SGUS012A – D3592, SEPTEMBER 1990 – REVISED MAY 1991

- **Military Temperature Range (–55°C to 125°C)**
- **Class B, High-Reliability Processing**
- **High-Performance Floating-Point RISC Processor Optimized for Graphics**
- **Two Operating Modes**
  - **Floating-Point Coprocessor for SMJ34020 Graphics System Processor**
  - **Independent Floating-Point Processor**
- **Direct Connection to SMJ34020 Coprocessor Interface**
  - **Direct Extension to the SMJ34020 Instruction Set**
  - **Multiple SMJ34082A Capability**
- **Fast Pipelined Instruction Cycle Time**
  - **SMJ34082A-30 . . . 66-ns Coprocessor Mode . . . 65-ns Host-Independent Mode**
  - **SMJ34082A-28 . . . 70-ns Coprocessor Mode . . . 70-ns Host-Independent Mode**
- **Sustained Data Transfer Rates of 120 Mbytes/s (SMJ34082A-30)**
- **Sequencer Executes Internal or User-Programmed Instructions**
- **22 64-Bit Data Registers**
- **Comprehensive Floating-Point and Integer Instruction Set**
- **Internal Programs for Vector, Matrix, and 3-D Graphics Operations**
- **Full IEEE Standard 754-1985 Compatibility**
  - **Addition, Subtration, Multiplication, and Comparison**
  - **Division and Square Root**
- **Selectable Data Formats**
  - **32-Bit Integer**
  - **32-Bit Single-Precision Floating-Point**
  - **64-Bit Double-Precision Floating-Point**
- **External Memory Addressing Capability**
  - **Program Storage (up to 64K Words)**
  - **Data Storage (up to 64K Words)**
- **0.8-µm EPIC™ CMOS Technology**
  - **High-Performance**
  - **Low Power (< 2 W)**

description

The SMJ34082A is a high-speed graphics floating-point processor implemented in Texas Instruments advanced 0.8-µm CMOS technology. The SMJ34082A combines a 16-bit sequencer and a 3-operand (source A, source B, and destination) 64-bit Floating-Point Unit (FPU) with 22 64-bit data registers on a single chip. The data registers are organized into two files of ten registers each, with two registers for internal feedback. In addition, it provides an instruction register to control FPU execution, a status register to retain the most recent FPU status outputs, eight control registers, and a two-deep stack (see functional block diagram).

The SMJ34082A is fully compatible with IEEE Standard 754-1985 for binary floating-point addition, subtraction, multiplication, division, square root, and comparison. Floating-point operands can be either in single- or double-precision IEEE format.

In addition to floating-point operations, the SMJ34082A performs 32-bit integer arithmetic, logical comparisons, and shifts. Integer operations may be performed on 32-bit 2s complement or unsigned operands. Integer results are 32-bits long (even for 32 x 32 integer multiplication). Absolute value conversions, floating-point to integer conversions, and integer to floating-point conversions are available.

The ALU and the multiplier are closely coupled and can be operated in parallel to perform sums of products or products of sums. During multiply/accumulate operations, both the ALU and the multiplier are active and the registers in the FPU core can be used to feedback products and accumulate sums without tying up locations in register files A and B.

When used with the SMJ34020, the SMJ34082A operates in the coprocessor mode. The SMJ34020 can control multiple SMJ34082A coprocessors. When used as a stand-alone or with processors other than the SMJ34020, the SMJ34082A operates in the host-independent mode. The SMJ34082A is fully programmable by the user

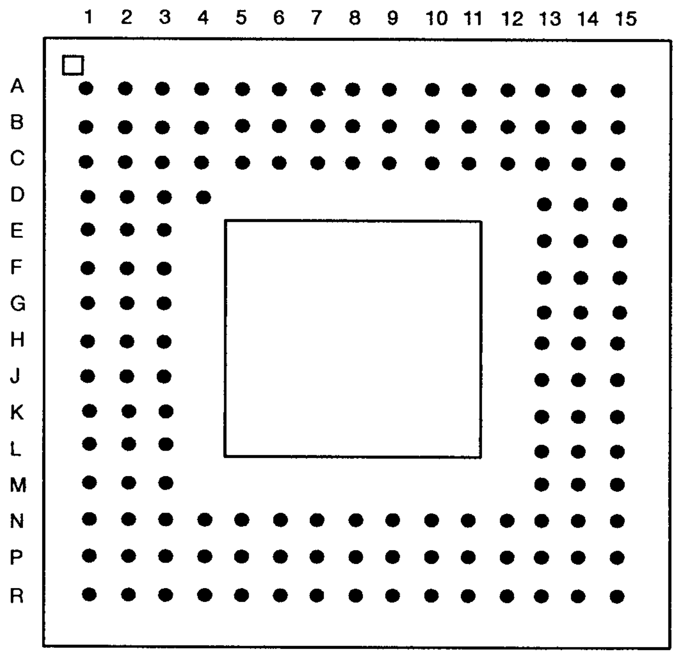
EPIC is a trademark of Texas Instruments Incorporated.

and can interface to other processors or floating-point subsystems through its two 32-bit bidirectional buses. In the coprocessor mode, the TMS340 family tools may be used to develop code for the SMJ34082A. The TMS34082A Software Tool Kit is used to develop code for host-independent mode applications or for external routines in the coprocessor mode.

pin descriptions

Pin descriptions and grid assignments for the SMJ34082A are given on the following pages. The pin at location D4 has been added for indexing purposes.

145-PIN GB PACKAGE
(TOP VIEW)

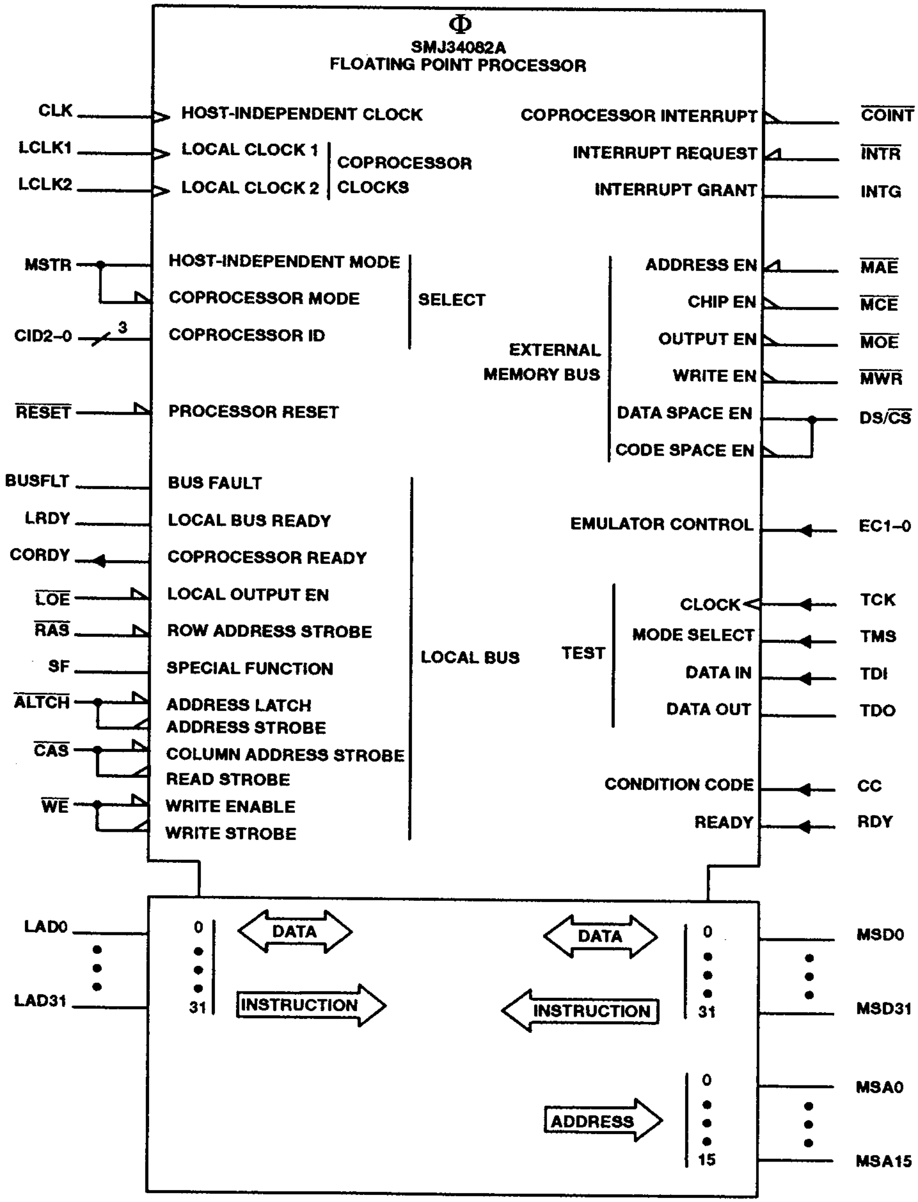


SGUS012A – D3592, SEPTEMBER 1990 – REVISED MAY 1991

Pin Grid Assignments

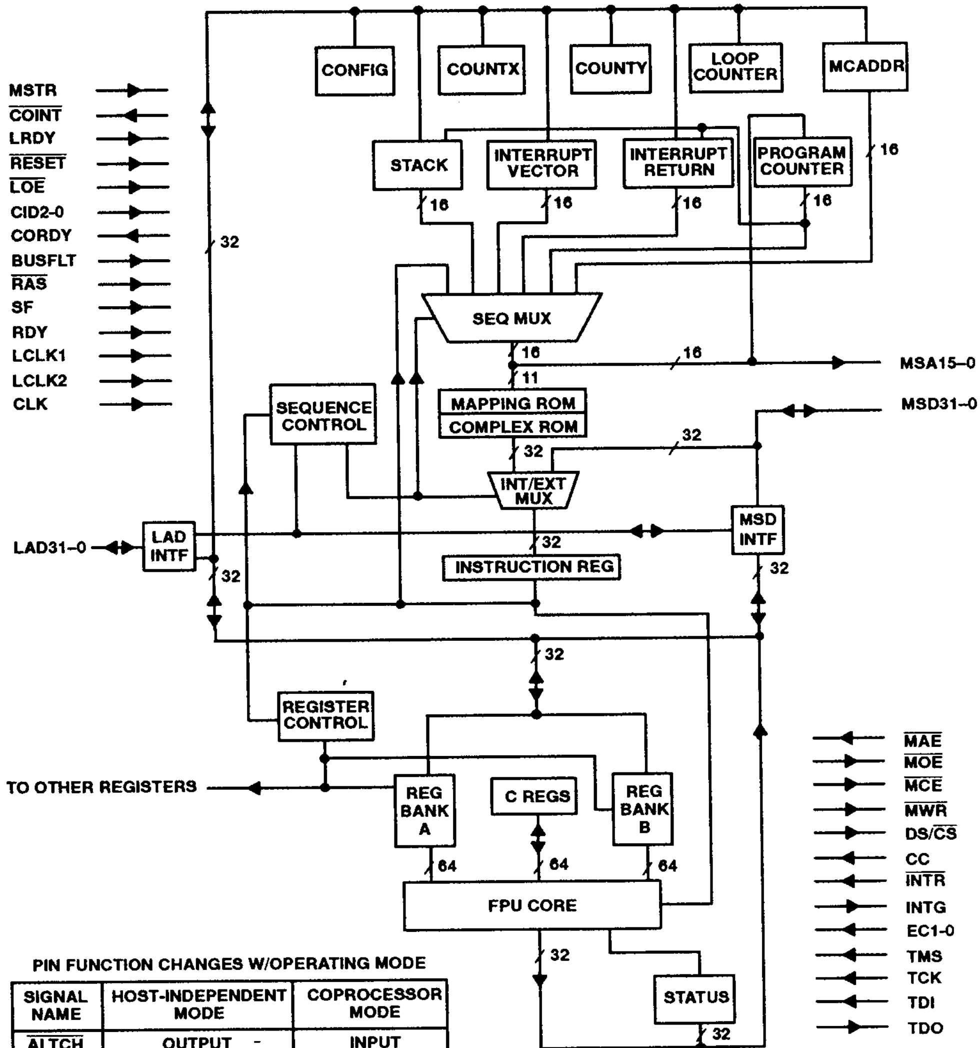
PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME
A1	NC	B15	LAD27	F1	MSD10	K15	RDY	P2	NC
A2	LAD1	C1	MSD4	F2	MSD9	L1	MSD18	P3	MSD29
A3	LAD3	C2	MSD3	F3	V_{CC}	L2	MSD21	P4	MSD31
A4	LAD5	C3	MSD0	F13	CORDY	L3	MSD23	P5	MSA1
A5	LAD8	C4	V_{SS}	F14	$\overline{ALTCH}$	L13	V_{SS}	P6	MSA3
A6	LAD9	C5	V_{CC}	F15	$\overline{CAS}$	L14	CID0	P7	MSA6
A7	LAD11	C6	LAD6	G1	MSD13	L15	CID2	P8	MSA8
A8	LAD12	C7	V_{SS}	G2	MSD12	M1	MSD20	P9	MSA10
A9	LAD13	C8	V_{CC}	G3	MSD11	M2	MSD24	P10	MSA13
A10	LAD15	C9	V_{SS}	G13	$\overline{WE}$	M3	V_{SS}	P11	$\overline{MWR}$
A11	LAD17	C10	V_{CC}	G14	EC1	M13	V_{CC}	P12	$\overline{MOE}$
A12	LAD19	C11	LAD21	G15	EC0	M14	LCLK1	P13	INTG
A13	LAD22	C12	V_{SS}	H1	MSD14	M15	LCLK2	P14	BUSFLT
A14	LAD24	C13	LAD25	H2	TDO	N1	MSD22	P15	$\overline{RAS}$
A15	NC	C14	LAD26	H3	V_{SS}	N2	MSD26	R1	NC
B1	MSD1	C15	LAD29	H13	V_{SS}	N3	V_{CC}	R2	MSD27
B2	NC	D1	MSD6	H14	$\overline{LOE}$	N4	MSD28	R3	MSD30
B3	LAD0	D2	MSD5	H15	TDI	N5	V_{SS}	R4	MSA0
B4	LAD2	D3	MSD2	J1	MSD15	N6	V_{CC}	R5	MSA2
B5	LAD4	D4	NC	J2	MSD16	N7	MSA5	R6	MSA4
B6	LAD7	D13	V_{CC}	J3	V_{CC}	N8	V_{SS}	R7	MSA7
B7	LAD10	D14	LAD28	J13	CC	N9	V_{CC}	R8	TCK
B8	TMS	D15	LAD31	J14	MSTR	N10	MSA14	R9	MSA9
B9	LAD14	E1	MSD8	J15	CLK	N11	V_{SS}	R10	MSA11
B10	LAD16	E2	MSD7	K1	MSD17	N12	$\overline{MAE}$	R11	MSA12
B11	LAD18	E3	V_{SS}	K2	MSD19	N13	LRDY	R12	MSA15
B12	LAD20	E13	V_{SS}	K3	V_{SS}	N14	SF	R13	DS/$\overline{CS}$
B13	LAD23	E14	LAD30	K13	CID1	N15	$\overline{RESET}$	R14	$\overline{MCE}$
B14	NC	E15	$\overline{COINT}$	K14	$\overline{INTR}$	P1	MSD25	R15	NC

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984.

functional block diagram



PIN FUNCTION CHANGES W/OPERATING MODE

SIGNAL NAME	HOST-INDEPENDENT MODE	COPROCESSOR MODE
ALTCH	OUTPUT	INPUT
WE	OUTPUT	INPUT
CAS	OUTPUT	INPUT

Terminal Functions

PIN NAME	PIN NO.	I/O†	DESCRIPTION
$\overline{ALTCH}$	F14	I [O]	Address Latch, active low. In the coprocessor mode, falling edge of $\overline{ALTCH}$ latches instruction and status present on the LAD bidirectional bus (LAD31-0). In the host-independent mode, $\overline{ALTCH}$ is address output strobe for memory accesses on LAD31-0.
BUSFLT	P14	I	Bus Fault. In the coprocessor mode, BUSFLT high indicates a data fault on the LAD bus (LAD31-0) during current bus cycle, which in turn causes SMJ34082A not to capture current data on LAD bus. Tied low if not used or in the host-independent mode.
$\overline{CAS}$	F15	I [O]	Column Address Strobe, active low. In the coprocessor mode, causes SMJ34082A to latch LAD bus data when $\overline{CAS}$ has a low-to-high transition if LRDY was high and BUSFLT was low at the previous LCLK2 rising edge. In the host-independent mode, this signal is the read strobe output.
CC	J13	I	Condition Code Input. In both modes, may be used as an external conditional input for branch conditions.
CID0 CID1 CID2	L14 K13 L15	I	Coprocessor ID. In the coprocessor mode, used to set a coprocessor ID so that a SMJ34020 Graphics System Processor controlling multiple SMJ34082A coprocessors can designate which coprocessor is being selected by the current instruction. Tied low in the host-independent mode.
CLK	J15	I	System Clock. In the coprocessor mode, tied low. In the host-independent mode, input is the system clock.
$\overline{COINT}$	E15	O	Coprocessor Interrupt Request, active low. In the coprocessor mode, signals an exception not masked out in the configuration register. Remains low until the status register is read. In the host-independent mode, user programmable I/O when LADCFG is low. When LADCFG is high, designates bus cycle boundaries on LAD31-0.
CORDY	F13	O	Coprocessor Ready. In the coprocessor mode, if the SMJ34020 sends an instruction before the SMJ34082A has completed a previous instruction, this signal goes low to indicate that the SMJ34020 should wait. In the host-independent mode, user programmable.
DS/$\overline{CS}$	R13	O	Data Space/Code Space. In both modes, when MEMCFG is low and DS/$\overline{CS}$ is low, selects program memory on MSD port. When MEMCFG is low and DS/$\overline{CS}$ is high, selects data memory on MSD port. When MEMCFG is high, DS/$\overline{CS}$ is memory chip select, active low.
EC0 EC1	G15 G14	I	Emulator Mode Control and Test. In both modes, tied high for normal operation.
INTG	P13	O	Interrupt Grant Output. In the coprocessor mode, INTG is low. In the host-independent mode, this signal is set high to acknowledge an interrupt request input.
$\overline{INTR}$	K14	I	Interrupt Request Input, active low. In the coprocessor mode, $\overline{INTR}$ is tied high. In the host-independent mode, causes call to subroutine address in interrupt vector register.

† The [ ]'s denote the type of buffer ut
ilized in the host-independent mode. If no [ ]'s appear, the buffer type is identical for both modes of operation.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Terminal Functions (Continued)

PIN NAME	PIN NO.	I/O†	DESCRIPTION
LAD0 LAD1 LAD2 LAD3 LAD4 LAD5 LAD6 LAD7 LAD8 LAD9 LAD10 LAD11 LAD12 LAD13 LAD14 LAD15 LAD16 LAD17 LAD18 LAD19 LAD20 LAD21 LAD22 LAD23 LAD24 LAD25 LAD26 LAD27 LAD28 LAD29 LAD30 LAD31	B3 A2 B4 A3 B5 A4 C6 B6 A5 A6 B7 A7 A8 A9 B9 A10 B10 A11 B11 A12 B12 C11 A13 B13 A14 C13 C14 B15 D14 C15 E14 D15	I/O	Local Address and Data Bus. In the coprocessor mode, used by SMJ34020 to input instructions and data operands to SMJ34082, and used by SMJ34082A to output results. In the host-independent mode, used by the SMJ34082A for address output and data I/O.
LCLK1 LCLK2	M14 M15	I	Local Clocks 1 and 2. In the coprocessor mode, two local clocks generated by the SMJ34020, 90 degrees out of phase, to provide timing inputs to SMJ34082A. In the host-independent mode, tied low.
$\overline{LOE}$	H14	I	Local Bus Output Enable, active low. In both modes, enables the local bus (LAD31-0) to be driven at the proper times when low. In addition during the host-independent mode when LADCFG is low, does not affect $\overline{ALTCH}$, $\overline{CAS}$, $\overline{WE}$, CORDY, or $\overline{COINT}$. When LADCFG is high, $\overline{ALTCH}$, $\overline{COINT}$, and CORDY are not disabled by $\overline{LOE}$ high; $\overline{CAS}$ and $\overline{WE}$ are disabled.
LRDY	N13	I	Local Bus Data Ready. In the coprocessor mode, when LRDY is high, indicates that data is available on LAD bus. When LRDY is low, indicates that the SMJ34082A should not load data from LAD31-0 and may also be used in conjunction with BUSFLT. In the host-independent mode, when LRDY is low, the device is stalled until LRDY is set high again and tied high if not used.
$\overline{MAE}$	N12	I	Memory Address and Data Output Enable, active low. In both modes, with $\overline{MAE}$ low, the SMJ34082A can output an address on MSA15-0 and data on MSD31-0. $\overline{MAE}$ high does not disable DS/$\overline{CS}$, $\overline{MCE}$, $\overline{MWR}$, or $\overline{MOE}$.
$\overline{MCE}$	R14	O	Memory Chip Enable. In both modes, when MEMCFG low, active (low) indicates access to external memory on MSD31-0. When MEMCFG is high, $\overline{MCE}$ low is external code memory chip select.
$\overline{MOE}$	P12	O	Memory Output Enable, active low. In both modes when low, enables output from external memory on to MSD port.

Terminal Functions (Continued)

PIN NAME	PIN NO.	I/O†	DESCRIPTION
MSA0	R4	O	Memory Address output. In both modes, addresses up to 64K words of external program memory and/or up to 64K words of data memory on the MSD port, depending on setting of DS/$\overline{CS}$ select.
MSA1	P5		
MSA2	R5		
MSA3	P6		
MSA4	R6		
MSA5	N7		
MSA6	P7		
MSA7	R7		
MSA8	P8		
MSA9	R9		
MSA10	P9		
MSA11	R10		
MSA12	R11		
MSA13	P10		
MSA14	N10		
MSA15	R12		
MSD0	C3	I/O	External Memory Data. In both modes, I/Os to external memory. Used to read from or write to external data or program memory on the MSD port.
MSD1	B1		
MSD2	D3		
MSD3	C2		
MSD4	C1		
MSD5	D2		
MSD6	D1		
MSD7	E2		
MSD8	E1		
MSD9	F2		
MSD10	F1		
MSD11	G3		
MSD12	G2		
MSD13	G1		
MSD14	H1		
MSD15	J1		
MSD16	J2		
MSD17	K1		
MSD18	L1		
MSD19	K2		
MSD20	M1		
MSD21	L2		
MSD22	N1		
MSD23	L3		
MSD24	M2		
MSD25	P1		
MSD26	N2		
MSD27	R2		
MSD28	N4		
MSD29	P3		
MSD30	R3		
MSD31	P4		
MSTR	J14	I	Host-Independent/Coprocessor Mode Select. In the coprocessor mode, MSTR must be tied low to operate properly. In the host-independent mode, MSTR must be tied high to operate properly.
$\overline{MWR}$	P11	O	Memory Write Enable. In both modes, when low, data on MSD31-0 can be written to external program or data memory.

Terminal Functions (Continued)

PIN NAME	PIN NO.	I/O†	DESCRIPTION
NC	A1 A15 B2 B14 D4 P2 R1 R15		No Internal Connection. These pins should be left floating.
$\overline{RAS}$	P15	I	Row Address Strobe, active low. In the coprocessor mode, $\overline{RAS}$ is high during all of coprocessor instruction cycle. In the host-independent mode, it is not used.
RDY	K15	I	Ready. In both modes, when RDY is low, it causes a nondestructive stall of sequencer and floating-point operations. All internal registers and status in the FPU core are preserved. Also, no output lines will change state.
$\overline{RESET}$	N15	I	Reset, active low. In both modes, resets sequencer output and clears pipeline registers, internal states, status, and exception disable registers in FPU core. Other registers are unaffected.
SF	N14	I	Special Function Input. In the coprocessor mode when SF is high, indicates the LAD bus input is an instruction or data from SMJ34020 registers. When SF is low, indicates the LAD input is a data operand from memory. In the host-independent mode, not used.
TCK	R8	I	Test Clock for JTAG four-wire boundary scan. In both modes, TCK is low for normal operation.
TDI	H15	I	Test Data Input for JTAG four-wire boundary scan. In both modes, TDI may be left floating.
TDO	H2	O	Test Data Output for JTAG four-wire boundary scan
TMS	B8	I	Test Mode Select for JTAG four-wire boundary scan. In both modes, SMJ may be left floating.
V_{CC}	C5 C8 C10 D13 F3 J3 M13 N3 N6 N9	I	5-V Power Supply. All pins must be connected and used.
V_{SS}	C4 C7 C9 C12 E3 E13 H3 H13 K3 L13 M3 N5 N8 N11	I	Ground Pins. All pions must be connected and used.
$\overline{WE}$	G13	I [O]	Write Enable, active low. In the coprocessor mode, the write strobe from the SMJ34020 to enable a write to or from the SMJ34082A LAD bus. In the host-independent mode, the SMJ34082A write strobe output.

† The [ ]'s denote the type of buffer utilized in the host-independent mode. If no [ ]'s appear, the buffer type is identical for both modes of operation.

data flow

The SMJ34082A has two bidirectional 32-bit buses, LAD31-0 and MSD31-0. Each bus can be used to pass instructions and data operands to the FPU core and to output results. A separate 16-bit bus, MSA15-0, provides memory addressing capability on the MSD bus.

When the SMJ34082A is used as a coprocessor for the SMJ34020 Graphics System Processor (GSP), data for the SMJ34082A can be transferred through the 32-bit bidirectional data bus (LAD31-0) and may be passed to any internal registers or to external memory on the memory expansion interface (MSD31-0). When the SMJ34082A is used as a standalone FPU, it can use both the LAD bus (LAD31-0) and the MSD bus (MSD31-0) to interface with external data memory or system buses.

In the host-independent mode, the SMJ34082A can be operated with the LAD bus as its single data bus and the MSD bus as the instruction source, or with data storage on either port and the program memory on the MSD bus.

The data space/code space (DS/$\overline{CS}$) output can be used to control access either to data memory or program memory on the MSD port. Up to 64K words of code space and 64K words of data space are directly supported. In the coprocessor mode, both instructions and data are transferred on the LAD bus with the option of accessing external user-generated programs on the MSD port.

One 32-bit operand can be input to the data registers each clock cycle. A 64-bit double-precision floating-point operand is input in two cycles. Transfers to or from the data registers can normally be programmed as block moves, loading one or more sets of operands with a single move instruction to minimize I/O overhead. Several modes for moving operands and instructions are available. Block transfers up to 512 words between the LAD and MSD buses can be programmed in either direction.

To permit direct input to or output from the LAD bus in the host-independent mode, other options for controlling the LAD bus have been implemented. When two 32-bit operands are being selected for input to the FPU core, one operand may be selected from LAD. On output from the FPU, a result may simultaneously be written to a register and to the LAD bus.

During initialization in the host-independent mode, a bootstrap loader can bring 65 32-bit words from the LAD bus and write them out to external program memory on the MSD bus, after which the device begins executing from the first memory location (zero). The first word is loaded into the configuration register. This option facilitates the initial loading of program memory on the MSD port upon power-up.

architecture

Because the sequencer, control and data registers, and FPU core are closely coupled, the SMJ34082A can execute a variety of complex floating-point or integer calculations rapidly, with a minimum of external data transfers. The internal architecture of the FPU core supports concurrent operation of the multiplier and the ALU, providing several options for storing or feeding back intermediate results. Also, several special registers are available to support specific calculations for graphics algorithms. Each of the main architectural elements of the SMJ34082A is discussed below.

The control functions of the SMJ34082A are provided by sequence control logic, register control logic, and bus interface control logic, together with user-programmed configuration settings stored in the configuration register. The on-board sequencer selects the next program execution address, either from internal code or from external program memory. Next-address sources include the program counter, stack, interrupt vector register, interrupt return register, or address register (for indirect jumps).

COUNTX, COUNTY, and MIN-MAX/LOOPCT registers are used for temporary storage by internal graphics routines. They may also serve as temporary storage for the user.

A separate FPU status register is provided, which can be used by test-and-branch instructions to control program execution. Because of the large number of status outputs, branches on status can be easily programmed. The status register contents are also important when dealing with status exceptions including such conditions as overflow, underflow, invalid operations (divide by zero), or illegal data formats such as infinity, Not a Number (NaN), or denormalized operands.

Register control logic permits all data and control registers to be accessed in accordance with applicable architectural restrictions. Register files A and B can be written to or read from the external buses, as can the control registers. Internal registers C and CT are embedded in the FPU core and can only be accessed by the FPU internal buses. The C and CT registers cannot be used as sources or destinations for MOVE instructions, and several registers (listed in Table 1) are not available as sources for FPU operations.

Table 1. Iinternal Registers

REGISTER ADDRESS	REGISTER NAME	RESTRICTIONS ON USE
00000	RA0	
00001	RA1	
00010	RA2	
00011	RA3	
00100	RA4	
00101	RA5	
00110	RA6	
00111	RA7	
01000	RA8	
01001	RA9	
01010	C†	Not a source or destination for moves
01011	CT†	Not a source or destination for moves
01100	STATUS	Not a source for FPU instructions
01101	CONFIG	Not a source for FPU instructions
01110	COUNTX	Not a source for FPU instructions
01111	COUNTY	Not a source for FPU instructions
10000	RB0	
10001	RB1	
10010	RB2	
10011	RB3	
10100	RB4	
10101	RB5	
10110	RB6	
10111	RB7	
11000	RB8	
11001	RB9	
11010	VECTOR	Not a source for FPU instructions
11011	MCADDR	Not a source for FPU instructions
11100	SUBADD0	Not a source for FPU instructions
11101	SUBADD1	Not a source for FPU instructions
11110	IRAREG	Not a source for FPU instructions
11111	MIN-MAX/LOOPCT	Not a source for FPU instructions

† C and CT registers cannot both be used for FPU operand sources in the same instruction.

register files A and B, feedback registers C and CT

SMJ34082A contains two register files, each with ten 64-bit registers and two 64-bit feedback registers. Most instructions will operate on one value from each of the RA and RB register files and return the result to either the RA or RB files or one of the feedback registers.

When the ONEFILE control bit is high in the configuration register, data written to a register in file RA is simultaneously written to the corresponding location in file RB. In this mode, the two register files act as a ten-word, two-read/one-write register file.

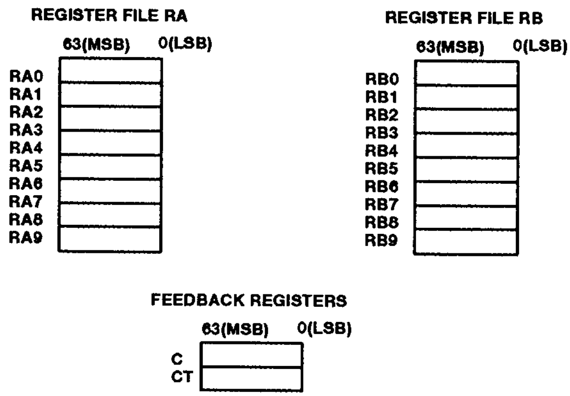


Figure 1. Data Registers

Two 64-bit feedback registers, C and CT, are embedded in the FPU core. FPU instructions may use the feedback registers as one of the operands, but the registers cannot be accessed for external moves. The C and CT registers can be used as either the A or B operand, but both cannot be used as operands during the same instruction. However, C (or CT) may be used for more than one operand in the same instruction. For example, C + CT is not a valid instruction, but C + C is.

The CT feedback register is used in integer divide operations as a temporary holding register. Any data stored in CT will be lost during an integer divide.

internal control/status register definitions

configuration register definition

The configuration register (CONFIG) is a special 32-bit register that the user loads to configure the SMJ34082A for exception handling, IEEE mode (vs. fast mode), rounding modes, and data-fetch operations. The configuration register is initialized to 'FFE00420' hex.

Table 2. Configuration Register Definition

BIT NO.	NAME	DESCRIPTION
31	MIVAL	Multiplier invalid operation (I) exception mask. Initialized to 1 (enabled).
30	MOVER	Multiplier overflow (V) exception mask. Initialized to 1 (enabled).
29	MUNDER	Multiplier underflow (U) exception mask. Initialized to 1 (enabled).
28	MINEX	Multiplier inexact (X) exception mask. Initialized to 1 (enabled).
27	MDIV0	Divide by zero (DIV0) exception mask. Initialized to 1 (enabled).
26	MDENORM	Multiplier denormal (DENORM) exception mask. Initialized to 1 (enabled).
25	AIVAL	ALU invalid operation (I) exception mask. Initialized to 1 (enabled).
24	AOVER	ALU overflow (V) exception mask. Initialized to 1 (enabled).
23	AUNDER	ALU underflow (U) exception mask. Initialized to 1 (enabled).
22	AINEX	ALU inexact (X) exception mask. Initialized to 1 (enabled).
21	ADENORM	ALU denormal (DENORM) exception mask. Initialized to 1 (enabled).
11-20	N/A	Reserved, set to all 0s.
10	REVISION	Revision number, read only. Set to 1.
9	LADCFG	When low, $\overline{CAS}$, $\overline{WE}$, CORDY, $\overline{COINT}$, and $\overline{ALTCH}$ are active signals not affected by $\overline{LOE}$. When high, $\overline{LOE}$ high places $\overline{CAS}$ and $\overline{WE}$ in high impedance, as well as the LAD bus. $\overline{COINT}$, which defines the LAD cycle boundaries, is controlled by bit 1 of the LAD move instruction instead of the set mask instruction. $\overline{COINT}$ will remain high unless a LAD move instruction (with bit 1 high) is in progress. The setting of this bit has no effect in the coprocessor mode. Initialized to 0.
8	MEMCFG	When high, $\overline{MCE}$ becomes code space chip enable and DS/$\overline{CS}$ becomes data space chip enable (eliminates need for external inverter). When low, $\overline{MCE}$ is chip select for external code and data space. DS/$\overline{CS}$ functions as an address bit which selects code space (when low) or data space (when high). Initialized to 0.
7	N/A	Reserved for later use. Initialized to 0. Must be loaded with 0.
6	ONEFILE	When high, causes simultaneous write to both register files (for example, to both RA0 and RB0 at once). The register files act as a single two-read, one-write register file. Initialized to 0.
5	PIPES2	When high, makes FPU output registers transparent. When low, registers are enabled. Initialized to 1.
4	PIPES1	When high, makes FPU internal pipeline registers transparent. When low, registers are enabled. Initialized to 0.
3	FAST	When high,fast mode is selected (all denormalized inputs and outputs are 0). When low, IEEE mode is selected. Initialized to 0.
2	LOAD	Load order. 0 = MSH, then LSH; 1 = LSH, then MSH. Initialized to 0.
1	RND1	Rounding mode select 1. Initialized to 0.
0	RND0	Rounding mode select 0. Initialized to 0.

LSH denotes least-significant half of a 64-bit word, MSH denotes most-significant half of a 64-bit word.

The mask bits serve as exception detect enables for the exception masks listed above. Setting the bit high (logic '1') enables the detection of the specific exception. When an enabled exception occurs, the ED bit in the status register will be set high and can be used to generate interrupts. The fast bit allows the SMJ34082A to control the handling of denormalized numbers. When the fast bit is set high, all denormalized numbers input to the device are flushed to zero, and all denormalized results are also flushed to zero (this is also called 'sudden underflow'). When the fast bit is low, IEEE mode is selected. Denormalized numbers may be generated by (or input to) the ALU. Denormalized numbers must first be wrapped before being used as operands for multiply or divide instructions.

The LOAD bit defines the expected order of double-precision operands. At reset, this bit will default to 0 indicating that the most significant 32 bits are transferred first. If the bit is set to a 1, then the expected order of 64-bit data transfers starts with the least significant 32 bits.

The RND0 and RND1 bits select the IEEE rounding mode, as shown in Table 3.

Table 3. Rounding Mode

RND1 - RND0	ROUNDING MODES
0 0	Round towards nearest
0 1	Round toward zero (truncated)
1 0	Round towards infinity (round up)
1 1	Round towards negative infinity (round down)

status register definition

The floating-point status register (STATUS) is a 32-bit register used for reporting the exceptions that occur during SMJ34082A operations and status codes set by the results of implicit and explicit compare operations. The status register is cleared upon reset, except for the INTENED flag, which is set to 1 in the coprocessor mode.

Table 4. Status Register Definition

BIT NO.	NAME	DESCRPTION
31	N	Sign bit (A < B flag for compare)
30	GT	A > B (valid on compare)
29	Z	Zero flag (A = B for compare)
28	V	IEEE overflow flag. The result is greater than the largest allowable value for the specified format.
27	I	IEEE invalid operation flag. A NaN has been input to the multiplier or the ALU, or an invalid operation [(0 * 1) or (∞−∞) or (−∞ +∞)] has been requested. This signal also goes high if an operation involves the square root of a negative number. When IVAL hoes high, the STX pins indicate which port had the NaN.
26	U	IEEE underflow flag. The result is inexact and less than the minimum allowable value for the specified format. In fast mode, this condition causes the result to go to zero.
25	X	IEEE inexact flag. The result of an operation is inexact.
24	DIV0	Divide by zero. An invalid operation involving a zero divisor has been detected by the multiplier.
23	RND	The mantissa of a number has been increased in magnitude by rounding. If the number generated was wrapped, then the 'unwrap rounded' instruction must be used to properly unwrap the wrapped number.
22	DENIN	Input to the multiplier is a denormalized number. When DENIN goes high, the STX pins indicate which port has the denormal input.
21	DENORM	The multiplier output is wrapped number or the ALU output is a denormalized number. In fast mode, this condition causes the result to go to zero. It also indicates an invalid integer operation with a negative unsigned integer result.
20	STX1	A NaN or a denormalized number has been input on the A port.
19	STX0	A NaN or a denormalized number has been input on the B port.
18	ED	Exception detect status signal representing logical OR of all enabled exceptions in the configuration register.
17	UNORD	The two inputs of a comparison operation are unordered, i.e.; one or both of the inputs is an NaN.
16	INTFLG	Software interrupt flag. Set by external code to signal a software interrupt.
15	INTENHW	Hardware interrupt ($\overline{INTR}$) enable, active high (initialized to zero)
14	NXOROV	N (negative) XOR V (overflow)
13	VANDZB	V (overflow) AND $\overline{Z}$ (NOT zero)
12	INTENED	ED interrrupt enable, active high (initialized to zero in the host-independent mode, one in the coprocessor mode)
11	INTENSW	Software interrupt (INTFLG) enable, active high (initialized to zero)
10	ZGT	Zn > Zmax (valid for 2-D MIN-MAX instruction)
9	ZLT	Zn < Zmin (valid for 2-D MIN-MAX instruction)
8	YGT	Yn > Ymax (valid for 1-D or 2-D MIN-MAX instruction)
7	YLT	Yn < Ymin (valid for 1-D or 2-D MIN-MAX instruction)
6	XGT	Xn > Xmax (valid for 1-D or 2-D MIN-MAX instruction)
5	XLT	Xn < Xmin (valid for 1-D or 2-D MIN-MAX instruction)
4	HINT	Hardware interrupt flag
3-0	N/A	Reserved

indirect address register (MCADDR) definition

The indirect address register (MCADDR) can be set to point to a memory location for indirect move or jump operations through the MSD port. MCADDR is cleared upon reset.

31															16	0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	V	INDIRECT ADDRESS

Figure 2. Indirect Address Definition

The function of bit 16 varies, depending on whether the instruction is a MOVE or JUMP. During a MOVE instruction, bit 16 selects data space when set high, or code space when low. During a JUMP instruction, bit 16 selects an internal instruction when set high, or an external instruction when low.

stack registers (SUBADD1-SUBADD0) definition

The stack contains two subroutine return address registers, SUBADD0 and SUBADD1, which serves as a two-deep LIFO (last-in, first-out) stack. A subroutine jump causes the program counter to be pushed onto the stack, and a return from subroutine pops the last address pushed on the stack. More than two pushes will overwrite the contents of SUBADD1.

Bit 31 (Pointer) is set high in the stack location that was written last and reset to zero in the other stack location. Setting bit 30 (Enable) high enables a write into bit 31 (set or reset the pointer) in either stack location. If bit 31 is zero in both SUBADD0 and SUBADD1 (as when the stack has been saved externally and later restored), SUBADD0 can be designated as top of stack by setting bit 31. The stack pointers (bit 31) are cleared upon reset.

Bit 16 (I) is set high when the address in a stack location points to an internal routine, or set low when the address is for an external instruction.

31															16	0
P	E	X	X	X	X	X	X	X	X	X	X	X	X	X	I	SUBADD0
P	E	X	X	X	X	X	X	X	X	X	X	X	X	X	I	SUBADD1

Figure 3. Stack Definition

interrupt vector register (VECTOR) definition

The interrupt vector register (VECTOR) serves as a pointer to an external program to be executed upon receipt of an interrupt. Bit 16 (I) is always set low to point to a routine in external code space. The interrupt vector is cleared on reset.

31															16	0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	I	INTERRUPT ADDRESS

Figure 4. Interrupt Vector Definition

interrupt return register (IRAREG) definition

The interrupt return register (IRAREG) retains a copy of the program counter at the time of an external interrupt. This address is used as the next execution address upon returning from the interrupt. Bit 16 (I) is set high when the address in the stack location points to an internal instruction, or set low when the address is for an external instruction. This register is not affected by the reset signal.

31															16	0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	I	INTERRUPT RETURN ADDRESS

Figure 5. Interrupt Return Definition

COUNTX and COUNTY registers definition

The counter registers (COUNTX, COUNTY) are used to store the current counts of the minimum and maximum values when executing MIN-MAX instructions. COUNTX and COUNTY are cleared on reset.

31 … 16	… 0
COUNT FOR MAX VALUE	COUNT FOR MIN VALUE

Figure 6. COUNTY and COUNTX Register Definition

The COUNTX register is updated on both the 1-D and 2-D MIN-MAX instruction such that the count of the current minimum value is in the lower 16 bits of the register and the count of the current maximum value is in the upper 16 bits. The COUNTY register is used only in the 2-D MIN-MAX instruction to keep track of the counts of the minimum and maximum for the second value of a pair. The COUNTX and COUNTY registers may also be used for temporary storage when not using the MIN-MAX instructions.

MIN-MAX/LOOPCT register

The MIN-MAX/LOOPCT register stores the current values of two separate counters. The LSH contains the current loop counter, and the MSH is used to hold the current minimum or maximum value of a MIN-MAX operation. The MIN-MAX/LOOPCT register is cleared upon reset. The MIN-MAX/LOOPCT register may also be used for temporary storage when not using the MIN-MAX instructions.

31 … 16	… 0
COUNT FOR MIN-MAX VALUE	LOOP COUNT

Figure 7. MIN-MAX/LOOPCT Register Definition

FPU core

The FPU core itself consists of a multiplier and an ALU, each with an intermediate pipeline register and an output register (see Figure 8, FPU core functional block diagram). Four multiplexers select the multiplier and ALU operands from the data registers, feedback registers, or previous multiplier or ALU result. Results are directed either to the internal feedback registers (C or CT), the 20 data registers in register files RA and RB, or the ten other miscellaneous registers.

Both the internal pipeline registers and the output registers can be enabled or made transparent (disabled) by setting the PIPES2-PIPES1 bits in the configuration register. When the device is powered up, the default settings of the internal registers are PIPES2 high (output registers transparent) and PIPES1 low (internal pipeline registers enabled).

When the FPU core is used for chained operations, the multiplier and ALU operate in parallel. Two data inputs are provided from the RA and RB input registers, while multiplier and ALU feedback are used as the other two operands. While in the chained mode, the output registers of the FPU must be enabled to latch feedback operands. The appropriate registers must be enabled by setting the PIPES2-PIPES1 controls in the configuration register at the beginning of chained operations, and the PIPES2-PIPES1 control should then be reinitialized upon termination.

Fully pipelined operation (both pipeline and output registers enabled) affects timing when writing results back to the RA and RB register files. To adjust writeback timing, it is possible to issue the NOP (no operation) instruction to the FPU core when the results are to be retained in the output registers for one or more additional cycles. The NOP instruction is only effective when the output registers are enabled, as each NOP causes the output register contents to be retained for one additional cycle.

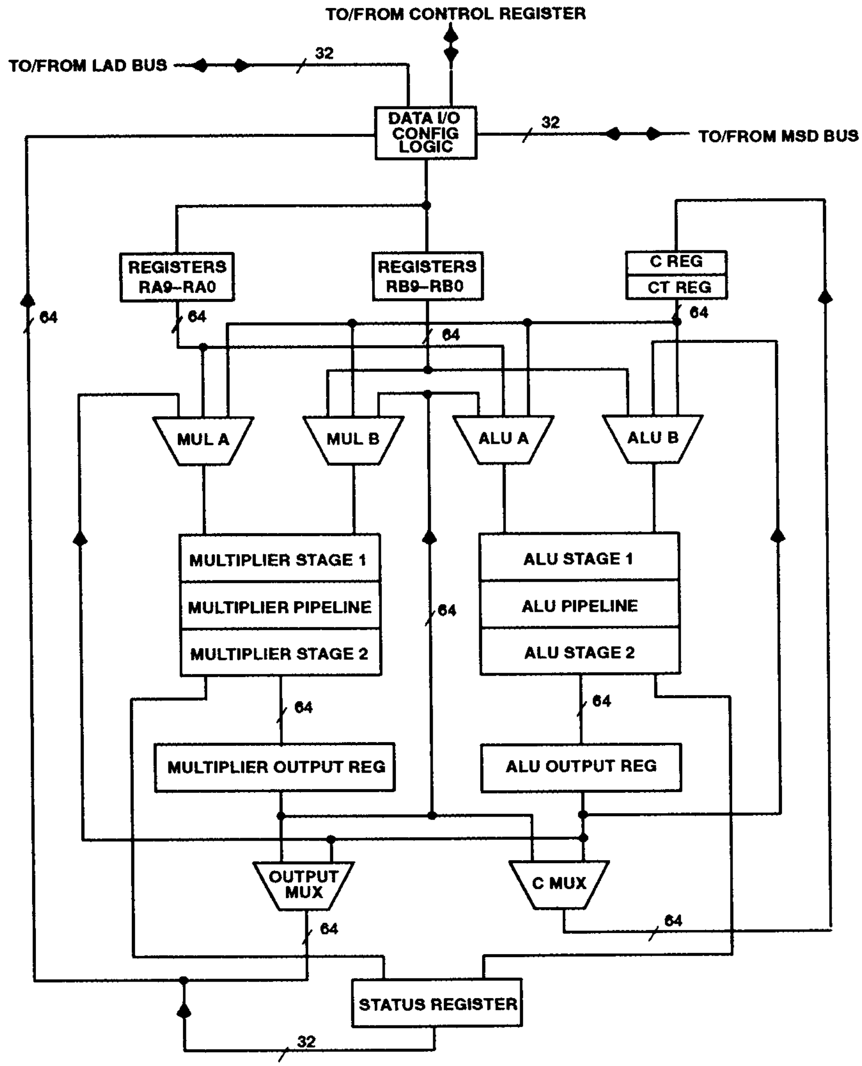


Figure 8. FPU Core Functional Block Diagram

SMJ34082A operating modes

The SMJ34082A can operate as a stand-alone floating-point processor or a graphics coprocessor to the SMJ34020 Graphics System Processor. Control of FPU operation is provided either from external program memory or from the SMJ34020. External instructions are addressed by address lines MSA15-0 and are input on MSD31-0. SMJ34020 instructions are input on LAD31-0.

Both the MSD and LAD buses can be used for data transfers as well. Combinations of control signals distinguish instruction fetches from data transfers. A single instruction may be used to transfer data and to perform an operation within the FPU.

The SMJ34082A supports external code and data storage with the memory expansion interface, MSD31-0. Up to 64K 32-bit data operands and 64K instructions may be added externally to the SMJ34082A. The signal DS/$\overline{CS}$ controls whether data space or code space is being accessed, and read/write control is provided with the chip enable ($\overline{MCE}$), output enable ($\overline{MOE}$), address enable ($\overline{MAE}$), write enable ($\overline{MWR}$), and address lines (MSA15-0).

The SMJ34082A also provides instructions that allow the SMJ34020 to read/write directly from/to external memory. The external code support permits full utilization of the SMJ34082A features and instruction set.

coprocessor-mode operation

Operation in the coprocessor mode assumes MSTR is low. In this mode, the SMJ34082A acts as a closely coupled coprocessor to the SMJ34020. The interface between the two devices consists of direct connections between pins. More than one coprocessor may be connected to the SMJ34020 by setting the appropriate coprocessor ID (CID2-CID0). Up to four coprocessors executing in parallel may be used with a single SMJ34020.

In the coprocessor mode, clock signals are provided by LCLK1 and LCLK2 from the SMJ34020. Internally, the FPU generates a rising clock edge from each LCLK1 edge (rising or falling). Thus, the SMJ34082A actually operates at twice the LCLK1 input clock frequency.

initialization (coprocessor mode)

On reset, the SMJ34082A clears all pipeline registers and internal states. The configuration register and status register return to their initialization values. When $\overline{RESET}$ returns high in the coprocessor mode, the SMJ34082A is in an idle state waiting for the next instruction from the SMJ34020.

LAD bus control (coprocessor mode)

Both data and instructions are transferred over the bidirectional LAD bus in the coprocessor mode. A unique combination of signal inputs distinguishes an instruction from data. SF, $\overline{ALTCH}$, $\overline{CAS}$, $\overline{RAS}$, and $\overline{WE}$ are used to designate coprocessor functions from other operations on the LAD bus.

Data may be transferred to or from SMJ34020 registers or memory via LAD31-0. Transfers between the LAD and MSD buses can also be programmed. A single coprocessor instruction may be used to transfer data to the SMJ34082A and then perform an FPU operation.

MSD bus control (coprocessor mode)

Use of the MSD bus in the coprocessor mode is optional. External memory on MSD31-0 can be used to store data, user-programmed subroutines, or both. Different combinations of control signals distinguish between data memory and code memory. Control signals for MSD and MSA buses operate the same in the host-independent and coprocessor modes.

interrupt handling (coprocessor mode)

A software interrupt to the SMJ34082A is generated by the set mask external instruction. When the interrupt is granted, the current program counter is stored in the interrupt return register, and a branch to the interrupt vector address is executed. Software interrupts may be disabled.

If the exception detect interrupt (ED) is enabled, a SMJ34082A exception causes $\overline{\mathrm{COINT}}$ to go low, signalling the exception to the SMJ34020. This exception does *not* cause a branch to the interrupt vector. If its interrupts are enabled, the SMJ34020 will branch to an interrupt vector to service the SMJ34082A request. Interrupts are cleared by reading the SMJ34082A status register.

host-independent mode operation

Operation in the host-independent mode assumes MSTR high. The SMJ34082A has several hardware control signals, as well as programmable features, which support system functions such as initialization, data transfer, or interrupts in the host-independent mode. CLK provides the input clock to the SMJ34082A. Details of initialization, LAD and MSD bus interface control, and interrupt handling are provided in the following sections.

initialization (host-independent mode)

To simplify initialization of external program memory, the SMJ34082A provides a bootstrap loader to perform an initial program load of 64 instructions. Once invoked, the loader causes the SMJ34082A to read 65 words from the LAD bus and write 64 words out to the external program memory on the MSD bus, beginning with location 0. The first word read is used to initialize the configuration register.

This loader is invoked by first setting $\overline{\mathrm{RESET}}$ low, and then $\overline{\mathrm{INTR}}$ low. A separate timing diagram for using the bootstrap loader is provided (see Figure 34). $\overline{\mathrm{INTR}}$ should be taken low after $\overline{\mathrm{RESET}}$ is already low, as shown in the diagram. When the bootstrap loader is started, the FPU core is reset (internal states and status are cleared, but not data registers) and the stack pointer, program counter, and interrupt vector register are all set to zero.

$\overline{\mathrm{RESET}}$ must be set high again before the loader operation can start (see Figure 34). Once the loader is active, an external interrupt (signalled by $\overline{\mathrm{INTR}}$ low) will not be granted until the load sequence is finished. However, $\overline{\mathrm{RESET}}$ going low terminates the load sequence, regardless of whether the sequence is complete. When the load sequence is finished, the device begins program execution at external address 0.

LAD bus control (host-independent mode)

Data transfer from the LAD bus (LAD31-0) is controlled primarily by output signals, $\overline{\mathrm{ALTCH}}$, $\overline{\mathrm{WE}}$, and $\overline{\mathrm{CAS}}$. $\overline{\mathrm{ALTCH}}$ is the address write strobe that signals an address is being output on the LAD bus. The $\overline{\mathrm{CAS}}$ signal is the read strobe, and $\overline{\mathrm{WE}}$ is the write enable output to memory.

If a bidirectional FIFO is used instead of memory, $\overline{\mathrm{CAS}}$ can be directly connected to the read clock and $\overline{\mathrm{WE}}$ to the write clock. The CC input can be used to signal the SMJ34082A when data is ready for input from the FIFO stack.

Data input on the LAD bus can be written to data registers, control registers, or passed through for output on the MSD bus. Alternatively, the LAD bus input can be selected directly as an FPU source operand without writing to a register.

An FPU result can be written to a data register and at the same time be passed out on the LAD bus. When this is done, the clock period may need to be extended up to 15 ns (SMJ34082-30) to allow for the propagation delay from the FPU core to the outputs.

Depending on the specific system implementation, transferring data to and from the LAD bus without intervening register operations may significantly improve throughput. In the host-independent mode, data moves to and from internal registers can be minimized at the cost of adjusting the clock period to assure integrity of FPU inputs to and output from the LAD bus.

MSD bus control (host-independent mode)

The MSD bus can be used to access either external data memory or external code memory, depending on the combination of control signals required. If the memory on the MSD port is shared with a host processor, the $\overline{\text{MAE}}$ and RDY signals can be used to prevent conflicts between the host and the SMJ34082A. When memory on the MSD port is shared, the host processor can monitor the state of the SMJ34082A memory chip enable ($\overline{\text{MCE}}$) to determine when the SMJ34082A is not accessing the memory.

Otherwise, the $\overline{\text{MAE}}$ signal may be tied low (if unused), and the SMJ34082A can use $\overline{\text{MOE}}$, $\overline{\text{MCE}}$, $\overline{\text{MWR}}$, and DS/$\overline{\text{CS}}$ to control external memory operations into either data space or code space, as selected by DS/$\overline{\text{CS}}$.

interrupt handling (host-independent mode)

Interrupts to the SMJ34082A can be signalled by setting the interrupt request input ($\overline{\text{INTR}}$) low. $\overline{\text{INTR}}$ is associated with the vector in the interrupt vector register. Software interrupts are signalled by setting the software interrupt flag in the status register.

In the event of an FPU status exception in the host-independent mode, an interrupt is generated that causes a branch to an exception handler routine. The address of the exception handler is stored in the interrupt vector register by the user prior to execution of the FPU program. Interrupts may be disabled by setting the appropriate bits in the status register.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC} (see Note 1) .... 6 V
Input voltage range, V_I .... −0.3 V to 6 V
Off-state output voltage range .... −2 V to 6 V
Operating free-air (minimum) and case (maximum) temperature range .... −55°C to 125°C
Storage temperature range .... −65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage levels are with respect to ground (V_{SS}).

recommended operating conditions

PARAMETER				MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage			4.5	5	5.5	V
V_{SS}	Supply voltage (see Note 2)				0		V
V_{IH}	High-level input voltage			2.4		V_{CC}+0.3	V
V_{IL}	Low-level input voltage			−0.3		0.6	V
I_{OH}	High-level output current					−8	mA
I_{OL}	Low-level output current					8	mA
f_{clock}	Clock frequency	Coprocessor mode	SMJ34082A-28			7.1	MHz
			SMJ34082A-30			7.6	
		Host-independent Mode	SMJ34082A-28			14.3	
			SMJ34082A-30			15.4	
T_A	Operating free-air temperature			−55			°C
T_C	Operating case temperature					125	°C

NOTE 2: In order to minimize noise on V_{SS}, care should be taken to provide a minimum-inductance path between the V_{SS} pins and system ground.

electrical characteristics over recommended operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP‡	MAX	UNIT
V_{OH}	High-level output voltage		V_{CC} = 4.5 V,	I_{OH} = −8 mA	2.6			V
V_{OL}	Low-level output voltage		V_{CC} = 4.5 V,	I_{OL} = 8 mA			0.6	V
I_O	High-impedance bidirectional pins output current		V_{CC} = 4.5 V,	V_O = 2.8 V			10	µA
			V_{CC} = 4.5 V,	V_O = 0.6 V			−10	
I_I	Input current		V_I = V_{SS} to V_{CC}				±10	µA
I_{CC}§	Supply current	Dynamic	V_{CC} = 5.5 V				325	mA
		Quiescent	V_I = V_{IL}max or V_{IH}min,	I_{OH} = I_{OL} = 0			50	mA
			V_I = 0.2 V or V_{CC} − 0.2 V,	I_{OH} = I_{OL} = 0			50	
C_i	Input capacitance					10		pF

‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.
§ I_{CC} is measured at maximum clock frequency. Inputs are presented with random logic highs and lows to assure the toggling of internal nodes.

coprocessor mode (MSTR low)

switching characteristics over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted)†

propagation delay times

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{p(ATCL-CORV)}$	Propagation delay time, $\overline{\text{ALTCH}}$ low to CORDY valid	11		40		40	ns
$t_{p(ATCH-LADV)}$	Propagation delay time, $\overline{\text{ALTCH}}$ high to LAD data valid	16		35		35	
$t_{p(CASL-LADV)}$	Propagation delay time, $\overline{\text{CAS}}$ low to LAD data valid	14		30		25	
$t_{p(CASH-LADZ)}$	Propagation delay time, $\overline{\text{CAS}}$ high to LAD disabled	14		30		25	
$t_{p(LC1-DCSL)ML}$	Propagation delay time, LCLK1 ↑ or ↓ to DS/$\overline{\text{CS}}$ low with MEMCFG low	17, 21, 23		25		25	
$t_{p(LC1-DCSH)ML}$	Propagation delay time, LCLK1 ↑ or ↓ to DS/$\overline{\text{CS}}$ high with MEMCFG low	17, 19, 21, 23, 24, 26		25		25	
$t_{p(LC1-DCSL)MH}$	Propagation delay time, LCLK1 ↑ or ↓ to DS/$\overline{\text{CS}}$ low with MEMCFG high	18, 20, 22, 25, 27		30	2	22	
$t_{p(LC1-DCSH)MH}$	Propagation delay time, LCLK1 ↑ or ↓ to DS/$\overline{\text{CS}}$ high with MEMCFG high	18, 20, 22, 25, 27		21	2	21	
$t_{p(LC1-MCEL)}$	Propagation delay time, LCK1 ↑ or ↓ to $\overline{\text{MCE}}$ low	17-19, 21-27		21	2	21	
$t_{p(LC1-MCEH)ML}$	Propagation delay time, LCLK1 ↑ or ↓ to $\overline{\text{MCE}}$ high with MEMCFG low	17, 19, 21, 23		23	2	23	
$t_{p(LC1-MCEH)MH}$	Propagation delay time, LCLK1 ↑ or ↓ to $\overline{\text{MCE}}$ high with MEMCFG high	18, 22, 25, 27		15	2	15	
$t_{p(LC1-MOEL)}$	Propagation delay time, LCLK1 ↑ or ↓ to $\overline{\text{MOE}}$ low	17, 18, 21-23, 26, 27	10	35	10	35	
$t_{p(LC1-MOEH)}$	Propagation delay time, LCLK1 ↑ or ↓ to $\overline{\text{MOE}}$ high	17, 18, 21-23, 26, 27	3	13	3	13	
$t_{p(LC1-MSAV)}$	Propagation delay time, LCLK1 ↑ or ↓ to MSA address valid	17-27		25		25	
$t_{p(LC1-MSDV)}$	Propagation delay time, LCLK1 ↑ or ↓ to MSD data valid	19, 20-22, 24, 25		40		40	
$t_{p(LC1-MWRL)}$	Propagation delay time, LCLK1 ↑ or ↓ to $\overline{\text{MWR}}$ low	19-22, 24, 25	10	35	10	35	
$t_{p(LC1-MWRH)}$	Propagation delay time, LCLK1 ↑ or ↓ to $\overline{\text{MWR}}$ high	20-22, 24, 25	3	13	3	13	
$t_{p(LC1H-COIL)}$	Propagation delay time, LCLK1 ↑ to $\overline{\text{COINT}}$ low	12		23		20	
$t_{p(LC1H-COIH)}$	Propagation delay time, LCLK1 ↑ to $\overline{\text{COINT}}$ high	12		23		20	
$t_{p(LC1H-LADV)}$	Propagation delay time, LCLK1 ↑ to LAD data valid	16		28		23	
$t_{p(MSDV-LADV)}$	Propagation delay time, MSD data valid to LAD data valid	26, 27		30		25	
$t_{p(RASH-LADXZ)}$	Propagation delay time, $\overline{\text{RAS}}$ high to LAD disabled	16		30		25	

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.

coprocessor mode (MSTR low)

switching characteristics over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted) (continued)†

enable and disable times

PARAMETER		FIGURE	SMJ34082A-28 MIN	SMJ34082A-28 MAX	SMJ34082A-30 MIN	SMJ34082A-30 MAX	UNIT
$t_{en(LOEL-LADZX)}$	Enable time, $\overline{LOE}$ low to LAD enabled	16	2	17	2	17	ns
$t_{en(MAEL-MSAZX)}$	Enable time, $\overline{MAE}$ low to MSA enabled	21, 22	2	17	2	17	
$t_{en(MAEL-MSDZX)}$	Enable time, $\overline{MAE}$ low to MSD enabled	22	2	17	2	17	
$t_{dis(LOEH-LADXZ)}$	Disable time, $\overline{LOE}$ high to LAD disabled	16	2	17	2	17	ns
$t_{dis(MAEH-MSAXZ)}$	Disable time, $\overline{MAE}$ high to MSA disabled	21, 22	2	17	2	17	
$t_{dis(MAEH-MSDXZ)}$	Disable time, $\overline{MAE}$ high to MSD disabled	21	2	17	2	17	

valid times

PARAMETER		FIGURE	SMJ34082A-28 MIN	SMJ34082A-28 MAX	SMJ34082A-30 MIN	SMJ34082A-30 MAX	UNIT
$t_{v(MWRH-MSA)}$	Valid time, MSA address after $\overline{MWR}$ high	20-22, 24, 25	0		0		ns
$t_{v(MWRH-MSD)}$	Valid time, MSD data output after $\overline{MWR}$ high	20-22, 24, 25	0		0		
$t_{v(LC1-MSA)}$	Valid time, MSA address valid after LCK ↑ or ↓	17-22, 24-27	3		3		
$t_{v(LC1L-COR)}$	Valid time, CORDY valid after LCLK1 low	11	0		0		

timing requirements over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted)†

clock period and pulse duration

PARAMETER		FIGURE	SMJ34082A-28 MIN	SMJ34082A-28 MAX	SMJ34082A-30 MIN	SMJ34082A-30 MAX	UNIT
$t_{c(LC1)}$	Clock period, LCLK1 ($1/f_{clock}$) with PIPES1 low	10, 17-22, 24-27	170		162		ns
$t_{c(LC2)}$	Clock period, LCLK2 ($1/f_{clock}$) with PIPES1 low	10	170		162		
$t_{w(LC1H)}$	Pulse duration, LCLK1 high	10	76		72		ns
$t_{w(LC1L)}$	Pulse duration, LCLK1 low	10	76		72		
$t_{w(LC2H)}$	Pulse duration, LCLK2 high	10	76		72		
$t_{w(LC2L)}$	Pulse duration, LCLK2 low	10	76		72		
$t_{w(DCSH)MH}$	Pulse duration, DS/$\overline{CS}$ high with MEMCFG high	20, 25, 27	5		5		
$t_{w(RSTL)}$	Pulse duration, $\overline{RESET}$ low	12	35		30		
$t_{w(MCEH)}$	Pulse duration, $\overline{MCE}$ high	18, 25, 27	5		5		
$t_{w(MOEH)}$	Pulse duration, $\overline{MOE}$ high	17, 18, 23, 26, 27	5		5		
$t_{w(MWRH)}$	Pulse duration, $\overline{MWR}$ high	20, 24, 25	5		5		

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.

coprocessor mode (MSTR low)

timing requirements over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted) (continued)†

transition times

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{t(LC1)}$	Transition time, LCLK1	10		15		15	ns
$t_{t(LC2)}$	Transition time, LCLK2	10		15		15	

setup and hold times

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{su(BUS-LC2H)}$	Setup time, BUSFLT valid before LCLK2 ↑	11	20		13		ns
$t_{su(CC-LC1)}$	Setup time, CC valid before LCLK1 ↑ or ↓	12	7		7		
$t_{su(LAD-ATCL)}$	Setup time, LAD address valid before $\overline{ALTCH}$ low	13-16, 23	17		17		
$t_{su(LAD-CASH)}$	Setup time, LAD address valid before $\overline{CAS}$ high	13, 15, 24, 25	15		15		
$t_{su(LRD-LC2H)}$	Setup time, LRDY valid before LCLK2 ↑	11	20		20		
$t_{su(MSD-LC1)}$	Setup time, MSD data valid before LCLK1 ↑ or ↓	17, 18, 23	12		12		
$t_{su(RASH-ATCL)}$	Setup time, $\overline{RAS}$ high before $\overline{ALTCH}$ low	13-15, 23	35		30		
$t_{su(RDYL-LC1)}$	Setup time, RDY low before LCLK1 ↑ or ↓	12	20		15		
$t_{su(RSTH-LC1)}$	Setup time, $\overline{RESET}$ high before LCLK1 ↑ or ↓	12	50		50		
$t_{su(SF-ATCL)}$	Setup time, SF valid before $\overline{ALTCH}$ low	13-16, 23	15		15		
$t_{su(WEL-CASL)}$	Setup time, $\overline{WE}$ low for data write before $\overline{CAS}$ low	13, 16	15		15		
$t_{h(ATCH-SF)}$	Hold time, SF valid after $\overline{ALTCH}$ high	13-15, 23	15		12		ns
$t_{h(ATCL-LAD)}$	Hold time, LAD address valid after $\overline{ALTCH}$ low	13-16, 23	21		17		
$t_{h(CASH-LAD)}$	Hold time, LAD data valid after $\overline{CAS}$ high	13, 15, 24, 25	0		0		
$t_{h(CASH-SF)}$	Hold time, SF valid after $\overline{CAS}$ high	13-15, 23	15		15		
$t_{h(LC1-CC)}$	Hold time, CC valid after LCLK1 ↑ or ↓	12	5		5		
$t_{h(LC1-MSD)}$	Hold time, MSD input data valid after LCLK1 ↑ or ↓	17, 18, 23	4		4		
$t_{h(LC1-RDY)}$	Hold time, RDY valid after LCLK1 ↑ or ↓	12	5		5		
$t_{h(LC1H-LC2L)}$	Hold time, LCLK2 low after LCLK1 high	10	20		20		
$t_{h(LC2H-BUS)}$	Hold time, BUSFLT valid after LCLK2 high	11	5		5		
$t_{h(LC2H-LC1H)}$	Hold time, LCLK1 high after LCLK2 high	10	20		20		
$t_{h(LC2H-LRD)}$	Hold time, LRDY valid after LCLK2 high	11	5		5		
$t_{h(WEH-SF)}$	Hold time, SF valid after $\overline{WE}$ high	13	20		20		

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.

SGUS012A – D3592, SEPTEMBER 1990 – REVISED MAY 1991

coprocessor mode (MSTR low)

timing requirements over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted) (continued)†

delay times

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{d(DCSH-MCEL)MH}$	Delay time, DS/$\overline{CS}$ high to $\overline{MCE}$ low with MEMCFG high	18, 22	4		4		ns
$t_{d(DCSH-MWRL)}$	Delay time, DS/$\overline{CS}$ high to $\overline{MWR}$ low	19, 24	5		5		
$t_{d(MCEH-DCSL)MH}$	Delay time, $\overline{MCE}$ high to DS/$\overline{CS}$ low with MEMCFG high	20	4		4		
$t_{d(MCEH-MWRL)}$	Delay time, $\overline{MCE}$ high to $\overline{MWR}$ low	25	5		5		
$t_{d(MOEH-MWRL)}$	Delay time, $\overline{MOE}$ high to $\overline{MWR}$ low	19	5		5		
$t_{d(MSAV-MWRL)}$	Delay time, MSA valid to $\overline{MWR}$ low	20-22, 24, 25	4		4		
$t_{d(MSDZ-MOEL)}$	Delay time, MSD disabled to $\overline{MOE}$ low	21, 22	2		2		
$t_{d(MWRH-MCEL)MH}$	Delay time, $\overline{MWR}$ high to $\overline{MCE}$ low with MEMCFG high	25	5		5		
$t_{d(MWRH-MOEL)}$	Delay time, $\overline{MWR}$ high to $\overline{MOE}$ low	19, 21, 22	5		5		
$t_{d(MWRH-MSDVZ)}$	Delay time, $\overline{MWR}$ high to MSD disabled	21	1	12	1	9	
$t_{d(MWRL-MSDZX)}$	Delay time, $\overline{MWR}$ low to MSD enabled	21, 22	1	13	1	13	

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.

host-independent mode (MSTR high)

switching characteristics over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted)†

propagation delay times

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{p(CLKH-ATCH)}$	Propagation delay time, CLK ↑ to $\overline{ALTCH}$ high	29, 30		10		10	ns
$t_{p(CLKH-ATCL)}$	Propagation delay time, CLK ↑ to $\overline{ALTCH}$ low	29, 30		28		28	
$t_{p(CLKH-CASH)}$	Propagation delay time, CLK ↑ to $\overline{CAS}$ high	29, 31, 32, 34-36		10		10	
$t_{p(CLKH-CASL)}$	Propagation delay time, CLK ↑ to $\overline{CAS}$ low	29, 31, 32, 34-36		28		28	
$t_{p(CLKH-COIH)}$	Propagation delay time, CLK ↑ to $\overline{COINT}$ high	29-31, 33, 35, 36, 46		20		20	
$t_{p(CLKH-COIL)}$	Propagation delay time, CLK ↑ to $\overline{COINT}$ low	29-31, 33, 35, 36, 46		20		20	
$t_{p(CLKH-CORH)}$	Propagation delay time, CLK ↑ to CORDY high	46		20		17	
$t_{p(CLKH-CORL)}$	Propagation delay time, CLK ↑ to CORDY low	46		20		17	
$t_{p(CLKH-DCSH)MH}$	Propagation delay time, CLK ↑ to DS/$\overline{CS}$ high with MEMCFG high	36, 38, 40, 42-44	1	10	1	10	
$t_{p(CLKH-DCSH)ML}$	Propagation delay time, CLK ↑ to DS/$\overline{CS}$ high with MEMCFG low	35, 37, 39, 41, 45, 46		23		20	
$t_{p(CLKH-DCSL)MH}$	Propagation delay time, CLK ↑ to DS/$\overline{CS}$ low with MEMCFG high	36, 38, 40, 42-44	1	23	1	20	
$t_{p(CLKH-DCSL)ML}$	Propagation delay time, CLK ↑ to DS/$\overline{CS}$ low with MEMCFG low	37, 41, 45-47		23		20	
$t_{p(CLKH-ITGH)}$	Propagation delay time, CLK ↑ to INTG high‡	47		20		15	
$t_{p(CLKH-ITGL)}$	Propagation delay time, CLK ↑ to INTG low	47		25		20	
$t_{p(CLKH-LADV)}$	Propagation delay time, CLK ↑ to LAD valid	29, 30, 33-35, 43, 44		35		35	
$t_{p(CLKH-MCEH)MH}$	Propagation delay time, CLK ↑ to $\overline{MCE}$ high with MEMCFG high	36, 38, 42-46	1	10	1	10	
$t_{p(CLKH-MCEH)ML}$	Propagation delay time, CLK ↑ to $\overline{MCE}$ high with MEMCFG low	37, 39, 41, 45-47	1	20	1	20	
$t_{p(CLKH-MCEL)}$	Propagation delay time, CLK ↑ to $\overline{MCE}$ low	35-39, 41-47	1	23	1	20	
$t_{p(CLKH-MOEH)}$	Propagation delay time, CLK ↑ to $\overline{MOE}$ high	37, 38, 41-47	1	11	1	11	
$t_{p(CLKH-MOEL)}$	Propagation delay time, CLK ↑ to $\overline{MOE}$ low	37, 38, 41-47	10	35	10	35	
$t_{p(CLKH-MSAV)}$	Propagation delay time, CLK ↑ to MSA address valid	35-47		20		20	
$t_{p(CLKH-MSDV)}$	Propagation delay time, CLK ↑ to MSD data valid	35, 36, 39-42		40		40	
$t_{p(CLKH-MWRH)}$	Propagation delay time, CLK ↑ to $\overline{MWR}$ high	35, 36, 40-42	1	10	1	10	
$t_{p(CLKH-MWRL)}$	Propagation delay time, CLK ↑ to $\overline{MWR}$ low	35, 36, 39-42	10	35	10	35	

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.
‡ Interrupts are not granted during multicycle instructions.

host-independent mode (MSTR high)

switching characteristics over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted) (continued)†

propagation delay times (continued)

PARAMETER		FIGURE	SMJ34082A-28 MIN	SMJ34082A-28 MAX	SMJ34082A-30 MIN	SMJ34082A-30 MAX	UNIT
$t_{p(CLKH-WEH)}$	Propagation delay time, CLK ↑ to $\overline{WE}$ high	30, 33, 43, 44		10		10	ns
$t_{p(CLKH-WEL)}$	Propagation delay time, CLK ↑ to $\overline{WE}$ low	30, 33, 43, 44		30		30	ns

enable and disable times

PARAMETER		FIGURE	SMJ34082A-28 MIN	SMJ34082A-28 MAX	SMJ34082A-30 MIN	SMJ34082A-30 MAX	UNIT
$t_{en(CLKH-LADZX)}$	Enable time, CLK high to LAD enabled	29, 30	5		5		ns
$t_{en(LOEL-LADZX)}$	Enable time, $\overline{LOE}$ low to LAD enabled	33	2	17	2	17	ns
$t_{en(MAEL-MSAZX)}$	Enable time, $\overline{MAE}$ low to MSA enabled	41, 42	2	17	2	17	ns
$t_{en(MAEL-MSDZX)}$	Enable time, $\overline{MAE}$ low to MSD enabled	42	2	17	2	17	ns
$t_{dis(CLKH-LADZX)}$	Disable time, CLK high to LAD disabled‡	29, 30		25		25	ns
$t_{dis(LOEH-LADXZ)}$	Disable time, $\overline{LOE}$ high to LAD disabled	33	2	17	2	17	ns
$t_{dis(MAEH-MSAXZ)}$	Disable time, $\overline{MAE}$ high to MSA disabled	41, 42	2	17	2	17	ns
$t_{dis(MAEH-MSDXZ)}$	Disable time, $\overline{MAE}$ high to MSD disabled	42	2	17	2	17	ns

valid times

PARAMETER		FIGURE	SMJ34082A-28 MIN	SMJ34082A-28 MAX	SMJ34082A-30 MIN	SMJ34082A-30 MAX	UNIT
$t_{v(ATCH-LAD)}$	Valid time, LAD output data after $\overline{ALTCH}$ high	29, 30	2		2		ns
$t_{v(CLKH-MSA)}$	Valid time, MSA address valid after CLK high	35-47	3		3		ns
$t_{v(MWRH-MSD)}$	Valid time, MSD data valid after $\overline{MWR}$ high	35, 36, 40-42	1		1		ns
$t_{v(MWRH-MSA)}$	Valid time, MSA address valid after $\overline{MWR}$ high	35, 36, 40-41	1		1		ns
$t_{v(WEH-LAD)}$	Valid time, LAD data valid after $\overline{WE}$	30, 33, 43, 44	2		2		ns

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.
‡ Valid only for last write in series. The LAD bus is not placed in high-impedance state between consecutive outputs.

host-independent mode (MSTR high)

timing requirements over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted)†

clock period and pulse duration

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{c(CLK)}$	Clock period time, CLK ($1/f_{clock}$) with PIPES1 low	28-31, 33-48	78		73		ns
$t_{w(ATCH)}$	Pulse duration, $\overline{\text{ALTCH}}$ high	30	5		5		ns
$t_{w(CASH)}$	Pulse duration, $\overline{\text{CAS}}$ high	29, 31, 32, 35, 36	5		5		
$t_{w(CLKH)}$	Pulse duration, CLK high	28	17		17		
$t_{w(CLKL)}$	Pulse duration, CLK low	28	22		22		
$t_{w(DCSH)}$	Pulse duration, DS/$\overline{\text{CS}}$ high	36, 40, 44	5		5		
$t_{w(ITRL)}$	Pulse duration, $\overline{\text{INTR}}$ low	34, 47	30		30		
$t_{w(MCEH)}$	Pulse duration, $\overline{\text{MCE}}$ high	36, 38, 44-46	5		5		
$t_{w(MOEH)}$	Pulse duration, $\overline{\text{MOE}}$ high	37, 38, 43-46	6		6		
$t_{w(MWRH)}$	Pulse duration, $\overline{\text{MWR}}$ high	35, 36, 40	6		6		
$t_{w(RSTL)}$	Pulse duration, $\overline{\text{RESET}}$ low	34	40		40		
$t_{w(WEH)}$	Pulse duration, $\overline{\text{WE}}$ high	30, 33, 43, 44	5		5		

transition time

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{t(CLK)}$	Transition time, CLK	28		15		15	ns

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.

host-independent mode (MSTR high)

timing requirements over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted) (continued)†

setup and hold times

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{su(CC-CLKH)}$	Setup time, CC before CLK high	45	7		7		ns
$t_{su(LADV-CLKL)}$	Setup time, LAD data valid before CLK low for immediate data input‡	32	15		15		
$t_{su(ITRL-CLKH)}$	Setup time, $\overline{INTR}$ before CLK high	47	20		15		
$t_{su(LAD-CLKH)}$	Setup time, LAD input data valid before CLK high	29, 31, 34-36	15		13		
$t_{su(LRD-CLKH)}$	Setup time, LRDY before CLK high	48	20		15		
$t_{su(MSD-CLKH)}$	Setup time, MSD data valid before CLK high	37, 38, 43-47	13		13		
$t_{su(RDYV-CLKH)}$	Setup time, RDY valid before CLK high	48	20		12		
$t_{su(RSTH-CLKH)}$	Setup time, $\overline{RESET}$ high before CLK high	34	45		45		
$t_{su(RSTL-ITRL)}$	Setup time, $\overline{RESET}$ low before $\overline{INTR}$ low for bootstrap loader	34	20		20		ns
$t_{h(CLKH-CC)}$	Hold time, CC after CLK high	45	3		3		
$t_{h(CLKH-ITR)}$	Hold time, $\overline{INTR}$ after CLK high	47	3		3		
$t_{h(CLKH-LAD)}$	Hold time, LAD input data valid after CLK high	29, 31, 35, 36	5		5		
$t_{h(CLKH-LRD)}$	Hold time, LRDY after CLK high	48	0		0		
$t_{h(CLKH-MSD)}$	Hold time, MSD input data valid after CLK high	37, 38, 43-47	4		4		
$t_{h(CLKH-RDY)}$	Hold time, RDY after CLK high	48	0		0		
$t_{h(CLKL-LAD)}$	Hold time, LAD data after CLK low for immediate data input‡	32	5		5		
$t_{h(ITRL-RSTH)}$	Hold time, $\overline{RESET}$ low after $\overline{INTR}$ low for bootstrap loader	34	15		15		

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.

‡ This mode permits data input that does not meet the minimum setup before CLK high. The clock period for this mode must be extended according to the equation:

$$\text{Adjusted clock period} = \text{Normal clock period} + \text{Data delay} + 5\ \text{ns}$$

The data delay is the delay from CLK high to valid data. This mode may not be used to input data for divides or square roots.

host-independent mode (MSTR high)

timing requirements over recommended ranges of supply voltage and operating free-air (minimum) and case (maximum) temperature range (unless otherwise noted) (continued)†

delay times

PARAMETER		FIGURE	SMJ34082A-28		SMJ34082A-30		UNIT
			MIN	MAX	MIN	MAX	
$t_{d(ATCH-CASL)}$	Delay time, $\overline{ALTCH}$ high to $\overline{CAS}$ low	29	5		5		
$t_{d(ATCH-WEL)}$	Delay time, $\overline{ALTCH}$ high to $\overline{WE}$ low	30	3		3		
$t_{d(CASH-ATCL)}$	Delay time, $\overline{CAS}$ high to $\overline{ALTCH}$ low	29	3		3		
$t_{d(CASH-WEL)}$	Delay time, $\overline{CAS}$ high to $\overline{WE}$ low	33	3		3		
$t_{d(COIL-ATCL)}$	Delay time, $\overline{COINT}$ low to $\overline{ALTCH}$ low	29, 30	0		0		
$t_{d(COIL-CASL)}$	Delay time, $\overline{COINT}$ low to $\overline{CAS}$ low	31, 35, 36	0		0		
$t_{d(COIL-WEL)}$	Delay time, $\overline{COINT}$ low to $\overline{WE}$ low	33	0		0		
$t_{d(DCSH-MCEL)MH}$	Delay time, DS/$\overline{CS}$ high to $\overline{MCE}$ low with MEMCFG high	38, 42	5		5		
$t_{d(DCSH-MWRL)}$	Delay time, DS/$\overline{CS}$ high to $\overline{MWR}$ low	35, 39	4		4		
$t_{d(MCEH-DCSL)MH}$	Delay time, $\overline{MCE}$ high to DC/$\overline{CS}$ low with MEMCFG high	40	5		5		
$t_{d(MCEH-MWRL)}$	Delay time, $\overline{MCE}$ high to $\overline{MWR}$ low	36	5		5		ns
$t_{d(MOEH-MWRL)}$	Delay time, $\overline{MOE}$ high to $\overline{MWR}$ low	39	5		5		
$t_{d(MSAV-MWRL)}$	Delay time, MSA valid to $\overline{MWR}$ low	35, 36, 40-42	4		4		
$t_{d(MSDZ-MOEL)}$	Delay time, MSD disabled to $\overline{MOE}$ low	41, 42	2		2		
$t_{d(MWRH-MCEL)MH}$	Delay time, $\overline{MWR}$ high to $\overline{MCE}$ low with MEMCFG high	36	5		5		
$t_{d(MWRH-MOEL)}$	Delay time, $\overline{MWR}$ high to $\overline{MOE}$ low	41, 42	5		5		
$t_{d(MWRH-MSDXZ)}$	Delay time, $\overline{MWR}$ high to MSD disabled	42	1	12	1	9	
$t_{d(MWRL-MSDZX)}$	Delay time, $\overline{MWR}$ low to MSD enabled	41, 42	1	13	1	13	
$t_{d(WEH-ATCL)}$	Delay time, $\overline{WE}$ high to $\overline{ALTCH}$ low	29	3		3		
$t_{d(WEH-CASL)}$	Delay time, $\overline{WE}$ high to $\overline{CAS}$ low	31	3		3		

† See Parameter Measurement Information for load circuit, voltage waveforms, and timing diagrams. The device parameters are measured for PIPES2 high and PIPES1 low. No other pipeline settings are specified.

EXPLANATION OF LETTER SYMBOLS

This data sheet uses a type of letter symbol based on JEDEC Std-100 and IEC Publication 748-2, 1985, to describe time intervals. The format is:

$$t_{A(BC\text{-}DE)F}$$

Where:

Subscript A indicates the type of dynamic parameter being represented. One of the following is used:

Switching Characteristics:
- p = Propagation delay time
- en = Enable time
- dis = Disable time

Timing Requirements:
- c = Clock period
- w = Pulse duration
- t = Transition time
- d = Delay time
- su = Setup time
- h = Hold time
- v = Valid time

Subscript B indicates the name of the signal or terminal for which a change of state or level (or establishment of a state or level) constitutes a signal event assumed to occur first, that is, at the beginning of the time interval.

Subscript C indicates the direction of the transistion and/or the final state or level of the signal represented by B. One or two of the following are used:

- H = High or transition to high
- L = Low or transition to low
- V = A valid steady-state level
- X = Unknown, changing, or "don't care" level
- Z = High-impedance (off) state

Subscript D indicates the name of the signal or terminal for which a change of state or level (or establishment of a state or level) constitutes a signal event assumed to occur last, that is, at the end of the time interval.

Subscript E indicates the direction of the transition and/or the final state or level of the signal represented by D. One or two of the symbols described in *Subscript C* are used.

Subscript F indicates additional information such as mode of operation, test conditions, etc.

The hyphen between the C and D subscripts is omitted when no confusion is likely to occur. For these letter symbols on this data sheet, the signal names are further abbreviated as follows:

SIGNAL NAME	B & D SUBSCRIPT	SIGNAL NAME	B & D SUBSCRIPT	SIGNAL NAME	B & D SUBSCRIPT	SIGNAL NAME	B & D SUBSCRIPT	SIGNAL NAME	B & D SUBSCRIPT
$\overline{\text{ALTCH}}$	ATC	CORDY	COR	LCLK2	LC2	MSA(0:15)	MSA	TCK	TCK
BUSFLT	BFT	DC/$\overline{\text{CS}}$	DCS	$\overline{\text{LOE}}$	LOE	MSD(0:31)	MSD	TDI	TDI
$\overline{\text{CAS}}$	CAS	EC(0:1)	EC	LRDY	LRD	$\overline{\text{MWR}}$	MWR	TDO	TDO
CC	CC	INTG	INT	$\overline{\text{MAE}}$	MAE	$\overline{\text{RAS}}$	RAS	TMS	TMS
CID(0:2)	CID	$\overline{\text{INTR}}$	ITR	MSTR	MST	RDY	RDY	V_{CC}/V_{SS}	—
CLK	CLK	LAD(0:31)	LAD	$\overline{\text{MCE}}$	MCE	$\overline{\text{RESET}}$	RST	$\overline{\text{WE}}$	WE
$\overline{\text{COINT}}$	COI	LCLK1	LC1	$\overline{\text{MOE}}$	MOE	SF	SF	MEMCFG	M

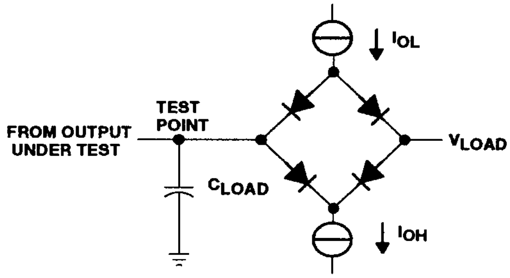
PARAMETER MEASUREMENT INFORMATION

LOAD CIRCUIT PARAMETERS

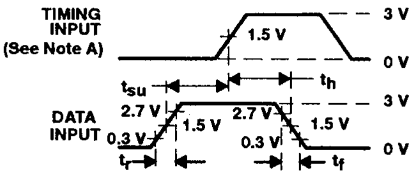
TIMING PARAMETERS		C_{LOAD}† (pF)	I_{OL} (mA)	I_{OH} (mA)	V_{LOAD} (V)
t_{en}	t_{PZH}	65	8	−8	0
	t_{PZL}				3
t_{dis}	t_{PHZ}	65	8	−8	1.5
	t_{PLZ}				
t_p		65	8	−8	‡

† C_{LOAD} includes the typical load circuit and distributed capacitance.

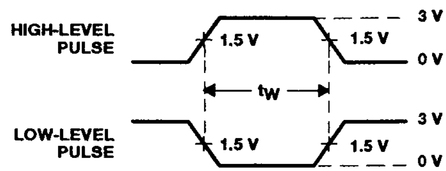
‡ $\frac{V_{LOAD} - V_{OL}}{I_{OL}} = 50\ \Omega$, where $V_{OL} = 0.6$ V, $I_{OL} = 8$ mA.



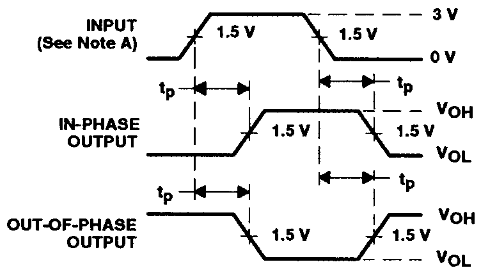
LOAD CIRCUIT



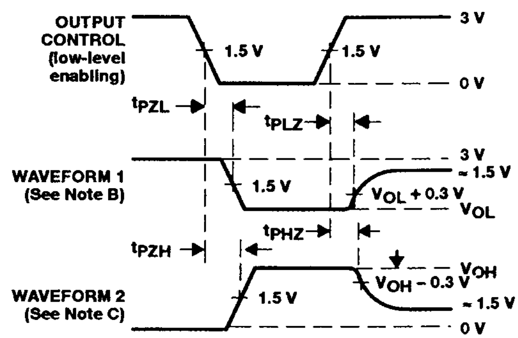
VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES
INPUT RISE AND FALL TIMES



VOLTAGE WAVEFORMS
PULSE DURATION



VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

NOTES: A. Phase relationships between waveforms were chosen arbitrarily. All input pulses are supplied by pulse generators having the following characteristics: PRR = 1 MHz, $Z_O = 50\ \Omega$, $t_r \le 6$ ns, $t_f \le 6$ ns.

B. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control.

C. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control. For t_{PLZ} and t_{PHZ}, V_{OL} and V_{OH} are measured values.

Figure 9

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

PARAMETER MEASUREMENT INFORMATION

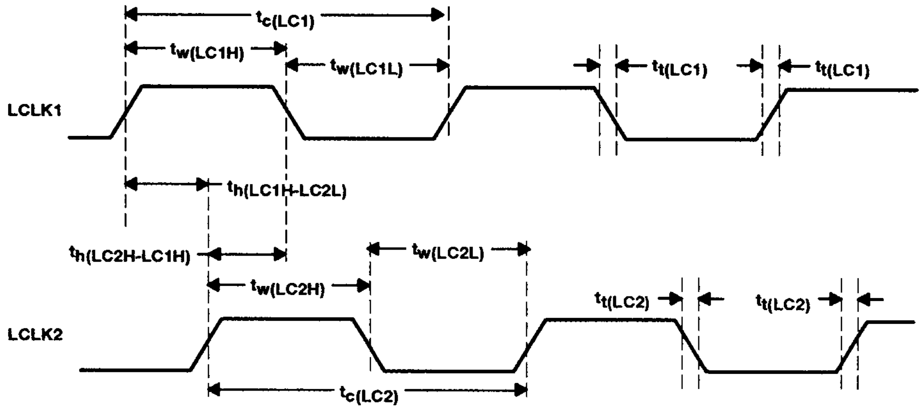
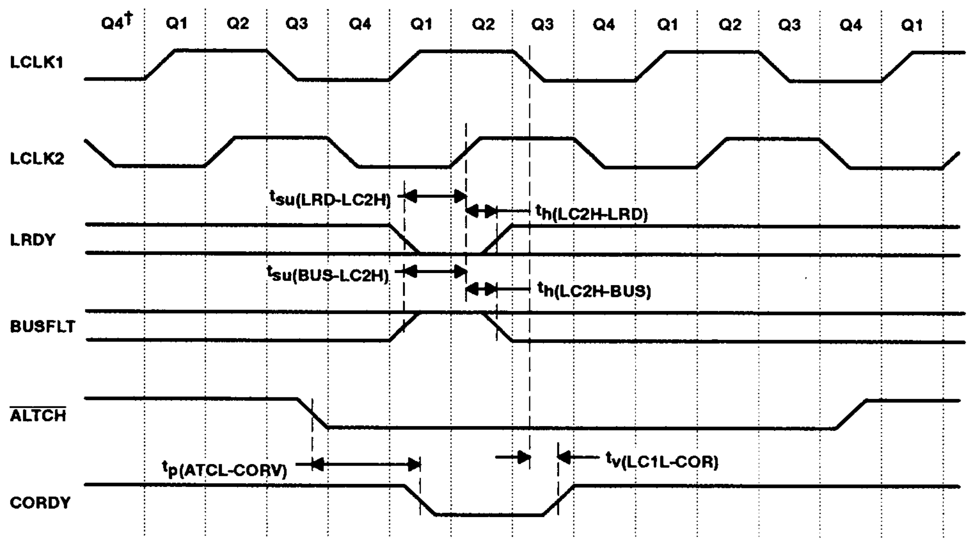


Figure 10. Coprocessor Mode, Input Clocks



† Q1, Q2, Q3, and Q4 represent the first, second, third, and fourth quarter clocks, respectively, of the LCLK1 clock period.

Figure 11. Coprocessor Mode, Bus Control Signals

PARAMETER MEASUREMENT INFORMATION

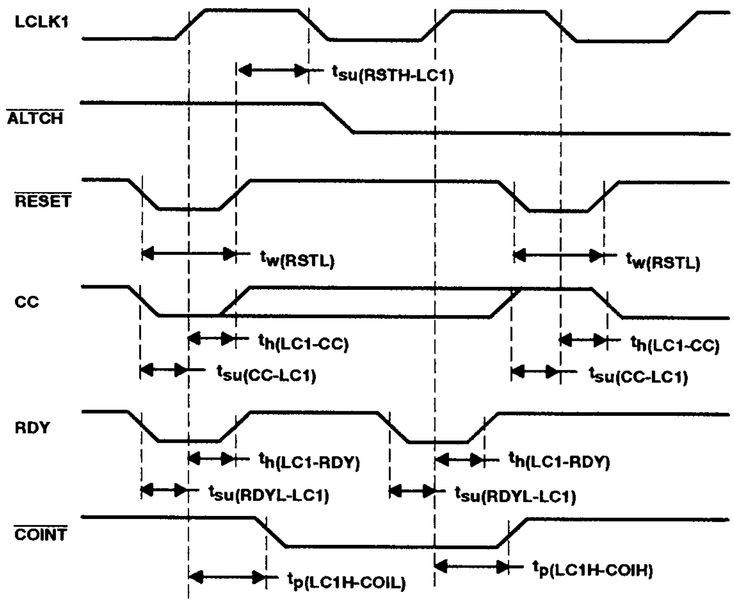
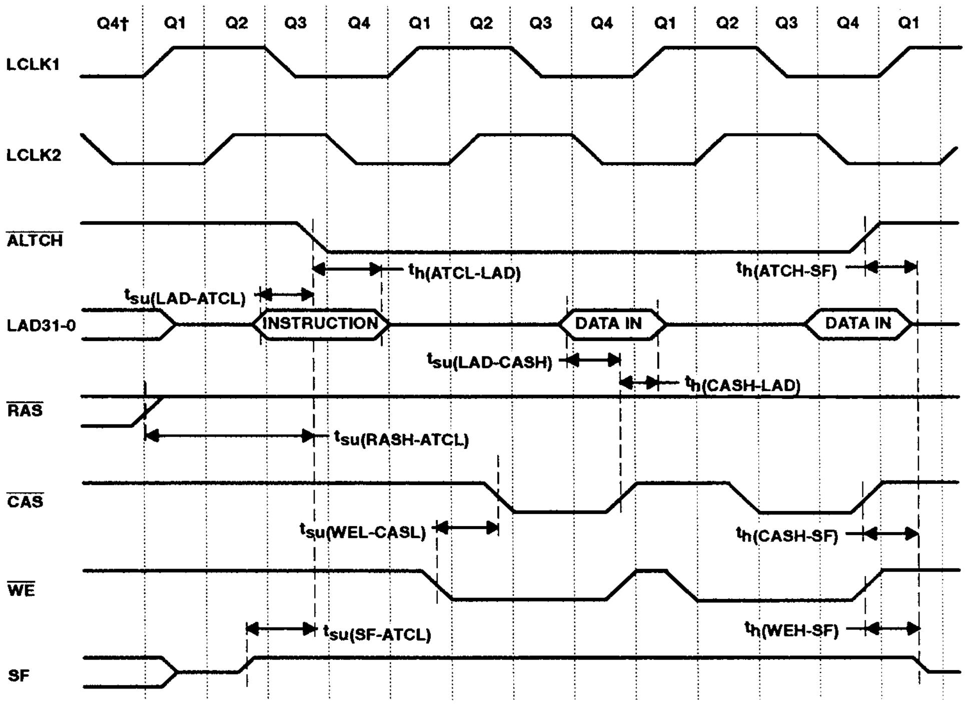


Figure 12. Coprocessor Mode, Control Signals

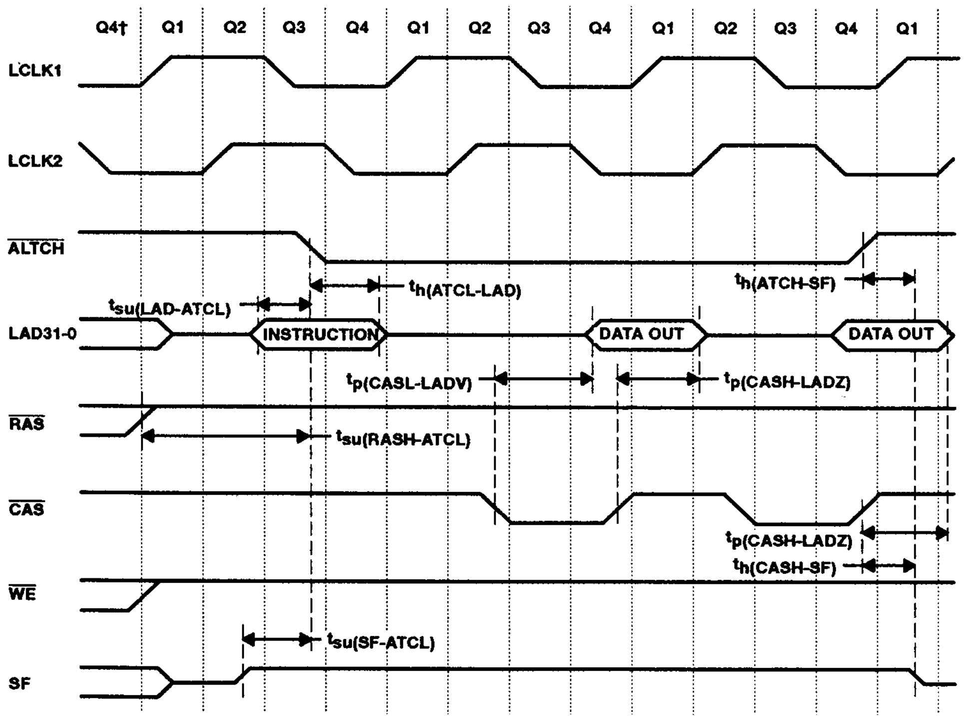
PARAMETER MEASUREMENT INFORMATION



† Q1, Q2, Q3, and Q4 represent the first, second, third, and fourth quarter clocks, respectively, of the LCLK1 clock period.

Figure 13. Coprocessor Mode, SMJ34020 GSP to SMJ34082

PARAMETER MEASUREMENT INFORMATION

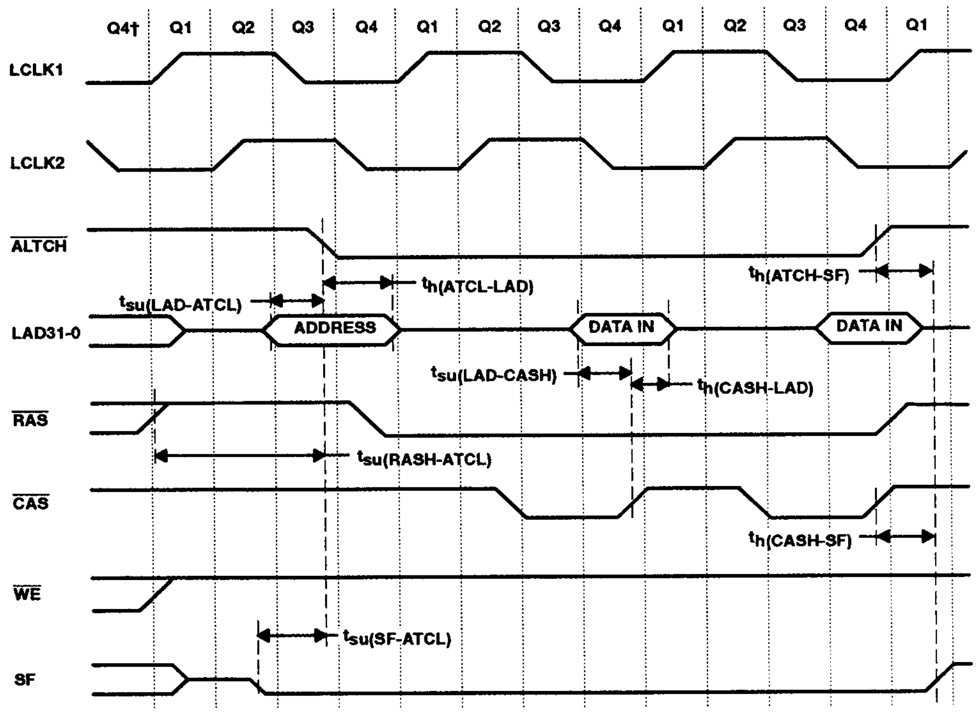


† Q1, Q2, Q3, and Q4 represent the first, second, third, and fourth quarter clocks, respectively, of the LCLK1 clock period.

Figure 14. Coprocessor Mode, SMJ34082A to SMJ34020 GSP Including Coprocessor Internal Cycle

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

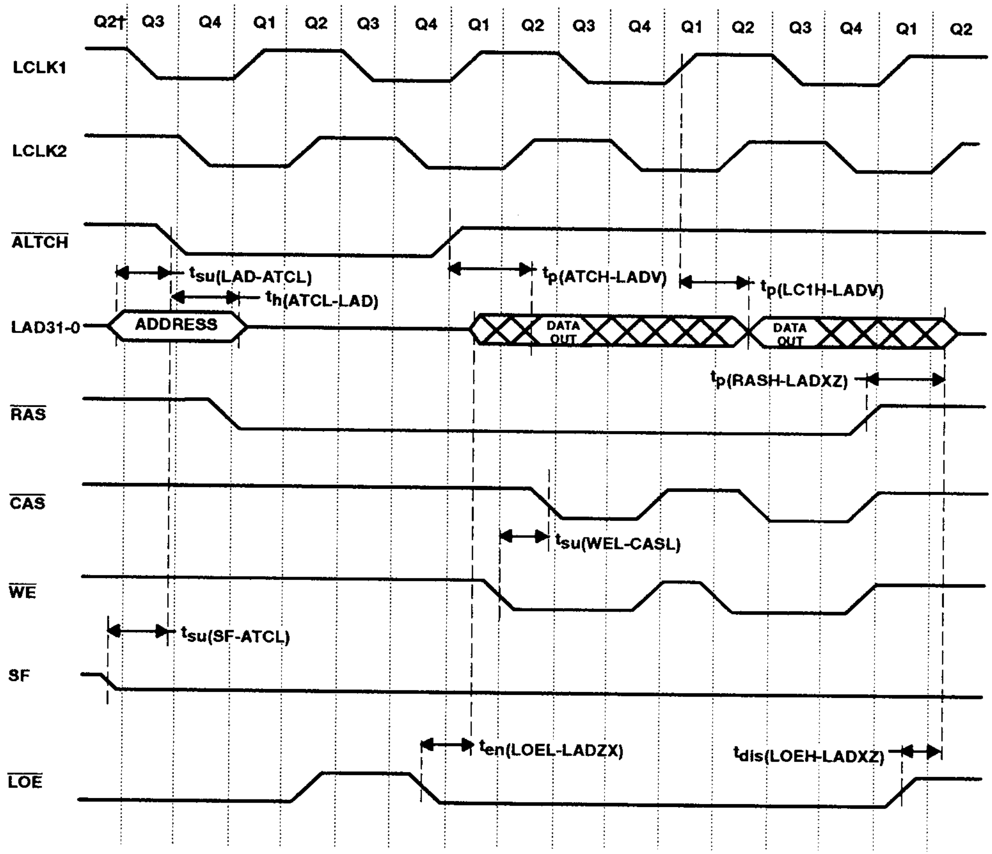
PARAMETER MEASUREMENT INFORMATION



† Q1, Q2, Q3, and Q4 represent the first, second, third, and fourth quarter clocks, respectively, of the LCLK1 clock period.

Figure 15. Coprocessor Mode, DRAM/VRAM Memory to SMJ34082

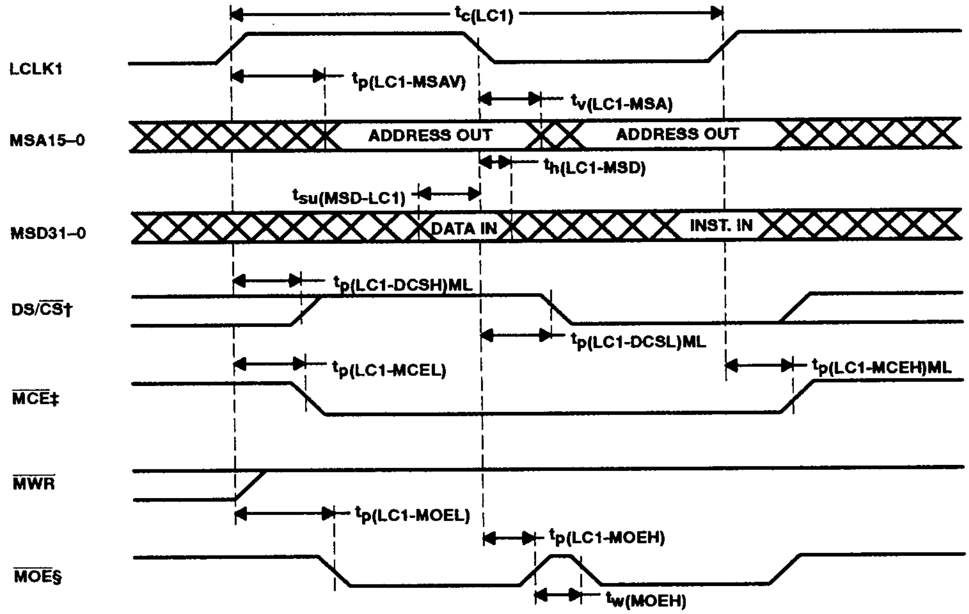
PARAMETER MEASUREMENT INFORMATION



† Q1, Q2, Q3, and Q4 represent the first, second, third, and fourth quarter clocks, respectively, of the LCLK1 clock period.

Figure 16. Coprocessor Mode, SMJ34082A to DRAM/VRAM Memory

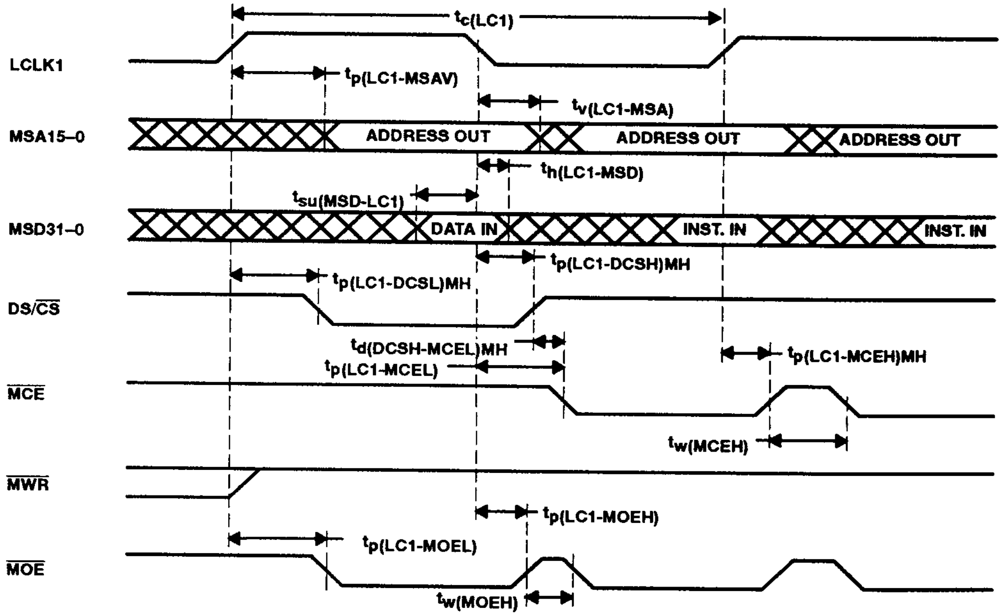
PARAMETER MEASUREMENT INFORMATION



† The setting of DS/$\overline{CS}$ determines whether the value on the MSD bus is an instruction or data.
‡ $\overline{MCE}$ dos not toggle at each clock edge.
§ $\overline{MOE}$ goes high at each clock edge.
NOTE: This example shows a data read followed by an instruction read.

Figure 17. Coprocessor Mode MSD Bus Timing, Memory to SMJ34082A with MEMCFG Low

PARAMETER MEASUREMENT INFORMATION

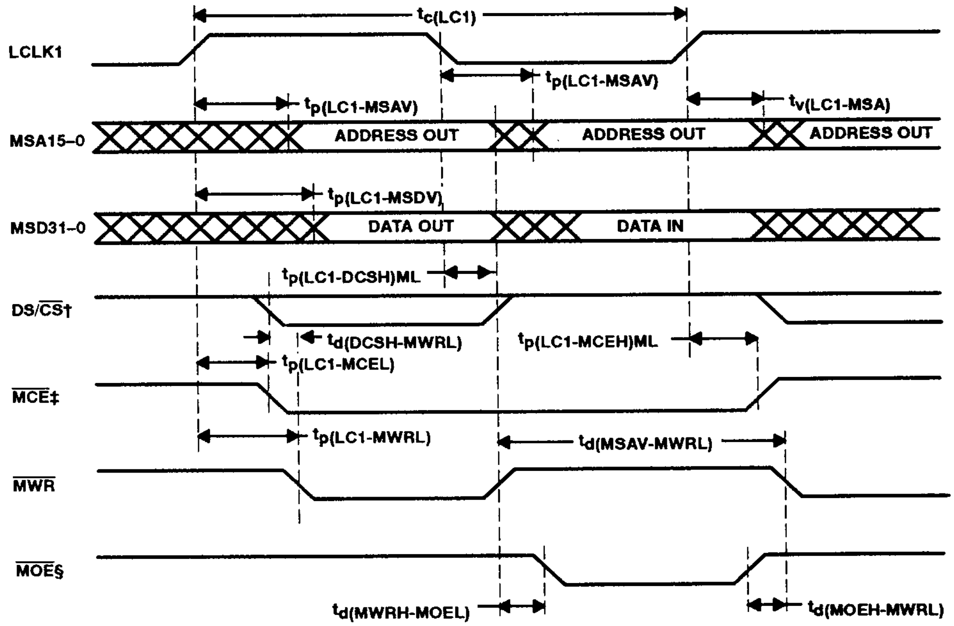


NOTE: This example shows a data read followed by an instruction read followed by an instruction read. This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register. When MEMCFG is high, DS/$\overline{CS}$ and $\overline{MCE}$ rise after every clock edge. In this mode, DS/$\overline{CS}$ and $\overline{MCE}$ may not both be active (low) at the same time.

Figure 18. Coprocessor Mode MSD Bus Timing, Memory to SMJ34082A with MEMCFG High

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

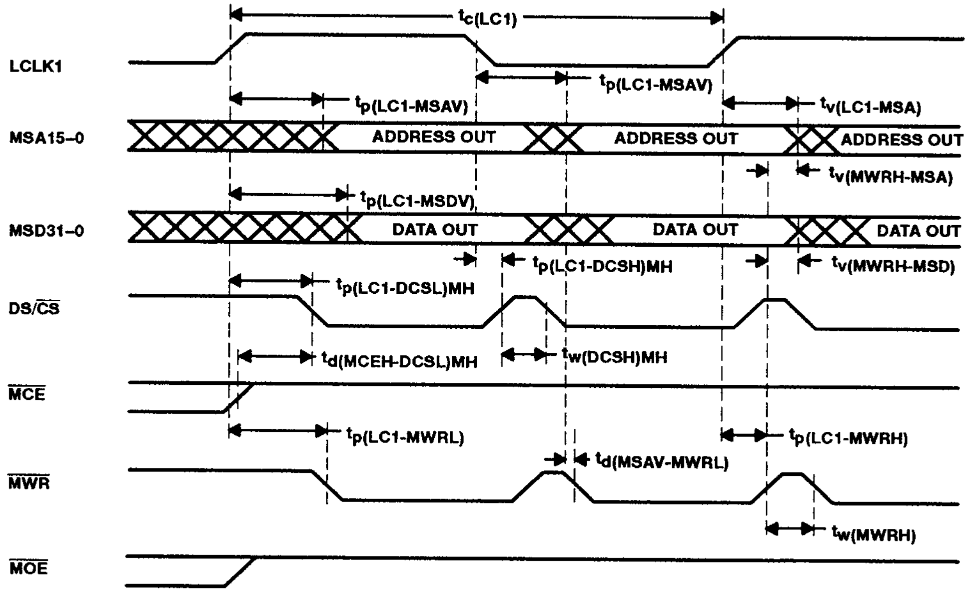
PARAMETER MEASUREMENT INFORMATION



† The setting of DS/$\overline{CS}$ determines whether the value on the MSD bus is an instruction or data.
‡ $\overline{MCE}$ does not toggle at each clock edge.
§ $\overline{MWR}$ goes high at each clock edge.
NOTE: This example shows a data write followed by a code read.

Figure 19. Coprocessor Mode MSD Bus Timing, SMJ34082A to Memory with MEMCFG Low

PARAMETER MEASUREMENT INFORMATION

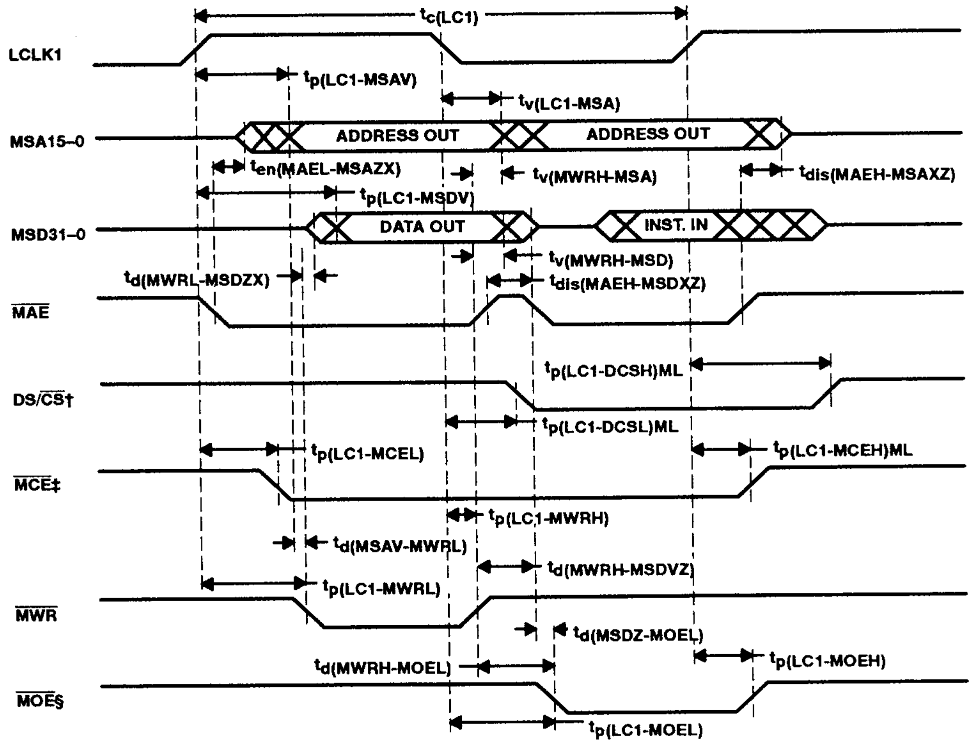


NOTE: This example shows multiple data writes. Timing for multiple code writes would be similar. This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register. When MEMCFG is high, DS/$\overline{CS}$ and $\overline{MCE}$ rise after every clock edge. In this mode, DS/$\overline{CS}$ and $\overline{MCE}$ may not both be active (low) at the same time.

Figure 20. Coprocessor Mode MSD Bus Timing, SMJ34082A to Memory with MEMCFG High

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

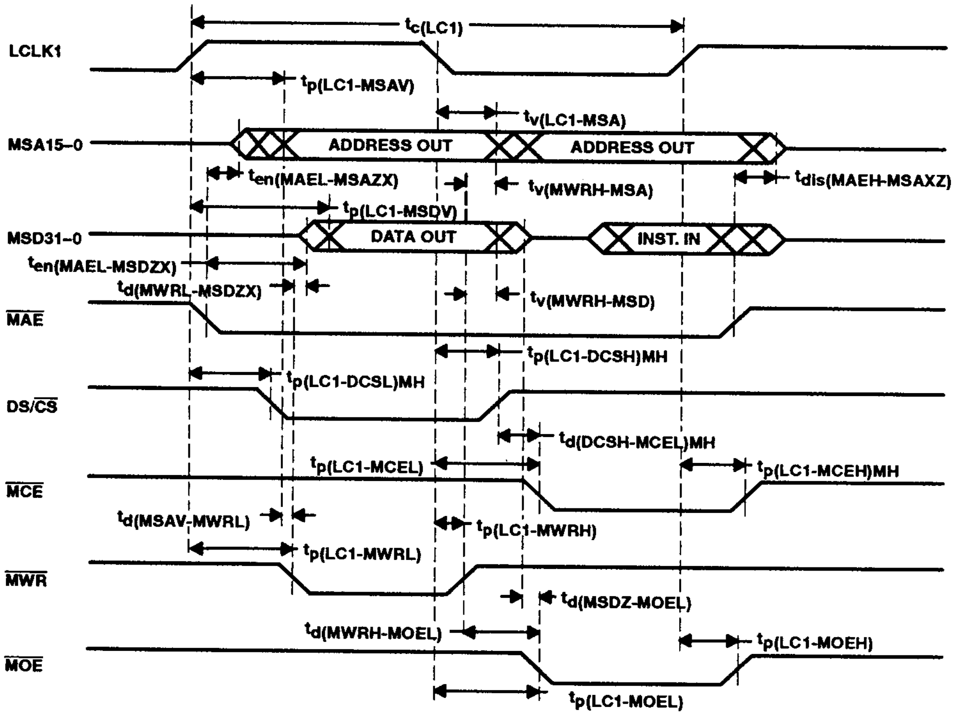
PARAMETER MEASUREMENT INFORMATION



† The setting of DS/$\overline{CS}$ determines whether the value on the MSD bus is an instruction or data.
‡ $\overline{MCE}$ does not toggle at each clock edge.
§ $\overline{MOE}$ goes high at each clock edge.
NOTE: This example shows a data write followed by an instruction read.

Figure 21. Coprocessor Mode, MSD Enable/Disable Timing with MEMCFG Low

PARAMETER MEASUREMENT INFORMATION

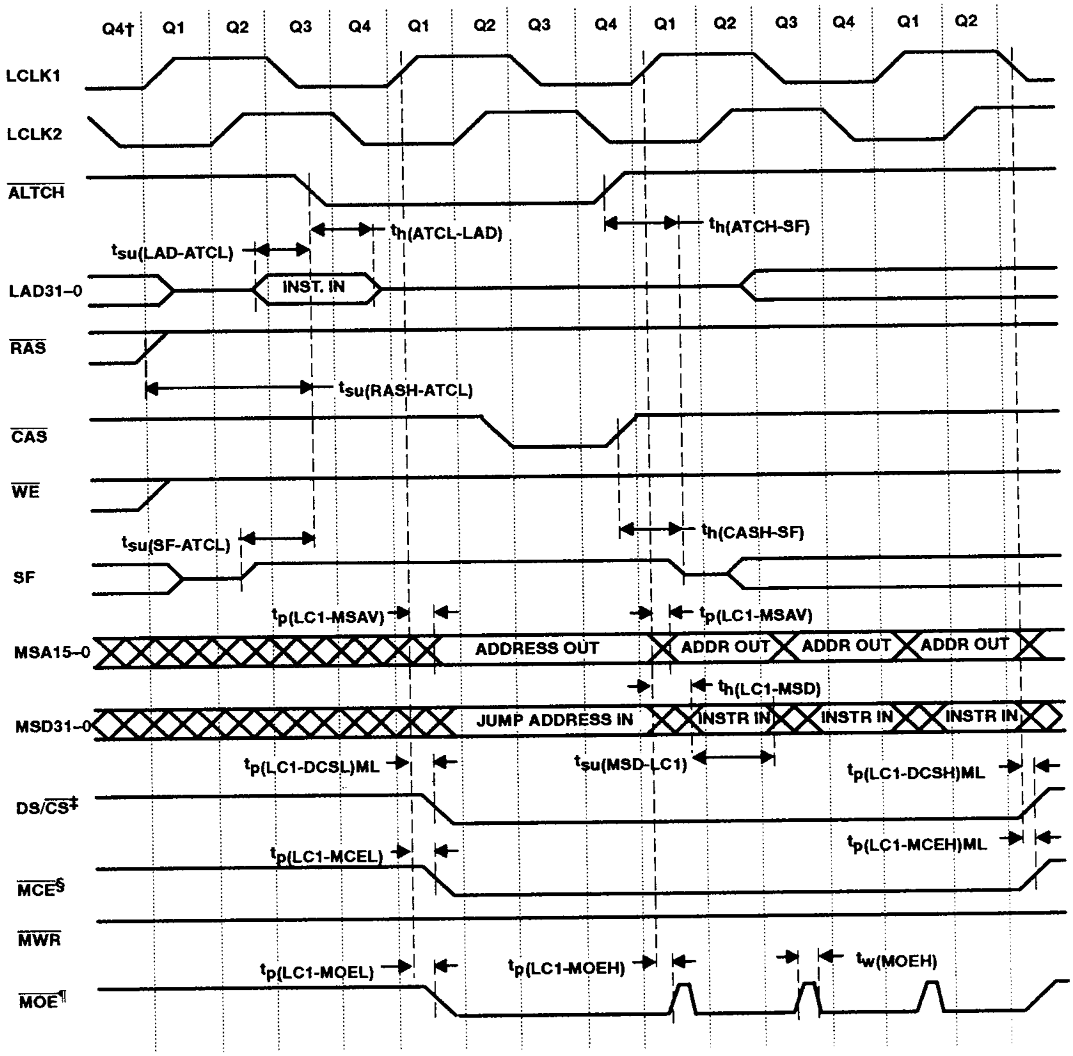


NOTE: This example shows a data write followed by an instruction read. Timing for multiple code writes would be similar. This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register. When MEMCFG is high, DS/$\overline{CS}$ and $\overline{MCE}$ rise after every clock edge. In this mode, DS/$\overline{CS}$ and $\overline{MCE}$ may not both be active (low) at the same time.

Figure 22. Coprocessor Mode, MSD Bus Enable/Disable Timing with MEMCFG High

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

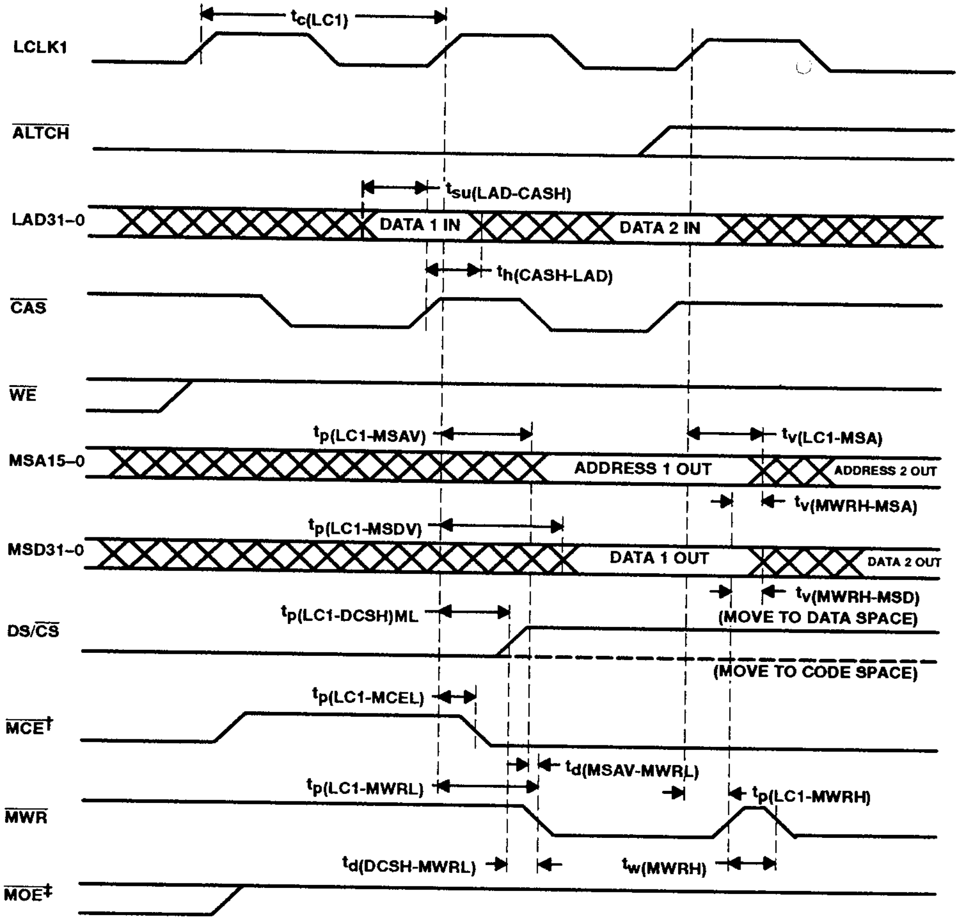
PARAMETER MEASUREMENT INFORMATION



† Q1, Q2, Q3, and Q4 represent the first, second, third, and fourth quarter clocks, respectively, of the LCLK1 clock period.
‡ The setting of DS/$\overline{CS}$ determines whether the value on the MSD bus in an instruction or data.
§ $\overline{MCE}$ does not toggle at each rising clock edge.
¶ $\overline{MOE}$ goes hiigh at each rising clock edge.

Figure 23. Coprocessor Mode, Jump to External Memory Subroutine with MEMCFG Low

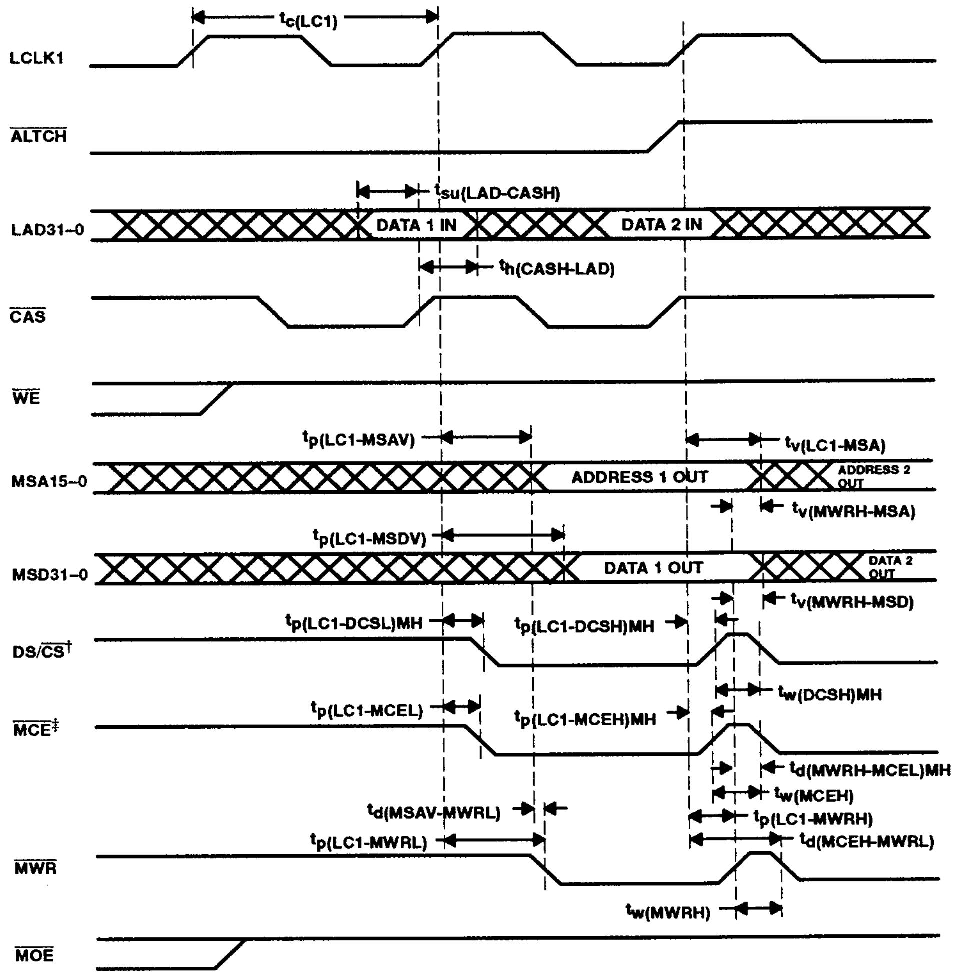
PARAMETER MEASUREMENT INFORMATION



† MCE does not toggle at each clock edge.
‡ MOE goes high at each clock edge.

Figure 24. Coprocessor Mode, LAD to MSD Bus Transfer Timing with MEMCFG Low

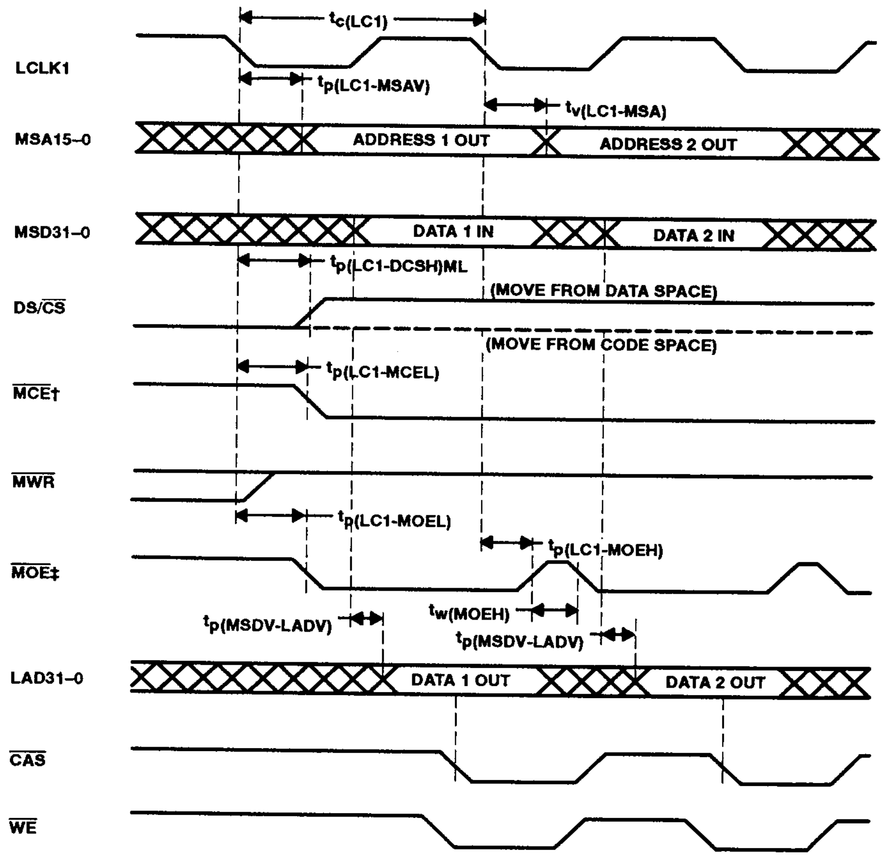
PARAMETER MEASUREMENT INFORMATION



† DS/CS valid for moves to data space; MCE valid for moves to code space. Only one of these would be valid for each move instruction.
‡ This option for using DS/CS as data space chip enable and MCE as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register.

Figure 25. Coprocessor Mode, LAD to MSD Bus Transfer Timing with MEMCFG High

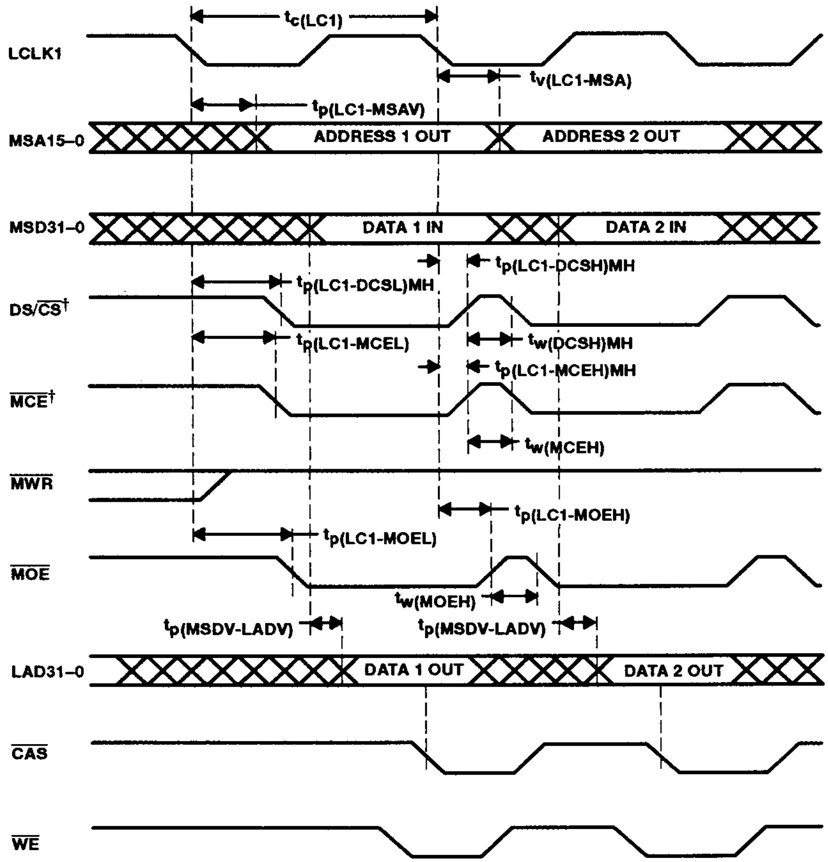
PARAMETER MEASUREMENT INFORMATION



† $\overline{MCE}$ dos not toggle at each clock edge.
‡ $\overline{MOE}$ goes high at each clock edge.

Figure 26. Coprocessor Mode, MSD to LAD Bus Transfer Timing with MEMCFG Low

PARAMETER MEASUREMENT INFORMATION



† DS/$\overline{CS}$ valid for moves to data space; $\overline{MCE}$ valid for moves to code space. Only one would be valid for each move instruction.

NOTE: This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is involved by setting the MEMCFG bit high in the configuration register.

Figure 27. Coprocessor Mode, MSD to LAD Bus Transfer Timing with MEMCFG High

PARAMETER MEASUREMENT INFORMATION

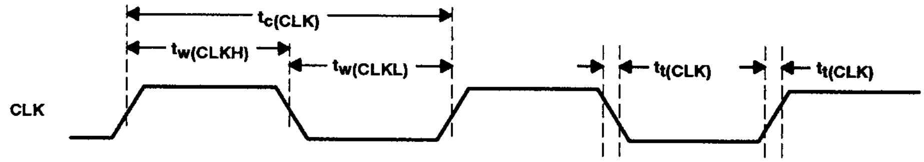
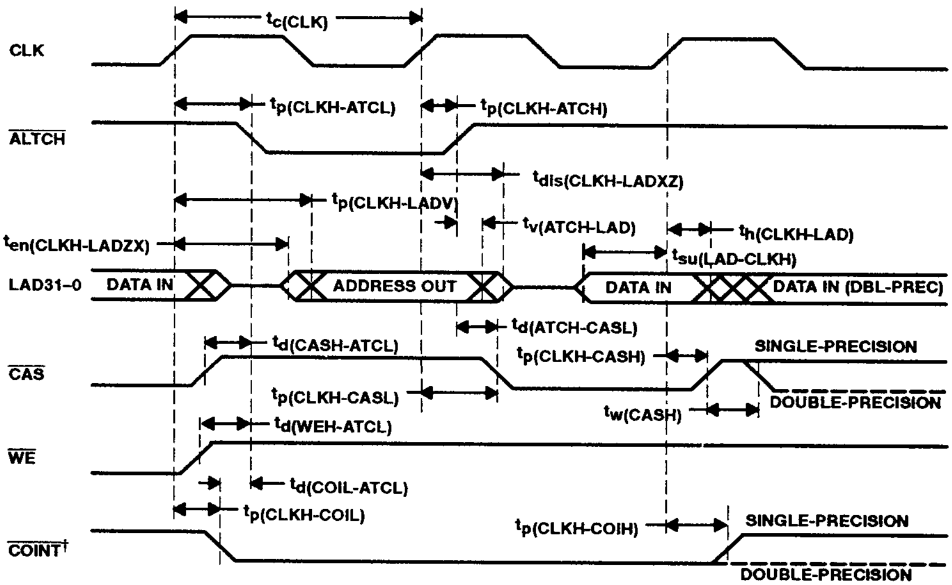


Figure 28. Host-Independent Mode, Input Clock



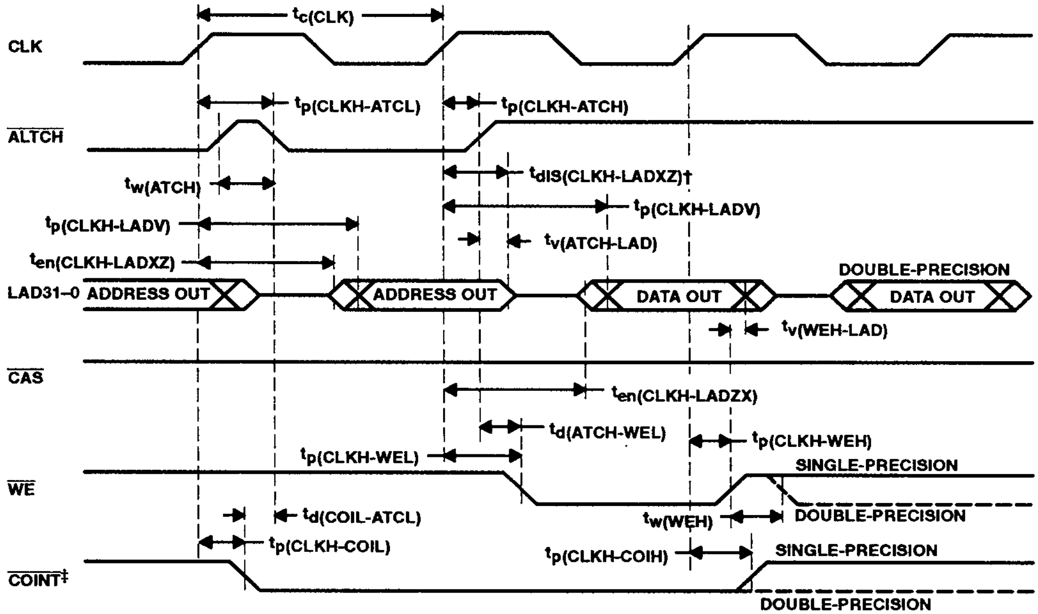
† COINT timing is for LADCFG high only. When the LADCFG bit is set high in the configuratin register, COINT is controlled by bit 1 of the LAD move instruction instead of the set mask instruction.

NOTE: This timing diagram assumes an external address latch to store address for external memory reads. Data input hold time on the latch is zero; data (or address) output hold time is nonzero.

Figure 29. Host-Independent Mode, LAD Bus Timing for Memory to SMJ34082A

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

PARAMETER MEASUREMENT INFORMATION

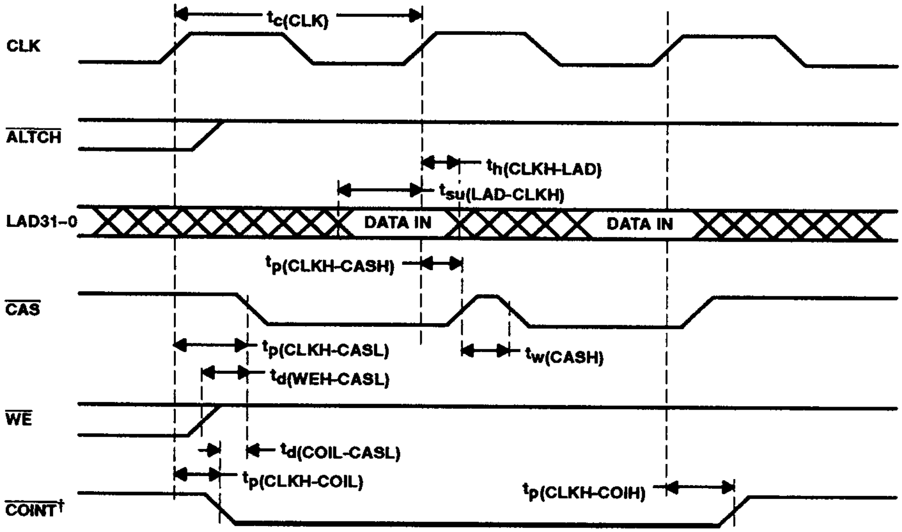


† Valid only for last write in series. The LAD bus is not placed in high-impedance state between consecutive outputs.
‡ COINT timing is for LADCFG high only. When the LADCFG bit is set high in the configuration register, COINT is controlled by bit 1 of the LAD move instruction instead of the set mask instruction.

NOTE: This timing diagram assumes an external address latch to store address for external memory reads. Data input hold time is zero. Data (or address) output hold time is nonzero. Valid only for last write in series. The LAD bus is not placed in high impedance between consecutive outputs.

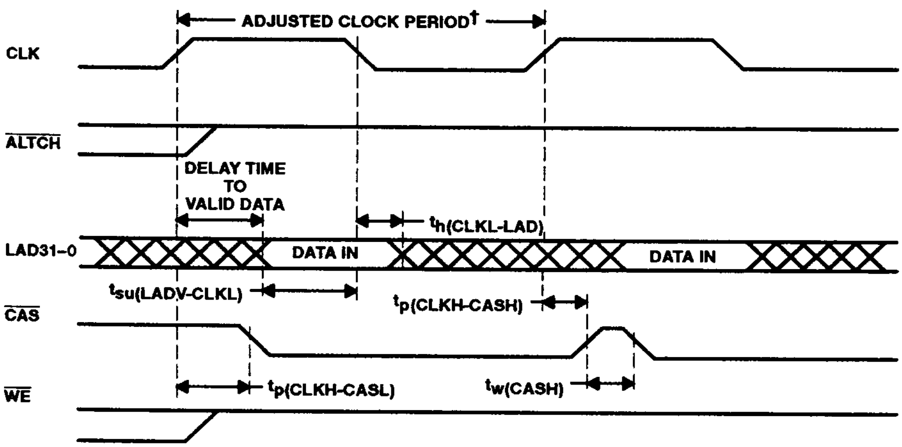
Figure 30. Host-Independent Mode, LAD Bus Timing for SMJ34082A to Memory

PARAMETER MEASUREMENT INFORMATION



† COINT timing is for LADCFG high only. When the LADCFG bit is set high in the configuration register, COINT is controlled by bit 1 of the LAD move instruction instead of the set mask instruction.

Figure 31. Host-Independent Mode, LAD Bus Timing Input to SMJ34082A

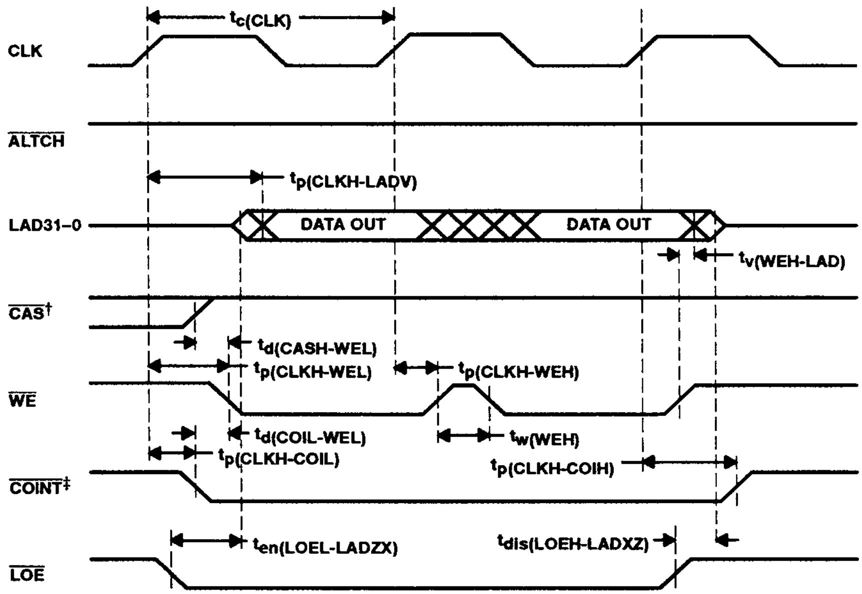


† This mode permits data input which does not meet the minimum setup before CLK high. For immediate data input, CLK must be high for more than 20 ns. This input mode cannot be used to input data for divides and square roots.

Adjusted clock period = Normal clock period + Data delay + 5 ns

Figure 32. Host-Independent Mode, LAD Bus Timing Input of Immediate Data to SMJ34082A

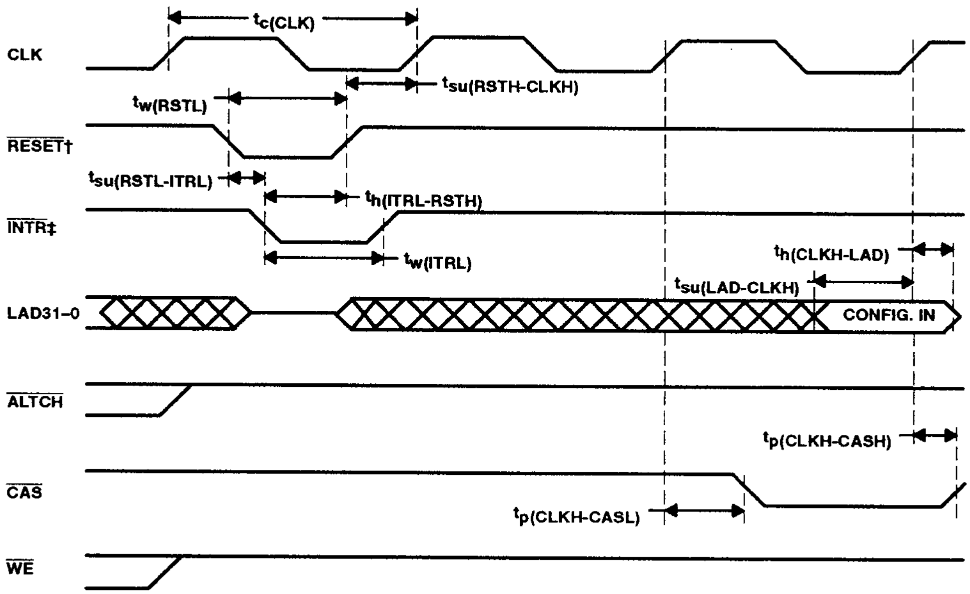
PARAMETER MEASUREMENT INFORMATION



† When the LADCFG bit is high, $\overline{\text{LOE}}$ high places $\overline{\text{CAS}}$ and $\overline{\text{WE}}$ (as well as the LAD bus) in high impedance.
‡ Valid only for LADCFG high. When the LADCFG bit is high in the configuration register, $\overline{\text{COINT}}$ is controlled by bit 1 of the LAD move instruction instead of the set mask instruction.
NOTE: If the instruction writes the result of an FPU operation to a register and outputs the result to the LAD bus, in the same cycle, the minimum clock period must be extended.

Figure 33. Host-Independent Mode, LAD Bus Timing Output from SMJ34082A

PARAMETER MEASUREMENT INFORMATION



† RESET is level sensitive. When RESET is set low, both LAD and MSD buses are placed in high-impedance state. When RESET is released, the sequencer forces a jump to address 0. If INTR goes low while RESET is low, the loader moves 64 words through to the external memory on MSD. Timing for the LAD to MSD move is shown in a later diagram, with the exception that the first word on LAD loads the configuration register and does not pass to the MSD bus.

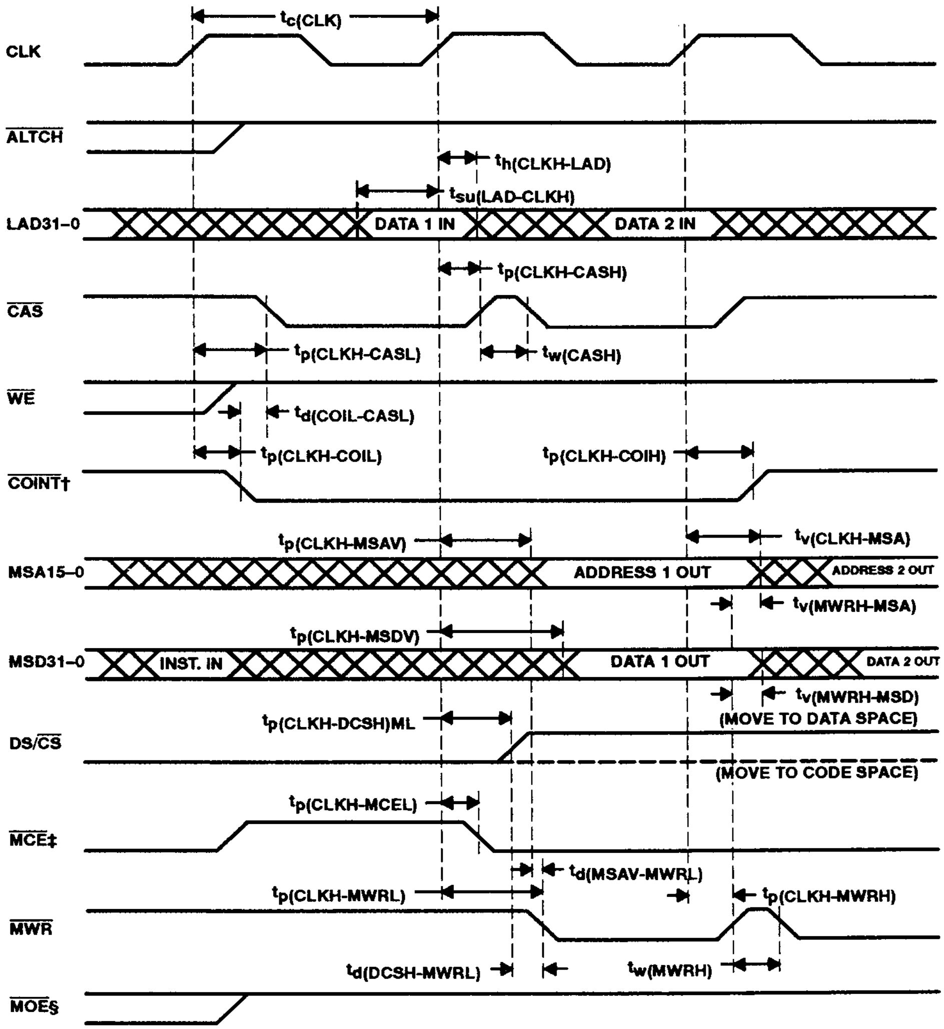
‡ INTR may be low one or more cycles after RESET goes low. RESET is held low, and then INTR is taken low. The bootstrap loader starts when RESET is set high, which may involve a delay of one or more cycles after INTR goes low.

NOTE: When the bootstrap loader is invoked, the first data word input on the LAD bus should be the configuration register settings, which will be written into the configuration register. This allows the user to select the MEMCFG setting, for reading or writing memory on the MSD port, as well as the LADCFG setting for the LAD bus interface.

Figure 34. Host-Independent Mode LAD Bus Timing, Bootstrap Loader Operation

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

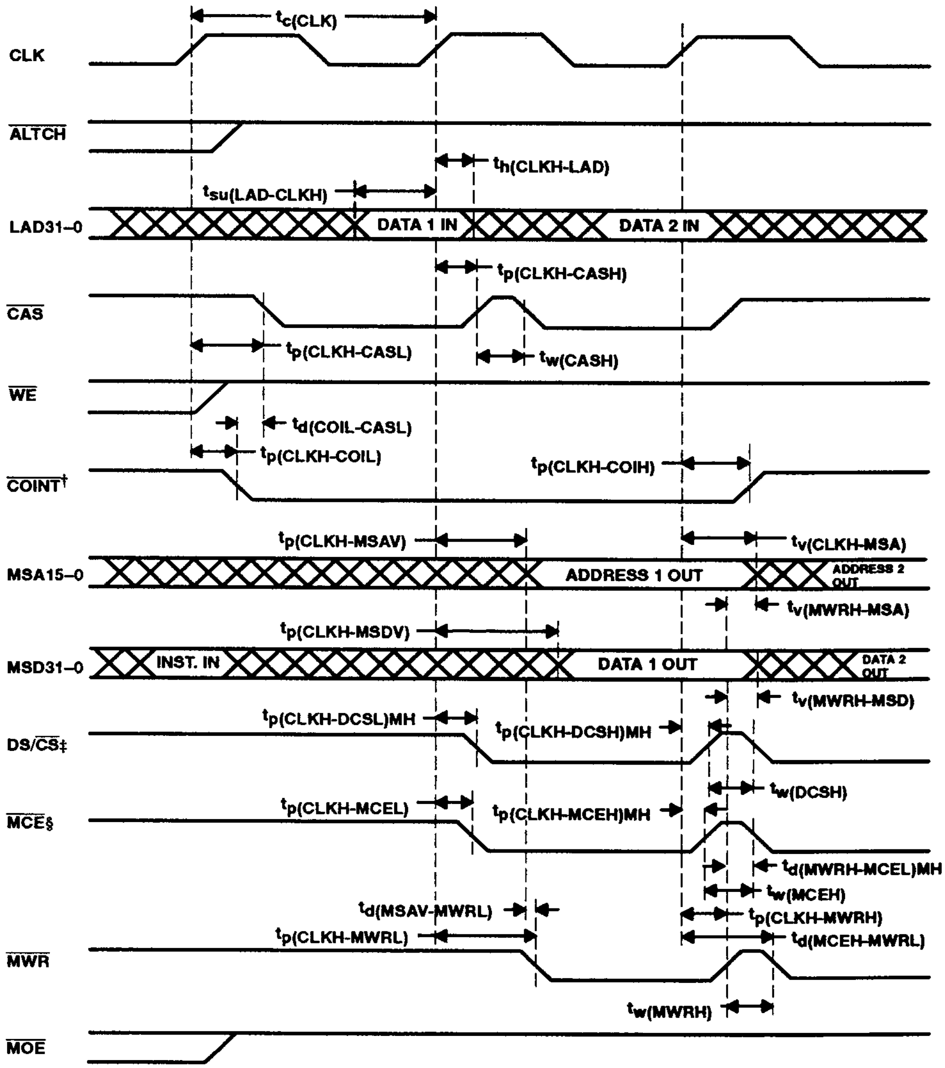
PARAMETER MEASUREMENT INFORMATION



† COINT timing is for LADCFG high only. When the LADCFG bit is set high in the configuration register, COINT is controlled by bit 1 of the LAD move instruction instead of the set mask instruction.
‡ MCE does not toggle at each rising clock edge.
§ MOE goes high at each rising clock edge.

Figure 35. Host-Independent Mode, LAD to MSD Bus Timing with MEMCFG Low

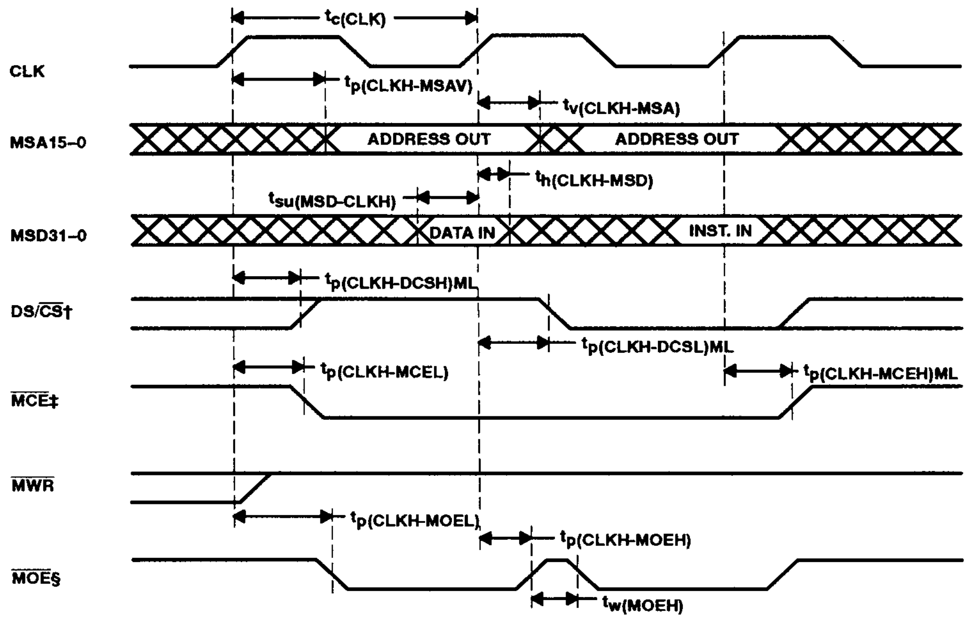
PARAMETER MEASUREMENT INFORMATION



† COINT timing is for LADCFG high only. When the LADCFG bit is set high in the configuration register, COINT is controlled by bit 1 of the LAD move instruction instead of the set mask instruction.
‡ DS/CS valid for moves to data space; MCE valid for moves to code space. Only one of these would be valid for each move instruction.
§ This option for using DS/CS as data space chip enable and MCE as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register.

Figure 36. Host-Independent Mode, LAD to MSD Bus Transfer Timing with MEMCFG High

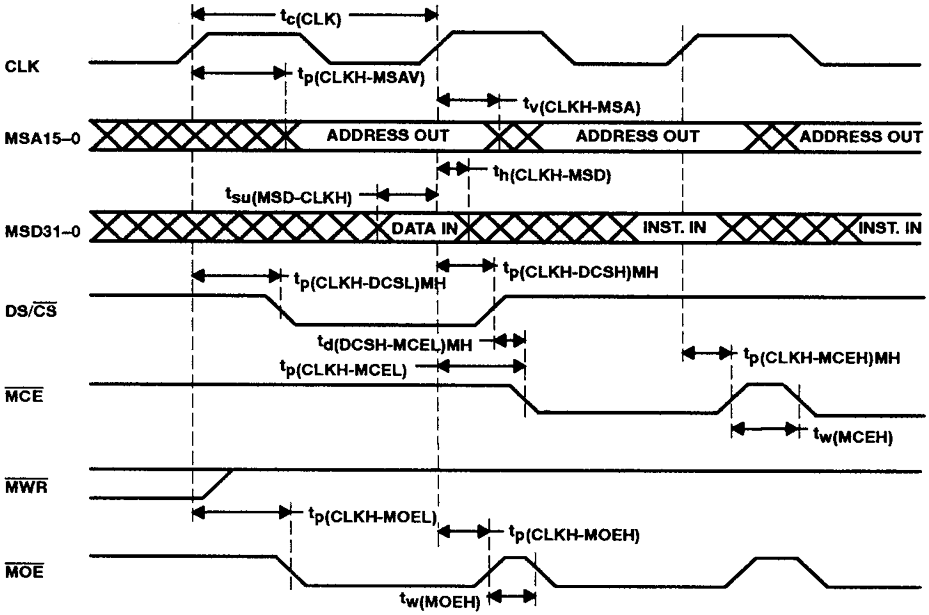
PARAMETER MEASUREMENT INFORMATION



† The setting of DS/$\overline{CS}$ determines whether the value on the MSD bus is an instruction or data.
‡ $\overline{MCE}$ dos not toggle at each rising clock edge.
§ $\overline{MOE}$ goes high at each rising clock edge.
NOTE; This example shows a data read followed by an instruction read.

Figure 37. Host-Independent Mode MSD Bus Timing, Memory to SMJ34082A with MEMCFG Low

PARAMETER MEASUREMENT INFORMATION

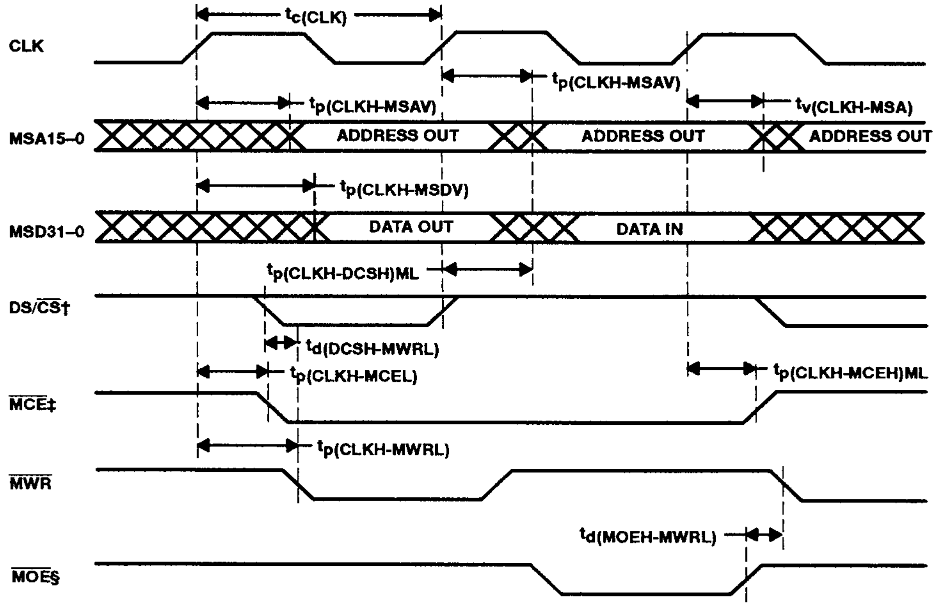


NOTE: This example shows a data read followed by an instruction read followed by an instruction read. This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register. When MEMCFG is high, DS/$\overline{CS}$ and $\overline{MCE}$ rise after every rising clock edge. In this mode, DS/$\overline{CS}$ and $\overline{MCE}$ may not both be active (low) at the same time.

Figure 38. Host-Independent Mode MSD Bus Timing, Memory to SMJ34082A with MEMCFG High

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

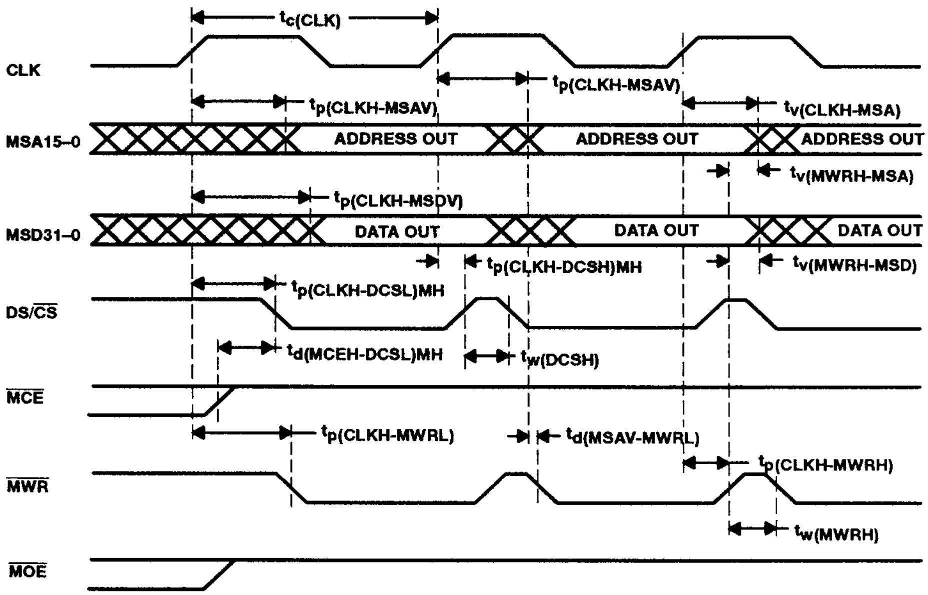
PARAMETER MEASUREMENT INFORMATION



† The setting of DS/CS determines whether the value on the MSD bus is an instruction or data.
‡ MCE dos not toggle at each rising clock edge.
§ MWR goes high at each rising clock edge.
NOTE: This example shows a data write followed by a code read.

Figure 39. Host-Independent Mode MSD Bus Timing, SMJ34082A to Memory with MEMCFG Low

PARAMETER MEASUREMENT INFORMATION

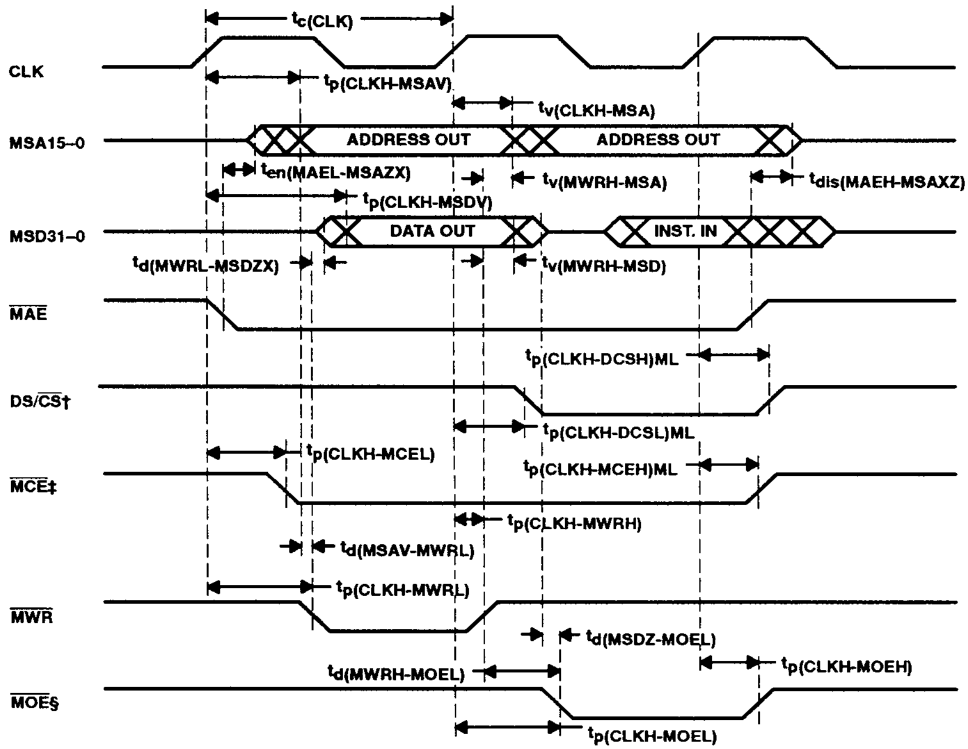


NOTE: This example shows multiple data writes. Timing for multiple code writes would be similar. This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register. When MEMCFG is high, DS/$\overline{CS}$ and $\overline{MCE}$ rise after every rising clock edge. In this mode, DS/$\overline{CS}$ and $\overline{MCE}$ may not both be active (low) at the same time.

Figure 40. Host-Independent Mode MSD Bus Timing, SMJ34082A to Memory with MEMCFG High

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

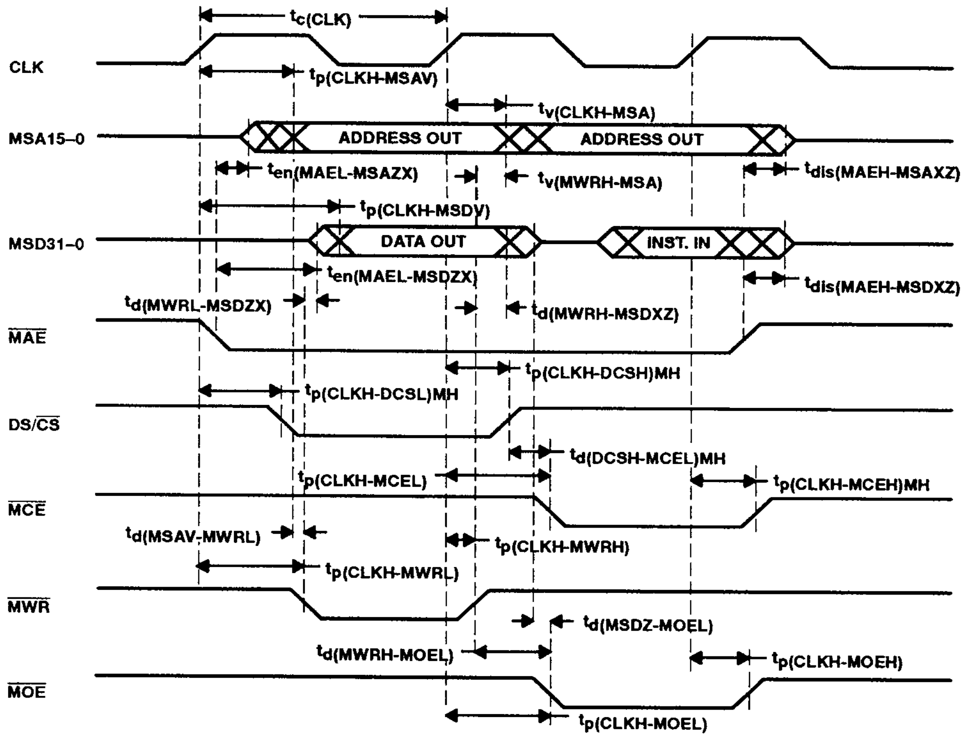
PARAMETER MEASUREMENT INFORMATION



† The setting of DS/$\overline{CS}$ determines whether the value on the MSD bus is an instruction or data.
‡ $\overline{MCE}$ dos not toggle at each rising clock edge.
§ $\overline{MOE}$ goes high at each rising clock edge.
NOTE: This example shows a data write followed by an instruction read.

Figure 41. Host-Independent Mode, MSD Enable/Disable Timing with MEMCFG Low

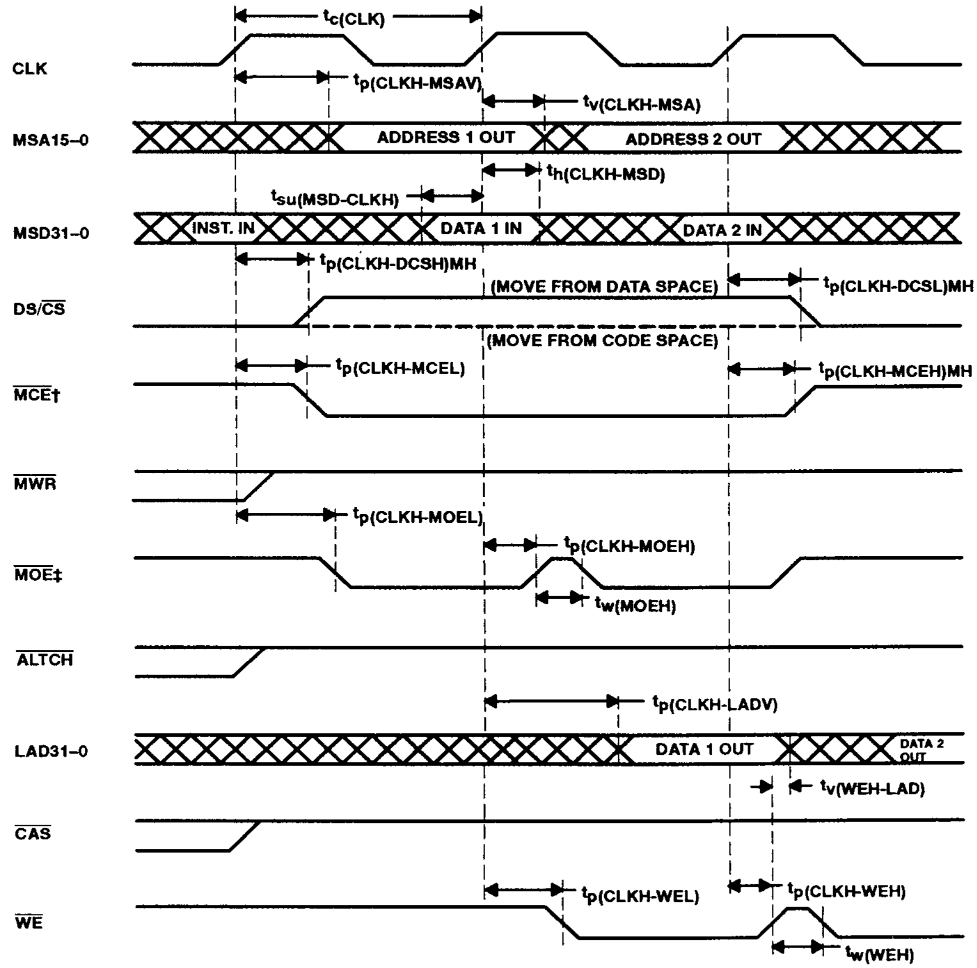
PARAMETER MEASUREMENT INFORMATION



NOTE: This example shows a data write followed by an instruction read. Timing for multiple code writes would be similar. This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is invoked by setting the MEMCFG bit high in the configuration register. When MEMCFG is high, DS/$\overline{CS}$ and $\overline{MCE}$ rise after every rising clock edge. In this mode, DS/$\overline{CS}$ and $\overline{MCE}$ may not both be low at the same time.

Figure 42. Host-Independent Mode, MSD Bus Enable/Disable Timing with MEMCFG High

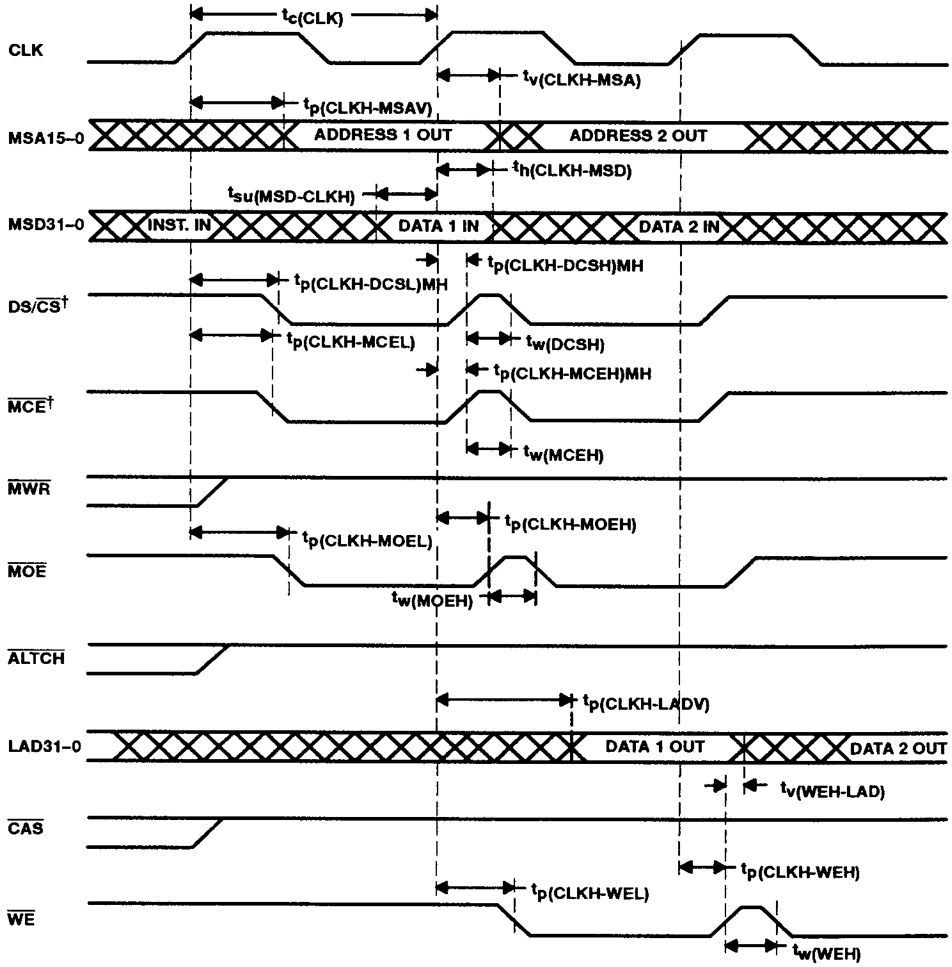
PARAMETER MEASUREMENT INFORMATION



† MCE dos not toggle at each rising clock edge.
‡ MOE goes high at each rising clock edge.

Figure 43. Host-Independent Mode, MSD to LAD Bus Transfer Timing with MEMCFG High

PARAMETER MEASUREMENT INFORMATION

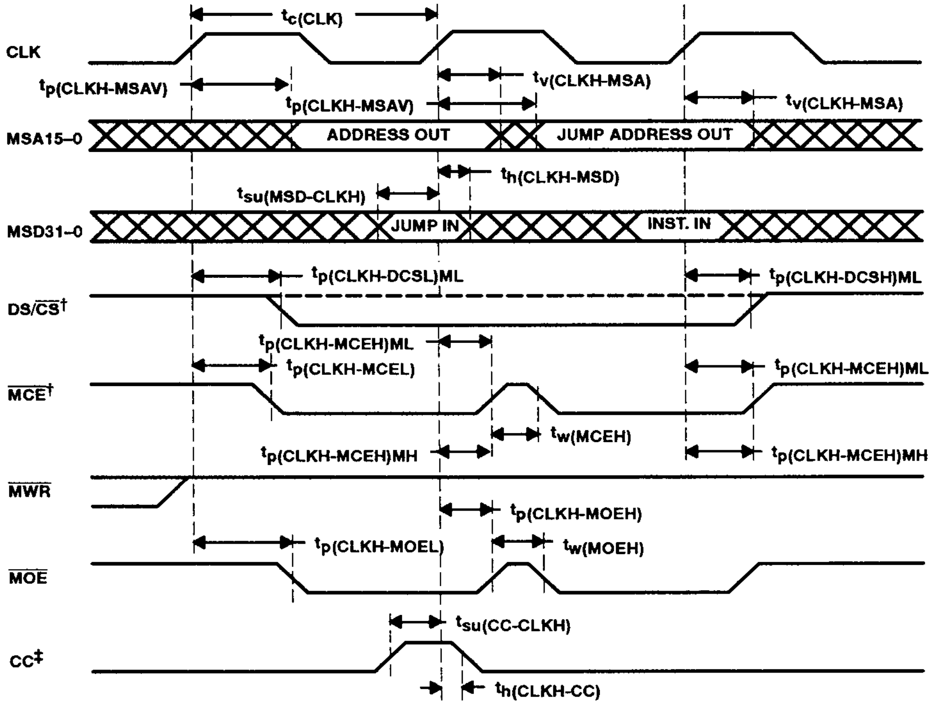


† DS/$\overline{CS}$ valid for moves to data space; $\overline{MCE}$ valid for moves to code space. Only one would be valid for each move instruction.

NOTE: This option for using DS/$\overline{CS}$ as data space chip enable and $\overline{MCE}$ as code space chip enable is involved by setting the MEMCFG bit high in the configuration register.

Figure 44. Host-Independent Mode, MSD to LAD Bus Transfer Timing with MEMCF High

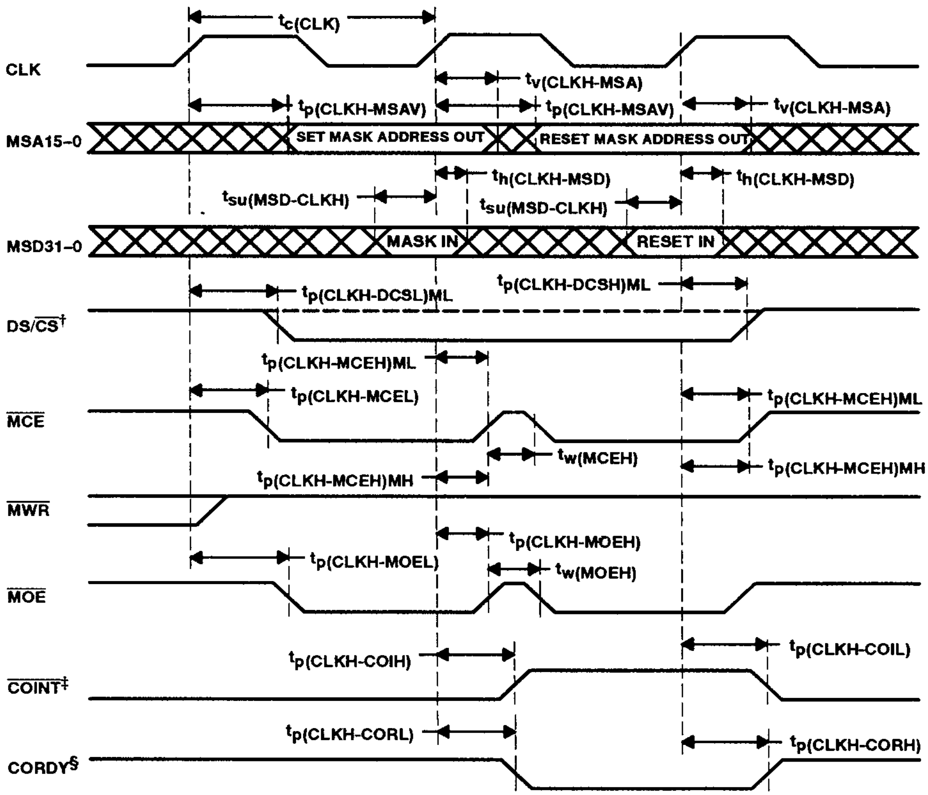
PARAMETER MEASUREMENT INFORMATION



† Dotted line shows DS/$\overline{CS}$ for MEMCFG high.
‡ The CC input is registered on each rising edge of the clock, so the CC bit can be latched one cycle and tested during the next cycle.

Figure 45. Host-Independent Mode, MSD Bus Timing Test Condition (CC) and Branch

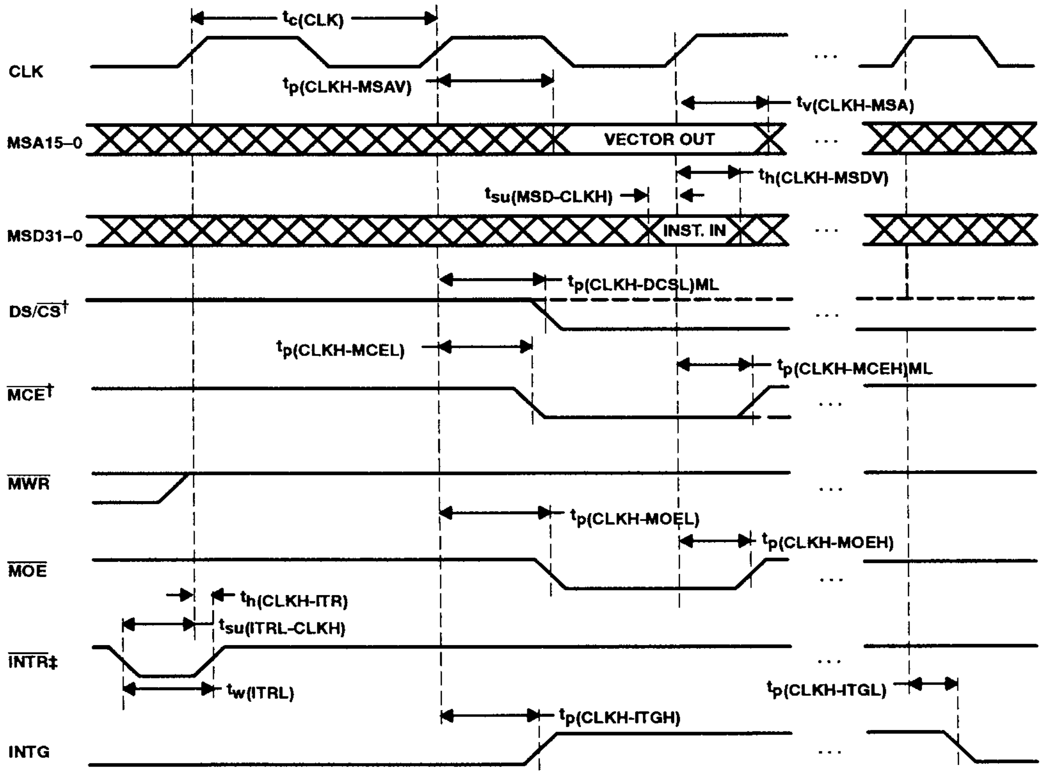
PARAMETER MEASUREMENT INFORMATION



† Dotted line shows DS/$\overline{CS}$ for MEMCFG high.
‡ Valid for MEMCFG low only. When MEMCFG low, $\overline{COINT}$ is set high by the set mask instruction, and it remains high until reset with another set mask instruction.
§ The CORDY output is set low by the set mask instruction, and it remains low until reset with another set mask instruction.

Figure 46. Host-Independent Mode MSD Bus Timing, SET/RESET $\overline{COINT}$ and CORDY

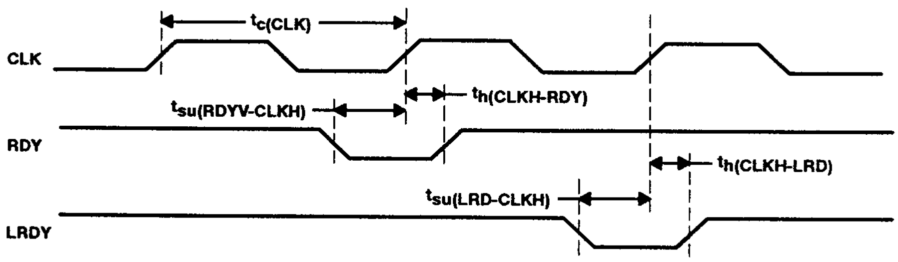
PARAMETER MEASUREMENT INFORMATION



† Dotted lines show DS/$\overline{CS}$ and $\overline{MCE}$ for MEMCFG high.
‡ $\overline{INTR}$ is negative-edged triggered.
NOTE: Interrupts are not granted during multi-cycle instructions. This example shows two interrupt requests. The first is granted immediately; the second, after the first is finished. INTG remains high after an interrupt is granted until interrupts are reenabled or a return from interrupt instruction is executed.

Figure 47. Host-Independent Mode, MSD Bus Timing External Interrupt to SMJ34082A

PARAMETER MEASUREMENT INFORMATION



NOTE: When either RDY or LRDY is set low and the setup time before CLK high is observed, the device is stalled for one or more clock cycles, until RDY or LRDY is set high again. During a wait state, internal states and status are preserved and output signals do not change. LRDY can be used in this manner only in the host-independent mode.

Figure 48. Host-Independent Mode, MSD Bus Timing Wait State Timing

PROGRAMMING INFORMATION

programming the SMJ34082A

The SMJ34082A is supported by a software development tool kit, including a C compiler and an assembler. Program development using the tools is described in the TMS34082A tool kit documentation. Information on internal instructions and listing of the external instructions are provided in the following sections.

In both the coprocessor and host-independent modes, the SMJ34082A instruction word is 32 bits long. The number, length, and arrangement of fields in the 32-bit word depends on the operating mode and operation selected. Internal microcode to the SMJ34082A is not restricted to the same 32-bit instruction formats so certain internal programs may execute faster than the same operations written with external code can achieve.

In the coprocessor mode, the SMJ34082A can execute instructions both from the SMJ34020 and from the program memory on the MSD bus (MSD31-0). In the host-independent mode the SMJ34082A is controlled from code input on the MSD bus. Internal instructions may be executed in the host-independent mode by performing a jump to the internal address.

PROGRAMMING INFORMATION

internal instructions

The SMJ34082A FPU performs a wide range of internal arithmetic and logical operations, as well as complex operations (flagged '†'), summarized below. Complex instructions are multi-cycle routines stored in the internal program ROM.

One-Operand Operations:
- Absolute Value
- Square Root
- Reciprocal†
- 1s Complement
- 2s Complement

Conversions:
- Integer to Single
- Integer to Double
- Single to Double
- Single to Integer
- Double to Integer
- Double to Single

Two-Operand Operations:
- Add
- Subtract
- Compare
- Multiply
- Divide

Matrix Operations:
- 4x4, 4x4 Multiply†
- 1x4, 4x4 Multiply†
- 3x3, 3x3 Multiply†
- 1x3, 3x3 Multiply†

Graphics Operations:
- Backface Testing†
- Polygon Clipping†
- 2-D Linear Interpolation†
- 2-D Window Compare†
- 2-Plane Clipping (X,Y,Z)†
- 2-D Cubic Spline†
- Polygon Elimination†
- Viewport Scaling and Conversion†
- 3-D Linear Interpolation†
- 3-D Volume Compare†
- 2-Plane Color Clipping (R,B,G,I)†
- 3-D Cubic Spline†

Image Processing:
- 3x3 Convolution†

Chained Operations :
- Polynomial Expansion†
- 1-D Min/Max†
- Multiply/Accumulate†
- 2-D Min/Max†

Vector Operations:
- Add†
- Subtract†
- Magnitude†
- Scaling†
- Dot Product†
- Cross Product†
- Normalization†
- Reflection†

The internal ROM routines may be used in either the coprocessor or host-independent mode. In the coprocessor mode, the internal routines are invoked by SMJ34020 instructions to its coprocessor(s).

In the host-independent mode, the internal programs can be called as subroutines by the externally stored code. External programs can call internal routines by executing a jump to subroutine with bit 16 (internal code select) set high and the address of the internal routine as the jump address.

The format of the SMJ34082A instruction in the coprocessor mode is shown in Figure 49. The instruction is issued by the SMJ34020 via the LAD bus.

† Indicates a complex instruction.

PROGRAMMING INFORMATION

31	28	24	20	15	13	8	7	6	5				0
ID	ra	rb	rd	md	fpuop	type	size	0	1	0	0	0	0	0

Figure 49. SMJ34082A Instruction

The 3-bit ID field identifies the coprocessor for which the instruction is intended. This coprocessor ID corresponds to the settings of the CID2-CID0 pins. To broadcast an instruction to all coprocessors, the ID is set to 4h.

Table 5. Coprocessor ID

ID	COPROCESSOR
000	FPU0
001	FPU1
010	FPU2
011	FPU3
100	FPU broadcast
101	Reserved
110	Reserved
111	User defined

Four coprocessor addressing modes are defined for the SMJ34082A. The md field indicates the addressing mode.

Table 6. Addressing Modes

MODE	MD FIELD	OPERATION
0	00	FPU internal operations with no jump or external moves
1	01	Transfer data to/from SMJ34020 registers
2	10	Transfer data to/from memory (controlled by SMJ34020)
3	11	External instructions

The type and size bits identify the type of operand; as shown below in Table 7. The I bit is used to indicate to the SMJ34082A that this is a reissue of a coprocessor instruction due to a bus interruption. The least significant four bits are the bus status bits, which will all be zero to indicate a coprocessor cycle.

Table 7. OPERAND Types

TYPE	SIZE	OPERAND TYPE
0	0	32-bit integer
0	1	Reserved
1	0	Single-precision floating-point (32-bit)
1	1	Double-precision floating-point (64-bit)

The ra, rb, and rd fields are for the two sources and destination within the FPU. Register addresses are listed in Table 1. For the ra and rb fields, only the four least significant bits of the register address are used. The ra field may only use the RA register file, C, and CT. The RB field may only use the RB register file, C and CT.

The Floating-Point Unit Operation (fpuop) field is the FPU opcode (5 bits) described in Tables 8, 9, and 10.

In the coprocessor mode, the SMJ34082A executes user-defined routines (stored in external memory on the MSD bus) by executing a jump to external code. For this instruction, the md field (bits 15-13) is set high and the fpuop field gives the routine number (0-31). The SMJ34082A multiplies the routine number by two to get the jump address. For example, routine number 14 would have a jump address of 28 decimal or 1C hex.

PROGRAMMING INFORMATION

The routines are coded using the external instruction format discussed in the next section. The last instruction should be a jump to internal instruction address 0FFFh with the I-bit(internal) set or a return from subroutine instruction. This puts the FPU in an idle state, waiting for the next instruction from the SMJ34020.

Table 8. Coprocessor Mode Instructions

FPUOP	TMS34020 ASSEMBLER OPCODE	DESCRIPTION
00000	ADDx	Sum of ra and rb, place in rd
00001	SUBx	Subtract rb from ra, place result in rd
00010	CMPx	Set status bits on result of ra minus rb
00011	SUBx	Subtract ra from rb, place result in rd
00100	ADDAx	Absolute value of sum of ra and rb, place result in rd
00101	SUBAx	Absolute value of (ra minus rb), place result in rd
00110	MOVE or MOVx	Load multiple FPU registers from SMJ34020 GSP or its memory
00111	MOVE or MOVx	Save multiple FPU registers to SMJ34020 GSP or its memory
01000	MPYx	Multiply ra and rb, place result in rd
01001	DIVx	Divide ra by rb, place result in rd
01010	INVx	Divide 1 by rb, place result in rd
01011	ASUBAx	Absolute value of ra minus absolute value of rb, place in rd
01100	reserved	
01101	MOVEx	Move ra to rd, multiple, for n registers
01110	MOVEx	Move rb to rd, multiple, for n registers
01111	(see Table 10)	Single operand instructions, rb field redefined
10000	CPWx	Compare point to window (set XLT, XGT, YLT, TGT)
10001	CPVx	Compare point to volume (set XLT, XGT, YLT, YGT, ZLT, ZGT)
10010	BACKFx	Test polygon for facing direction (backface test)
10011	INMNMXx	Setup FPU registers for MNMX1 or MNMX2 instruction
10100	LINTx	Given [X1, Y1, Z1], [X2, Y2, Z2], and a plane, find [X3, Y3, Z3]
10101	CLIPFx	Clip a line to a plane pair boundary (start with point 1)
10110	CLIPRx	Clip a line to a plane pair boundary (start with point 2)
10111	CLIPCFx	Clip color values to a plane pair boundary (start with point 1)
11000	SCALEx	Scale and convert coordinates for viewpoint
11001	MTRANx	Transpose a matrix
11010	CKVTXx	Compare a polygon vertex to a clipping volume
11011	CONVx	3x3 convolution
11100	CLIPCRx	Clip color values to a plane pair boundary (start with point 2)
11101	OUTC3x	Compare a line to a clipping value
11110	CSPLNx	Calculate cubic spline for given coefficients
11111	(see Table 11)	Vector and matrix instructions, rb field redefined

F denotes single-precision, D denotes double-precision floating-point, x denotes operand type, and a blank designates signed integer

PROGRAMMING INFORMATION

Table 9. Coprocessor Mode Instructions, FPUOP = 01111_2

RB	TMS34020 ASSEMBLER OPCODE	DESCRIPTION
0000	PASS	Copy ra to rd
0001	NOT	Place 1s complement of ra in rd
0010	ABS	Place absolute value of ra in rd
0011	NEG	Place negated value of ra in rd
0100	CVDF	Convert double in ra to single in rd (T and S define ra)
0100	CVFD	Convert single in ra to double in rd (T and S define ra)
0101	CVDI	Convert double in ra to integer in rd (T and S define ra)
0101	CVFI	Convert single in ra to integer in rd (T and S define ra)
0110	CVID	Convert integer in ra to double in rd (T and S define ra)
0110	CVIF	Convert integer in ra to single in rd (T and S define ra)
0111	VSCLx	Multiply each component of a velocity by a scaling factor
1000	SQARx	Place (ra * ra) in rd
1001	SQRTx	Extract square root or ra, place in rd
1010	SQRTAx	Extract square root of absolute value of ra, place in rd
1011	ABORT	Stop execution of any FPU instruction
1100	CKVTXI	Initialize check vertex instruction
1101	CHECK	Check for previous instruction completion
1110	MOVMEM	Move data from system memory to external memory @ MCADDR
1111	MOVMEM	Move data to system memory from external memory @ MCADDR

Table 10. Coprocessor Mode Instructions, FPUOP = 11111_2

RB	TMS34020 ASSEMBLER OPCODE	DESCRIPTION
0000	POLYx	Polynomial expansion
0001	MACx	Multiply and accumulate
0010	MNMX1x	Determine 1-D minimum and maximum of a series
0011	MNMX2x	Determine 2-D minimum and maximum of a series of pairs
0100	MMPY0x	Multiply matrix elements 0, 1, 2, 3 by vector element 0
0101	MMPY1x	Multiply matrix elements 4, 5, 6, 7 by vector element 1
0110	MMPY2x	Multiply matrix elements 8, 9, 10, 11 by vector element 2
0111	MMPY3x	Multiply matrix elements 12, 13, 14, 15 by vector element 3
1000	MADDx	Add matrix elements 12, 13, 14, 15 to vector
1001	VADDx	Add two vectors
1010	VSUBx	Subtract a vector from a vector
1011	VDOTx	Compute scalar dot product of two vectors
1100	VCROSx	Compute cross product of two vectors
1101	VMAGx	Determine the magnitude of a vector
1110	VNORMx	Normalize a vector to unit magnitude
1111	VRFLCTx	Given normal and incident vectors, find the reflection

F denotes single-precision, D denotes double-precision floating-point, x denotes operand type, and a blank designates signed integer

PROGRAMMING INFORMATION

external instructions

External instructions are 32 bits long, and their formats (number, length, and function of fields) depend on the operations being selected. Separate formats are provided for data transfers, FPU processing, test and branch operations, and subroutine calls.

Instructions that control FPU operations can select operands from input registers, internal feedback, or from the LAD bus (32-bit operations only). The format for an FPU processing instruction is shown in Figure 50.

31	28	24	20	15	11 0
OP	RA	RB	RD	SEL_OP	INSTRUCTION

Figure 50. FPU Processing External Instruction Format

The op field selects the sequencer operation. Three continue instructions are available to permit control of the $\overline{WE}$ and $\overline{ALTCH}$ strobe outputs, which enable LAD output in the host-independent mode. The ra, rb, and rd fields are for the two sources and destination in the SMJ34082A register file. The sel_op field selects the source of the operands: register file or feedback registers. The instruction field designates the operation to be performed.

External instructions and cycle counts are listed in Table 11. Absolute values of operands or results, negated results, and wrapped number inputs are selectable options. Chained operations, using the multiplier and ALU in parallel, and other instructions to control program flow and move data are included.

External instruction timing depends on the pipeline registers setting, controlled by the PIPES2-1 bits in the configuration register. Most FPU processing instructions (with the exception of divide, square root, and double-precision multiply) execute in one cycle per pipeline stage.

PROGRAMMING INFORMATION

Table 11. External Instructions and Timing

SMJ34082A ASSEMBLER OPCODE	DESCRIPTION OF ROUTINE	PIPES2–1 11	PIPES2–1 10	PIPES2–1 01	PIPES2–1 00
ADD	Add A + B	1(1)	2(1)	2(1)	3(1)
AND	Logical AND A, B	1(1)	2(1)	2(1)	3(1)
ANDNA	Logical AND NOT A, B	1(1)	2(1)	2(1)	3(1)
ANDNB	Logical AND A, NOT B	1(1)	2(1)	2(1)	3(1)
CJMP	Conditional jump	1(1)	1(1)	1(1)	1(1)
CSJR	Conditional jump to subroutine	1(1)	1(1)	1(1)	1(1)
CMP	Compare A, B	1(1)	2(1)	2(1)	3(1)
COMPL	Pass 1s complement of A	1(1)	2(1)	2(1)	3(1)
DIV	Divide A / B				
	SP	8(8)	8(7)	9(7)	9(7)
	DP	13(13)	13(12)	15(12)	15(12)
	integer	16(16)	16(15)	17(15)	17(15)
DTOF	Convert from DP to SP	1(1)	2(1)	2(1)	3(1)
DTOI	Convert from DP to integer	1(1)	2(1)	2(1)	3(1)
DTOU	Convert from DP to unsigned integer	1(1)	2(1)	2(1)	3(1)
FTOD	Convert from SP to DP	1(1)	2(1)	2(1)	3(1)
FTOI	Convert from SP to integer	1(1)	2(1)	2(1)	3(1)
FTOU	Convert from SP to unsigned integer	1(1)	2(1)	2(1)	3(1)
ITOD	Convert from integer to DP	1(1)	2(1)	2(1)	3(1)
ITOF	Convert from integer to SP	1(1)	2(1)	2(1)	3(1)
LD	Load n words into register				
	SP	n + 1	n + 1	n + 1	n + 1
	DP	2n + 1	2n + 1	2n + 1	2n + 1
	integer	n + 1	n + 1	n + 1	n + 1
LDLCT	Load loop counter with value	1(1)	1(1)	1(1)	1(1)
LDMCADDR	Load MCADDR with value	1(1)	1(1)	1(1)	1(1)
MASK	Set programmable mask	1(1)	1(1)	1(1)	1(1)
MOVA	Move A (no status flags active)	1(1)	2(1)	2(1)	3(1)
MOVLM	Move n words from LAD bus to MSD bus				
	SP	n + 1	n + 1	n + 1	n + 1
	DP	2n + 1	2n + 1	2n + 1	2n + 1
	integer	n + 1	n + 1	n + 1	n + 1
MOVML	Move n words from MSD bus to LAD bus				
	SP	n + 1	n + 1	n + 1	n + 1
	DP	2n + 1	2n + 1	2n + 1	2n + 1
	integer	n + 1	n + 1	n + 1	n + 1
MOVRR	Multiple move, register to register				
	SP	n + 1	n + 1	n + 1	n + 1
	DP	2n + 1	2n + 1	2n + 1	2n + 1
	integer	n + 1	n + 1	n + 1	n + 1
MULT.ADD	Multiply $A_1 * B_1$, Add $A_2 + B_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)

DP denotes double-precision, and SP denotes single-precision.

PROGRAMMING INFORMATION

Tabale 11. External Instructions and Timing (Continued)

SMJ34082A ASSEMBLER OPCODE	DESCRIPTION OF ROUTINE	PIPES2–1 11	PIPES2–1 10	PIPES2–1 01	PIPES2–1 00
MULT.NEG	Multiply $A_1 * B_1$, Subtract $0 - A_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
MULT	Multiply A * B				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
MULT.PASS	Multiply $A_1 * B_1$, Add $A_2 + 0$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
MULT.SUB	Multiply $A_1 * B_1$, Subtract $A_2 - B_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
MULT.2SUBA	Multiply $A_1 * B_1$, Subtract $2 - A_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
MULT.SUBRL	Multiply $A_1 * B_1$, Subtract $B_2 - A_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
NEG	Pass –A (2s Complement)	1(1)	2(1)	2(1)	3(1)
NOR	Logical NOR A, B	1(1)	2(1)	2(1)	3(1)
OR	Logical OR A, B	1(1)	2(1)	2(1)	3(1)
PASS	Pass A	1(1)	2(1)	2(1)	3(1)
PASS	Pass B	1(1)	2(1)	2(1)	3(1)
PASS.ADD	Multiply $A_1 * 1$, Add $A_2 + B_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
PASS.NEG	Multiply $A_1 * 1$, Subtract $0 - A_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
PASS.PASS	Multiply $A_1 * 1$, Add $A_2 + 0$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
PASS.SUB	Multiply $A_1 * 1$, Subtract $A_2 - B_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
PASS.2SUBA	Multiply $A_1 * 1$, Subtract $2 - A_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)

DP denotes double-precision, and SP denotes single-precision.

PROGRAMMING INFORMATION

Table 11. External Instructions and Timing (Continued)

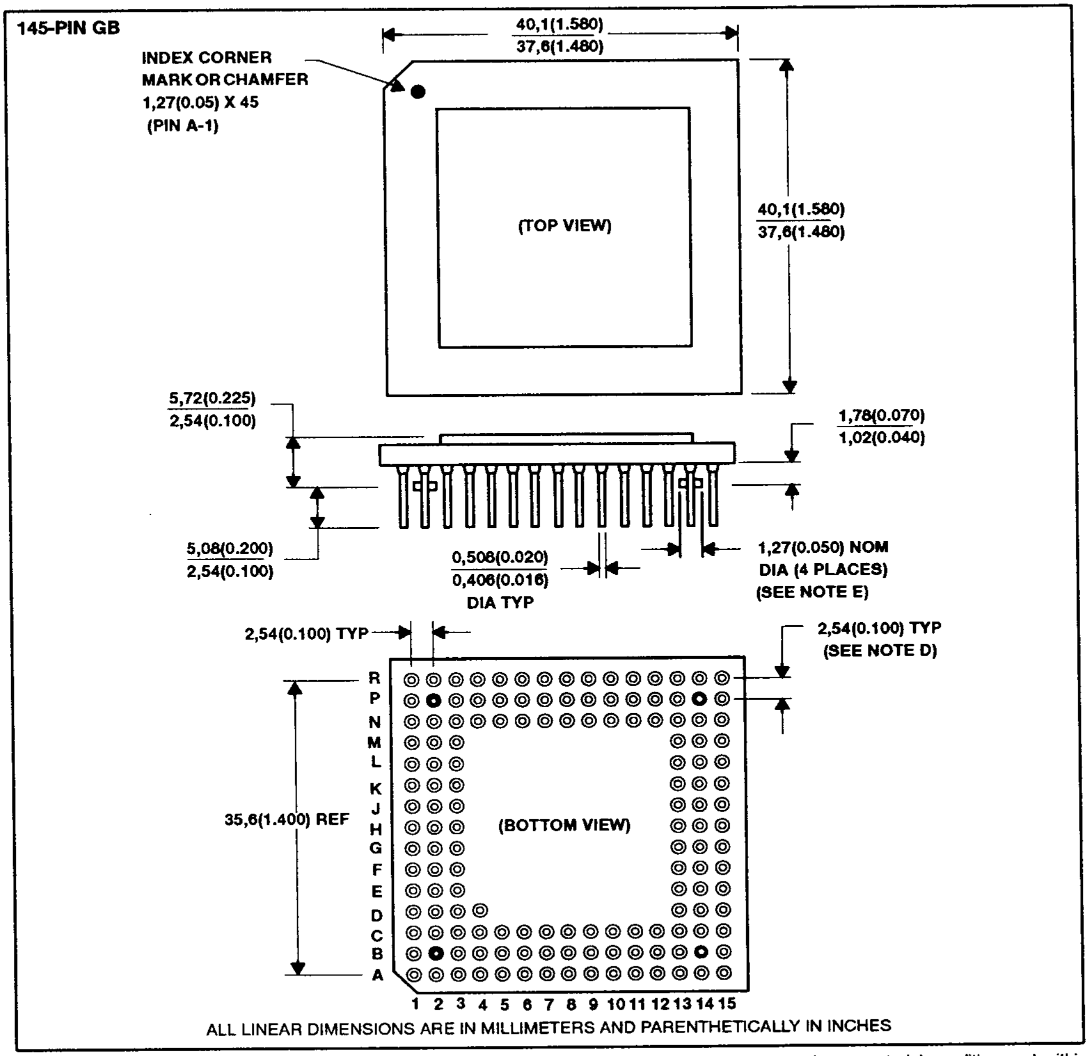
SMJ34082A ASSEMBLER OPCODE	DESCRIPTION OF ROUTINE	CYCLE COUNTS			
		PIPES2–1 11	PIPES2–1 10	PIPES2–1 01	PIPES2–1 00
RTS	Return from subroutine	1(1)	1(1)	1(1)	1(1)
SLL	Logical shift left A by B bits	1(1)	2(1)	2(1)	3(1)
SQRT	Square root of A				
	SP	11(11)	11(10)	12(10)	12(10)
	DP	16(16)	16(15)	17(15)	17(15)
	integer	20(20)	20(19)	21(19	21(19)
PASS.SUBRL	Multiply $A_1 * 1$, Subtract $B_2 - A_2$				
	SP	1(1)	2(1)	2(1)	3(1)
	DP	2(2)	3(2)	3(2)	4(2)
	integer	1(1)	2(1)	2(1)	3(1)
SRA	Arithmetic shift right A by B bits	1(1)	2(1)	2(1)	3(1)
SRL	Logical shift right A by B bits	1(1)	2(1)	2(1)	3(1)
ST	Store n words from register				
	SP	n + 1	n + 1	n + 1	n + 1
	DP	2n + 1	2n + 1	2n + 1	2n + 1
	integer	n + 1	n + 1	n + 1	n + 1
SUB	Subtract A – B	1(1)	2(1)	2(1)	3(1)
SUBRL	Subtract B – A	1(1)	2(1)	2(1)	3(1)
UTOD	Convert from unsigned integer to DP	1(1)	2(1)	2(1)	3(1)
UTOF	Convert from unsigned integer to SP	1(1)	2(1)	2(1)	3(1)
UWRAPI	Unwrap inexact operand	1(1)	2(1)	2(1)	3(1)
UWRAPR	Unwrap rounded operand	1(1)	2(1)	2(1)	3(1)
UWRAPX	Unwrap exact operand	1(1)	2(1)	2(1)	3(1)
WRAP	Wrap denormalized operand	1(1)	2(1)	2(1)	3(1)
XOR	Logical exclusive OR A, B	1(1)	2(1)	2(1)	3(1)

DP denotes double-precision, and SP denotes single-precision.

MECHANICAL DATA

GB pin-grid-array ceramic package

This is a hermetically sealed package.



NOTES: D. Pins are located within 0,13 (0.005) radius of true position relative to each other at maximum meterial condition and within 0,457 (0.018) radius of the center of the ceramic.

E. Dimensions do not include solder finish.

This Page Intentionally Left Blank