1980

MOS

DATABOOK

National Semiconductor

MOS
DATABOOK

NATIONAL
SEMICONDUCTOR

National Semiconductor

Electronic Data Processing

MM57109 MOS/LSI Number-Oriented Processor

General Description

The MM57109 is an MOS/LSI number-oriented processor (actually, a pre-programmed single-chip microcomputer member of National's COP family) intended for use in number processing applications. Scientific calculator functions, test and branch capability, internal number storage, and input/output instructions have been combined in this single chip device. Programming is done in calculator keyboard level language which simplifies software development. Generated code is more reliable because algorithms are preprogrammed in an on-chip ROM. Data or instructions can be synchronous or asynchronous; I/O digit count, I/O notation mode, and error control are user programmable; a sense input and flag outputs are available for single bit control.

The MM57109 can be used as a stand-alone processor with external ROM/PROM and program counter (PC). Alternatively it can be configured as a peripheral device on the bus of a microprocessor or minicomputer.

Applications

- Instruments
- Microprocessor/minicomputer peripheral
- Test equipment
- Process controllers

Features

- Scientific calculator instructions (RPN)
  - Up to 8-digit mantissa, 2-digit exponent
  - Four-register stack, one memory register
  - Trigonometric functions, logarithmic functions, Y^X, e^X, π, etc.
  - Error flag generation and recovery
- Flexible input/output
  - HOLD input allows asynchronous instructions or single stepping
  - Asynchronous digit input instruction (AIN) with data ready ($\overline{ADR}$) input
  - Multidigit I/O instructions (IN, OUT) with floating point or scientific notations
  - Programmable mantissa digit count for IN, OUT instructions
  - Sense input and flag outputs
- Branch control
  - Conditional and unconditional program branching
  - Increment/decrement branch on non-zero for program loops
- Interface simplicity
  - Single ϕ clock
  - Low power operation
  - Generates all I/O control signals
  - Separate digit input, output, and address buses

TABLE I. FEATURE COMPARISON OF LSI NUMBER PROCESSING CHIPS

FUNCTION	CALCULATOR	MM57109	MICRO-PROCESSOR
I/O	Keyboard display	Multidigit asynchronous digit single bit	Data bytes single bit
Data format	Floating point Scientific Notation	Floating point Scientific Notation	Binary
Data length	Fixed	Variable (1 to 8 digit mantissa)	Fixed
Program	Key sequence	External ROM/ PC, μP or FIFO	External ROM Internal PC
Speed (math or I/O operations)	14–1500 ms	0.5–1000 ms	0.5–1000 ms
Minimum number of chips for CPU and RAM	1–3	1 (external PC and program source)	2–6

Note: This data sheet is complete. It contains all necessary programming information and electrical interconnect details. The user should read this document *thoroughly* before proceeding.

Absolute Maximum Ratings

Voltage at Any Pin Relative to V_{SS} (All Other Pins Connected to V_{SS})	V_{SS} + 0.3V to V_{SS} − 12V
Ambient Operating Temperature	0°C to +70°C
Ambient Storage Temperature	−55°C to +125°C
Lead Temperature (Soldering, 10 seconds)	300°C

DC Electrical Characteristics $0°C \leq T_A \leq +70°C$, $7.9V \leq V_{SS} - V_{DD} \leq 9.5V$ unless otherwise stated

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Operating Voltage ($V_{SS} - V_{DD}$)		7.9	9.0	9.5	V
Operating Supply Current (I_{DD})	$V_{SS} - V_{DD}$ = 9.5V, T_A = 25°C (Excluding Outputs)		12	18	mA
Osc. Input Voltage Levels	Internal 6k Resistor to V_{SS}				
Input High Level (V_{IH})	$V_{SS} - V_{DD}$ = 7.9V	V_{SS}−1.0			V
Input Low Level (V_{IL})	$V_{SS} - V_{DD}$ = 9.5V			V_{DD}+1.5	V
HOLD, POR Input Voltage Levels	No Internal Resistors				
Input High Level (V_{IH})		V_{SS}−3.0			V
Input Low Level (V_{IL})				V_{DD}+1.5	V
I_1–I_6 Input Voltage Levels	Internal Resistors to V_{SS} (No Resistor for I_6), (Note 1)				
Input High Level (V_{IH})		V_{SS}−1.0			V
Input Low Level (V_{IL})				V_{SS}−4.0	V
INTERFACING WITH MOS OR CMOS					
All Outputs	External Resistor to V_{DD} = 10k–20k				
Output High Voltage (V_{OH})		V_{SS}−1		V_{SS}	V
Output Low Voltage (V_{OL})		V_{DD}		V_{DD}+1	V

AC Electrical Characteristics $0°C \leq T_A \leq +70°C$, $7.9V \leq V_{SS} - V_{DD} \leq 9.5V$ unless otherwise stated

PARAMETER	CONDITIONS (Note 2)	MIN	TYP	MAX	UNITS
Osc. Input Frequency		320	400	400	kHz
Osc. Duty Cycle		46	56	66	%
Osc. Input					
Rise Time (t_r)	C_{LOAD} = 25 pF, R_{LOAD} = 6 kΩ			350	ns
Fall Time (t_f)	RC = 0.15 μs			50	ns
Sync. Output Timing	C_{LOAD} = 250 pF				
t_B (1 Microcycle)		10.0		12.5	μs
t_{pdsL}		0.1		1.65	μs
t_{pdsH}		0.1		1.25	μs
t_{HS}		0.1		0.8	μs
R/$\overline{W}$, ISEL Output Timing					
t_{pdf}	C_{LOAD} = 100 pF			4.4	μs
$\overline{DAS}$ Output Timing					
t_{pdDAS}	C_{LOAD} = 50 pF			4.4	μs
t_{rDAS}	$C_{LOAD} \leq$ 20 pF	0.3			μs
DA1–DA4, $\overline{BR}$ Output Timing	C_{LOAD} = 100 pF (DA1–DA4)				
t_{pdg}	C_{LOAD} = 250 pF ($\overline{BR}$)	0.5		4.0	μs
DO1–DO4, F1, F2, RDY, ERROR Output Timing					
t_{pdK}				6.0	μs

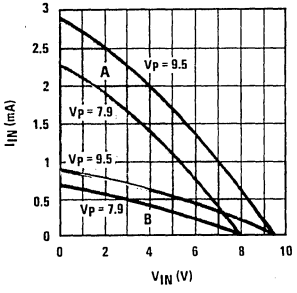
Note 1: An external resistor (5k–20k) can be tied at the I_6 input to V_{SS} to overcome internal load device to V_{DD}.

Note 2: See *Figure 2* for timing diagrams of each of the following inputs/outputs.

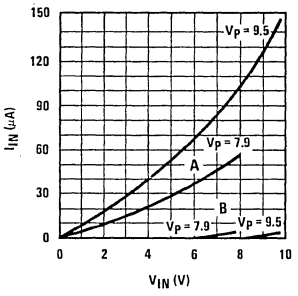
Typical Performance Characteristics

DC Voltage-Current Curves for Chip Inputs and Outputs

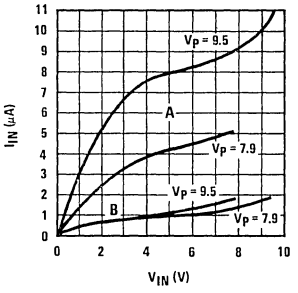
I_5–I_1



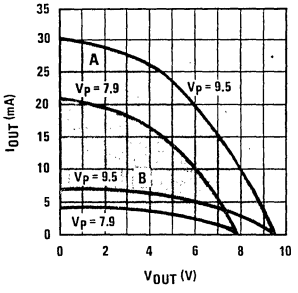
I_6



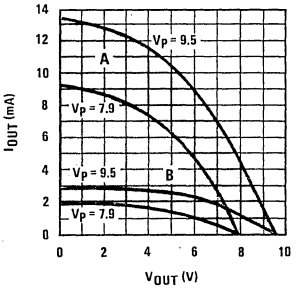
POR



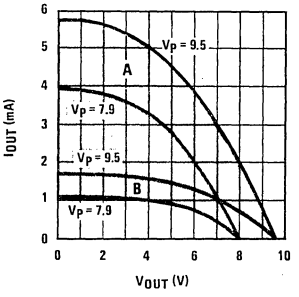
SYNC



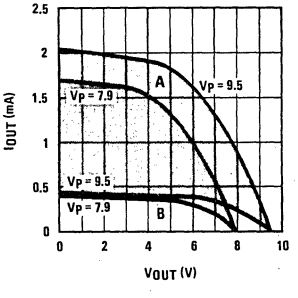
ISEL, R/$\overline{W}$



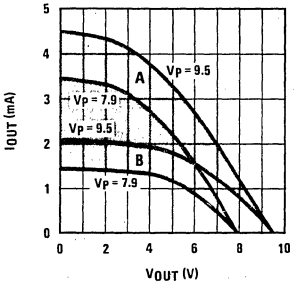
$\overline{DAS}$



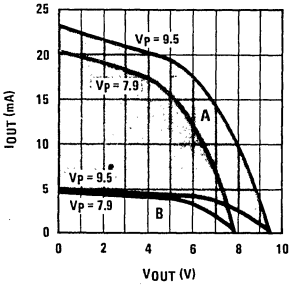
DA1–DA4



$\overline{BR}$



DO1–DO4, F1, F2, ERROR and RDY



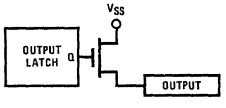
Note 1: Curves "A" represent a maximum current device at 0°C. Curves "B" represent a minimum current device at +70°C.

Note 2: Vp = V_{SS}–V_{DD} (volts)
V_{IN} = Input voltage at pin relative to V_{DD}
I_{IN} = Current *into* pin
V_{OUT} = Output voltage at pin relative to V_{DD}
I_{OUT} = Current *out of* pin

Note 3: HOLD input current is essentially zero at all valid input voltages.

Note 4: OSC input loading consists of a resistor to V_{SS} whose minimum possible value is 4k and whose maximum possible value is 8k.

Note 5: Output currents (I_{OUT}) are shown for logic "1" output conditions only. Any output which is at logic "0" is in a high-impedance state. Equivalent circuit for each output is:



Pinout and Block Diagrams

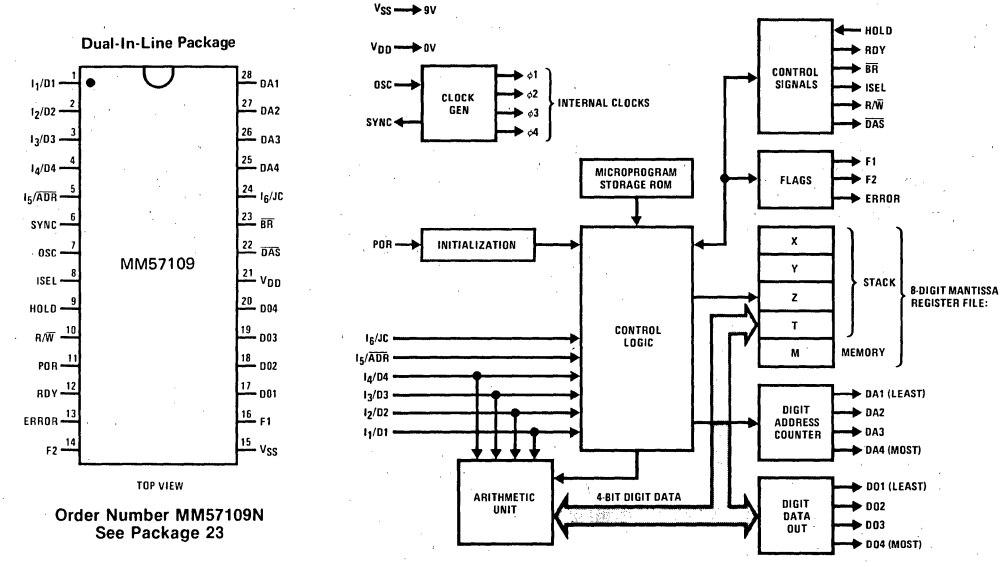
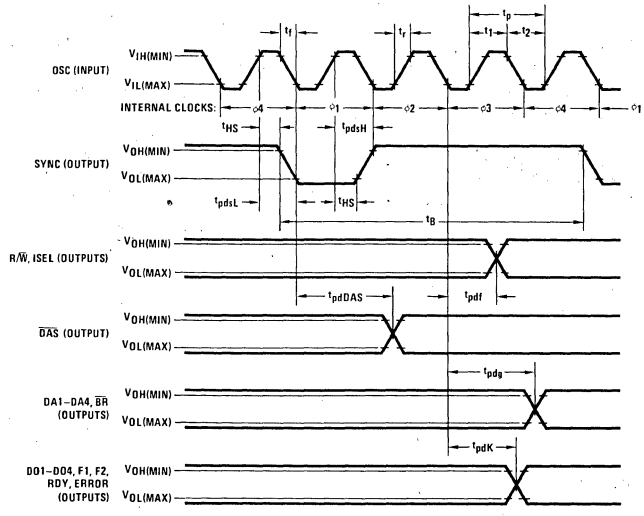


FIGURE 1

Switching Time Waveforms



Note 1: See discussion of timing diagrams on page 5.

Note 2: Osc. Duty Cycle = t_1 $(t_1 + t_2) = t_1/t_p$.

Note 3: The last four timing diagrams indicate that, *if* the output changes, it will change within the indicated time. This is not meant to imply that these signals will change every microcycle. The conditions which will cause the various outputs to change are explained in detail in the functional description section of this manual.

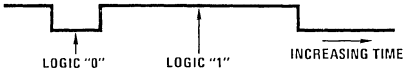
FIGURE 2. Timing Diagrams for Oscillator Input and Chip Outputs

Timing Diagram Conventions

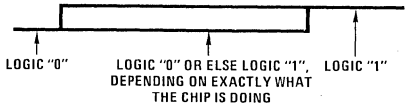
This data sheet makes extensive use of timing diagrams to illustrate electrical and logical characteristics of signal inputs and outputs. To avoid confusion concerning these diagrams, the following conventions have been adopted:

1. Time axis is horizontal, increasing to the right.

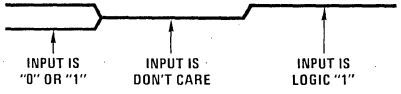
2. Upper side of waveform represents logic "1" (V_{SS}). Lower side of waveform represents logic "0" (V_{DD}).



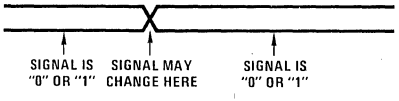
3. Lines appearing simultaneously at both logic "0" and logic "1" indicate that the state of the input or output is either "0" or "1", and does not change during this time. This is used when the logic state depends on exactly what the user is doing with the chip at the time, and thus is unknown to the person drawing the timing waveform.



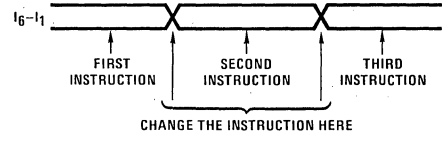
4. Lines located at a level between logic "1" and logic "0" appear on input signals only, and indicate that the state of the input is a don't care during this time.



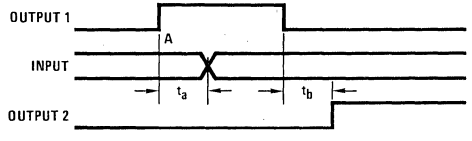
5. An "X" in a waveform indicates that the input or output may change state at this time.



This representation is often used for a group of inputs or outputs whose logic states are unknown, but the time at which a signal may change must be shown. Example:



6. Minimum and/or maximum time values are sometimes specified. The interpretation of these values depends on whether the waveforms are inputs or outputs. Example:



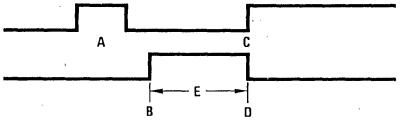
The following table shows three different ways in which these timing diagrams can be interpreted, depending on whether min, max or both are specified.

PARAMETER		MIN	MAX	COMMENT
(A)	t_a	3.7 μs		Input must be changed later than 3.7 μs after point "A"
	t_b		1.9 ms	Output 2 will go high no later than 1.9 ms after output 1 goes low
(B)	t_a		2.4 ns	Input must be changed before 2.4 ns after point "A"
	t_b	0.7 s		Output 2 will not go high until at least 0.7 s after output 1 goes low
(C)	t_a	0.6 μs	3.0 μs	Input must be changed between 0.6 and 3.0 μs after point "A", no sooner and no later
	t_b	0.01 μs	0.9 μs	Output 2 will go high between 0.01 and 0.9 μs after output 1 goes low

This illustrates the various meanings that must be attached to time values, depending on whether they refer to inputs or outputs and whether minimum, maximum or min/max limits are placed on the value.

7. Rise and fall times are measured from maximum logic low to minimum logic high. For example, if the maximum logic low ("0") level ($V_{L(MAX)}$) of a signal is V_{DD} + 1V, and if the minimum logic high ("1") level ($V_{H(MIN)}$) is V_{SS} – 1.5V, then the rise time would be the time it takes the signal to go from V_{DD} + 1V to V_{SS} – 1.5V.

Timing diagrams are seldom shown to scale because of space limitations. However, they do show the proper relationship between waveforms. Consequently, the reader, when studying a timing diagram, should exercise care in understanding what information the timing diagram is meant to show, and ignore the time scale distortions that are necessarily introduced. Example:



Waveform relationships are maintained, so pulse A does come before B, and C and D occur at the same time. However, the time axis may be distorted, so pulse width E may not be twice that of pulse A. It may be 10 times, or even 1000 times wider.

Pinout Description

The MM57109 is intended for microprocessor number processing applications, either as a microcomputer peripheral chip or as a stand-alone processor. *Figure 1* shows a pinout diagram of the MM57109, giving the pin numbers and names of the signal lines. It also shows a functional block diagram illustrating the internal organization of the MM57109 and the origin of the signal lines that are used to communicate with the external world.

The MM57109 operates on a 9V power supply. In order to make it TTL compatible, it can be operated from supplies of 5V and −4V. The signal inputs are designed to respond properly to LPTTL logic levels (with the exception of OSC, HOLD and POR) when the MM57109 is operated in this fashion. (See electrical specifications and *Figure 3* for details on LPTTL interface).

A 400 kHz oscillator operating between V_{DD} and V_{SS} is required. The rise and fall times and frequency of this oscillator are not critical, making it relatively easy to generate. The MM57109 provides a SYNC output, which is a signal that goes active low once every 4 oscillator cycles. A single SYNC pulse corresponds to a single "microcycle" (about 10 μs). The execution of a single MM57109 instruction involves thousands of microcycles. A later section of this manual will contain a tabulation of instruction execution times listed in microcycles.

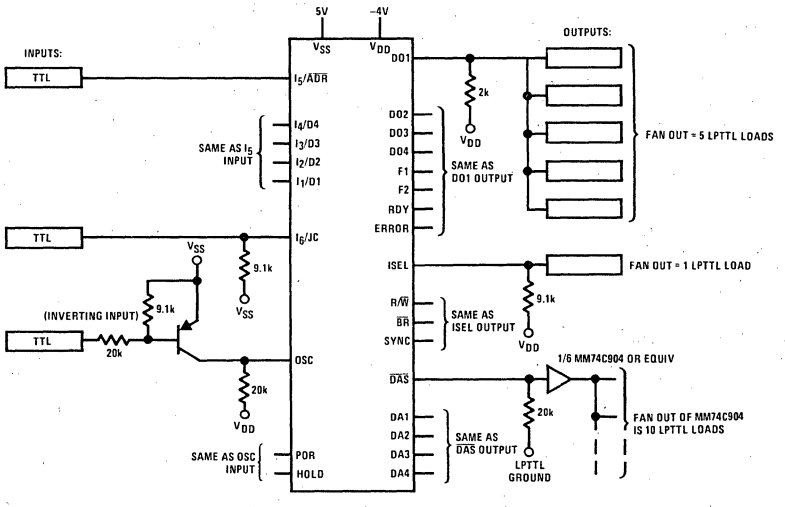
The processor is reset by applying a reset pulse to the POR pin as shown in *Figure 6*. The chip will then set the various outputs to their proper levels and generate three ready pulses (RDY). These ready pulses are designed to provide for automatic processing of an error in stand-alone systems. (See section titled ERROR CONTROL.) A microcomputer system would ignore the first two RDY pulses and use the third one as a "Ready for Instruction" signal.

The MM57109 has 6 instruction inputs (I_6–I_1) which are used to provide it with a 6-bit instruction code (commonly referred to as an "op code"). Each op code corresponds to one of the MM57109 instructions. A list of instructions, their op codes, and a description of the operations they perform will be given later in this manual. The 6 instruction lines are shared by 6 data lines. The output ISEL identifies which function the 6 lines are performing. When ISEL = 1, the 6 lines are instruction lines (I_6–I_1). When ISEL = 0, the 6 lines are data lines (JC, $\overline{ADR}$, D4–D1). In many cases the data lines, which are associated with the IN, AIN, and TJC instructions, will not be used. In these instances ISEL can be ignored. If the data lines are used, ISEL is the select input for six 2–1 multiplexers or an enable input to buffers, latches or ROMs. Later in this manual sample systems will be shown illustrating use of ISEL.

A ready output (RDY) goes high when the processor is ready to read a 6-bit instruction code. This output operates in conjunction with the HOLD input. When RDY goes high, it will remain high if HOLD = 1. If processor instructions are not always ready when RDY goes high, some method must be provided to set HOLD = 1. A microprocessor might have a flag output which holds HOLD = 1 until it is ready to pass an instruction to the processor. (If instructions are always ready when RDY goes high, the HOLD input can be tied to "0".) After RDY goes high, it will wait for HOLD = 0 and then go low again. At this time the 6-bit instruction code is read and the instruction is performed.

The branch output ($\overline{BR}$) is a 4-microcycle active low pulse which signals that the result of a test instruction (e.g. TEST X = 0) was true. This pulse starts prior to RDY = 1 for the next instruction, and ends slightly after RDY = 1.

The four signals RDY, HOLD, ISEL and $\overline{BR}$ were carefully chosen to allow the MM57109 to be used as a stand-alone processor or as a microcomputer peripheral. In a stand-alone system, RDY would be a clock for an external program counter (PC) whose outputs would address a ROM containing the MM57109 instructions. $\overline{BR}$ would parallel load the PC, resulting in a program branch. In a microcomputer system, RDY would inform



Note 1: I_5–I_1 have an internal resistor to V_{SS}. I_6 has an internal resistor to V_{DD}.
Note 2: All inputs must be driven by TTL or LPTTL outputs with no other loads on them.

FIGURE 3. Low Power TTL Interface

Pinout Description (Continued)

the microcomputer that the MM57109 is ready for a new instruction. HOLD would be used to inform the MM57109 that the microcomputer is not ready to respond to the MM57109.

Several instructions have conditions which will cause an error to occur. Table VI enumerates these conditions. When an error does occur, the MM57109 will set the ERROR output high. This output can be tested with the TERR instruction and cleared with the ECLR instruction.

The two outputs F1 and F2 are flags which are set by the instructions SF1 and SF2. They can also be pulsed active high with the instructions PF1 and PF2. These flag outputs could be used as single bit outputs from the MM57109.

The JC input is a general purpose single bit input. A TJC instruction will branch (i.e., result in a true condition causing a $\overline{BR}$ pulse) if the input JC is high. Otherwise the TJC instruction will do nothing.

Table II summarizes this description of the MM57109 signal lines.

INPUTTING DATA

As shown in *Figure 1*, the MM57109 has an internal register file. Each of the 5 registers (X, Y, Z, T and M) has 8 mantissa digits, 2 exponent digits, a decimal point position indicator, and mantissa and exponent sign bits. Instructions operate on these registers. The instructions IN and OUT input and output numbers to and from the X register. There are two possible modes of operation for IN and OUT instructions. Floating point mode transfers mantissa digits, a mantissa sign digit, and a decimal point position digit. Scientific notation mode transfers mantissa digits, 2 exponent digits, a digit containing mantissa and exponent sign bits, and a decimal point position indicator. Initially the MM57109 is in the floating point mode. The TOGM instruction toggles to the opposite mode. The number of mantissa digits input or output by an IN or OUT instruction is equal to the mantissa digit count (MDC). The MDC is initially 8 and can be set to any value from 1 to 8 using the SMDC instruction. When an IN or OUT instruction is executed, the four DA outputs will sequence through values indicating which digit is to be input or output. The section of this manual entitled DATA FORMATS gives the values of the DA lines for each of the digits input or output by an IN or OUT instruction. During an OUT instruction, the four DO outputs provide the digit outputs, coded in BCD. The R/$\overline{W}$ output is pulsed active low once for each digit. This R/$\overline{W}$ pulse can be used to write the data into a RAM or clock it into a latch. During an IN instruction, the four I lines (I_4–I_1), are data input lines for the digits to be input (and so are also named D4–D1). The same data format is used for IN as is used for OUT. The $\overline{DAS}$ output is pulsed active low prior to reading each digit. This $\overline{DAS}$ pulse can be used as a data request signal or to clock data into a latch.

The IN and OUT instructions have been designed to allow easy expansion of the internal register file. A 256 x 4 RAM will add an additional 16 registers for data storage. The DA lines are used to provide part of the RAM address. The rest of the address, which would specify one of the 16 registers, comes from the external instruction storage (ROM, microprocessor, etc). The DO lines are the input to the RAM, while the RAM outputs are multiplexed to the I lines, using ISEL to select between instructions or data. The processor R/$\overline{W}$ line is the RAM R/$\overline{W}$ signal.

There are three ways to input data to the MM57109. The first is the IN instruction which has already been described. Second is the AIN instruction, which inputs a single digit into the X register. Multiple AIN instructions will input more than one digit to the X register, since the AIN instruction does not cause termination of the number entry mode (number entry mode will be fully described later in this manual). The DA lines provide a digit address from 0 to 7 for multiple AIN instructions. The $\overline{ADR}$ input (shared with I_6) is a data hold signal for AIN. If $\overline{ADR}$ is high during an AIN instruction, the processor will wait till it goes low, and then read the digit on D4–D1. Finally, the F2 output of the MM57109 will be pulsed active low as a read acknowledge signal. Note that only mantissa digits, not exponents or signs, can be entered with AIN.

For systems using a microcomputer with the MM57109 as a peripheral, it is likely that neither the IN nor the AIN instruction would be used. Instead, the third method of inputting data to the processor would be used. This method involves entering numbers as instructions. Using the instructions "0", "1", "2", . . . "9", the decimal point instruction, etc., a number can be entered directly into the processor in the same manner as one presses keys to enter numbers into a calculator. The decimal point is to the right of the last digit entered unless it was fixed by a DP instruction. The EE instruction causes the next digits to be entered into the exponent. The CS instruction changes the sign of the mantissa (or exponent, if EE instruction was entered).

2-WORD INSTRUCTIONS

Several instructions are 2-word instructions, of which there are 4 types. Each type generates two RDY pulses, one for each word. The first type are the inverse instructions (inverse SIN, COS, TAN and inverse +, –, x, / for memory operations). These instructions require that the INV instruction first be executed, followed by the desired instruction (SIN, COS, etc.). The second type is the SMDC instruction. The second word of this instruction is the mantissa digit count, a BCD number from 1 to 8. The third type is the IN and OUT instructions. The second word of these instructions is a high order address for a RAM or a device select code. It is not necessary to use the second word of IN or OUT instructions because the MM57109 ignores it, providing only a RDY pulse that may or may not be used by external hardware. The final type of 2-word instructions are the branch instructions. The second word of these instructions is intended to be a branch address to be loaded into an external program counter in stand-alone systems. For a microprocessor system, the second RDY pulse can be used to clock the $\overline{BR}$ output into a latch. The latch can then be tested to discover if the branch condition was true ($\overline{BR}$ = 0) or false ($\overline{BR}$ = 1). Many microcomputer applications will not use the branch instructions, since testing and branching is often more easily done within the external microprocessor itself.

Pinout Description (Continued)

TABLE II. MM57109 PIN DESCRIPTION

ABBREVIATED NAME	PIN NUMBER	FUNCTIONAL NAME	DESCRIPTION
V_{SS}, V_{DD}	15, 21	V_{SS}, V_{DD} (Power Supply)	V_{SS} = V_{DD} + 9V nominally (see electrical specifications) (V_{SS} = Logic "1", V_{DD} = Logic "0").
POR	11	Power On Reset (Input)	An 8-microcycle or longer active high pulse at this input will initialize the MM57109. It then sets R/$\overline{W}$ = 1, other outputs = 0, and generates three RDY pulses before reading first instruction. HOLD must be 0 to complete each RDY pulse.
OSC	7	Oscillator (Input)	Single ϕ clock with frequency 4X microcycle time. Typical frequency is 400 kHz.
SYNC	6	Sync (Output)	Active low output pulse once each microcycle.
RDY	12	Ready (Output)	Rising edge indicates processor is ready to execute next instruction or get second word of 2-word instruction. If HOLD = 0, RDY goes low again and next instruction is executed. If HOLD = 1, RDY stays high until HOLD = 0. (See *Figure 8*). RDY can be used to clock an external program counter or to request an instruction from another CPU.
HOLD	9	Hold (Input)	When set high prior to or at the rising edge of RDY, RDY will be held high and instruction execution delayed until HOLD is set low.
$\overline{BR}$	23	Branch (Output)	A 4-microcycle active low pulse indicates a program branch. RDY goes high during this pulse. $\overline{BR}$ may be used as a load signal for an external PC or as a sense input to a microprocessor.
ISEL	8	Instruction Select (Output)	Selects 6 bit instruction code (ISEL = 1) or JC, $\overline{ADR}$, D4–D1 (ISEL = 0) on I_6–I_1 (the 6 input lines).
R/$\overline{W}$	10	Read/$\overline{Write}$ (Output)	Pulsed active low during OUT instruction to write data digits into a RAM or register. Address and data are valid at both edges. R/$\overline{W}$ is also pulsed during a PRW1 or PRW2 instruction.
I_6, JC	24	Input 6, Jump Condition (Input)	Most significant instruction bit when ISEL = 1. Jump condition for TJC instruction when ISEL = 0. (JC = 1 indicates jump condition true.)
I_5, $\overline{ADR}$	5	Input 5, AIN Data Ready (Input)	Instruction bit 5 when ISEL = 1. AIN Data Ready ($\overline{ADR}$) for AIN instruction when ISEL = 0, ($\overline{ADR}$ = 0 for data ready).
I_4–I_1, D4–D1	4, 3, 2, 1	Inputs 4–1, Data 4–1 (Inputs)	Instruction bits 4–1, or mantissa digit count on second word of SMDC instruction, when ISEL = 1. Digit data (AIN or IN instructions) when ISEL = 0. Bit 4 is the most significant bit.
DA4–DA1	25, 26, 27, 28	Digit Address 4–1 (Outputs)	Digit address for AIN, IN, and OUT instructions. Used as multiplex selector (AIN) or as low order address (IN, OUT) for RAM or other I/O device. Bit 4 is the most significant bit. Set to 0 after each IN, OUT or AIN instruction.
$\overline{DAS}$	22	Digit Address Strobe (Output)	Active low pulse indicates digit address is changing. New address is valid on second (positive-going) edge.
DO4–DO1	20, 19, 18, 17	Digit Outputs 4–1 (Outputs)	BCD digit output for OUT instruction. Set to 0 after each OUT instruction. Bit 4 is the most significant bit.
F1	16	Flag 1 (Output)	User controlled flag can be set or pulsed (reset if set).
F2	14	Flag 2 (Output)	User controlled flag can be set or pulsed (reset if set). Active low pulse (set if reset) generated after each AIN data read. This can be used as an acknowledge signal to clear a flip-flop.
ERROR	13	Error Flag (Output)	Set on an arithmetic or OUT error. Reset by ECLR instruction. See ERROR CONTROL for more information.

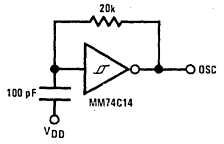
Functional Description

OSCILLATOR GENERATION

Figure 4 shows a simple circuit for generation of the MM57109 oscillator.

INITIALIZATION SEQUENCE

Figure 5 shows a flowchart and *Figure 6* a timing diagram of the MM57109 initialization sequence which occurs when the POR input is set high for at least 8 clock periods.



$f_{OSC} \cong 400$ kHz

$T_{OSC} \cong 2.5\ \mu s$

FIGURE 4. MM57109 Oscillator Circuit

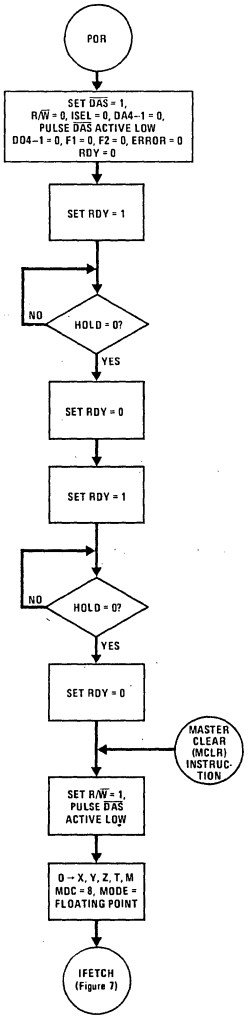
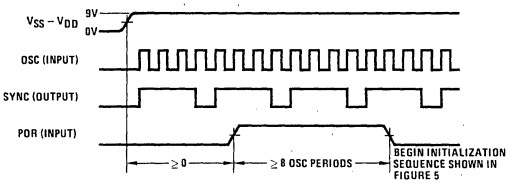
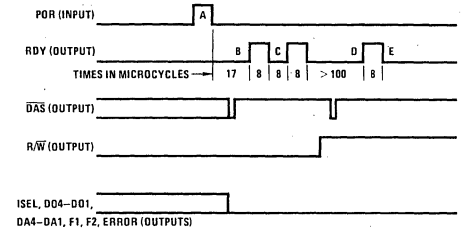


FIGURE 5. MM57109 Initialization Flowchart



(a) POR Sequence for Initialization



Note: Hold = 0

POINTS	DESCRIPTION (Assume External PC)	
A	Initialize PC to 0	Error Recovery Locations
B	Advance PC to location 1	
C	Advance PC to location 2	
D	Advance PC to location 3, MM57109 initialized	
E	Begin execution of instruction at location 3	

(b) Output Timing During Initialization

FIGURE 6. MM57109 Initialization Timing Diagram

Functional Description (Continued)

INSTRUCTION FETCH AND EXECUTION

Figure 7 shows a flowchart and *Figure 8* shows a timing diagram of the instruction fetch and execution sequence.

After initialization (POR) or the completion of an instruction, the processor raises RDY to signal the instruction store device that the processor is ready for the next instruction word. The instruction store device could be a semiconductor memory, host CPU, or an asynchronous device of some kind. If the instruction store is not ready to respond within the required access time (8 microcycles), it must raise the HOLD input to delay the instruction word fetch. HOLD may be set high any time while RDY is low, or at the leading edge of RDY. When HOLD goes low the processor will lower RDY and begin instruction execution. The instruction word must remain valid while RDY is low.

During program branches, skips, or fetching of the second word of a 2-word instruction, the RDY/HOLD sequence is the same as discussed above. (See flowchart in *Figure 7* and timing diagram in *Figure 8(e)*.)

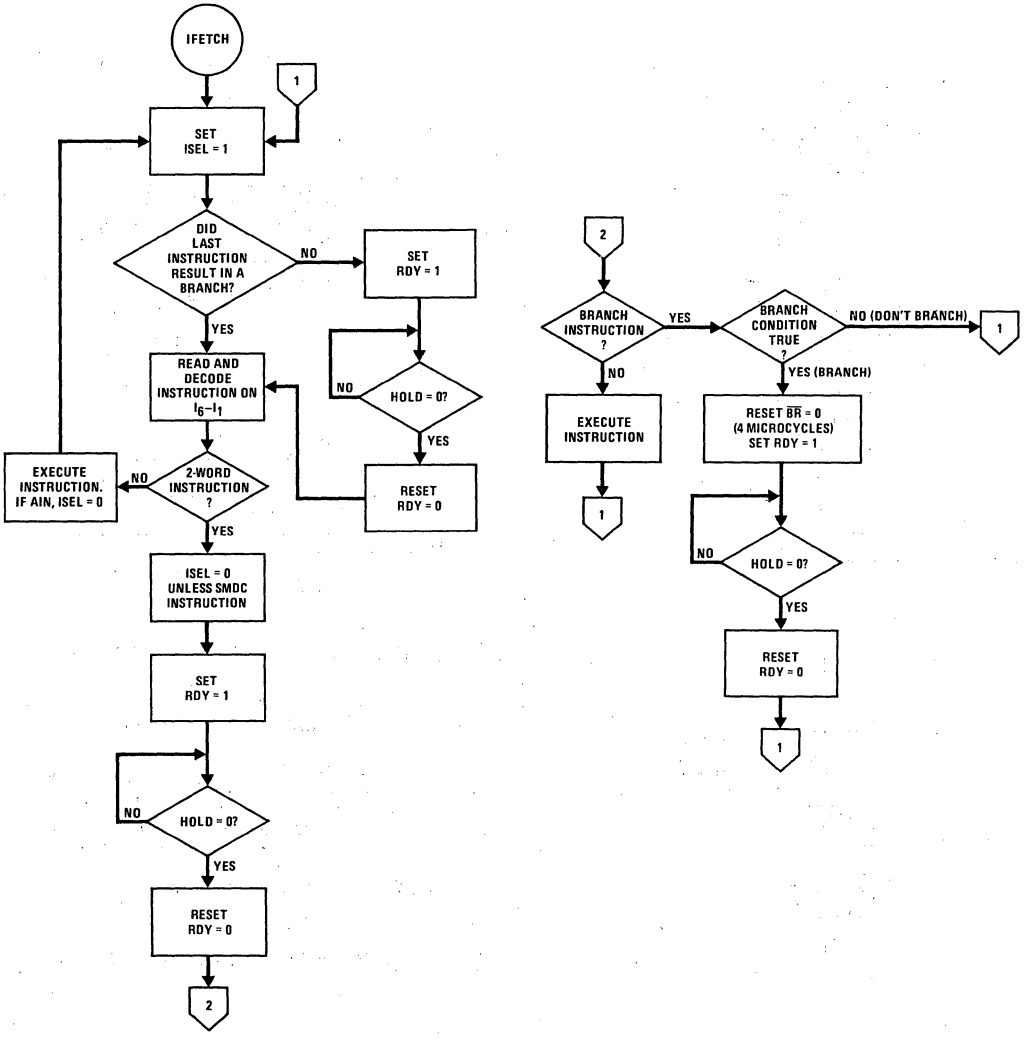
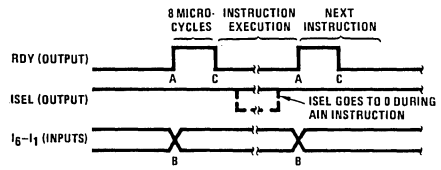


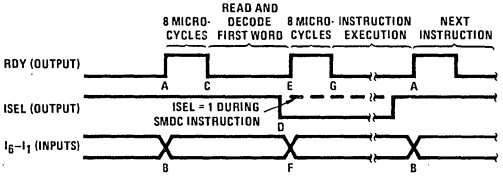
FIGURE 7. MM57109 Instruction Fetch and Execution Flowchart

Functional Description (Continued)



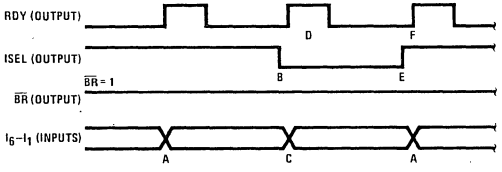
POINTS	DESCRIPTION
A	RDY goes high
B	Next instruction becomes available on I lines
C	RDY goes low, instruction is read and executed by MM57109

(a) 1-Word Instruction with HOLD = 0



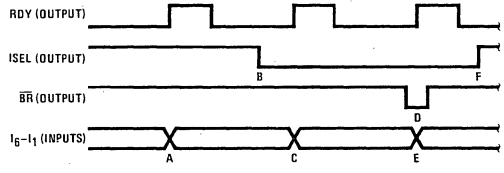
POINTS	DESCRIPTION
A	RDY goes high for first instruction word
B	First instruction word becomes available on I lines
C	RDY goes low. Instruction is read and decoded by MM57109
D	ISEL goes low for second instruction word
E	RDY goes high for second instruction word
F	Second instruction word becomes available on I lines
G	Instruction is executed by MM57109

(b) 2-Word Instruction with HOLD = 0



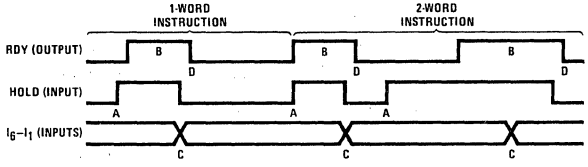
POINTS	DESCRIPTION
A	First word of next instruction becomes available on I lines
B	ISEL = 0 for second word
C	Second word (branch address) becomes available to external program counter
D	Data becomes available on JC for TJC instruction
E	ISEL = 1 for next instruction
F	RDY pulse for next instruction occurs

(c) 2-Word Branch Instruction that Doesn't Branch (HOLD = 0)



POINTS	DESCRIPTION
A	First word of branch instruction becomes available on I lines
B	ISEL = 0 for second word
C	Second word (branch address) becomes available to external program counter
D	Branch condition is true so 4-microcycle active low $\overline{BR}$ pulse occurs and RDY is pulsed with leading edge occurring during $\overline{BR}$ pulse. This clocks branch address into program counter
E	New instruction at branch address becomes available to MM57109
F	ISEL = 1 for next instruction. RDY pulse that usually follows ISEL = 1 is suppressed

(d) 2-Word Branch Instruction that Does Branch (HOLD = 0)



POINTS	DESCRIPTION
A	HOLD = 1 at or prior to RDY leading edge (RDY leading edge may be used to set HOLD = 1)
B	RDY stays high until HOLD = 0
C	When HOLD goes low, next instruction word must be available on I lines
D	RDY goes low after HOLD goes low

(e) RDY-HOLD Relationship for 1 and 2-Word Instructions

FIGURE 8. MM57109 Instruction Fetch and Execution Timing Diagrams

Functional Description (Continued)

MM57109 INSTRUCTION SET

The MM57109 has 70 instructions available to the user. The 70 instructions are classed into digit entry, move, math, clear, branch, input/output, and mode control instructions. Table III contains a detailed description of these instructions. *Figure 9* shows the instruction format. Table IV contains a summary of the instructions. Note that all arithmetic instructions operate on 8-digit mantissa/2-digit exponent numbers, regardless of mantissa digit count or notation mode. Accuracy of all instructions is 7 digits. Computations are performed internally with 9 digits, then rounded to 8. The eighth digit is accurate to ±5.

NUMBER ENTRY

When a digit, decimal point, or π is entered with an AIN, 0–9, DP, or PI instruction, the stack is first pushed and the X register cleared: $Z \rightarrow T$, $Y \rightarrow Z$, $X \rightarrow Y$, $0 \rightarrow X$. This process is referred to as "initiation of number entry." Following this, the entered digit and future digits are loaded into the X mantissa. Subsequent entry of digits or DP, EE, or CS instructions do not cause initiation of number entry. Digits following the eighth mantissa digit are ignored. (CAUTION: An internal error will occur if more than 8 digits are entered with AIN instruction, or if a non-BCD digit is entered. Note that the ERROR flag with *not* be set if this happens. A POR sequence will be necessary to restart the processor.) This number entry mode is terminated by any instruction except 0–9, DP, EE, CS, PI, AIN or HALT. Termination of number entry means two things. First, the number is normalized by adjusting the exponent and decimal point position so that the decimal point is to the right of the first mantissa digit. Second, the next digit, decimal point, or π entered will again cause initiation of number entry, as already described. There is one exception to this number entry initiation rule: the stack is *not* pushed if the instruction prior to the entered digit was an ENTER. However, the X register is still cleared and the entered digit put in X.

The IN instruction enters *all* digits of a number. Therefore, IN does not cause initiation of number entry. However, it does terminate number entry mode if the processor is in this mode before the IN instruction is executed. This means the user can mix 0–9, AIN and IN instructions without performing an ENTER before an IN.

The IN instruction will always push the stack prior to inputting digits unless the previous instruction was ENTER. This allows multiple IN instructions to be executed without performing an ENTER between them.

INSTRUCTION TIMING

Table V shows execution times of each instruction. These times are shown in microcycles, 1 microcycle being equal to 1 SYNC period (approximately 10 μs). *Figure 10* shows timing diagrams illustrating the dynamic characteristics of execution of each type of instruction, assuming HOLD = 0.

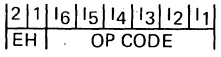
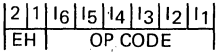
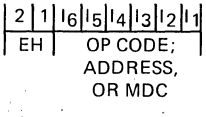
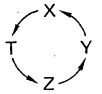
SINGLE WORD INSTRUCTIONS	2-WORD INSTRUCTIONS
	 
The "OP CODE" is a 6-bit operation code (see Table III) which specifies which instruction is to be performed. The MM57109 requires only the OP CODE to function. However, memory devices with 8-bit word sizes are often used. The extra two bits, here designated "EH" for "external hardware", could be used for device selection on AIN instructions, etc.	The first word has the same format as a 1-word instruction. The function and format of the second word depends on the OP CODE of the first word: (a) First word = SMDC instruction. The second word contains the MDC (1–8). (b) First word = IN or OUT instruction. The second word contains a high-order address for RAM or I/O device (low-order address from DA lines). (c) First word = INV instruction. The second word contains the second OP CODE for the instruction (M+, M–, MX, M/, SIN^{-1}, COS^{-1}, TAN^{-1}). (d) First word = branch instruction. The second word contains the branch address to be loaded into PC on branch.

FIGURE 9. MM57109 Instruction Format

Functional Description (Continued)

TABLE III. MM57109 INSTRUCTION DESCRIPTION TABLE (* INDICATES 2-WORD INSTRUCTION)

CLASS	SUBCLASS	MNEMONIC*	OCTAL OP CODE	FULL NAME	DESCRIPTION
Digit Entry		0	00	0	Mantissa or exponent digits. On first digit (d) the following occurs: Z → T, Y → Z, X → Y, d → X. See description of number entry on page 12.
		1	01	1	
		2	02	2	
		3	03	3	
		4	04	4	
		5	05	5	
		6	06	6	
		7	07	7	
		8	10	8	
		9	11	9	
		DP	12	Decimal Point	Digits that follow will be mantissa fraction.
		EE	13	Enter Exponent	Digits that follow will be exponent.
		CS	14	Change Sign	Change sign of exponent or mantissa. Xm = X mantissa. Xe = X exponent. CS causes −Xm → Xm or −Xe → Xe depending on whether or not an EE instruction was executed after last number entry initiation.
		PI	15	Constant π	3.1415927 → X, stack not pushed.
		EN	41	Enter	Terminates digit entry and pushes the stack. The argument entered will be in X and Y. Z → T, Y → Z, X → Y
		NOP	77	No Operation	Do nothing instruction that will terminate digit entry.
		HALT	17	Halt	External hardware detects HALT op code and generates HOLD = 1. Processor waits for HOLD = 0 before continuing. HALT acts as a NOP and may be inserted between digit entry instructions since it does not terminate digit entry.
Move		ROLL	43	Roll	Roll Stack. 
		POP	56	Pop	Pop Stack. Y → X, Z → Y, T → Z, 0 → T
		XEY	60	X exchange Y	Exchange X and Y. X ⟷ Y
		XEM	33	X exchange M	Exchange X with memory. X ⟷ M
		MS	34	Memory Store	Store X in Memory. X → M
		MR	35	Memory Recall	Recall Memory into X. Stack is pushed. M → X → Y → Z → T
		LSH	36	Left Shift Xm	X mantissa is left shifted while leaving decimal point in same position. Former most significant digit is saved in link digit. Least significant digit is zero. Former link digit is lost.
		RSH	37	Right Shift Xm	X mantissa is right shifted while leaving decimal point in same position. Link digit, which is normally zero except after a left shift, is shifted into the most significant digit. Least significant digit is lost.

Functional Description (Continued)

TABLE III. MM57109 INSTRUCTION DESCRIPTION TABLE (CONTINUED) (*INDICATES 2-WORD INSTRUCTION)

CLASS	SUBCLASS	MNEMONIC*	OCTAL OP CODE	FULL NAME	DESCRIPTION
Math	F (X,Y)	+	71	Plus	Add X to Y. X + Y → X. On +, −, x, / and YX instructions, stack is popped as follows: Z → Y, T → Z, O → T. Former X, Y are lost.
		−	72	Minus	Subtract X from Y. Y − X → X
		x	73	Times	Multiply X times Y. Y x X → X
		/	74	Divide	Divide X into Y. Y ÷ X → X
		YX	70	Y to X	Raise Y to X power. $Y^X \rightarrow X$
	F (X,M)	INV +*	40, 71	Memory Plus	Add X to memory. M + X → M. On INV +, −, x and / instructions, X, Y, Z, and T are unchanged. Former M is lost.
		INV −*	40, 72	Memory Minus	Subtract X from memory. M − X → M
		INV x*	40, 73	Memory Times	Multiply X times memory. M x X → M
		INV /*	40, 74	Memory Divide	Divide X into memory. M ÷ X → M
	F (X) Math	1/X	67	One Divided by X	1 ÷ X → X. On all F (X) math instructions Y, Z, T and M are unchanged and previous X is lost.
		SQRT	64	Square Root	$\sqrt{X} \rightarrow X$
		SQ	63	Square	$X^2 \rightarrow X$
		10X	62	Ten to X	$10^X \rightarrow X$
		EX	61	e to X	$e^X \rightarrow X$
		LN	65	Natural log of X	ln X → X
		LOG	66	Base 10 log of X	log X → X
	F (X) Trig	SIN	44	Sine X	SIN(X) → X. On all F(X) trig functions, Y, Z, T, and M are unchanged and the previous X is lost.
		COS	45	Cosine X	COS(X) → X
		TAN	46	Tangent X	TAN(X) → X
		INV SIN*	40, 44	Inverse sine X	$SIN^{-1}(X) \rightarrow X$
		INV COS*	40, 45	Inverse cosine X	$COS^{-1}(X) \rightarrow X$
		INV TAN*	40, 46	Inverse tan X	$TAN^{-1}(X) \rightarrow X$
		DTR	55	Degrees to radians	Convert X from degrees to radians.
		RTD	54	Radians to degrees	Convert X from radians to degrees.
Clear		MCLR	57	Master Clear	Clear all internal registers and memory; initialize I/O control signals, MDC = 8, MODE = floating point. (See INITIALIZATION.)
		ECLR	53	Error flag clear	O → Error flag
Branch	Test	JMP*	25	Jump	Unconditional branch to address specified by second instruction word. On all branch instructions, second word contains branch address to be loaded into external PC.
		TJC*	20	Test jump condition	Branch to address specified by second instruction word if JC (I_6) is true (=1). Otherwise, skip over second word.
		TERR*	24	Test error flag	Branch to address specified by second instruction word if error flag is true (=1). Otherwise, skip over second word. May be used for detecting specific errors as opposed to using the automatic error recovery scheme dealt with in the section on Error Control.
		TX = 0*	21	Test X = 0	Branch to address specified by second instruction word if X = 0. Otherwise, skip over second word.
		TXF*	23	Test $\|X\| < 1$	Branch to address specified by second instruction word if $\|X\| < 1$. Otherwise, skip over second word. (i.e. branch if X is a fraction.)
		TXLT0*	22	Test $X < 0$	Branch to address specified by second instruction word if $X < 0$. Otherwise, skip over second word.

Functional Description (Continued)

TABLE III. MM57109 INSTRUCTION DESCRIPTION TABLE (CONTINUED) (* INDICATES 2-WORD INSTRUCTION)

CLASS	SUBCLASS	MNEMONIC*	OCTAL OP CODE	FULL NAME	DESCRIPTION
Branch	Count	IBNZ	31	Increment memory and branch if $M \neq 0$	$M + 1 \rightarrow M$. If $M = 0$, skip second instruction word. Otherwise, branch to address specified by second instruction word.
		DBNZ	32	Decrement memory and branch if $M \neq 0$	$M - 1 \rightarrow M$. If $M = 0$, skip second instruction word. Otherwise, branch to address specified by second instruction word.
I/O	Multi-digit	IN*	27	Multidigit input to X	The processor supplies a 4-bit digit address (DA4–$\overline{DA1}$) accompanied by a digit address strobe ($\overline{DAS}$) for each digit to be input. The high order address for the number to be input would typically come from the second instruction word. The digit is input on D4–D1, using ISEL = 0 to select digit data instead of instructions. The number of digits to be input depends on the calculation mode (scientific notation or floating point) and the mantissa digit count (See DATA FORMATS and INSTRUCTION TIMING). Data to be input is stored in X and the stack is pushed ($X \rightarrow Y \rightarrow Z \rightarrow T$). At the conclusion of the input, DA4–DA1 = 0.
		OUT*	26	Multidigit output from X	Addressing and number of digits is identical to IN instruction. Each time a new digit address is supplied, the processor places the digit to be output on DO4–DO1 and pulses the R/$\overline{W}$ line active low. At the conclusion of output, DO4–DO1 = 0 and DA4–DA1 = 0.
I/O	Single-digit	AIN	16	Asynchronous Input	A single digit is read into the processor on D4–D1. ISEL = 0 is used by external hardware to select the digit instead of instruction. It will not read the digit until $\overline{ADR}$ = 0 (ISEL = 0 selects $\overline{ADR}$ instead of I_5), indicating data valid. F2 is pulsed active low to acknowledge data just read.
I/O	Flags	SF1	47	Set Flag 1	Set F1 high, i.e. F1 = 1.
		PF1	50	Pulse Flag 1	F1 is pulsed active high. If F1 is already high, this results in it being set low.
		SF2	51	Set Flag 2	Set F2 high, i.e. F2 = 1.
		PF2	52	Pulse Flag 2	F2 is pulsed active high. If F2 is already high, this results in it being set low.
		PRW1	75	Pulse R/$\overline{W}$ 1	Generates R/$\overline{W}$ active low pulse which may be used as a strobe or to clock extra instruction bits into a flip-flop or register.
		PRW2	76	Pulse R/$\overline{W}$ 2	Identical to PRW1 instruction. Advantage may be taken of the fact that the last 2 bits of the PRW1 op code are 01 and the last 2 bits of the PRW2 op code are 10. Either of these bits can be clocked into a flip-flop using the R/$\overline{W}$ pulse.
Mode Control		TOGM	42	Toggle Mode	Change mode from floating point to scientific notation or vice-versa, depending on present mode. The mode affects only the IN and OUT instructions. Internal calculations are always in 8-digit scientific notation.
		SMDC*	30	Set Mantissa Digit Count	Mantissa digit count is set to the contents of the second instruction word (=1 to 8).
		INV*	40	Inverse Mode	Set inverse mode for trig or memory function instruction that will immediately follow. Inverse mode is for next instruction only.

Functional Description (Continued)

TABLE IV. MM57109 INSTRUCTION SUMMARY TABLE (* INDICATES 2-WORD INSTRUCTION)

I_4–I_1	I_6I_5 = 00	01	10	11
0000	0	TJC*	INV*	XEY
0001	1	TX=0*	EN	EX
0010	2	TXLT0*	TOGM	10X
0011	3	TXF*	ROLL	SQ
0100	4	TERR*	SIN (SIN^{-1}*)	SQRT
0101	5	JMP*	COS (COS^{-1}*)	LN
0110	6	OUT*	TAN (TAN^{-1}*)	LOG
0111	7	IN*	SF1	1/X
1000	8	SMDC*	PF1	YX
1001	9	IBNZ*	SF2	+ (M+*)
1010	DP	DBNZ*	PF2	– (M–*)
1011	EE	XEM	ECLR	x (MX*)
1100	CS	MS	RTD	/ (M/*)
1101	PI	MR	DTR	PRW1
1110	AIN	LSH	POP	PRW2
1111	HALT	RSH	MCLR	NOP

Note 1: HALT is same as NOP except it does not terminate number entry. External hardware must generate HOLD = 1 to halt.

Note 2: ISEL = 0 for AIN, all 2-word instructions except SMDC.

Note 3: All instructions with I_6 I_5 = 00, do not terminate number entry. Other instructions do terminate number entry.

TABLE V. INSTRUCTION EXECUTION TIMES

INSTRUCTION MNEMONIC	EXECUTION TIME (MICROCYCLES) (AVERAGE)	EXECUTION TIME (MICROCYCLES) (WORST-CASE VALUES)	INSTRUCTION MNEMONIC	EXECUTION TIME (MICROCYCLES) (AVERAGE)	EXECUTION TIME (MICROCYCLES) (WORST-CASE VALUES)
0–9		238	OUT		583
DP		152	IN		395
EE		151	SF1		163
CS		166	PF1		185
PI		1312	SF2		163
HALT		134	PF2		185
AIN		284	PRW1		130
TJC		208	PRW2		130
TX=0		278	SIN	56200	95900
TXLT0		197	COS	56200	95900
TXF		277	TAN	35000	97600
TERR		191	INV SIN	54000	93900
JMP		186	INV COS	54000	93900
IBNZ		2314	INV TAN	30200	92900
DBNZ		2314	LN	24800	92000
SMDC		163	LOG	30700	92600
XEM		812	EX	30800	93900
MS		839	10X	27400	96500
MR		1385	+, –	2200	6600
LSH		168	INV+, INV–	1700	5000
RSH		173	(M+, M–)		
INV		166	x	3200	22700
EN		552	INV x (MX)	2700	21400
TOGM		157	/	7800	22300
ROLL		905	INV / (M/)	7300	21100
ECLR		163	1/X	4500	22800
POP		448	YX	55400	95500
MCLR		734	SQRT	7000	30200
XEY		652	SQ	3000	21900
NOP		122	DTR, RTD	9600	41700

Note 1: All times are measured from leading edge of ready for first word of the instruction to leading edge of ready for first word of the next instruction. (Hold = 0).

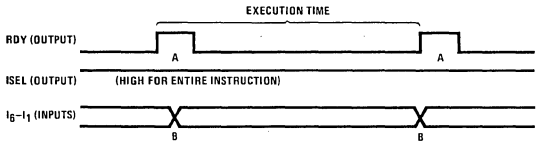
Note 2: Add 67 microcycles to the execution time of any instruction which initiates number entry and is preceded by an ENTER instruction.

Note 3: Add 282 microcycles to the execution time of any instruction which initiates number entry and is not preceded by an ENTER instruction.

Note 4: Add 1003 microcycles to the execution time of any instruction which terminates number entry.

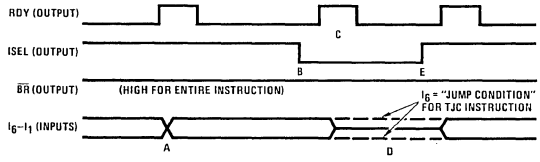
Note 5: The execution time of each instruction is a function of the internal state of the device and is not necessarily related to the number of digits in the operand. It is not possible to predict precisely what the execution time will be for any given instruction. This table shows worst-case values for basic instructions, and both average and worst-case values for mathematical instructions.

Functional Description (Continued)



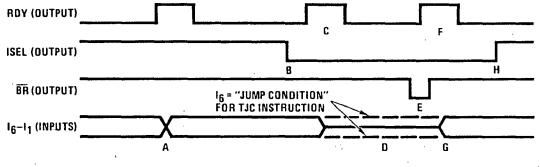
POINTS	DESCRIPTION
A	RDY pulsed to indicate ready for next instruction.
B	Instruction is placed on I lines before RDY goes low.

(a) Timing Characteristics for All Instructions Except AIN, HALT, TJC, TX = 0, TXLT0, TXF, TERR, JMP, OUT, IN, SMDC, IBNZ, DBNZ, PF1, PF2, SF1, SF2, PRW1, PRW2, and ECLR



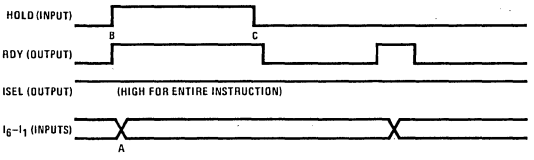
POINTS	DESCRIPTION
A	Instruction is placed on I lines.
B	ISEL goes low for second word.
C	Second instruction word (= branch address) becomes available to external program counter.
D	I lines at this time are don't care, (exception: for TJC instruction I_6 (= JC) must contain the jump condition signal during this time).
E	ISEL goes high prior to RDY pulse for next instruction.

(b) Timing Characteristics for TJC, TX = 0, TXLT0, TXF, TERR, IBNZ, and DBNZ with Branch Condition False



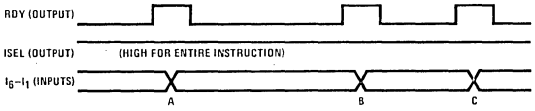
POINTS	DESCRIPTION
A	Instruction is placed on I lines.
B	ISEL goes low for second word.
C	Second instruction word (= branch address) becomes available to external program counter.
D	I lines at this time are don't-care, (exception: for TJC instruction I_6 (= JC) must contain the jump condition signal during this time).
E	4-microcycle $\overline{BR}$ active low pulse provides load signal for external program counter.
F	RDY leading edge occurs during $\overline{BR}$ pulse, thus loading program counter with branch address.
G	Next instruction appears on I lines prior to RDY going low.
H	ISEL goes high for first word of next instruction, RDY pulse that usually occurs at this time is suppressed.

(c) Timing Characteristics for TJC, TX = 0, TXLT0, TXF, TERR, IBNZ, and DBNZ with Branch Condition True. Also for JMP Instruction.



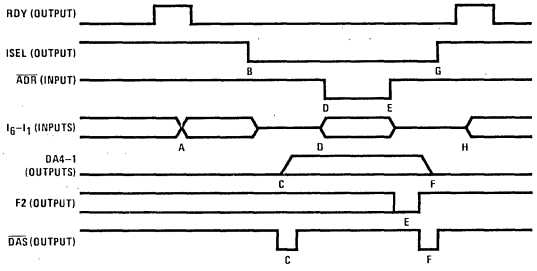
POINTS	DESCRIPTION
A	Halt instruction is placed on I lines.
B	External hardware decodes halt instruction and puts HOLD high. (This is necessary only if the user wishes to have a HALT instruction).
C	HOLD is released, RDY goes low, HALT instruction is executed as a do-nothing instruction.

(d) Timing Characteristics for HALT Instruction



POINTS	DESCRIPTION
A	SMDC instruction is placed on I lines.
B	At second RDY pulse second word of SMDC instruction is placed on I lines. This word is the new MDC. (1–8).
C	Next instruction is placed on I lines.

(e) Timing Characteristics for SMDC Instruction

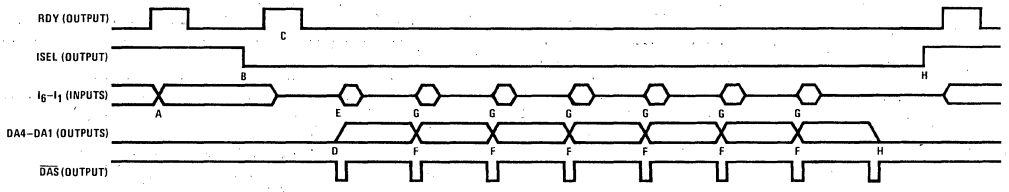


POINTS	DESCRIPTION
A	AIN instruction is placed on I lines.
B	ISEL goes low to select data digit on I lines. I_6 and I_4–I_1 are don't care as long as $\overline{ADR}$ (I_5) = 1.
C	Digit address appears on DA lines. $\overline{DAS}$ provides 1 microcycle active low pulse which frames DA change. (Negative going edge occurs while DA lines are still 0. Positive going edge occurs after DA lines have changed to their new value). The first AIN instruction will have a digit address of 0000. Consecutive AIN instructions have digit addresses of 0001, 0010, etc., up to 0111. Instructions which terminate number entry reset the internal digit address to 0000.
D	Data digit is placed on D4–D1. $\overline{ADR}$ goes low indicating valid data digit.
E	F2 is pulsed active low to indicate read of data on D4–D1. $\overline{ADR}$ (= I_5) must be low for this pulse to occur. If $\overline{ADR}$ is high, the MM57109 will wait till it goes low before reading the I lines. $\overline{ADR}$ may go high again anytime after negative going edge of F2. I lines are don't care after F2 is pulsed.
F	DA lines are reset to 0. $\overline{DAS}$ provides a 1 microcycle pulse framing DA change.
G	ISEL goes high prior to RDY pulse for next instruction.
H	Next instruction appears on I lines.

(f) Timing Characteristics for AIN Instruction

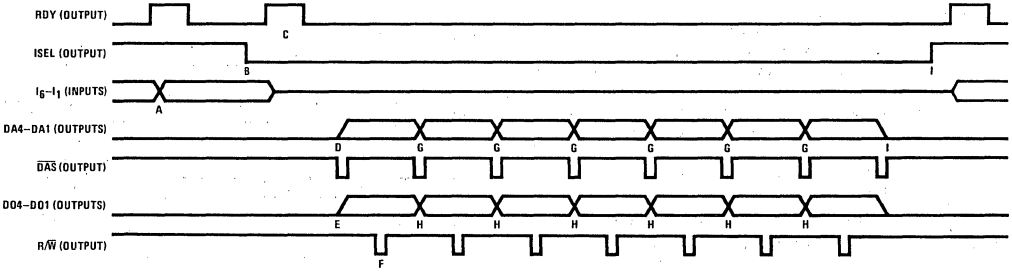
FIGURE 10. Instruction Timing Diagrams

Functional Description (Continued)



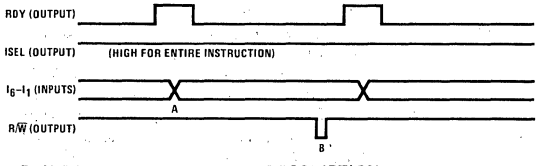
POINTS	DESCRIPTION
A	IN instruction is placed on I lines.
B	ISEL goes low for second instruction word. Data digits multiplexed onto I lines when ISEL = 0. I lines are don't care until $\overline{DAS}$ pulse occurs.
C	Second word of instruction becomes available to external hardware as a high-order address for a RAM or other device.
D	First digit address appears on DA lines. This digit address is 0 or 2 depending on whether mode is scientific notation or floating point, respectively. Each time a new digit address appears, a 1 microcycle active low pulse on $\overline{DAS}$ frames this change so that at the negative going edge of $\overline{DAS}$ the old digit address is valid while at the positive going edge of $\overline{DAS}$ the new digit address is valid.
E	Digit data becomes valid on D4–D1 within 1/2 microcycle after $\overline{DAS}$ negative edge. Digit data must remain valid for 1 microcycle. During this time data is read.
F	Digit address advances to next digit, i.e., 0, 1, 2, 3, . . ., N scientific notation or 2, 3, 4, . . ., N floating point where N = MDC + 3 scientific notation N = MDC + 1 floating point (See DATA FORMATS).
G	Next digit is placed on D4–D1, again within 1/2 microcycle after $\overline{DAS}$ negative edge.
H	All digits have been read in. Digit Address goes to 0000. $\overline{DAS}$ pulse occurs. ISEL goes high before RDY pulse for next instruction. Number of digits read depends on notation mode and mantissa digit count (see DATA FORMATS).

(g) Timing Characteristics for IN Instruction



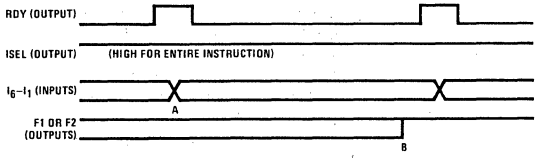
POINTS	DESCRIPTION
A	OUT instruction is placed on I lines.
B	ISEL goes low for second instruction word.
C	Second word of instruction becomes available to external hardware as a high order address for a RAM or other device.
D	First digit address appears on DA lines, the same as for the IN instruction. $\overline{DAS}$ frames DA changes, as with IN instruction.
E	First digit output appears on DO4–DO1 within 2 microcycles after the negative edge of $\overline{DAS}$.
F	R/$\overline{W}$ active low pulse occurs to write data into RAM or other device. This is a 2-microcycle pulse occurring within 3 microcycles after the negative edge of $\overline{DAS}$.
G	Digit address advances to next digit. (See DATA FORMATS).
H	Next digit output appears on DO4–DO1, 14 microcycles after the last digit appeared.
I	All digits have been output. Digit Address goes to 0000. ISEL goes high before RDY pulse for next instruction. Number of digits read depends on notation mode and mantissa digit count (see DATA FORMATS).

(h) Timing Characteristics for OUT Instruction



POINTS	DESCRIPTION
A	PRW1 or PRW2 instruction is placed on I lines
B	2-microcycle R/$\overline{W}$ active low pulse occurs.

(i) Timing Characteristics for PRW1, PRW2 Instructions

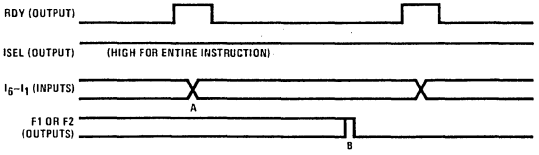


POINTS	DESCRIPTION
A	SF1 or SF2 instruction is placed on I lines
B	F1 or F2 is set high.

(j) Timing Characteristics for SF1, SF2 Instructions

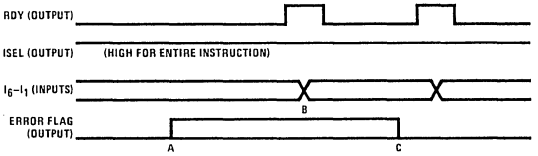
FIGURE 10. Instruction Timing Diagrams (Continued)

Functional Description (Continued)



POINTS	DESCRIPTION
A	PF1 or PF2 instruction is placed on I lines
B	F1 or F2 is pulsed active high for 2 microcycles.

(k) Timing Characteristics for PF1, PF2 Instructions



POINTS	DESCRIPTION
A	If an error occurs, ERROR is set high (see Table VI).
B	ECLR instruction is placed on I lines.
C	Error flag is set low.

(l) Timing Characteristics for ECLR Instruction and Error Occurrences

FIGURE 10. Instruction Timing Diagrams (Continued)

DATA FORMATS

IN/OUT Instructions

Mantissa digit count and notation mode determine data format. Table VII shows the contents of the D, D0 and DA lines for an IN or OUT instruction. Anywhere from 4 to 11 digits will be input or output by a single instruction.

AIN Instruction

One digit is input per AIN instruction. A maximum of 8 digits may be entered into the X mantissa by using consecutive AIN instructions. Digit entry is terminated by EN or any function instruction, (see NUMBER ENTRY). Table VII shows the DA lines for consecutive AIN instructions.

ERROR CONTROL

The error flag, which can drive an LED indicator, is set high upon detection of an arithmetic or output error. (See Table VI).

TABLE VI. ERROR CONDITIONS

Error flag is set when:

1. LN X when $X \leq 0$
   LOG X when $X \leq 0$
2. Any result $< 10^{-99}$
   Any result $\geq 10^{100}$
3. TAN 90°, 270°, 450°, etc.
4. SIN X, COS X, TAN X when $|X| \geq 9000°$
5. SIN^{-1} X, COS^{-1} X when $|X| > 1$ or $|X| \leq 10^{-50}$
6. SQRT X when $X < 0$
7. /, INV/, 1/X when X = 0
8. In floating point mode OUT instruction if number of mantissa digits to left of decimal point is greater than the mantissa digit count.

The error flag can be tested by the TERR instruction (which branches if ERROR = 1) or it can be used to clear the external program counter, resulting in a hardware jump to location 0, the error recovery location. In either case, an ECLR instruction must be executed to clear the error flag.

The occurrence of an error does not affect the operation of the processor in any way. The OUT instruction will not output digits if error condition 8 is true, but otherwise will output digits, even if the error flag is set.

For automatic error recovery, ERROR is wired to the asynchronous clear input of the external program counter (PC). The instruction at location 0 is an ECLR to clear ERROR so that the next RDY pulse will advance the PC to location 1. A JMP instruction at location 1 with the branch address at location 2 of an error routine is then executed, which results in a transfer of program control to the error routine. These first 3 error recovery locations are skipped over upon reset (POR) as can be seen in the initialization and instruction fetch flowcharts. The program shown in Table VIII shows typical error recovery coding.

SAMPLE SYSTEMS

Figures 11–14 show sample systems using the MM57109. *Figure 11* shows a simple demonstrator system using switches to enter instructions. An LED display is used to demonstrate the OUT instruction, with a switch to force an OUT instruction on the I lines and to hold the HOLD input low for 1 second for repeated execution of the OUT instruction, resulting in a multiplexed display. A flip-flop latches the $\overline{BR}$ pulse which occurs when a test and branch instruction is true. LED lamps provide visual indication of the various flags. An enter button allows single instruction words to be entered one at a time in the ENTER mode and causes the display to light for 1 second in the DISPLAY mode.

Figure 12 shows a stand-alone system with external program counter and a RAM to expand memory.

Figure 13 shows the MM57109 used as a microprocessor peripheral. Latches contain instructions for the MM57109 and digit data for the microprocessor.

Figure 14 shows a data acquisition system which obtains data from a 3-digit A/D converter. *Figure 14(b)* shows a program which reads data from the A/D converter. This coding should be studied as a general example of an MM57109 program.

Figure 15 shows a microprocessor to MM57109 interface using 2 FIFO's for instruction and data buffering.

These sample systems are not intended to be detailed drawings of a complete system (except *Figure 11*). Their purpose is to provide the designer with some ideas as to how to use the MM57109 in an actual system.

Functional Description (Continued)

TABLE VII. DATA FORMATS

IN/OUT INSTRUCTIONS (A) MODE = SCIENTIFIC NOTATION

DA4–DA1	IN: / OUT:	D4 / DO4	D3 / DO3	D2 / DO2	D1 / DO1
0		Most significant exponent digit			
1		Least significant exponent digit			
2		Sm	0	0	Se
3		Not used			
4		Most significant mantissa digit (Decimal point follows this digit) This digit *must* be non-zero on the AIN instruction unless the entire number is zero. Failure to do this will result in errors in calculations. This digit will *always* be non-zero on the OUT instruction, for non zero numbers.			
5		Second most significant mantissa digit.			
.		.			
.		.			
.		.			
MDC + 3		Least significant mantissa digit			

IN/OUT INSTRUCTIONS (B) MODE = FLOATING POINT

DA4–DA1	DPX	IN: / OUT:	D4 / DO4	D3 / DO3	D2 / DO2	D1 / DO1
2			Sm	0	0	0
3			DP POS			
4	11		Most significant mantissa digit = 0–9. On the OUT instruction, this digit will be non-zero unless $\|X\| < 1$, in which case it will be zero and DP POS will be 11. Leading zeroes are not blanked.			
5	10		Second most significant mantissa digit			
.	.		.			
.	.		.			
.	.		.			
MDC + 3	12 − MDC		Least significant mantissa digit = 0–9			

Notes:

MDC = Mantissa digit count, set by SMDC instruction, initially = 8
Sm = Sign of mantissa, 0 = positive, 1 = negative
Se = Sign of exponent (Se = 0 in floating point mode)
DP POS = Decimal point position indicator is a value in the range from 11 down to 12 − MDC, which indicates a digit, as given by the DPX column in the table. The decimal point is located immediately following this digit. Example: If DP POS = 10, then the decimal point follows the second most significant mantissa digit (DPX = 10).

AIN INSTRUCTION

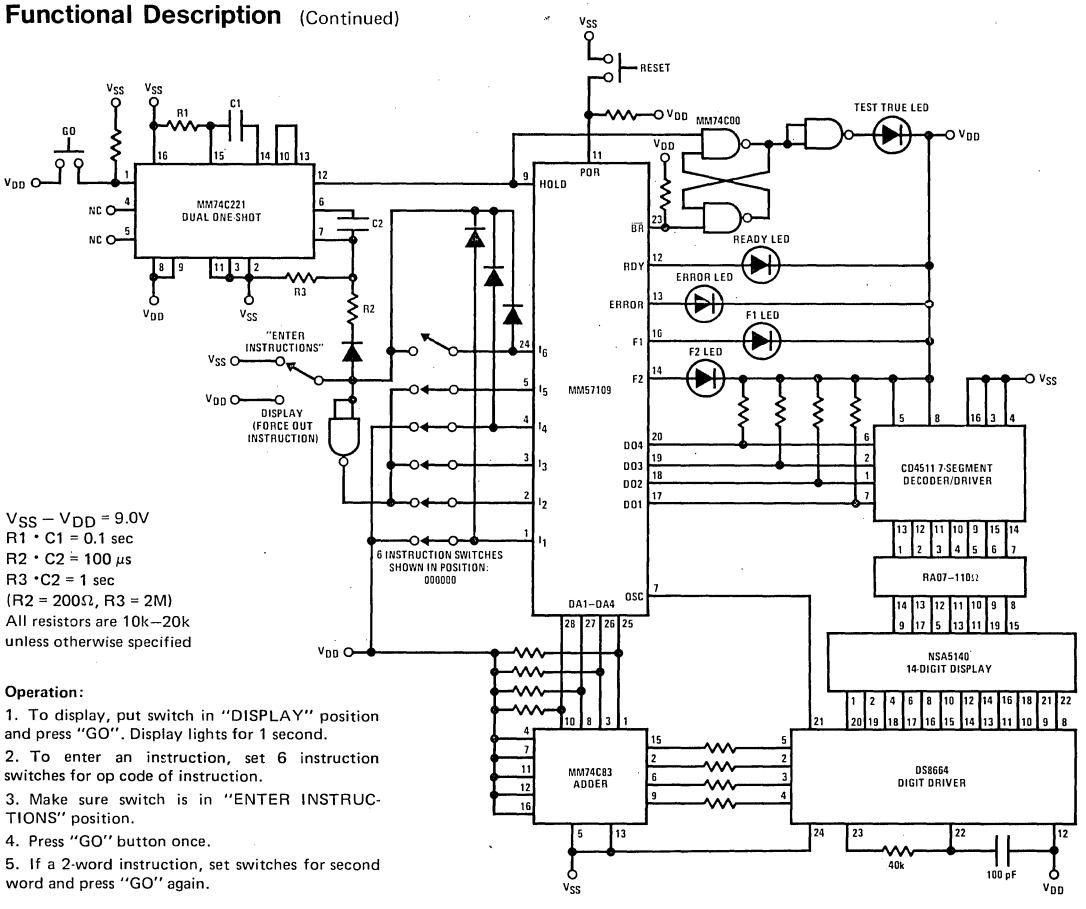
DA4–DA1	
0	Most significant digit Xm (first AIN instruction)
.	.
.	.
.	.
7	Least significant digit Xm (eighth AIN instruction)

Note. X_m = X register mantissa. Decimal point follows last digit entered. An irrecoverable internal error will occur if more than 8 digits are entered in a row with AIN, or if a non-BCD digit is entered.

TABLE VIII. ERROR RECOVERY CODING

OCTAL ADDRESS	OCTAL OP CODE	LABEL	INSTRUCTION MNEMONIC	OPERAND	COMMENT
00	53		ECLR		Clear error flag
01	25		JMP	ERROR	Jump to error routine
02	75				Address of label 'ERROR'
03	.		User Program		
.	.		.		
.	.		.		
.	.		.		
.	.		.		
.	.		.		
75	.	ERROR	—	—	User error recovery routine
.	.		.	.	.
.	.		.	.	.

Functional Description (Continued)



$V_{SS} - V_{DD} = 9.0V$
$R1 \cdot C1 = 0.1$ sec
$R2 \cdot C2 \doteq 100\ \mu s$
$R3 \cdot C2 = 1$ sec
(R2 = 200Ω, R3 = 2M)
All resistors are 10k–20k unless otherwise specified

Operation:

1. To display, put switch in "DISPLAY" position and press "GO". Display lights for 1 second.
2. To enter an instruction, set 6 instruction switches for op code of instruction.
3. Make sure switch is in "ENTER INSTRUCTIONS" position.
4. Press "GO" button once.
5. If a 2-word instruction, set switches for second word and press "GO" again.

FIGURE 11. MM57109 Sample System with Switches for Instruction Entry

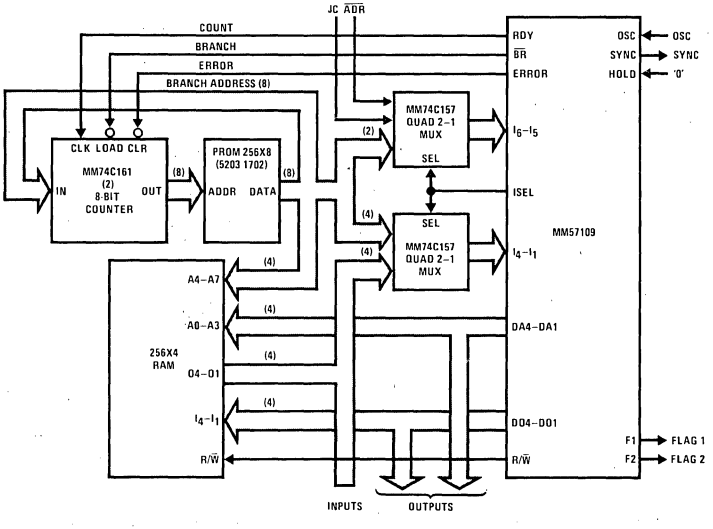


FIGURE 12. MM57109 Stand-Alone System

Functional Description (Continued)

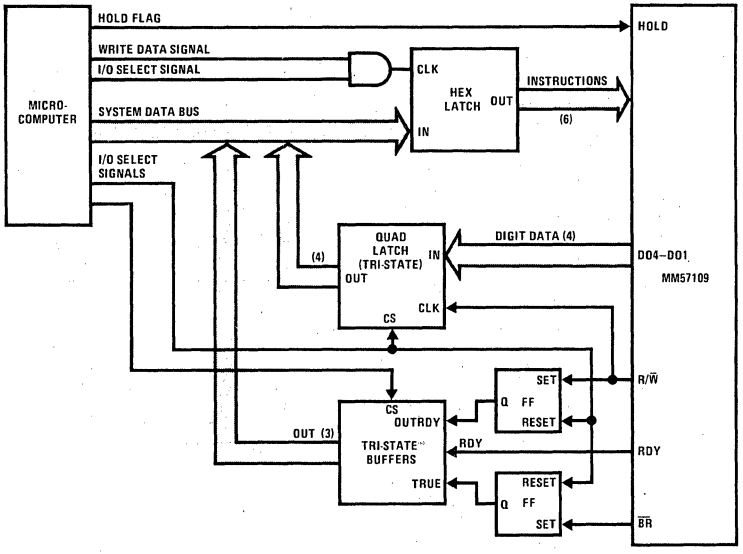


FIGURE 13(a). MM57109 as a Microcomputer Peripheral

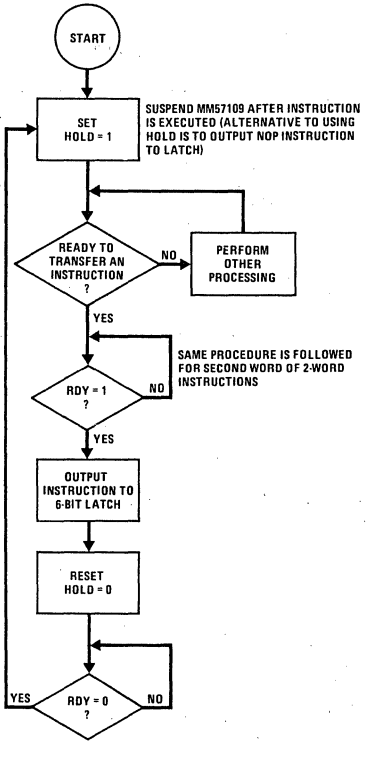


FIGURE 13(b). Microcomputer Software for MM57109 Peripheral Interface

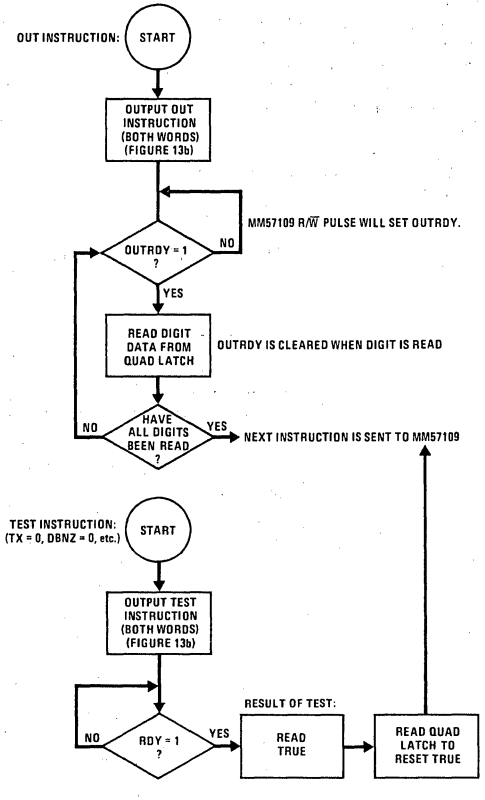


FIGURE 13(c). Microcomputer Software for MM57109 OUT, Test Instructions

Functional Description (Continued)

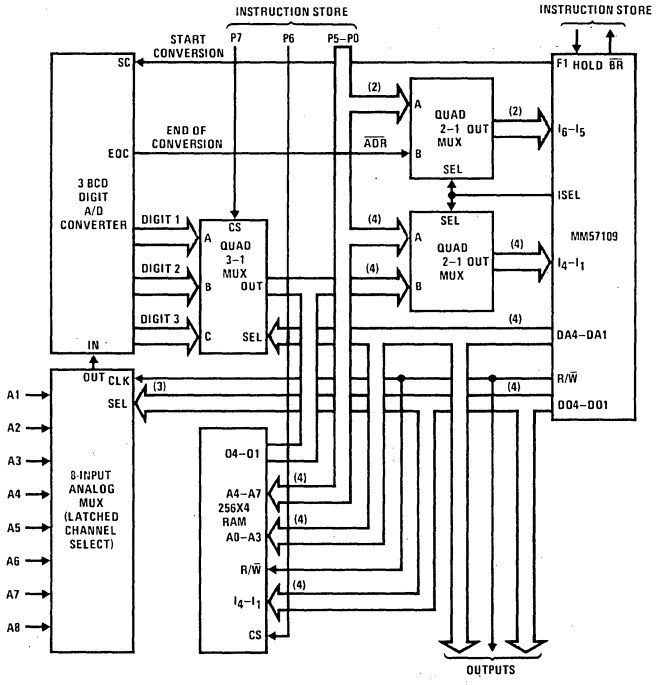


FIGURE 14(a). MM57109 Analog Data Acquisition System Block Diagram

Acquisition System Instruction Format

TYPE OF INSTRUCTION	P_7	P_6	$P_5P_4P_3P_2P_1P_0$
Select Analog Channel	1	1	OUT instruction
A/D Input	0	1	AIN instruction
RAM I/O	1	0	IN/OUT instructions for second word. Second word P_0–P_3 are high order RAM addresses
Others	1	1	Other instructions

Acquisition System Coding

	P_7	P_6	P_5–P_0	COMMENT
	1	1	4	Number of channels to be input
	1	1	MS	Store in M
	1	1	SMDC	Mantissa Digit Count = 1
	1	1	1	
LOOP	1	1	MR	Retrieve channel number
	1	1	OUT	Select analog channel
	1	1	0	
	1	1	PF1	Start A/D converter
	1	1	EN	Push stack
	0	1	AIN	Read A/D converter digit 1 when EOC = 0
	0	1	AIN	Read A/D converter digit 2
	0	1	AIN	Read A/D converter digit 3
	1	1	DBNZ	Update channel number and check if 0
	1	1	LOOP	
	1	1	X	Channel 1 times channel 2 (C1 x C2)
	1	1	/	(C1 x C2) ÷ C3
	1	1	COS	COSINE ((C1 x C2) ÷ C3)
	1	1	+	C4 + COSINE ((C1 x C2) ÷ C3)
	1	1	SMDC	Mantissa Digit Count = 3
	1	1	3	
	1	0	OUT	Result to RAM (0)
	1	0	0	

FIGURE 14(b). MM57109 Analog Data Acquisition System Input Coding

Functional Description (Continued)

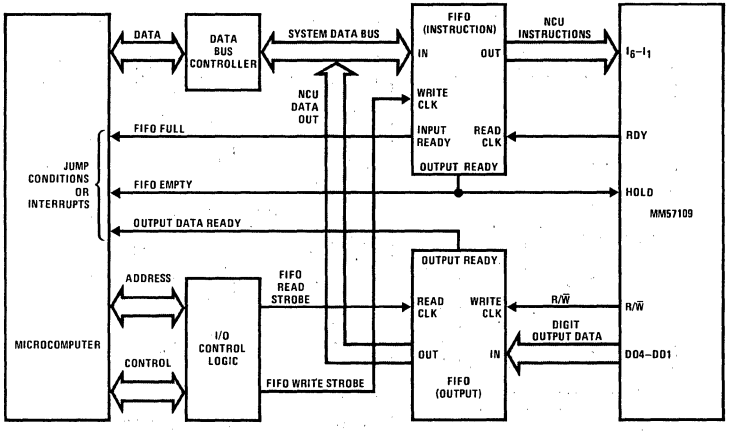


FIGURE 15. MM57109 Microcomputer Interface Using 2 FIFO's

Getting Your MM57109 Going

After wiring up a system using an MM57109, the following steps should be followed to verify that the processor is operating properly:

1. Check power supply for proper level, polarity, and absence of noise.
2. Check oscillator frequency, levels, duty cycle and rise and fall times.
3. Verify presence of SYNC output.
4. Check POR reset pulse duration, levels and rise and fall times.
5. Put HOLD input high and verify that RDY stays high.
6. Put HOLD input low and verify that RDY is pulsing active high.
7. Check that the system places the proper instructions on the I lines.
8. Force an OUT instruction on the I lines, put HOLD low, apply a reset pulse, and verify that DO4, DO2, DO1 DA–DA1, $\overline{DAS}$ and R/$\overline{W}$ are changing.