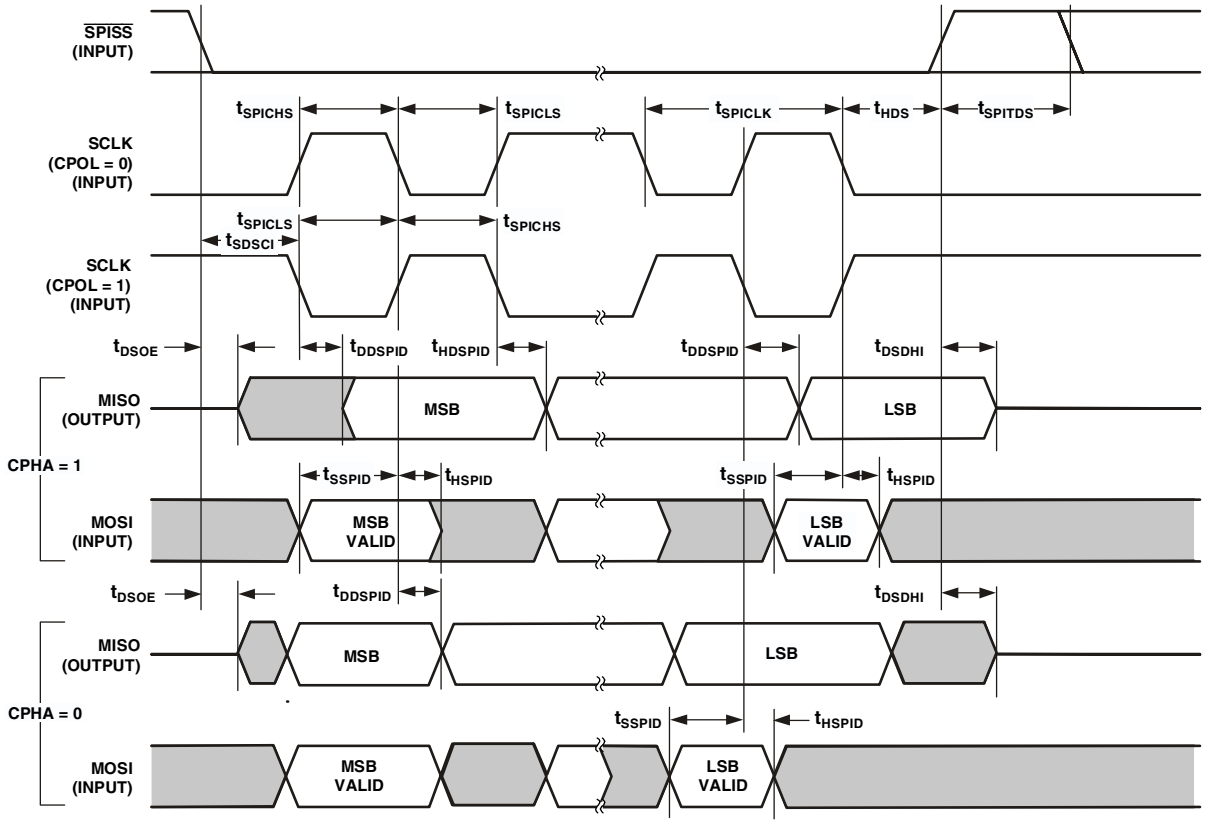


Figure 22. Serial Peripheral Interface (SPI) Port—Slave Timing

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.

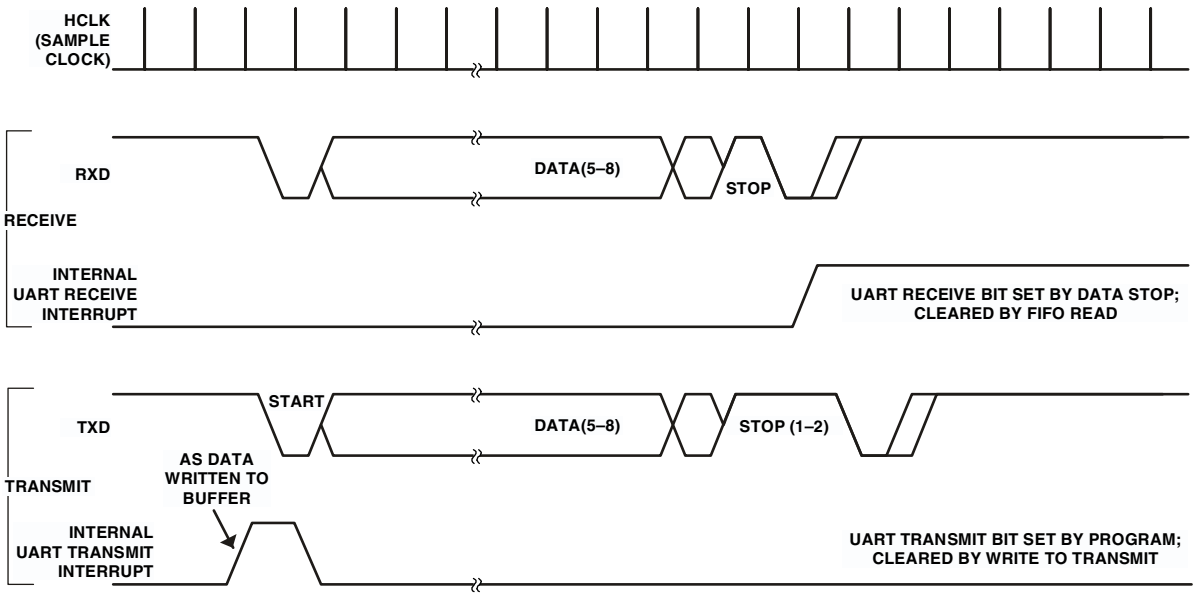


Figure 23. UART Port—Receive and Transmit Timing

144-Lead Mini-BGA Pinout

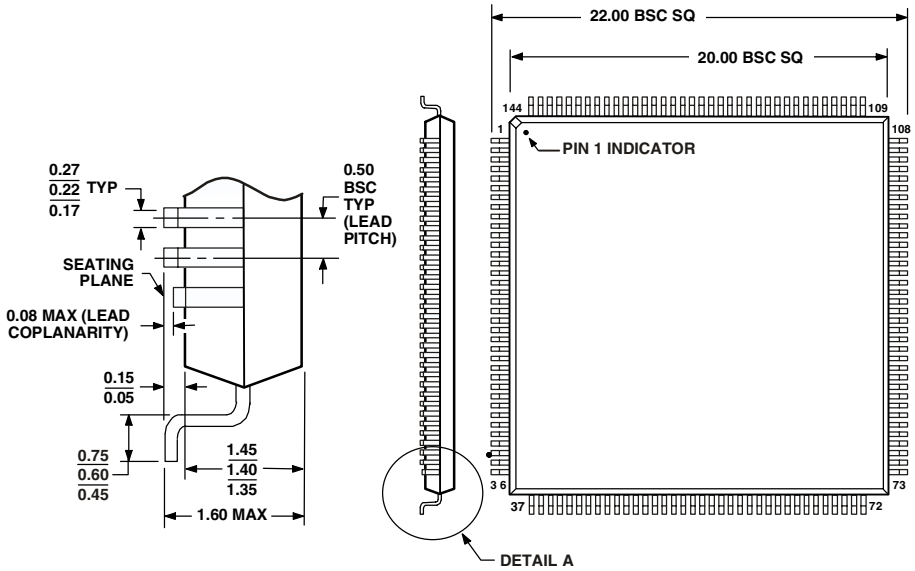
Table 27 lists the mini-BGA pinout by signal name. Table 28 lists the mini-BGA pinout by ball number.

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		

OUTLINE DIMENSIONS

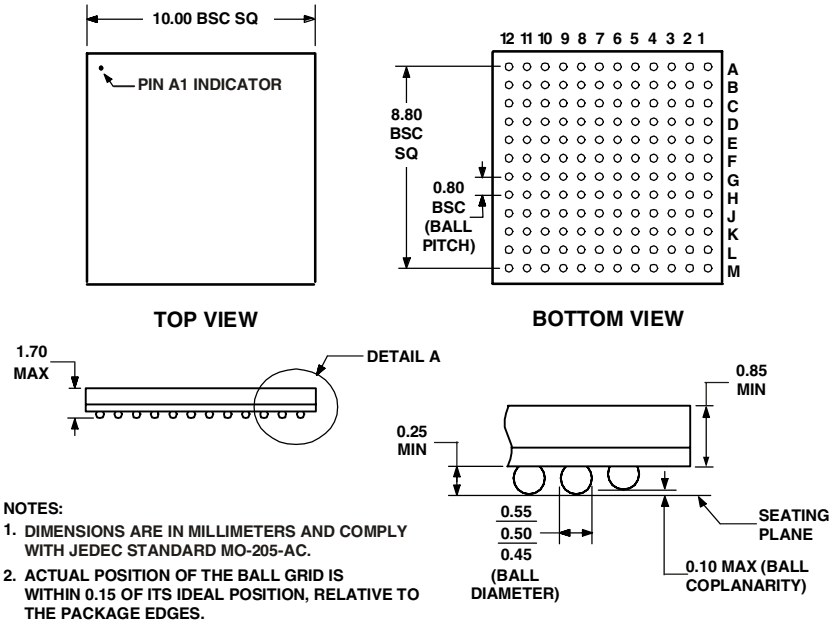
144-Lead Metric Thin Plastic Quad Flatpack [LQFP]
(ST-144)



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.

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 (PBGA)

Revision History

Location	Page
7/02—Changed from Rev. 0 to Rev. A	
Changes to formatting only	Global

C02936–0–7/02(A)

PRINTED IN U.S.A.