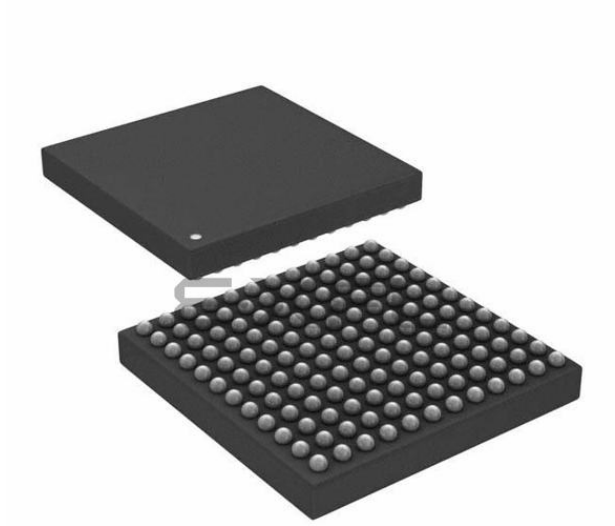


**Welcome to E-XFL.COM**

## 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 | Obsolete |
| Type | Fixed Point |
| Interface | Host Interface, Serial Port |
| Clock Rate | 75MHz |
| Non-Volatile Memory | External |
| On-Chip RAM | 40kB |
| Voltage - I/O | 3.30V |
| Voltage - Core | 2.50V |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 144-LFBGA |
| Supplier Device Package | 144-MiniBGA (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/analog-devices/adsp-2186mkca-300 |

# ADSP-2186M* PRODUCT PAGE QUICK LINKS

**Last Content Update: 02/23/2017**

## COMPARABLE PARTS

View a parametric search of comparable parts.

## DOCUMENTATION

**Application Notes**

- AN-227: Digital Control System Design with the ADSP-2100 Family
- AN-227: Digital Control System Design with the ADSP-2100 Family
- AN-334: Digital Signal Processing Techniques
- AN-524: ADV601/ADV611 Bin Width Calculation in ADSP-21xx DSP
- 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-103: Performing Level Conversion Between 5v and 3.3v IC's
- EE-104: Setting Up Streams with the VisualDSP Debugger
- EE-11: ADSP-2181 Priority Chain & IDMA Holdoffs
- EE-110: A Quick Primer on ELF and DWARF File Formats
- 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-33: Programming The ADSP-21xx Timer In C
- EE-35: Troubleshooting your ADSP-218x EZ-ICE
- EE-356: Emulator and Evaluation Hardware Troubleshooting Guide for CCES Users
- EE-36: ADSP-21xx Interface to the IOM-2 bus
- EE-38: ADSP-2181 IDMA Port - Cycle Steal Timing
- EE-39: Interfacing 5V Flash Memory to an ADSP-218x (Byte Programming Algorithm)
- EE-48: Converting Legacy 21xx Systems To A 218x System Design
- 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-90: Using the 21xx C-FFT Library
- EE-96: Interfacing Two AD73311 Codecs to the ADSP-218x

**Data Sheet**

- ADSP-2186M: 16-Bit, 75 MIPS, 2.5V, 2 Serial Ports, Host Port, 40 KB RAM Data Sheet

**Evaluation Kit Manuals**

- ADSP-218x DSP family and ADSP-2192 EZ-KIT Lite® Installation Procedure -Non-USB

**Integrated Circuit Anomalies**

- ADSP-2186M Anomaly List for Revision 2.0

**Processor Manuals**

- ADSP 21xx Processors: Manuals
- ADSP-218x DSP Hardware Reference
- ADSP-218x DSP Instruction Set Reference
- Using the ADSP-2100 Family Volume 1
- Using the ADSP-2100 Family Volume 2

**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 User's Guide for 16-Bit Processors

## SOFTWARE AND SYSTEMS REQUIREMENTS

- Software and Tools Anomalies Search

## TOOLS AND SIMULATIONS

- ADSP-218xM IBIS Datafile (LQFP Package)

## DESIGN RESOURCES

- ADSP-2186M Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

## DISCUSSIONS

View all ADSP-2186M 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.

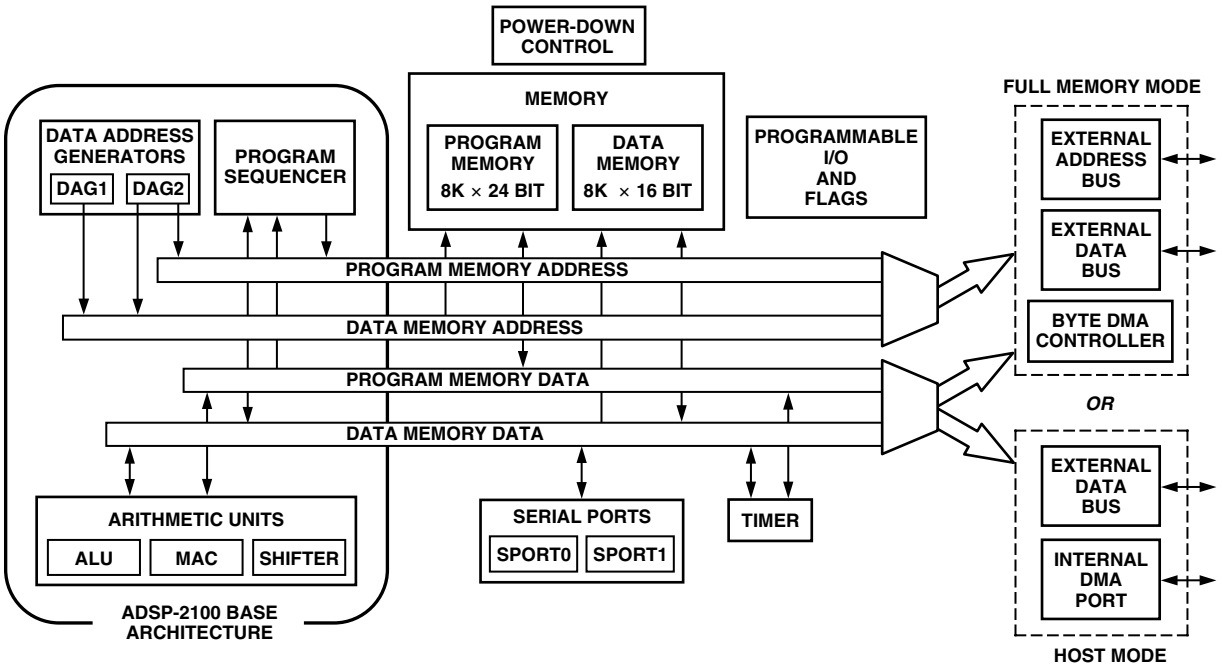



*Figure 1. Functional Block Diagram*

**ARCHITECTURE OVERVIEW**

The ADSP-2186M instruction set provides flexible data moves and multifunction (one or two data moves with a computation) instructions. Every instruction can be executed in a single processor cycle. The ADSP-2186M assembly language uses an algebraic syntax for ease of coding and readability. A comprehensive set of development tools supports program development.

Figure 1 is an overall block diagram of the ADSP-2186M. The processor contains three independent computational units: the ALU, the multiplier/accumulator (MAC), and the shifter. The computational units process 16-bit data directly 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 with 40 bits of accumulation. 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.

The internal result (R) bus connects the computational units so that the output of any unit may be the input of any unit on the next cycle.

A powerful program sequencer and two dedicated data address generators ensure efficient delivery of operands to these computational units. The sequencer supports conditional jumps, subroutine calls, and returns in a single cycle. With internal loop counters and loop stacks, the ADSP-2186M 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 address pointers. Whenever the pointer is used to access data (indirect addressing), it is post-modified by the value of one of four possible modify registers. A length value may be associated with each pointer to implement automatic modulo addressing for circular buffers.

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

- Program Memory Address (PMA) Bus
- Program Memory Data (PMD) Bus
- Data Memory Address (DMA) Bus
- Data Memory Data (DMD) Bus
- Result (R) 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. Byte memory space and I/O memory space also share the external buses.

Program memory can store both instructions and data, permitting the ADSP-2186M to fetch two operands in a single cycle, one from program memory and one from data memory. The ADSP-2186M can fetch an operand from program memory and the next instruction in the same cycle.

In lieu of the address and data bus for external memory connection, the ADSP-2186M may be configured for 16-bit Internal DMA port (IDMA port) connection to external systems. The IDMA port is made up of 16 data/address pins and five control pins. The IDMA port provides transparent, direct access to the DSPs on-chip program and data RAM.

An interface to low-cost byte-wide memory is provided by the Byte DMA port (BDMA port). The BDMA port is bidirectional and can directly address up to four megabytes of external RAM or ROM for off-chip storage of program overlays or data tables.

The byte memory and I/O memory space interface supports slow memories and I/O memory-mapped peripherals with programmable wait state generation. External devices can gain control of

**Memory Interface Pins**

The ADSP-2186M processor can be used in one of two modes: Full Memory Mode, which allows BDMA operation with full external overlay memory and I/O capability, or Host Mode, which allows IDMA operation with limited external addressing capabilities. The operating mode is determined by the state of the Mode C pin during $\overline{\text{RESET}}$ and cannot be changed while the processor is running.

The following tables list the active signals at specific pins of the DSP during either of the two operating modes (Full Memory or Host). A signal in one table shares a pin with a signal from the other table, with the active signal determined by the mode set. For the shared pins and their alternate signals (e.g., A4/IAD3), refer to the package pinout tables.

**Full Memory Mode Pins (Mode C = 0)**

| Pin Name | # of Pins | I/O | Function |
|---|---|---|---|
| A13:0 | 14 | O | Address Output Pins for Program, Data, Byte, and I/O Spaces |
| D23:0 | 24 | I/O | Data I/O Pins for Program, Data, Byte, and I/O Spaces (8 MSBs are also used as Byte Memory Addresses.) |

**Host Mode Pins (Mode C = 1)**

| Pin Name | # of Pins | I/O | Function |
|---|---|---|---|
| IAD15:0 | 16 | I/O | IDMA Port Address/Data Bus |
| A0 | 1 | O | Address Pin for External I/O, Program, Data, or Byte Access[1] |
| D23:8 | 16 | I/O | Data I/O Pins for Program, Data, Byte, and I/O Spaces |
| $\overline{\text{IWR}}$ | 1 | I | IDMA Write Enable |
| $\overline{\text{IRD}}$ | 1 | I | IDMA Read Enable |
| IAL | 1 | I | IDMA Address Latch Pin |
| $\overline{\text{IS}}$ | 1 | I | IDMA Select |
| $\overline{\text{IACK}}$ | 1 | O | IDMA Port Acknowledge Configurable in Mode D; Open Drain |

NOTE

[1]In Host Mode, external peripheral addresses can be decoded using the A0, $\overline{\text{CMS}}$, $\overline{\text{PMS}}$, $\overline{\text{DMS}}$, and $\overline{\text{IOMS}}$ signals.

**Interrupts**

The interrupt controller allows the processor to respond to the 11 possible interrupts and reset with minimum overhead. The ADSP-2186M provides four dedicated external interrupt input pins: $\overline{IRQ2}$, $\overline{IRQL0}$, $\overline{IRQL1}$, and $\overline{IRQE}$ (shared with the PF7:4 pins). In addition, SPORT1 may be reconfigured for $\overline{IRQ0}$, $\overline{IRQ1}$, FI and FO, for a total of six external interrupts. The ADSP-2186M also supports internal interrupts from the timer, the byte DMA port, the two serial ports, software, and the power-down control circuit. The interrupt levels are internally prioritized and individually maskable (except power- down and reset). The $\overline{IRQ2}$, $\overline{IRQ0}$, and $\overline{IRQ1}$ input pins can be programmed to be either level- or edge-sensitive. $\overline{IRQL0}$ and $\overline{IRQL1}$ are level-sensitive and $\overline{IRQE}$ is edge-sensitive. The priorities and vector addresses of all interrupts are shown in Table I.

**Table I. Interrupt Priority and Interrupt Vector Addresses**

| Source Of Interrupt | Interrupt Vector Address (Hex) |
|---|---|
| Reset (or Power-Up with PUCR = 1) | 0000 (Highest Priority) |
| Power-Down (Nonmaskable) | 002C |
| $\overline{IRQ2}$ | 0004 |
| $\overline{IRQL1}$ | 0008 |
| $\overline{IRQL0}$ | 000C |
| SPORT0 Transmit | 0010 |
| SPORT0 Receive | 0014 |
| $\overline{IRQE}$ | 0018 |
| BDMA Interrupt | 001C |
| SPORT1 Transmit or $\overline{IRQ1}$ | 0020 |
| SPORT1 Receive or $\overline{IRQ0}$ | 0024 |
| Timer | 0028 (Lowest Priority) |

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 power-down interrupt is nonmaskable.

The ADSP-2186M masks all interrupts for one instruction cycle following the execution of an instruction that modifies the IMASK register. This does not affect serial port autobuffering or DMA transfers.

The interrupt control register, ICNTL, controls interrupt nesting and defines the $\overline{IRQ0}$, $\overline{IRQ1}$, and $\overline{IRQ2}$ external interrupts to be either edge- or level-sensitive. The $\overline{IRQE}$ pin is an external edge sensitive interrupt and can be forced and cleared. The $\overline{IRQL0}$ and $\overline{IRQL1}$ pins are external level sensitive interrupts.

The IFC register is a write-only register used to force and clear interrupts. On-chip stacks preserve the processor status and are automatically maintained during interrupt handling. The stacks are twelve levels deep to allow interrupt, loop, and subroutine nesting. The following instructions allow global enable or disable servicing of the interrupts (including power down), regardless of the state of IMASK. Disabling the interrupts does not affect serial port autobuffering or DMA.

*ENA INTS;*
*DIS INTS;*

When the processor is reset, interrupt servicing is enabled.

**LOW POWER OPERATION**

The ADSP-2186M has three low power modes that significantly reduce the power dissipation when the device operates under standby conditions. These modes are:

- Power-Down
- Idle
- Slow Idle

The CLKOUT pin may also be disabled to reduce external power dissipation.

**Power-Down**

The ADSP-2186M processor has a low power feature that lets the processor enter a very low-power dormant state through hardware or software control. Following is a brief list of power-down features. Refer to the *ADSP-2100 Family User's Manual*, "System Interface" chapter, for detailed information about the power-down feature.

- Quick recovery from power-down. The processor begins executing instructions in as few as 200 CLKIN cycles.
- Support for an externally generated TTL or CMOS processor clock. The external clock can continue running during power-down without affecting the lowest power rating and 200 CLKIN cycle recovery.
- Support for crystal operation includes disabling the oscillator to save power (the processor automatically waits approximately 4096 CLKIN cycles for the crystal oscillator to start or stabilize), and letting the oscillator run to allow 200 CLKIN cycle start-up.
- Power-down is initiated by either the power-down pin ($\overline{PWD}$) or the software power-down force bit. Interrupt support allows an unlimited number of instructions to be executed before optionally powering down. The power-down interrupt also can be used as a nonmaskable, edge-sensitive interrupt.
- Context clear/save control allows the processor to continue where it left off or start with a clean context when leaving the power-down state.
- The $\overline{RESET}$ pin also can be used to terminate power-down.
- Power-down acknowledge pin indicates when the processor has entered power-down.

**Idle**

When the ADSP-2186M is in the Idle Mode, the processor waits indefinitely in a low-power state until an interrupt occurs. When an unmasked interrupt occurs, it is serviced; execution then continues with the instruction following the IDLE instruction. In Idle mode IDMA, BDMA and autobuffer cycle steals still occur.

**Active Configuration**
Active Configuration involves the use of a three-statable external driver connected to the Mode C pin. A driver's output enable should be connected to the DSP's $\overline{\text{RESET}}$ signal such that it only drives the PF2 pin when $\overline{\text{RESET}}$ is active (low). When $\overline{\text{RESET}}$ is deasserted, the driver should three-state, thus allowing full use of the PF2 pin as either an input or output. To minimize power consumption during power-down, configure the programmable flag as an output when connected to a three-stated buffer. This ensures that the pin will be held at a constant level, and will not oscillate should the three-state driver's level hover around the logic switching point.

**$\overline{\text{IACK}}$ Configuration**
Mode D = 0 and in host mode: $\overline{\text{IACK}}$ is an active, driven signal and cannot be "wire OR'd."

Mode D = 1 and in host mode: $\overline{\text{IACK}}$ is an open drain and requires an external pull-down, but multiple $\overline{\text{IACK}}$ pins can be "wire OR'd" together.

**MEMORY ARCHITECTURE**
The ADSP-2186M provides a variety of memory and peripheral interface options. The key functional groups are Program Memory, Data Memory, Byte Memory, and I/O. Refer to the following figures and tables for PM and DM memory allocations in the ADSP-2186M.

**Program Memory**
**Program Memory (Full Memory Mode)** is a 24-bit-wide space for storing both instruction opcodes and data. The ADSP-2186M has 8K words of Program Memory RAM on chip, and the capability of accessing up to two 8K external memory overlay spaces using the external data bus.

**Program Memory (Host Mode)** allows access to all internal memory. External overlay access is limited by a single external address line (A0). External program execution is not available in host mode due to a restricted data bus that is 16 bits wide only.

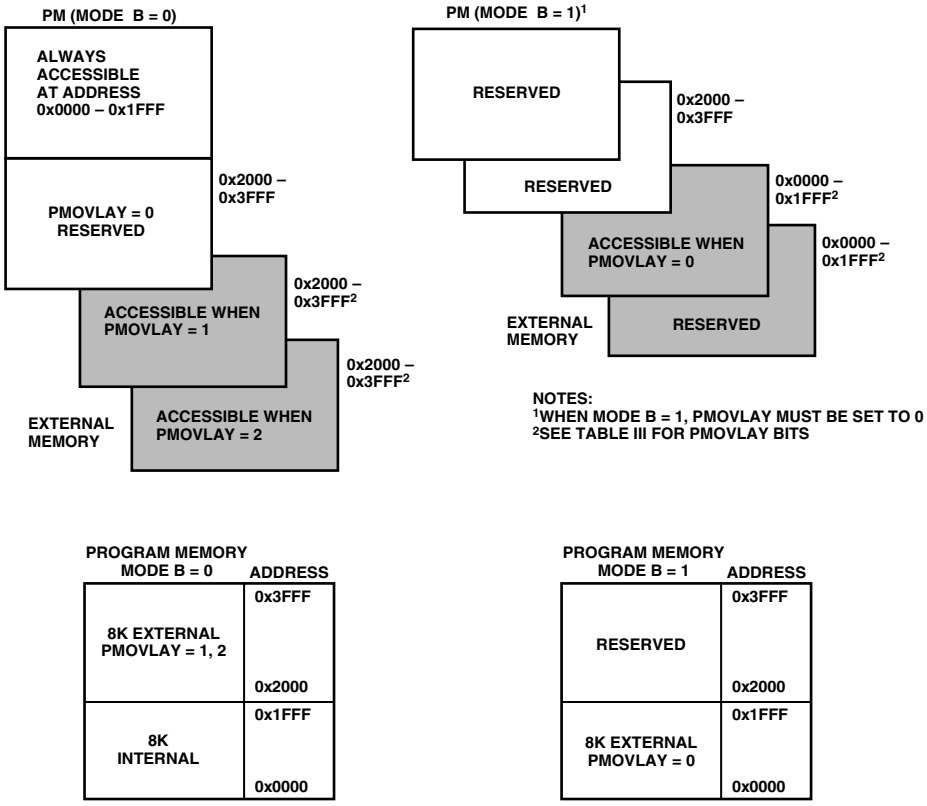



*Figure 4. Program Memory*

**Table III. PMOVLAY Bits**

| PMOVLAY | Memory | A13 | A12:0 |
|---|---|---|---|
| 0 | Reserved | Not Applicable | Not Applicable |
| 1 | External Overlay 1 | 0 | 13 LSBs of Address Between 0x2000 and 0x3FFF |
| 2 | External Overlay 2 | 1 | 13 LSBs of Address Between 0x2000 and 0x3FFF |

**Data Memory**

**Data Memory (Full Memory Mode)** is a 16-bit-wide space used for the storage of data variables and for memory-mapped control registers. The ADSP-2186M has 8K words on Data Memory RAM on-chip. Part of this space is used by 32 memory-mapped registers. Support also exists for up to two 8K external memory overlay spaces through the external data bus. All internal accesses complete in one cycle. Accesses to external memory are timed using the wait states specified by the DWAIT register and the wait state mode bit.

**Data Memory (Host Mode)** allows access to all internal memory. External overlay access is limited by a single external address line (A0).

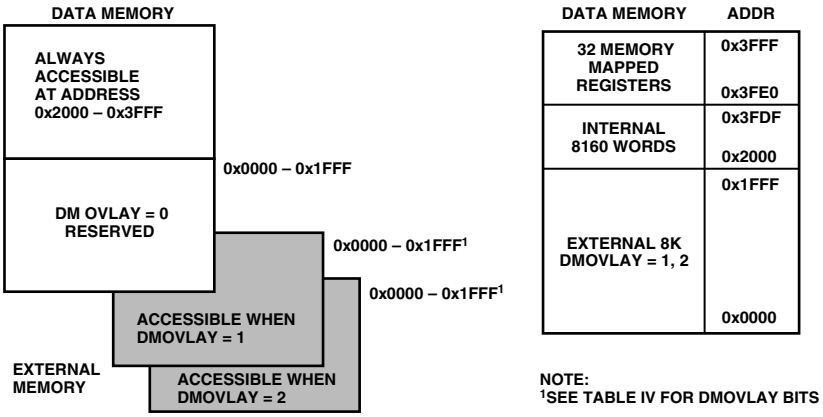



*Figure 5. Data Memory Map*

**Table IV. DMOVLAY Bits**

| DMOVLAY | Memory | A13 | A12:0 |
|---|---|---|---|
| 0 | Reserved | Not Applicable | Not Applicable |
| 1 | External Overlay 1 | 0 | 13 LSBs of Address Between 0x2000 and 0x3FFF |
| 2 | External Overlay 2 | 1 | 13 LSBs of Address Between 0x2000 and 0x3FFF |

**Memory Mapped Registers (New to the ADSP-2186M)**

The ADSP-2186M has three memory mapped registers that differ from other ADSP-21xx Family DSPs. The slight modifications to these registers (Wait State Control, Programmable Flag and Composite Select Control, and System Control) provide the ADSP-2186M's wait state and $\overline{\text{BMS}}$ control features. Default bit values at reset are shown; if no value is shown, the bit is undefined at reset. Reserved bits are shown on a grey field. These bits should always be written with zeros.

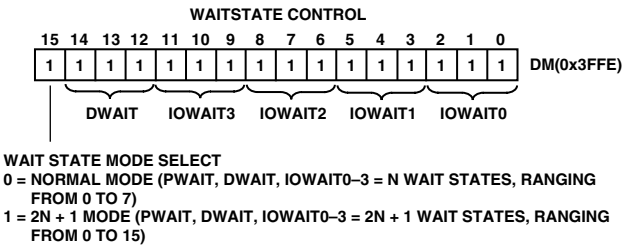



*Figure 6. Wait State Control Register*

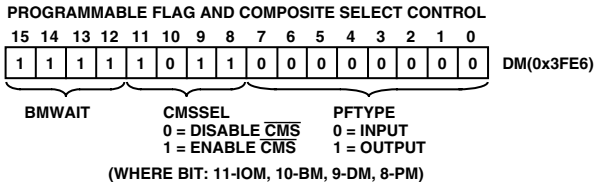



*Figure 7. Programmable Flag and Composite Control Register*

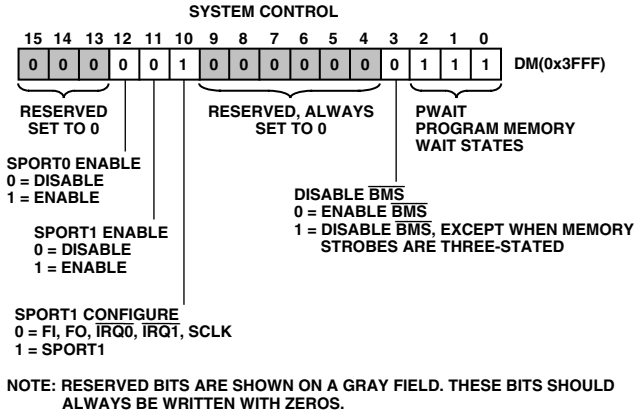



*Figure 8. System Control Register*

**I/O Space (Full Memory Mode)**

The ADSP-2186M supports an additional external memory space called I/O 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 2048 locations of 16-bit wide data. The lower eleven bits of the external address bus are used; the upper three bits are undefined. Two instructions were added to the core ADSP-2100 Family instruction set to read from and write to I/O memory space. The I/O space also has four dedicated three-bit wait state registers, IOWAIT0–3, which in combination with the wait state mode bit, specify up to 15 wait states to be automatically generated for each of four regions. The wait states act on address ranges as shown in Table V.

**Table V. Wait States**

| Address Range | Wait State Register |
|---|---|
| 0x000–0x1FF | IOWAIT0 and Wait State Mode Select Bit |
| 0x200–0x3FF | IOWAIT1 and Wait State Mode Select Bit |
| 0x400–0x5FF | IOWAIT2 and Wait State Mode Select Bit |
| 0x600–0x7FF | IOWAIT3 and Wait State Mode Select Bit |

**Composite Memory Select ($\overline{\text{CMS}}$)**
The ADSP-2186M has a programmable memory select signal that is useful for generating memory select signals for memories mapped to more than one space. The $\overline{\text{CMS}}$ signal is generated to have the same timing as each of the individual memory select signals ($\overline{\text{PMS}}$, $\overline{\text{DMS}}$, $\overline{\text{BMS}}$, $\overline{\text{IOMS}}$) but can combine their functionality.

Each bit in the CMSSEL register, when set, causes the $\overline{\text{CMS}}$ signal to be asserted when the selected memory select is asserted. For example, to use a 32K word memory to act as both program and data memory, set the $\overline{\text{PMS}}$ and $\overline{\text{DMS}}$ bits in the CMSSEL register and use the $\overline{\text{CMS}}$ pin to drive the chip select of the memory, and use either $\overline{\text{DMS}}$ or $\overline{\text{PMS}}$ as the additional address bit.

The $\overline{\text{CMS}}$ pin functions like the other memory select signals with the same timing and bus request logic. A 1 in the enable bit causes the assertion of the $\overline{\text{CMS}}$ signal at the same time as the selected memory select signal. All enable bits default to 1 at reset, except the $\overline{\text{BMS}}$ bit.

**Byte Memory Select ($\overline{\text{BMS}}$)**
The ADSP-2186M's $\overline{\text{BMS}}$ disable feature combined with the $\overline{\text{CMS}}$ pin allows use of multiple memories in the byte memory space. For example, an EPROM could be attached to the $\overline{\text{BMS}}$ select, and an SRAM could be connected to $\overline{\text{CMS}}$. Because at reset $\overline{\text{BMS}}$ is enabled, the EPROM would be used for booting. After booting, software could disable $\overline{\text{BMS}}$ and set the $\overline{\text{CMS}}$ signal to respond to $\overline{\text{BMS}}$, enabling the SRAM.

**Byte Memory**
The byte memory space is a bidirectional, 8-bit-wide, external memory space used to store programs and data. Byte memory is accessed using the BDMA feature. The byte memory space consists of 256 pages, each of which is 16K × 8.

The byte memory space on the ADSP-2186M supports read and write operations as well as four different data formats. The byte memory uses data bits 15:8 for data. The byte memory uses data bits 23:16 and address bits 13:0 to create a 22-bit address. This allows up to a 4 meg × 8 (32 megabit) ROM or RAM to be used without glue logic. All byte memory accesses are timed by the BMWAIT register and the wait state mode bit.

**Byte Memory DMA (BDMA, Full Memory Mode)**
The byte memory DMA controller allows loading and storing of program instructions and data using the byte memory space. The BDMA circuit is able to access the byte memory space while the processor is operating normally and steals only one DSP cycle per 8-, 16- or 24-bit word transferred.

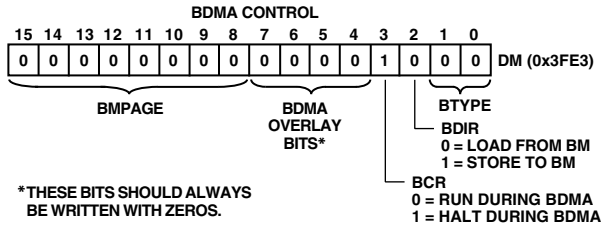



*Figure 9. BDMA Control Register*

The BDMA circuit supports four different data formats that are selected by the BTYPE register field. The appropriate number of 8-bit accesses are done from the byte memory space to build the word size selected. Table VI shows the data formats supported by the BDMA circuit.

**Table VI. Data Formats**

| BTYPE | Internal Memory Space | Word Size | Alignment |
|---|---|---|---|
| 00 | Program Memory | 24 | Full Word |
| 01 | Data Memory | 16 | Full Word |
| 10 | Data Memory | 8 | MSBs |
| 11 | Data Memory | 8 | LSBs |

Unused bits in the 8-bit data memory formats are filled with 0s. The BIAD register field is used to specify the starting address for the on-chip memory involved with the transfer. The 14-bit BEAD register specifies the starting address for the external byte memory space. The 8-bit BMPAGE register specifies the starting page for the external byte memory space. The BDIR register field selects the direction of the transfer. Finally, the 14-bit BWCOUNT register specifies the number of DSP words to transfer and initiates the BDMA circuit transfers.

BDMA accesses can cross page boundaries during sequential addressing. A BDMA interrupt is generated on the completion of the number of transfers specified by the BWCOUNT register.

The BWCOUNT register is updated after each transfer so it can be used to check the status of the transfers. When it reaches zero, the transfers have finished and a BDMA interrupt is generated. The BMPAGE and BEAD registers must not be accessed by the DSP during BDMA operations.

The source or destination of a BDMA transfer will always be on-chip program or data memory.

When the BWCOUNT register is written with a nonzero value the BDMA circuit starts executing byte memory accesses with wait states set by BMWAIT. These accesses continue until the count reaches zero. When enough accesses have occurred to create a destination word, it is transferred to or from on-chip memory. The transfer takes one DSP cycle. DSP accesses to external memory have priority over BDMA byte memory accesses.

The BDMA Context Reset bit (BCR) controls whether the processor is held off while the BDMA accesses are occurring. Setting the BCR bit to 0 allows the processor to continue operations. Setting the BCR bit to 1 causes the processor to stop execution while the BDMA accesses are occurring, to clear the context of the processor, and start execution at address 0 when the BDMA accesses have completed.

The BDMA overlay bits specify the OVLAY memory blocks to be accessed for internal memory. For ADSP-2186M, set to zero BDMA overlay bits in BDMA control register.

The BMWAIT field, which has four bits on ADSP-2186M, allows selection of up to 15 wait states for BDMA transfers.

**Internal Memory DMA Port (IDMA Port; Host Memory Mode)**

The IDMA Port provides an efficient means of communication between a host system and the ADSP-2186M. The port is used to access the on-chip program memory and data memory of the DSP with only one DSP cycle per word overhead. The IDMA port cannot, however, be used to write to the DSP's memory-mapped control registers. A typical IDMA transfer process is described as follows:

1. Host starts IDMA transfer.
2. Host checks $\overline{\text{IACK}}$ control line to see if the DSP is busy.
3. Host uses $\overline{\text{IS}}$ and IAL control lines to latch either the DMA starting address (IDMAA) or the PM/DM OVLAY selection into the DSP's IDMA control registers. If Bit 15 = 1, the value of bits 7:0 represent the IDMA overlay: bits 14:8 must be set to 0. If Bit 15 = 0, the value of Bits 13:0 represent the starting address of internal memory to be accessed and Bit 14 reflects PM or DM for access. For ADSP-2186M, IDDMOVLAY and IDPMOVLAY bits in IDMA overlay register should be set to zero.
4. Host uses $\overline{\text{IS}}$ and $\overline{\text{IRD}}$ (or $\overline{\text{IWR}}$) to read (or write) DSP internal memory (PM or DM).
5. Host checks $\overline{\text{IACK}}$ line to see if the DSP has completed the previous IDMA operation.
6. Host ends IDMA transfer.

The IDMA port has a 16-bit multiplexed address and data bus and supports 24-bit program memory. The IDMA port is completely asynchronous and can be written while the ADSP-2186M is operating at full speed.

The DSP memory address is latched and then automatically incremented after each IDMA transaction. An external device can therefore access a block of sequentially addressed memory by specifying only the starting address of the block. This increases throughput as the address does not have to be sent for each memory access.

IDMA Port access occurs in two phases. The first is the IDMA Address Latch cycle. When the acknowledge is asserted, a 14-bit address and 1-bit destination type can be driven onto the bus by an external device. The address specifies an on-chip memory location, the destination type specifies whether it is a DM or PM access. The falling edge of the IDMA address latch signal (IAL) or the missing edge of the IDMA select signal ($\overline{\text{IS}}$) latches this value into the IDMAA register.

Once the address is stored, data can be read from, or written to, the ADSP-2186M's on-chip memory. Asserting the select line ($\overline{\text{IS}}$) and the appropriate read or write line ($\overline{\text{IRD}}$ and $\overline{\text{IWR}}$ respectively) signals the ADSP-2186M that a particular transaction is required. In either case, there is a one-processor-cycle delay for synchronization. The memory access consumes one additional processor cycle.

Once an access has occurred, the latched address is automatically incremented, and another access can occur.

Through the IDMAA register, the DSP can also specify the starting address and data format for DMA operation. Asserting the IDMA port select ($\overline{\text{IS}}$) and address latch enable (IAL) directs the ADSP-2186M to write the address onto the IAD0–14 bus into the IDMA Control Register. If Bit 15 is set to 0, IDMA latches the address. If Bit 15 is set to 1, IDMA latches into the OVLAY register. This register, shown below, is memory mapped at address DM (0x3FE0). Note that the latched address (IDMAA) cannot be read back by the host. When Bit 14 in 0x3FE7 is set to 1, timing in Figure 31 applies for short reads. When Bit 14 in 0x3FE7 is set to zero, short reads use the timing shown in Figure 32. For ADSP-2186M, IDDMOVLAY and IDPMOVLAY bits in IDMA overlay register should be set to zero.

Refer to the following figures for more information on IDMA and DMA memory maps.

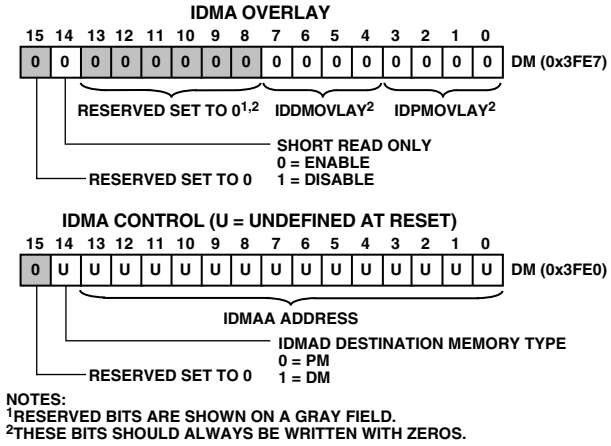
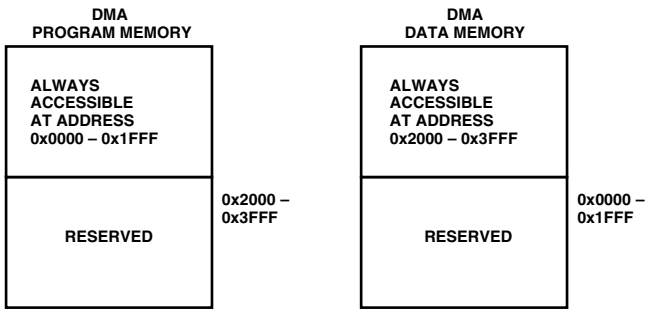



*Figure 10. IDMA Control/OVLAY Registers*




NOTE: IDMA AND BDMA HAVE SEPARATE DMA CONTROL REGISTERS.

*Figure 11. Direct Memory Access—PM and DM Memory Maps*

**Bootstrap Loading (Booting)**

The ADSP-2186M has two mechanisms to allow automatic loading of the internal program memory after reset. The method for booting is controlled by the Mode A, B, and C configuration bits.

When the MODE pins specify BDMA booting, the ADSP-2186M initiates a BDMA boot sequence when reset is released.

The BDMA interface is set up during reset to the following defaults when BDMA booting is specified: the BDIR, BMPAGE, BIAD, and BEAD registers are set to 0, the BTYPE register is set to 0 to specify program memory 24-bit words, and the BWCOUNT register is set to 32. This causes 32 words of on-chip program memory to be loaded from byte memory.

**ABSOLUTE MAXIMUM RATINGS**[1]

| Parameter | Value Min | Value Max |
|---|---|---|
| Internal Supply Voltage ($V_{DDINT}$) | –0.3 V | +3.0 V |
| External Supply Voltage ($V_{DDEXT}$) | –0.3 V | +4.0 V |
| Input Voltage[2] | –0.5 V | +4.0 V |
| Output Voltage Swing[3] | –0.5 V | $V_{DDEXT}$ + 0.5 V |
| Operating Temperature Range | –40°C | +85°C |
| Storage Temperature Range | –65°C | +150°C |
| Lead Temperature (5 sec) LQFP | | 280°C |

NOTES
[1]Stresses greater than those listed 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.
[2]Applies to Bidirectional pins (D0–D23, RFS0, RFS1, SCLK0, SCLK1, TFS0, TFS1, A1–A13, PF0–PF7) and Input only pins (CLKIN, $\overline{\text{RESET}}$, $\overline{\text{BR}}$, DR0, DR1, $\overline{\text{PWD}}$).
[3]Applies to Output pins ($\overline{\text{BG}}$, $\overline{\text{PMS}}$, $\overline{\text{DMS}}$, $\overline{\text{BMS}}$, $\overline{\text{IOMS}}$, $\overline{\text{CMS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, PWDACK, A0, DT0, DT1, CLKOUT, FL2–0, $\overline{\text{BGH}}$).

**ESD SENSITIVITY**

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-2186M 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.


# TIMING SPECIFICATIONS

**GENERAL NOTES**

Use the exact timing information given. Do not attempt to derive parameters from the addition or subtraction of others. While addition or subtraction would yield meaningful results for an individual device, the values given in this data sheet reflect statistical variations and worst cases. Consequently, you cannot meaningfully add up parameters to derive longer times.

**TIMING NOTES**

Switching characteristics specify how the processor changes its signals. You have no control over this timing—circuitry external to the processor must be designed for compatibility with these signal characteristics. Switching characteristics tell you what the processor will do in a given circumstance. You can also use switching characteristics 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.

**MEMORY TIMING SPECIFICATIONS**

The table below shows common memory device specifications and the corresponding ADSP-2186M timing parameters, for your convenience.

| Memory Device Specification | Parameter | Timing Parameter Definition[1] |
|---|---|---|
| Address Setup to Write Start | $t_{ASW}$ | A0–A13, $\overline{\text{xMS}}$ Setup before $\overline{\text{WR}}$ Low |
| Address Setup to Write End | $t_{AW}$ | A0–A13, $\overline{\text{xMS}}$ Setup before $\overline{\text{WR}}$ Deasserted |
| Address Hold Time | $t_{WRA}$ | A0–A13, $\overline{\text{xMS}}$ Hold before $\overline{\text{WR}}$ Low |
| Data Setup Time | $t_{DW}$ | Data Setup before $\overline{\text{WR}}$ High |
| Data Hold Time | $t_{DH}$ | Data Hold after $\overline{\text{WR}}$ High |
| OE to Data Valid | $t_{RDD}$ | $\overline{\text{RD}}$ Low to Data Valid |
| Address Access Time | $t_{AA}$ | A0–A13, $\overline{\text{xMS}}$ to Data Valid |

NOTE
[1]$\overline{\text{xMS}}$ = $\overline{\text{PMS}}$, $\overline{\text{DMS}}$, $\overline{\text{BMS}}$, $\overline{\text{CMS}}$ or $\overline{\text{IOMS}}$.

### FREQUENCY DEPENDENCY FOR TIMING SPECIFICATIONS

$t_{CK}$ is defined as 0.5 $t_{CKI}$. The ADSP-2186M uses an input clock with a frequency equal to half the instruction rate. For example, a 37.50 MHz input clock (which is equivalent to 26.6 ns) yields a 13.3 ns processor cycle (equivalent to 75 MHz). $t_{CK}$ values within the range of 0.5 $t_{CKI}$ period should be substituted for all relevant timing parameters to obtain the specification value.

Example: $t_{CKH}$ = 0.5 $t_{CK}$ – 2 ns = 0.5 (15 ns) – 2 ns = 5.5 ns

### ENVIRONMENTAL CONDITIONS[1]

| Rating Description | Symbol | LQFP | Mini-BGA |
|---|---|---|---|
| Thermal Resistance (Case-to-Ambient) | $\theta_{CA}$ | 48°C/W | 63.3°C/W |
| Thermal Resistance (Junction-to-Ambient) | $\theta_{JA}$ | 50°C/W | 70.7°C/W |
| Thermal Resistance (Junction-to-Case) | $\theta_{JC}$ | 2°C/W | 7.4°C/W |

NOTE
[1]Where the Ambient Temperature Rating ($T_{AMB}$) is:
$T_{AMB} = T_{CASE} - (PD \times \theta_{CA})$
$T_{CASE}$ = Case Temperature in °C
PD = Power Dissipation in W

### POWER DISSIPATION

To determine total power dissipation in a specific application, the following equation should be applied for each output:

$$C \times V_{DD}2 \times f$$

$C$ = load capacitance, $f$ = output switching frequency.

Example:

In an application where external data memory is used and no other outputs are active, power dissipation is calculated as follows:

Assumptions:

- External data memory is accessed every cycle with 50% of the address pins switching.
- External data memory writes occur every other cycle with 50% of the data pins switching.
- Each address and data pin has a 10 pF total load at the pin.
- The application operates at $V_{DDEXT}$ = 3.3 V and $t_{CK}$ = 30 ns.

*Total Power Dissipation* = $P_{INT} + (C \times V_{DDEXT}^2 \times f)$

$P_{INT}$ = internal power dissipation from Power vs. Frequency graph (Figure 15).

$(C \times V_{DDEXT}^2 \times f)$ is calculated for each output:

| Parameters | # of Pins | × C pF | × $V_{DDEXT}^2$ V | × f MHz | PD mW |
|---|---|---|---|---|---|
| Address | 7 | 10 | $3.3^2$ | 16.67 | 12.7 |
| Data Output, $\overline{WR}$ | 9 | 10 | $3.3^2$ | 16.67 | 16.3 |
| $\overline{RD}$ | 1 | 10 | $3.3^2$ | 16.67 | 1.8 |
| CLKOUT, $\overline{DMS}$ | 2 | 10 | $3.3^2$ | 33.3 | 7.2 |
| | | | | | 38.0 |

Total power dissipation for this example is $P_{INT}$ + 38.0 mW.

### Output Drive Currents

Figure 14 shows typical I-V characteristics for the output drivers on the ADSP-2186M. The curves represent the current drive capability of the output drivers as a function of output voltage.

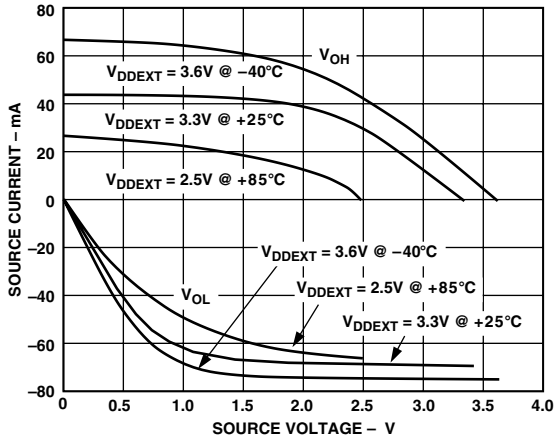



*Figure 14. Typical Output Driver Characteristics*

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| **Clock Signals and Reset** | | | | |
| *Timing Requirements:* | | | | |
| $t_{CKI}$ | CLKIN Period | 26.6 | 80 | ns |
| $t_{CKIL}$ | CLKIN Width Low | 8 | | ns |
| $t_{CKIH}$ | CLKIN Width High | 8 | | ns |
| *Switching Characteristics:* | | | | |
| $t_{CKL}$ | CLKOUT Width Low | $0.5t_{CK} - 2$ | | ns |
| $t_{CKH}$ | CLKOUT Width High | $0.5t_{CK} - 2$ | | ns |
| $t_{CKOH}$ | CLKIN High to CLKOUT High | 0 | 13 | ns |
| *Control Signals Timing Requirements*: | | | | |
| $t_{RSP}$ | $\overline{\text{RESET}}$ Width Low | $5t_{CK}$[1] | | ns |
| $t_{MS}$ | Mode Setup before $\overline{\text{RESET}}$ High | 2 | | ns |
| $t_{MH}$ | Mode Hold after $\overline{\text{RESET}}$ High | 5 | | ns |

NOTE
[1]Applies after power-up sequence is complete. Internal phase lock loop requires no more than 2000 CLKIN cycles assuming stable CLKIN (not including crystal oscillator start-up time).

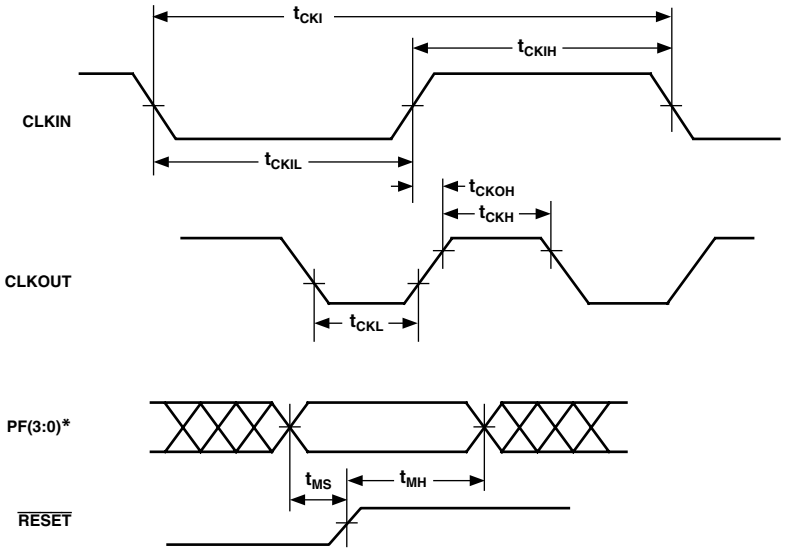



*Figure 21. Clock Signals*

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| **Interrupts and Flags** | | | | |
| *Timing Requirements:* | | | | |
| $t_{IFS}$ | $\overline{IRQx}$, FI, or PFx Setup before CLKOUT Low[1, 2, 3, 4] | $0.25t_{CK}$ + 10 | | ns |
| $t_{IFH}$ | $\overline{IRQx}$, FI, or PFx Hold after CLKOUT High[1, 2, 3, 4] | $0.25t_{CK}$ | | ns |
| *Switching Characteristics:* | | | | |
| $t_{FOH}$ | Flag Output Hold after CLKOUT Low[5] | $0.5t_{CK}$ – 5 | | ns |
| $t_{FOD}$ | Flag Output Delay from CLKOUT Low[5] | | $0.5t_{CK}$ + 4 | ns |

NOTES
[1]If $\overline{IRQx}$ and FI inputs meet $t_{IFS}$ and $t_{IFH}$ setup/hold requirements, they will be recognized during the current clock cycle; otherwise the signals will be recognized on the following cycle. (Refer to "Interrupt Controller Operation" in the Program Control chapter of the *ADSP-2100 Family User's Manual* for further information on interrupt servicing.)
[2]Edge-sensitive interrupts require pulsewidths greater than 10 ns; level-sensitive interrupts must be held low until serviced.
[3]$\overline{IRQx}$ = $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{IRQL0}$, $\overline{IRQL1}$, $\overline{IRQLE}$.
[4]PFx = PF0, PF1, PF2, PF3, PF4, PF5, PF6, PF7.
[5]Flag Outputs = PFx, FL0, FL1, FL2, FO.

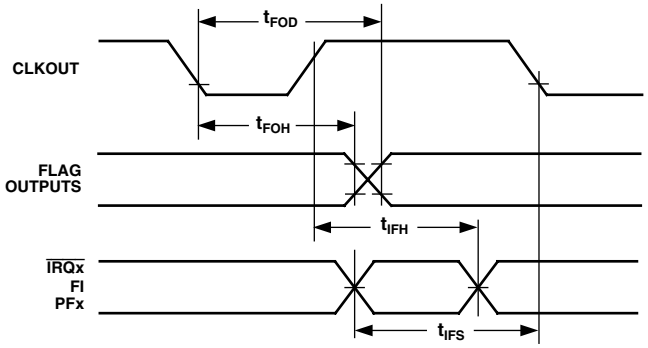



*Figure 22. Interrupts and Flags*

**Serial Ports**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| **Serial Ports** | | | | |
| *Timing Requirements:* | | | | |
| $t_{SCK}$ | SCLK Period | 26.6 | | ns |
| $t_{SCS}$ | DR/TFS/RFS Setup before SCLK Low | 4 | | ns |
| $t_{SCH}$ | DR/TFS/RFS Hold after SCLK Low | 7 | | ns |
| $t_{SCP}$ | SCLKIN Width | 12 | | ns |
| *Switching Characteristics*: | | | | |
| $t_{CC}$ | CLKOUT High to SCLKOUT | $0.25t_{CK}$ | $0.25t_{CK}$ + 6 | ns |
| $t_{SCDE}$ | SCLK High to DT Enable | 0 | | ns |
| $t_{SCDV}$ | SCLK High to DT Valid | | 12 | ns |
| $t_{RH}$ | $TFS/RFS_{OUT}$ Hold after SCLK High | 0 | | ns |
| $t_{RD}$ | $TFS/RFS_{OUT}$ Delay from SCLK High | | 12 | ns |
| $t_{SCDH}$ | DT Hold after SCLK High | 0 | | ns |
| $t_{TDE}$ | TFS (Alt) to DT Enable | 0 | | ns |
| $t_{TDV}$ | TFS (Alt) to DT Valid | | 12 | ns |
| $t_{SCDD}$ | SCLK High to DT Disable | | 12 | ns |
| $t_{RDV}$ | RFS (Multichannel, Frame Delay Zero) to DT Valid | | 12 | ns |

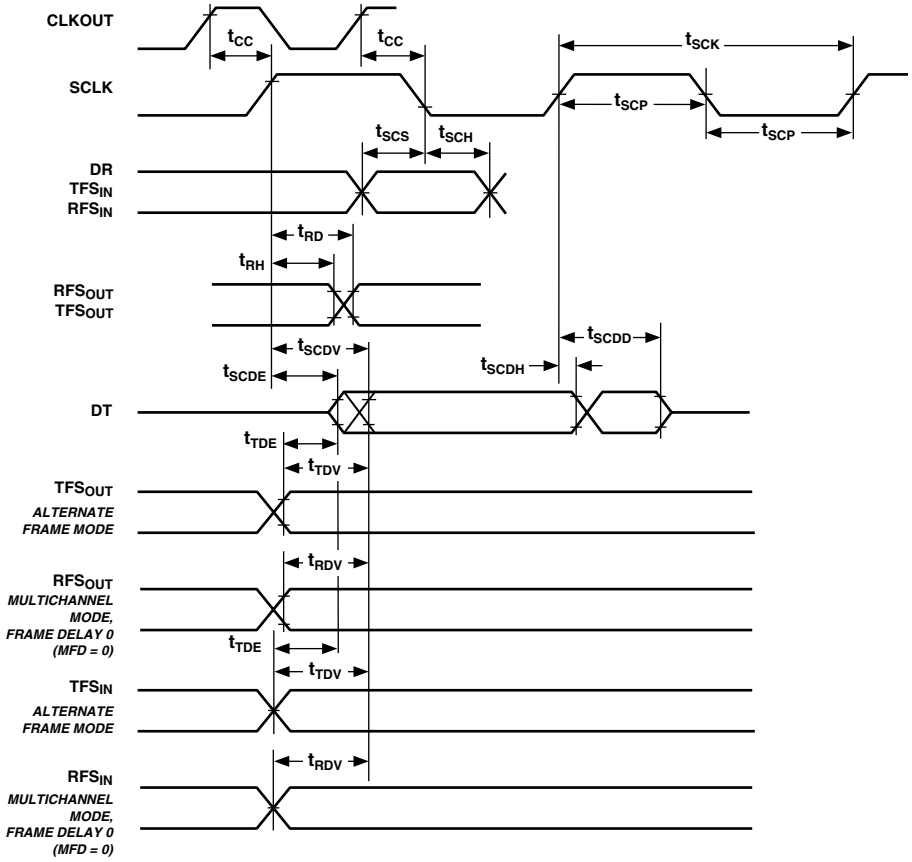



*Figure 26. Serial Ports*

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| **IDMA Address Latch** | | | | |
| *Timing Requirements:* | | | | |
| $t_{IALP}$ | Duration of Address Latch[1, 2] | 10 | | ns |
| $t_{IASU}$ | IAD15–0 Address Setup before Address Latch End[2] | 5 | | ns |
| $t_{IAH}$ | IAD15–0 Address Hold after Address Latch End[2] | 3 | | ns |
| $t_{IKA}$ | $\overline{IACK}$ Low before Start of Address Latch[2, 3] | 0 | | ns |
| $t_{IALS}$ | Start of Write or Read after Address Latch End[2, 3] | 3 | | ns |
| $t_{IALD}$ | Address Latch Start after Address Latch End[1, 2] | 2 | | ns |

NOTES
[1]Start of Address Latch = $\overline{IS}$ Low and IAL High.
[2]End of Address Latch = $\overline{IS}$ High or IAL Low.
[3]Start of Write or Read = $\overline{IS}$ Low and $\overline{IWR}$ Low or $\overline{IRD}$ Low.

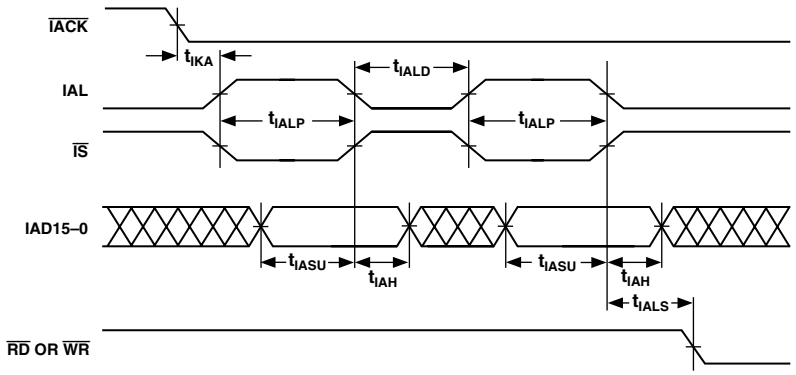



*Figure 27. IDMA Address Latch*

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| **IDMA Write, Short Write Cycle** | | | | |
| *Timing Requirements:* | | | | |
| $t_{IKW}$ | $\overline{IACK}$ Low before Start of Write[1] | 0 | | ns |
| $t_{IWP}$ | Duration of Write[1, 2] | 10 | | ns |
| $t_{IDSU}$ | IAD15–0 Data Setup before End of Write[2, 3, 4] | 3 | | ns |
| $t_{IDH}$ | IAD15–0 Data Hold after End of Write[2, 3, 4] | 2 | | ns |
| *Switching Characteristic*: | | | | |
| $t_{IKHW}$ | Start of Write to $\overline{IACK}$ High | | 10 | ns |

NOTES
[1]Start of Write = $\overline{IS}$ Low and $\overline{IWR}$ Low.
[2]End of Write = $\overline{IS}$ High or $\overline{IWR}$ High.
[3]If Write Pulse ends before $\overline{IACK}$ Low, use specifications $t_{IDSU}$, $t_{IDH}$.
[4]If Write Pulse ends after $\overline{IACK}$ Low, use specifications $t_{IKSU}$, $t_{IKH}$.

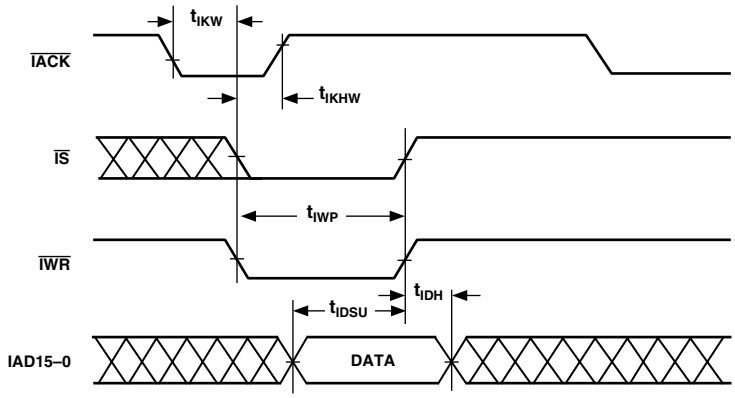



*Figure 28. IDMA Write, Short Write Cycle*

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| **IDMA Read, Long Read Cycle** | | | | |
| *Timing Requirements:* | | | | |
| $t_{IKR}$ | $\overline{IACK}$ Low before Start of Read[1] | 0 | | ns |
| $t_{IRK}$ | End of Read after $\overline{IACK}$ Low[2] | 2 | | ns |
| *Switching Characteristics:* | | | | |
| $t_{IKHR}$ | $\overline{IACK}$ High after Start of Read[1] | | 10 | ns |
| $t_{IKDS}$ | IAD15–0 Data Setup before $\overline{IACK}$ Low | $0.5t_{CK} - 2$ | | ns |
| $t_{IKDH}$ | IAD15–0 Data Hold after End of Read[2] | 0 | | ns |
| $t_{IKDD}$ | IAD15–0 Data Disabled after End of Read[2] | | 10 | ns |
| $t_{IRDE}$ | IAD15–0 Previous Data Enabled after Start of Read | 0 | | ns |
| $t_{IRDV}$ | IAD15–0 Previous Data Valid after Start of Read | | 11 | ns |
| $t_{IRDH1}$ | IAD15–0 Previous Data Hold after Start of Read (DM/PM1)[3] | $2t_{CK} - 5$ | | ns |
| $t_{IRDH2}$ | IAD15–0 Previous Data Hold after Start of Read (PM2)[4] | $t_{CK} - 5$ | | ns |

NOTES
[1]Start of Read = $\overline{IS}$ Low and $\overline{IRD}$ Low.
[2]End of Read = $\overline{IS}$ High or $\overline{IRD}$ High.
[3]DM read or first half of PM read.
[4]Second half of PM read.

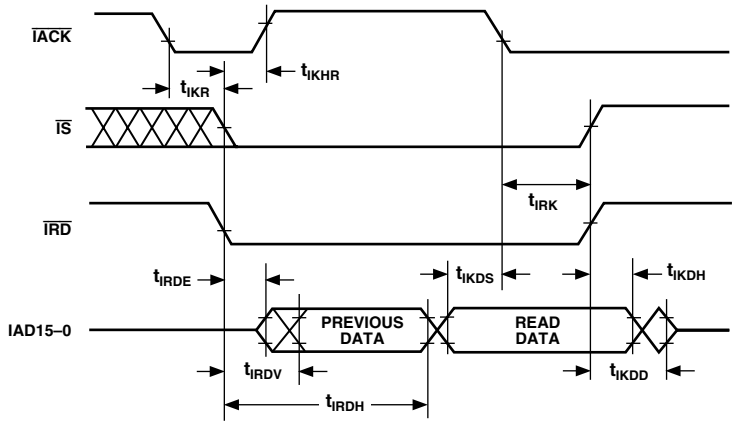
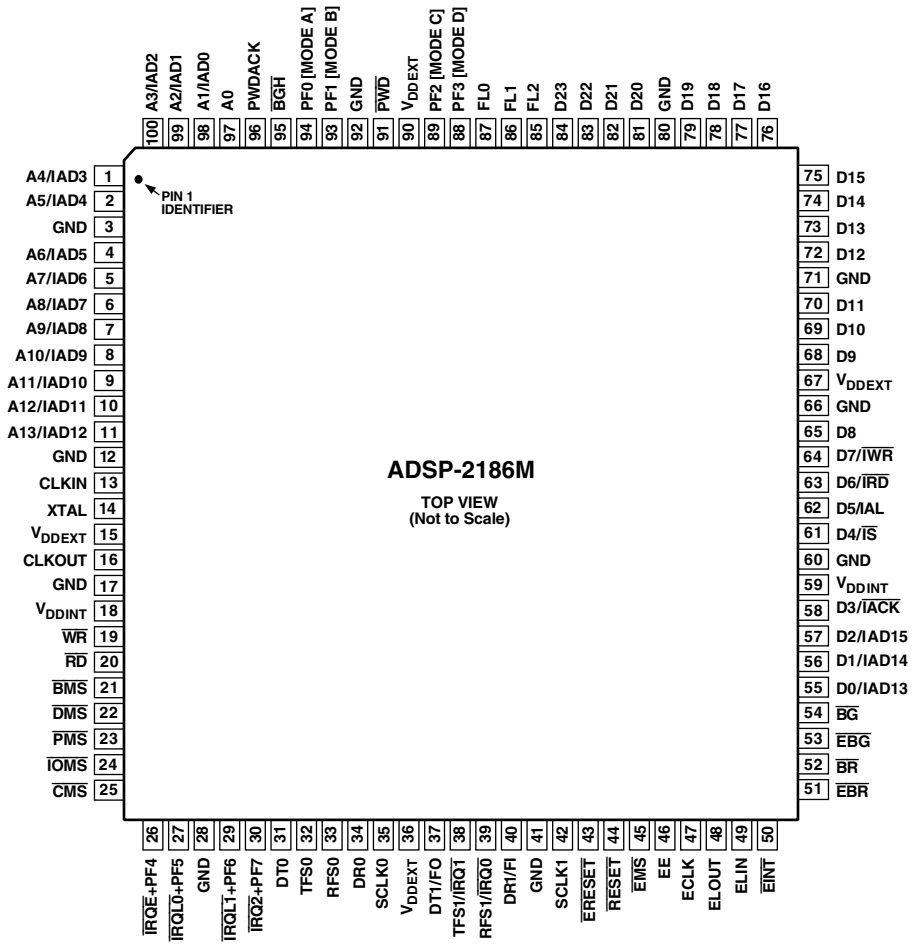



*Figure 30. IDMA Read, Long Read Cycle*

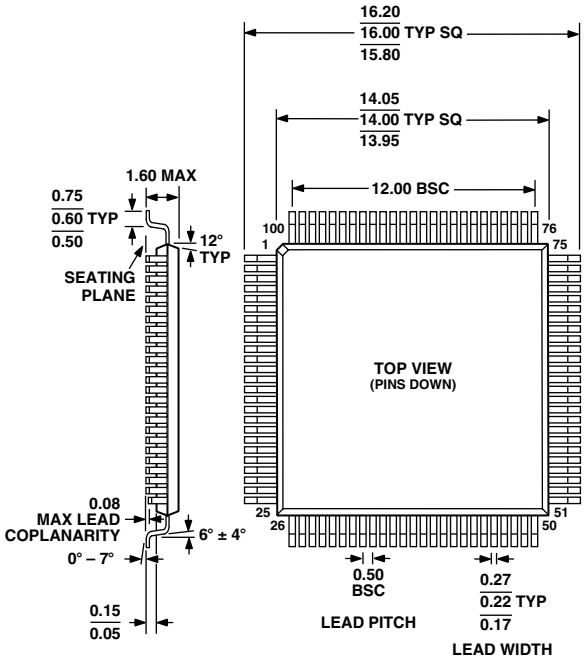
**100-LEAD LQFP PIN CONFIGURATION**



## OUTLINE DIMENSIONS

Dimensions shown in millimeters.

### 100-Lead Metric Thin Plastic Quad Flatpack (LQFP) (ST-100)




NOTE:
THE ACTUAL POSITION OF EACH LEAD IS WITHIN 0.08 FROM ITS IDEAL POSITION WHEN MEASURED IN THE LATERAL DIRECTION.