F3852
Dynamic Memory Interface (DMI)/ F3853 Static Memory Interface (SMI)

Microprocessor Product

Description

The Fairchild F3852 Dynamic Memory Interface (DMI) and F3853 Static Memory Interface (SMI) provide all interface logic needed to include up to 64K bytes of dynamic or static RAM in an F8 microcomputer system. In response to control signals output by the F3850 Central Processing Unit (CPU), the F3852 DMI and F3853 SMI generate address and control signals needed by standard static and dynamic RAM devices.

The DMI and SMI, like all other F8 memory devices, contain their own memory address generation logic.

The DMI and SMI are manufactured with N-channel, isoplanar MOS technology. Therefore, power dissipation is very low, typically less than 335 mW.

- **Interface Logic for 64K Bytes of Dynamic or Static RAM**
- **Four 16-Bit Registers for Addressing Logic**
- **Generates Address and Control for Standard Static and Dynamic RAM Devices**
- **DMA and Memory Refresh (F3852)**
- **Programmable Timer (F3853)**
- **N-channel Isoplanar MOS Technology**
- **Power Dissipation <335 mW**

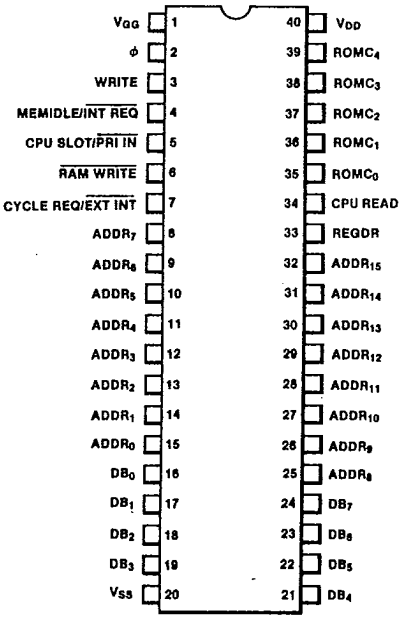
The F3852 DMI automatically refreshes dynamic RAM and provides interface logic to support the F3854 Direct Memory Access Controller (DMAC) (see *Figure 1*). The DMI may control static memory, in which case refresh would not be necessary. If I/O port OD (or ED) is set to 01, turning off DMA and memory refresh, the DMI has the characteristics of an SMI. Static RAM may be accessed via DMA if the control port is set to 00 and the data bus is properly buffered, as in any DMA system.

The F3852 DMI refresh logic is not implemented on the F3853 SMI; therefore, the SMI cannot support DMA, because memory refresh and DMA logic are interdependent. Since the SMI has no memory refresh or DMA logic, it includes a programmable timer and interrupt control circuitry (refer to *Figure 2*).

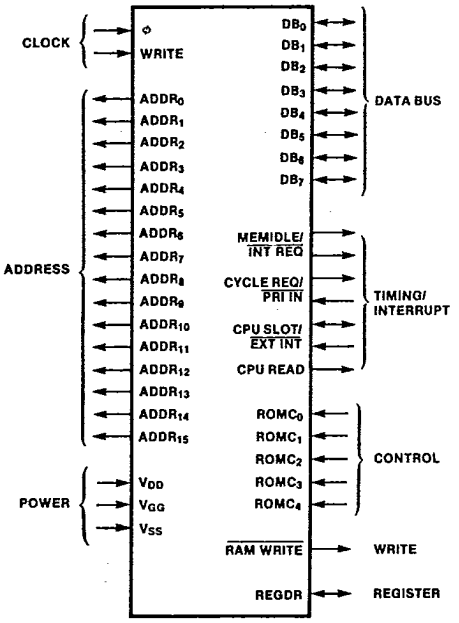
An F8 system is initialized by power-on, or by the EXT RESET line being pulsed low at the CPU. When an F8 system is initialized, DMA is turned off and memory refresh is on, with refresh every fourth cycle selected. Contents of all other registers are indeterminate; reading control port OD (or ED) also gives indeterminate results, although the DMA/refresh state of the DMI has been initialized.

Connection Diagram

40-Pin DIP



Signal Functions



Device Organization

Refer to the logic organization diagrams (*Figures 1* and *2*) for the following discussion of device organization.

Program Counter, Data Counter, and Stack Register
Addressing logic for both the DMI and SMI consists of four 16-bit registers: program counter PC0, data counters DC0 and DC1, and stack register PC1.

The program counter always addresses the memory word from which the next object program code must be fetched.

The data counters address memory words containing individual data bytes, or bytes within data tables to be used as operands.

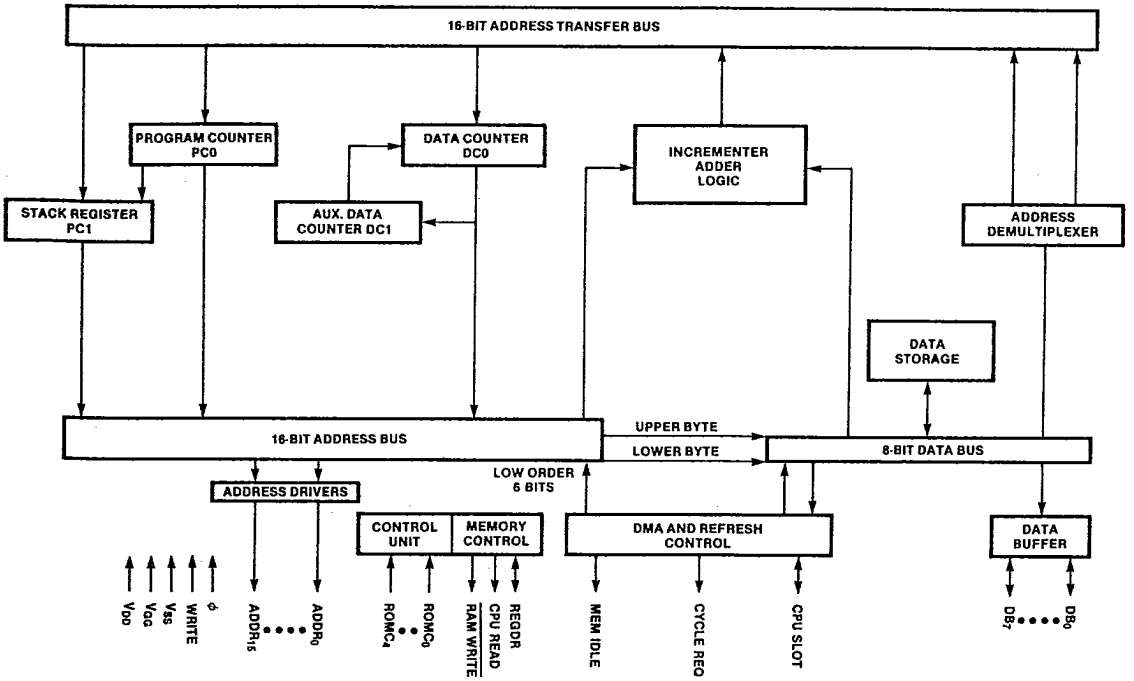
The stack register is a buffer for the program counter; its contents are never used directly to address memory.

Address decoding is identical, whether originating in PC0 or DC0. When an interrupt is acknowledged, the contents of PC0 are saved in PC1 (the previous contents of PC1 are lost). Since PC0 always addresses the memory location from which the next object program instruction byte is to be read, DC0 must address memory if the instruction requires data (i.e., an operand) to be accessed. The program counter cannot be used to address data, since it is saving the address of the next instruction code.

I/O Ports
The SMI and DMI each have four I/O ports, addressed OC, OD, OE, and OF for both devices when used separately. Option port addresses are used for the DMI in F8 systems that include both DMI and SMI; the F3852 DMI is designated by an SL31116 marking.

The implemented I/O ports are accessed via IN, INS, OUT and OUTS instructions, as with any I/O port. However, the DMI I/O ports are internal latches, having no connection to I/O pins or external interface. The REGDR signal, if not clamped low by an external device, goes high during IN or INS instructions that select either of the DMI ports. Clamping the REGDR signal low does not inhibit data bus driving during I/O as it does during the output of address registers.

Fig. 1 F3852 DMI Logic Organization



The OD (or ED) I/O port controls memory refresh and DMA as follows:

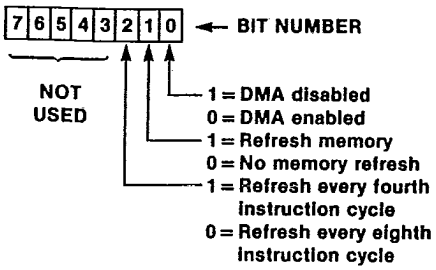


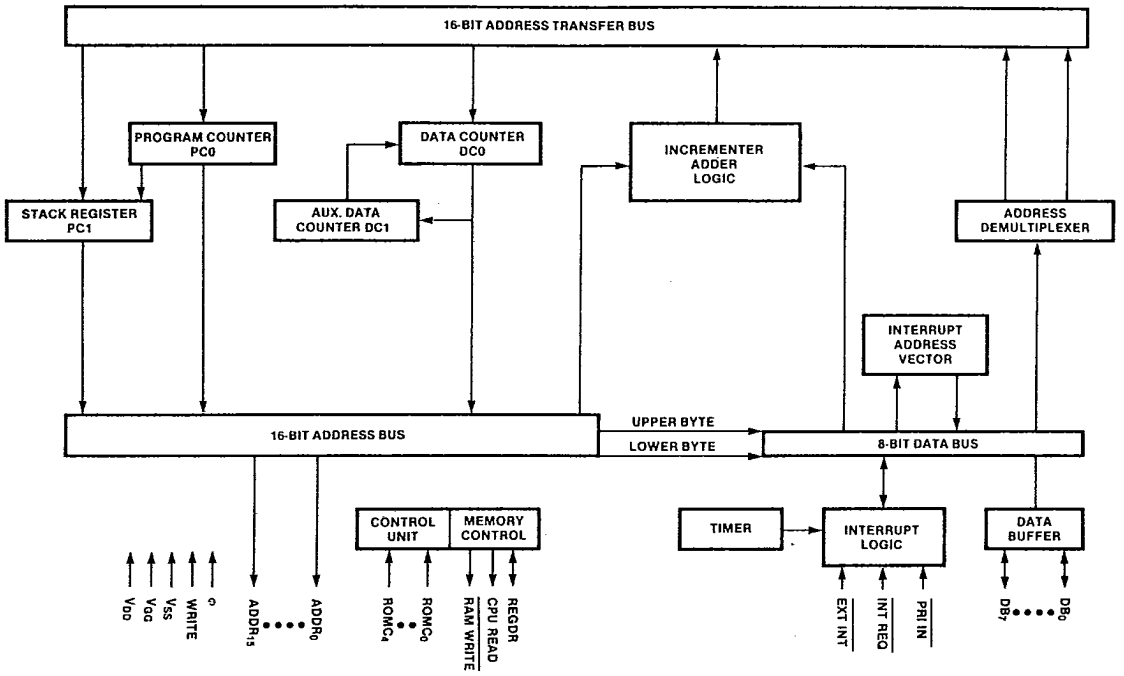
Table 1 lists the I/O ports for the SMI and DMI devices.

Table 1 I/O Ports for the SMI and DMI

Port	F3852	F3853
OC (EC*)	General-purpose, 8-bit data storage buffer, loaded with the OUT or OUTS instruction and read using the IN or INS instruction; EC for DMI/SMI option.	Interrupt address vector upper byte; written into and read from using the I/O instructions.
OD (ED*)	Control register for memory refresh and DMA; ED for DMI/SMI option.	Interrupt address vector lower byte; written into and read from using the I/O instructions.
OE (EE*)	Not used; EE for DMI/SMI option.	Interrupt control port
OF (EF*)	Not used; EF for DMI/SMI option.	Programmable timer

*F3852 with SL31116 marking.

Fig. 2 F3853 SMI Logic Organization



Direct Memory Access and Memory Refresh (F3852 Only) Within every instruction execution cycle there is a period when the CPU is not accessing memory because of the organization of the F8 microcomputer system. The F3852 DMI on-chip direct memory access and refresh control logic generates timing and control signals that identify time periods when the CPU is not accessing memory. During these time periods, memory is refreshed or DMA data accesses occur.

The DMA and memory refresh are similar operations in terms of the DMI logic. As described in the "DMI/SMI Signal Timing" section, CYCLE REQ identifies two or three memory access periods within an instruction cycle.

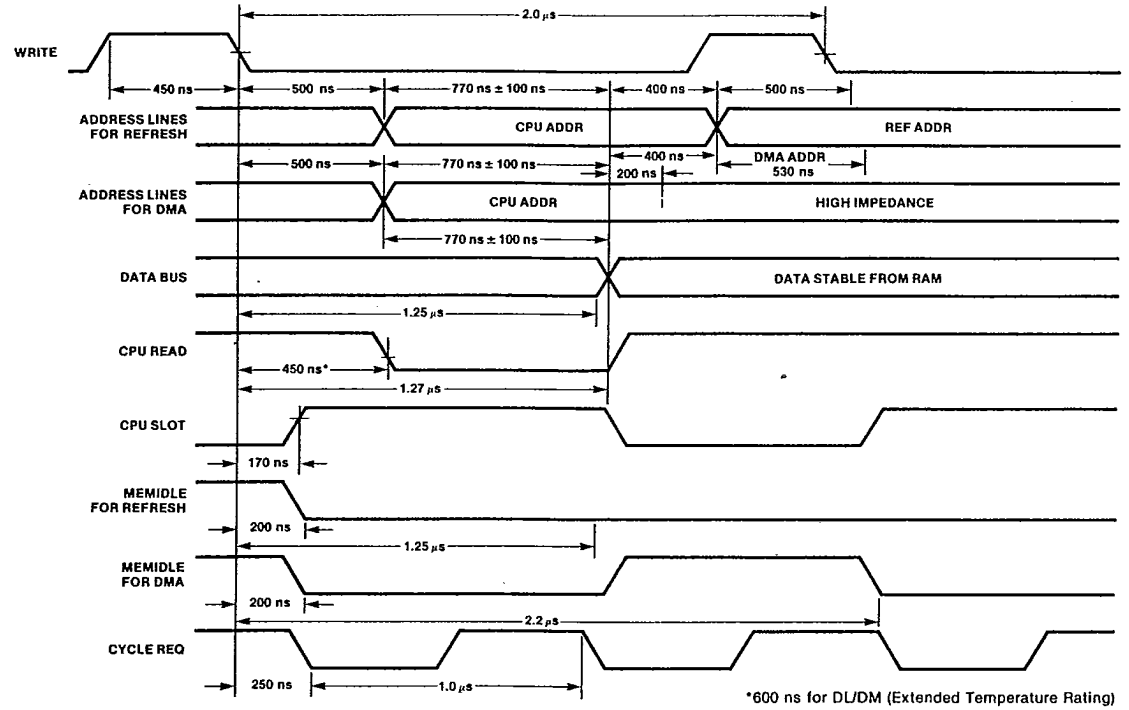
Either the first or the second access period is reserved for the instruction cycle being decoded. If the ROMC state for the instruction cycle requires data to be read out of a RAM, the read occurs during this reserved access period. If the ROMC state for the instruction cycle requires data to be input to an address register, or if no data movement occurs on the data bus, the reserved access period is not used for any memory access—it is, in effect, wasted.

One more memory access may occur within the instruction cycle, during either the second or third access period, while the data bus latches hold data accessed during the first period. This is a free access period.

Some available free access periods must be used to refresh dynamic RAM. A refresh uses logic within the DMI; therefore, a refresh occurs in parallel with other activity.

If the free access period is not used to refresh dynamic RAM, it may be used by an F3854 DMA device to perform direct memory accesses. The DMA device uses a separate data channel to access memory, so DMA can parallel other activity in the F8 system.

Fig. 3 Memory Refresh and DMA Timing During Short-Cycle Memory Read, with Address Out of Program Counter



*600 ns for DL/DM (Extended Temperature Rating)

Figures 3, 4, and *5* indicate worst-case timing for DMA and memory refresh. In an F8 write-to-memory cycle (ST), no refresh or DMA may take place. (Refer to the "DMI/SMI Signal Timing" and "Timing Characteristics" sections for further information.)

A complete memory refresh cycle executes in F milliseconds, where F is given by:

$$F = (2^6)T \cdot R$$

where T is the instruction cycle time, either 2 or 3 microseconds

R is the refresh rate, either 4 (for one slot in four) or 8 (for one slot in eight)

Programmable Timer (F3853 Only)
The SMI has an 8-bit shift register, addressable as I/O port OF, that may be used as a programmable timer.

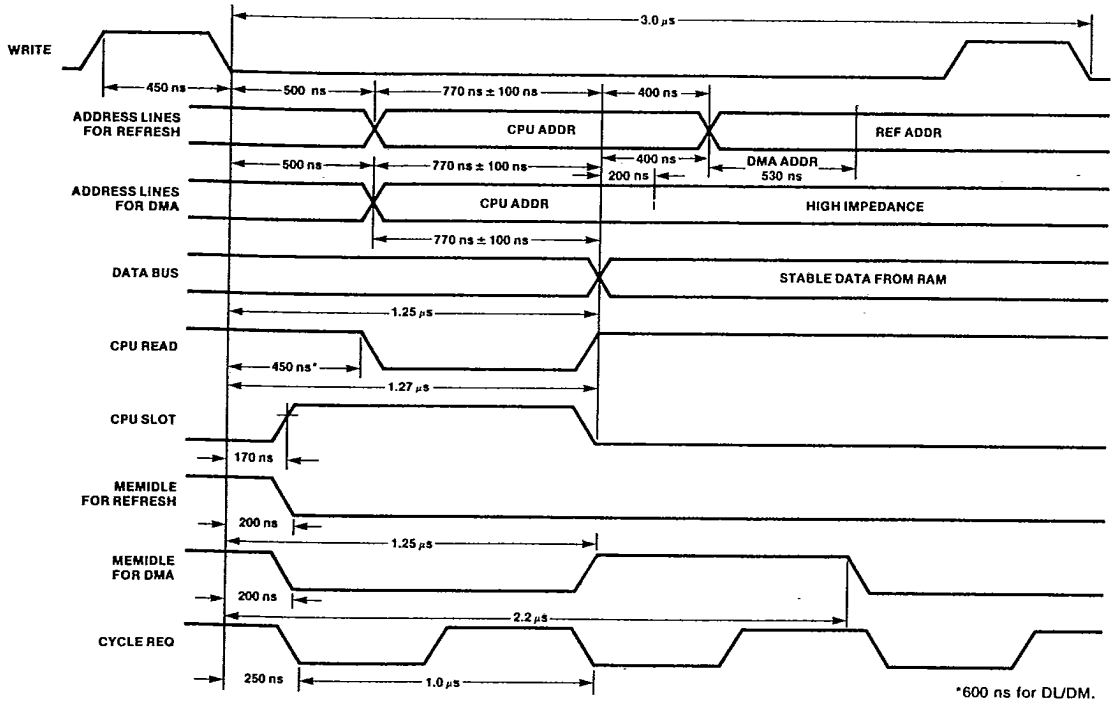
Figure 6 illustrates the shift register logic and the exclusive-OR feedback path.

Binary values in the range 0 through 254, when loaded into the timer, are converted into "timer counts." "Timer contents" is the actual binary value loaded into a timer, and "timer counts" is the corresponding number of time intervals the timer takes to time out. Data cannot be read out of the programmable timer I/O port.

As described in the *F8 and F3870 Guide to Programming*, an assembly language program specifies timer counts, which the assembler converts into the binary value that must be loaded into the programmable timer.

It is possible to write small subroutines that calculate time values one count faster or slower than a given value. Such subroutines would be used if programmed delays are required.

Fig. 4 Memory Refresh and DMA Timing During Long-Cycle Memory Read, with Address Out of Program Counter



The OUT or OUTS instruction is used to load timer counts into the programmable timer. The contents of the programmable timer cannot be read using an IN or INS instruction. The timer times out after a time interval given by the product:

(period of clock ϕ) × (timer counts) × (31)

For example, a value of 200 (11001000, or H'C8') loaded into the programmable timer becomes 215 timer counts. The timer therefore times out in 3.33 milliseconds, if the period of clock signal ϕ is 500 nanoseconds.

A value of 255 (H'FF') loaded into a programmable timer stops the timer.

All timers run continuously, unless they have been stopped by loading H'FF' into the timer. Upon timing out, the timer transmits an interrupt request to the interrupt logic (refer to the "Interrupt Logic" section for more details). If proper interrupt logic conditions exist, the timer interrupt request is passed to the CPU through the $\overline{\text{INT REQ}}$ output.

After a programmable timer has timed out, it again times out after 255 timer counts; therefore, if the timer is allowed to run continuously, it times out every 7905 ϕ clock periods, or every 3.953 milliseconds for a 500 nanosecond clock.

If the timer is actually loaded with a zero value, it times out in 24 counts, whereas once it has timed out it next times out in 255 counts (i.e., a time-out is not the same as counting down to zero).

Fig. 5 Memory Refresh and DMA Timing During Long-Cycle Memory Read, with Address Out of Data Counter

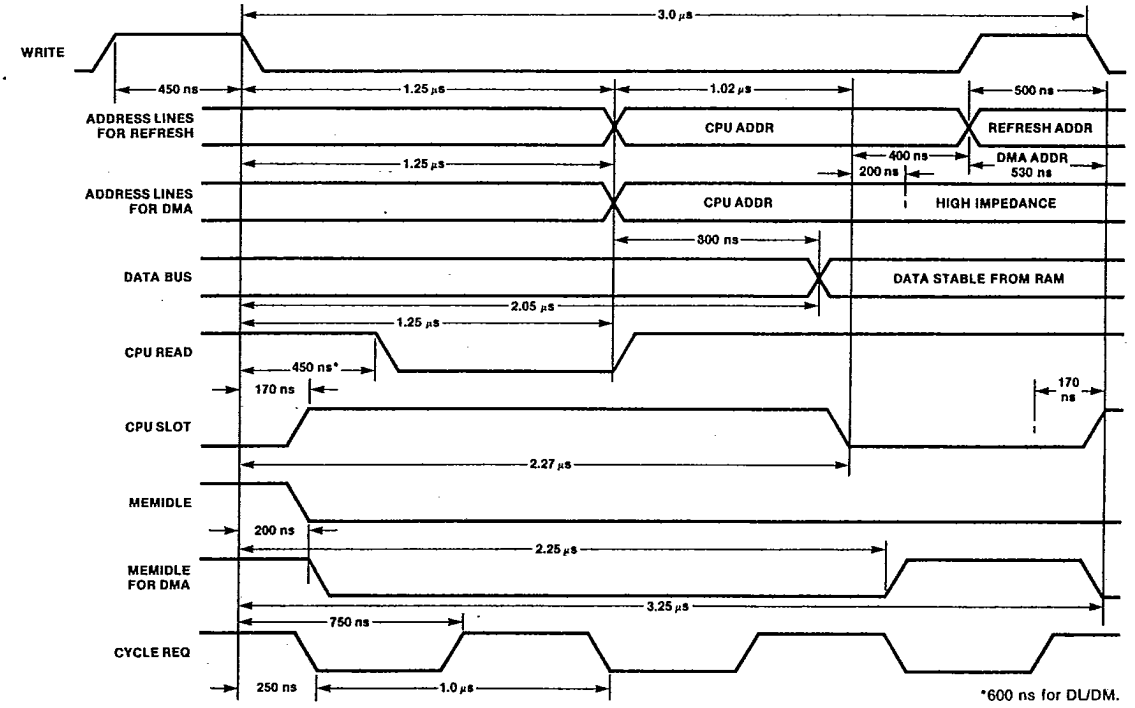
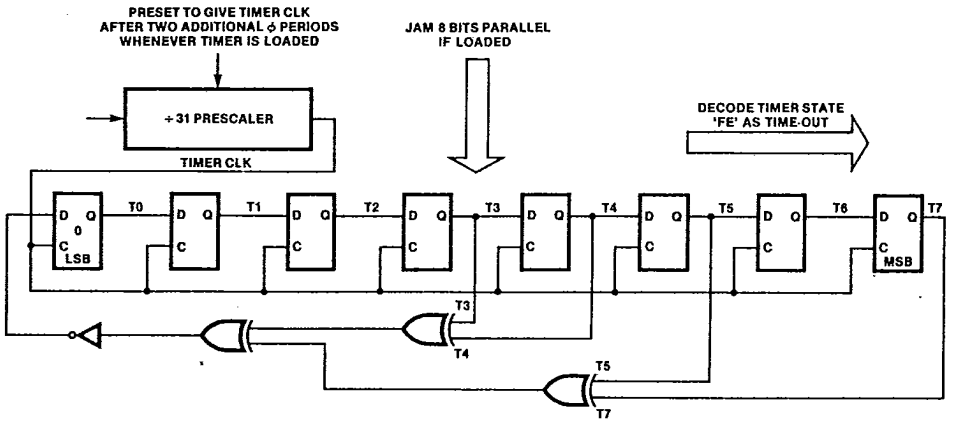


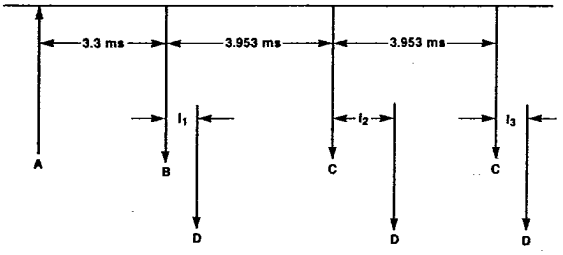
Fig. 6 Timer Block Diagram



When the timer and timer interrupt are being set to time a new interval, the timer must always be loaded before enabling the timer interrupt; loading the timer clears any pending timer interrupts. When the timer interrupt is enabled, any pending timer interrupt is acknowledged and forwarded to the CPU. Since the timer runs continuously, unless stopped under program control, enabling the timer before loading a time count can cause errors.

Figure 7 illustrates a possible signal sequence for a timer that is initially loaded with 200, then allowed to run continuously.

Fig. 7 Time-Out and Interrupt Request Timing



A — 200 LOADED INTO TIMER
B — FIRST TIME OUT
C — SECOND AND SUBSEQUENT TIME-OUTS
D — INTERRUPT SERVICE ROUTINES BEING ENTERED BY CPU
I_1, I_2, I_3 — INTERVALS BETWEEN TIME-OUT INTERRUPT REQUEST REACHING INTERRUPT LOGIC AND SERVICE ROUTINES BEING ENTERED BY CPU

Interrupt Logic

The interrupt control register or I/O port (OE) can be used to enable or disable interrupts. Data is loaded into the register or placed at the OE port using an OUT or OUTS instruction; data cannot be read out of the register or port.

The data at the I/O port is interpreted as follows:

B'XXXXXX00'	Disable all interrupts
B'XXXXXX01'	Enable external interrupt, disable timer interrupt

The contents of the interrupt control register are interpreted as:

B'XXXXXX10'	Disable all interrupts
B'XXXXXX11'	Disable external interrupt, enable timer interrupt

In the above, "X" represents "don't care" binary digits.

Depending on the contents of the interrupt control register, the SMI interrupt control logic can be accepting timer interrupts or external interrupts, or neither, but never both.

The OC and OD I/O ports are used for the interrupt address vector upper and lower bytes, respectively. They can be written into and read from using the I/O instructions.

Since only three device pins are available for use by interrupt logic, there is no priority out signal. This means that if an SMI is in an interrupt priority daisy chain, it must be the last device in the chain or external logic must generate a $\overline{PRI}$ $\overline{OUT}$ signal as shown in *Figure 8*.

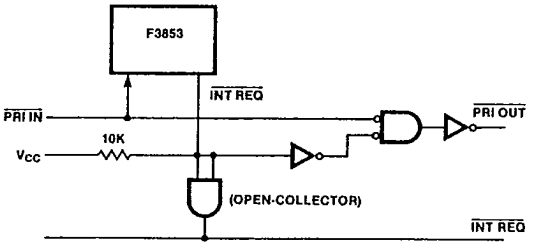
The SMI interrupt address vector consists of two programmable I/O ports. The interrupt address vector is set under program control, rather than being a mask option (as with the F3851 PSU). Even though the SMI interrupt address vector is programmable, bit 7 is still set to 0 for a timer interrupt, or to 1 for an external interrupt.

Address Contentions

When a DMI or SMI is present in an F8 system that includes a PSU, address contentions occur while using the XDC instruction.

The XDC instruction (ROMC state TD) causes the contents of data counters DC0 and DC1 to be exchanged; having no DC1 register, the PSU does not respond to this instruction. Therefore, the PSU and memory interface devices can have different values in their DC0 registers, and each value can be within the different address spaces of the two memory devices. An instruction that requires data to be output from the DC0 register may then cause two devices to simultaneously place different data on the data bus.

Fig. 8 $\overline{PRI}$ $\overline{OUT}$ Signal Generation



Signal Descriptions

The SMI and DMI signals are described in *Table 2.*

Table 2 SMI/DMI Signal Descriptions

Mnemonic	Pin No.	Name	Description
Clock			
ϕ	2	Clock	Clock inputs from the F3850 CPU.
WRITE	3	Clock	
Address			
$ADDR_0$–$ADDR_{15}$	15-8, 25-32	Address	Sixteen outputs through which an address is transmitted to dynamic RAM. The address may come from the PC0 or DC0 registers.
Data Bus			
DB_0–DB_7	16-19, 21-24	Data Bus	Eight bidirectional lines that link the DMI/SMI with all other devices in the F8 system. Only data moving to or from the address registers and I/O ports uses the data bus pins.
Timing			
CPU READ	34	CPU Read	An output signal that, when high, specifies that data is to be read out of a RAM location. When low, the signal is off; this does not specify a write operation (the write operation is specified by a $\overline{\text{RAM WRITE}}$ low signal).
CPU SLOT	5	CPU Slot	A DMI bidirectional signal that, when high, identifies portions of an instruction execution cycle during which the F3850 CPU is reading out of or writing into RAM. When 0 is loaded into port D and the CPU SLOT signal is held low by external logic, the address line drivers and $\overline{\text{RAM WRITE}}$ driver go to a high-impedance state.
CYCLE REQ	7	Cycle Request	An output signal that identifies each memory access period within an instruction cycle by making a high-to-low transition at the start of the memory access period. Does not identify events that are to occur during the memory access period. The CYCLE REQ signal is a divide-by-2 of ϕ during all ROMC states except ROMC state 05 (store-in-memory); it can be used to generate the clock signals required by many dynamic RAMs.
MEMIDLE	4	Memory Idle	A DMI output that identifies portions of an instruction execution cycle during which the F8 system is not accessing memory to read, write, or refresh. The MEMIDLE signal therefore identifies the portion of an instruction cycle that is available for DMA operations. The DMI can inhibit DMA by holding MEMIDLE constantly low. The address drivers and $\overline{\text{RAM WRITE}}$ driver are always in a high-impedance state when MEMIDLE is high, so that a DMA device may drive the address lines at this time.

Table 2 SMI/DMI Signal Descriptions (Cont.)

Mnemonic	Pin No.	Name	Description
Interrupt			
$\overline{EXT}$ $\overline{IN}$	7	External Interrupt	An SMI input signal; a high-to-low transition is interpreted as an interrupt request from an external device.
$\overline{INT}$ $\overline{REQ}$	4	Interrupt Request	An SMI output signal that becomes the $\overline{INT}$ $\overline{REQ}$ input to the F3850 CPU. Must be output low to interrupt the CPU; this only occurs if $\overline{PRI}$ $\overline{IN}$ is low and SMI interrupt control logic is requesting an interrupt.
$\overline{PRI}$ $\overline{IN}$	5	Priority In Line	An SMI input signal that, when low, sets the $\overline{INT}$ $\overline{REQ}$ output low in response to an interrupt.
Control			
$ROMC_0$-$ROMC_4$	35-39	Read Only Memory Control	Five input lines that are the control signals output by the F3850 CPU.
Write			
$\overline{RAM}$ $\overline{WRITE}$	6	Random Access Memory Write	An output signal that, when low, specifies data is to be written into a RAM location. When high, the signal is off; this does not necessarily specify a read operation.
Register			
REGDR	33	Register Drive	A bidirectional line that, when used as an input, can be clamped low by an external open-collector gate to prevent the DMI or SMI from placing a byte from its PC1 or DC0 register onto the data bus. The DMI supplies an internal pull-up resistor. When used as an output, REGDR can control data bus buffers. The DMI internally clamps REGDR low except during those ROMC states in which the DMI is required to place a byte out of the PC1 or DC0 registers, or either of its two control registers (I/O ports), onto the data bus.
Power			
V_{DD}	40	Power Supply	Nominal +5 Vdc
V_{GG}	1	Power Supply	Nominal +12 Vdc
V_{SS}	20	Ground	Common power and signal return

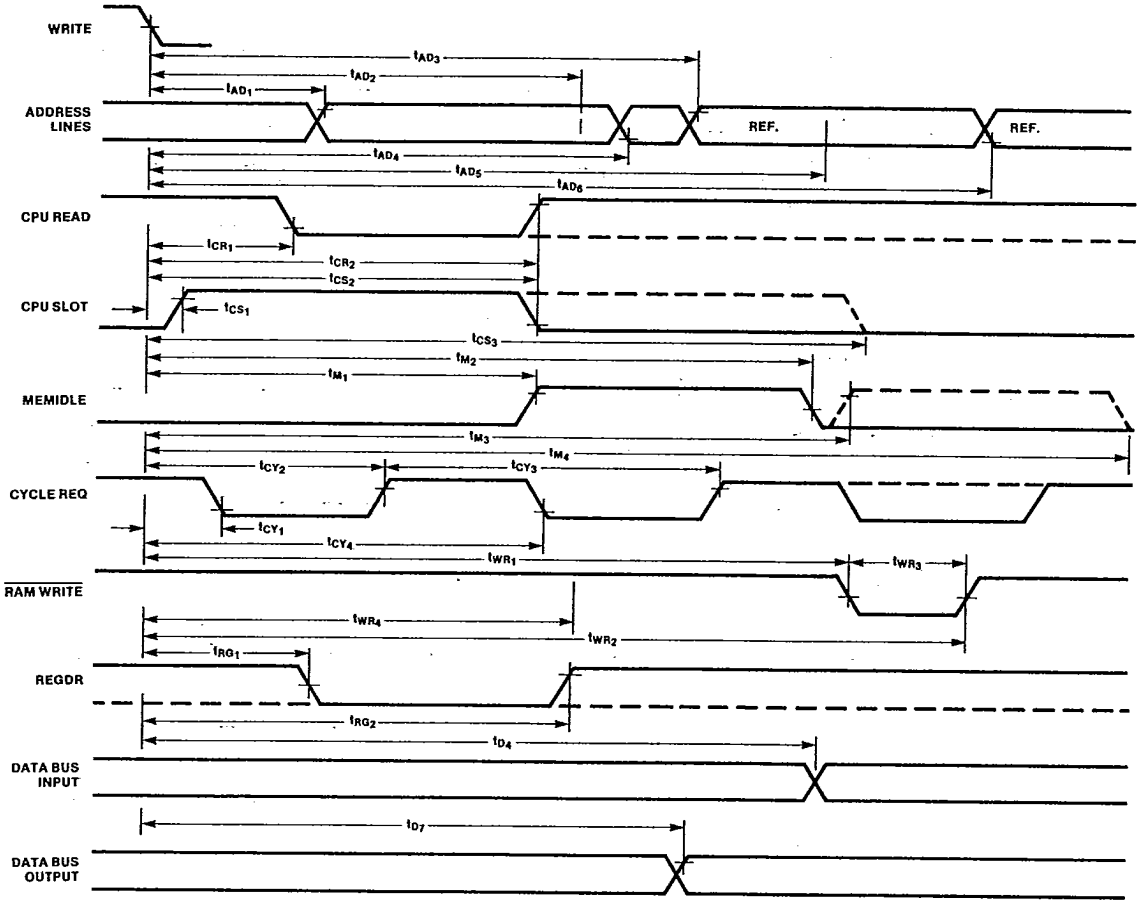
DMI/SMI Signal Timing

The DMI and SMI receive timing signals from the F3850 CPU, then output timing signals used by the dynamic RAM and an F3854 DMA device, if present. *Figures 9* and *10* illustrate the timing signals for the DMI and SMI, respectively.

Timing of the F3853 SMI $\overline{INT}$ $\overline{REQ}$, $\overline{PRI}$ $\overline{IN}$, and $\overline{EXT}$ $\overline{IN}$ interrupt signals is identical to that of the F3851 PSU; the remaining signals have the same timing as the DMI, except for the address lines.

Within an instruction cycle, there may be either two or three memory access periods, depending on whether the instruction cycle is long or short. A memory access period is equivalent to two ϕ clock periods, and is identified by a CYCLE REQ signal, which is a divide-by-2 of ϕ. Whether the instruction cycle is short or long depends on the source and destination of the data being transmitted during instruction execution.

Fig. 9 F3852 DMI Timing Signals



During the first memory access period, the device outputs the contents of PC0 onto the address lines ($ADDR_0$–$ADDR_{15}$).

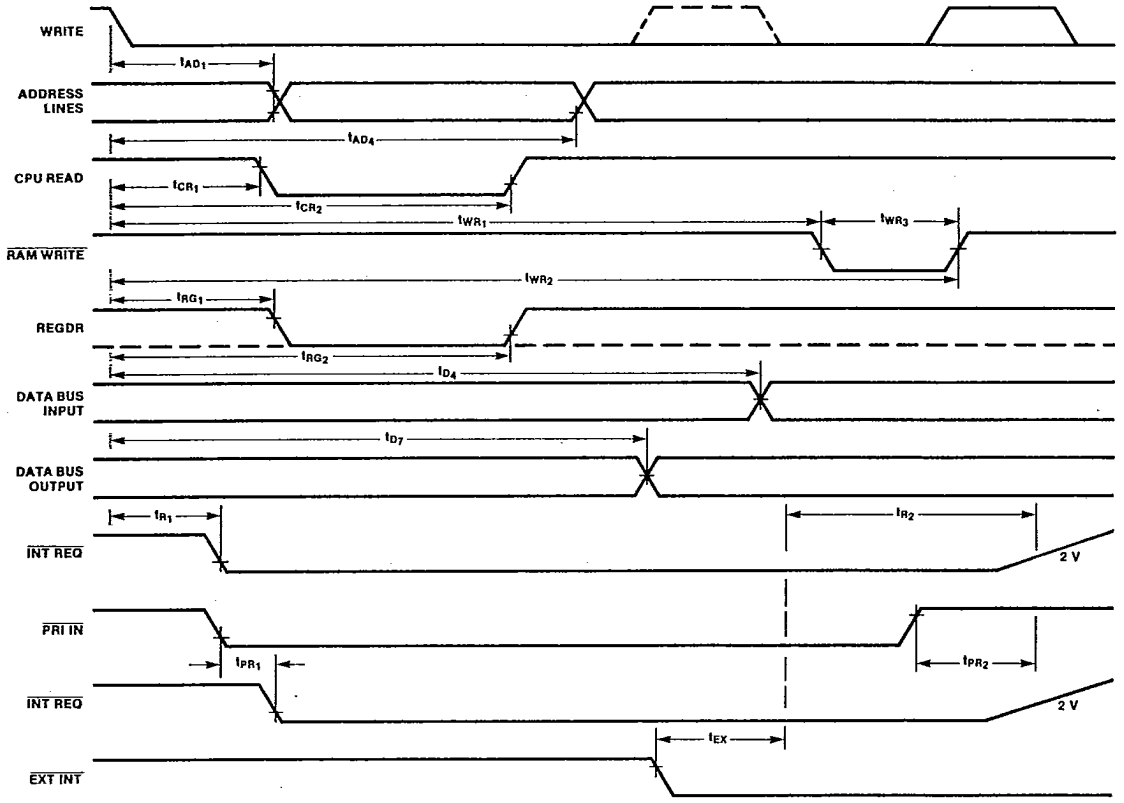
In effect, the logic begins by assuming that a memory read is to occur, with the memory address provided by PC0.

While the PC0 contents are being output on the address lines, the control unit (in parallel) decodes the ROMC state that has been received from the CPU.

If the assumed logic proves to be correct, or if no memory access is to occur, the second access period can be used for memory refresh or DMA.

If the instruction decoded by the CPU specifies a memory read with another memory address, the DMI wastes the first access period; in this case, the instruction cycle is always long. The required memory access is performed during the second access and memory refresh occurs, or DMA is implemented, in the third access period.

Fig. 10 F3853 SMI Timing Signals



If a memory write instruction is decoded, no access periods are available for memory refresh or DMA.

Four variations of the instruction cycle result. The timing diagrams (*Figures 11-18*) illustrating the four variations represent worst cases and assume $td_2 = 150$ ns. The four variations are:

1. The instruction fetch. The memory address originates in program counter PC0, and the instruction cycle is short. (Timing is shown in *Figures 11* and *12*.)
2. An immediate operand fetch. The memory address originates in program counter PC0, and the instruction cycle is long. (Timing is shown in *Figures 13* and *14*.)
3. A data fetch. A data byte is output from an address register, or the memory address originates in data counter DC0; therefore, the instruction cycle is long. (Timing is shown in *Figures 15* and *16*.)
4. A memory write. Data is written into the RAM location addressed by DC0. (Timing is shown in *Figures 17* and *18*.)

The CPU SLOT and MEMIDLE signals identify the way a memory access period is being used. When the F3850 CPU is accessing memory, the CPU SLOT signal is high; $\overline{\text{RAM WRITE}}$ and the address lines are driven at this time.

When memory is available for DMA access, the CPU SLOT signal is low and the MEMIDLE signal is high. When the DMI is refreshing dynamic memory, the CPU SLOT and MEMIDLE signals are both low.

The DMI logic is able to achieve two memory accesses within one instruction cycle.

Buffer/latches are placed on the F8 data bus lines between the RAM and the F8 system to hold the data fetched during the first access.

The SMI, without memory refresh and DMA capabilities, does not generate three timing signals (CPU SLOT, CYCLE REQ, and MEMIDLE). These device pins are instead used by interrupt logic.

Since the DMI does not access memory within a single memory access period, as identified by the CPU SLOT signal, data bus timing is relaxed when using the SMI as compared to the DMI. Worst-case timing for the four possible machine cycles is explained in the "DMI/SMI Signal Timing" section.

The implications of this relaxed data bus timing parameter are that slower static memories may be used with the SMI; on the other hand, memory controlled by an SMI cannot be accessed by an F3854 DMA. Another implication is that a latching type buffer is not needed between memory and the data bus.

Instruction Execution

The DMI and SMI respond to signals that are output by the F3850 CPU in the course of implementing instruction cycles. The actions taken by the DMI and SMI during instruction execution are a function of the ROMC state.

Data Output by RAM

Figures 11 through *16* illustrate the worst-case timing when RAM, controlled by the DMI, outputs data onto the data bus. (In these figures, it is assumed that the CPU SLOT signal is used to strobe the RAM data into the data bus latches.)

The CPU READ signal is output high by the SMI/DMI to enable transfer of data from the data bus buffers to the data bus. Dynamic RAM has its own connection to the data bus through buffer/latches; data is not transferred through the DMI/SMI. (A CPU READ high signal is active when its respective data bus drivers are turned on.)

Data Output by the DMI/SMI

The REGDR signal defines the address space of the address registers. If an ROMC state received by the DMI/SMI requires data to be output from an address register, the device becomes the selected data source if the REGDR signal is allowed to go high.

Data Input to RAM

Figures 15 and *16* illustrate timing when data is written into RAM. Data is transferred through 3-state buffers on the data bus and into RAM. The $\overline{\text{RAM WRITE}}$ signal is pulsed low by the DMI/SMI to enable the transfer of data off the data bus into RAM. The 3-state buffers or multiplexers between the data bus and $\overline{\text{RAM WRITE}}$ data lines are necessary if DMA sources are also allowed to write into RAM.

Data Input to the DMI/SMI

Address contention is created by having duplicate address registers. One method for resolving the contention is to force every memory device to read data into its address registers whenever an ROMC state specifies any such operation. Address space concepts therefore do not apply when data is read into the address registers.

Fig. 11 DMI Timing Signals Output During Short-Cycle Memory Read, with Address from Program Counter

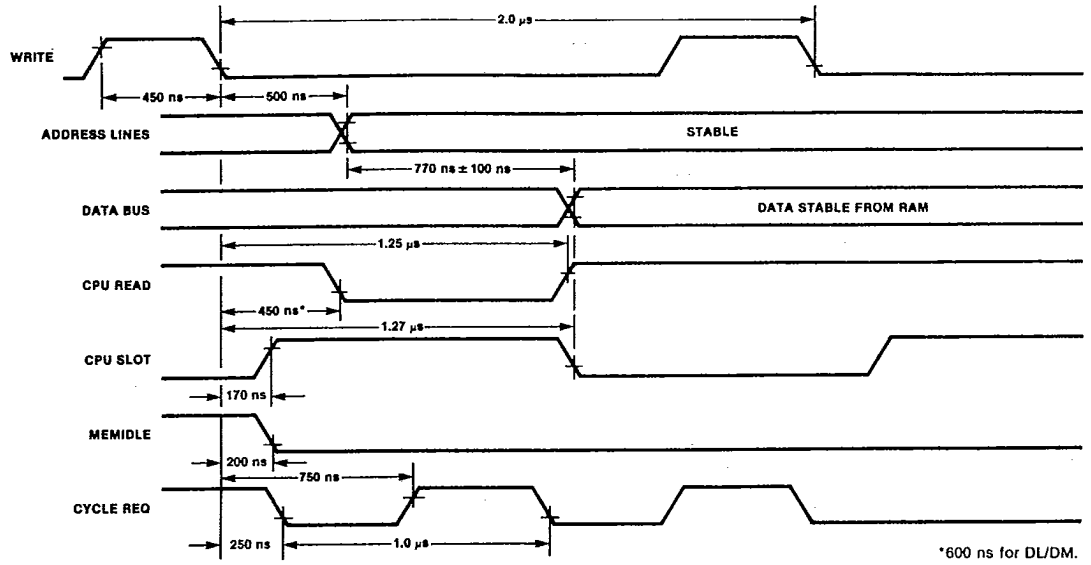


Fig. 12 SMI Timing Signals Output During Short-Cycle Memory Read, with Address from Program Counter

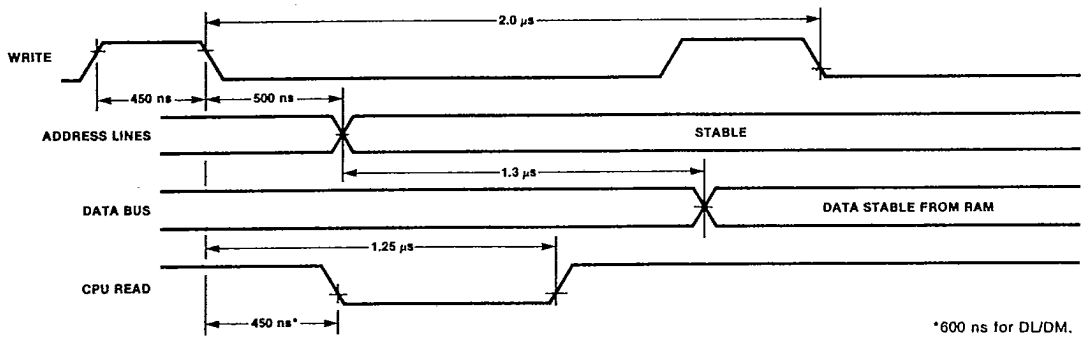
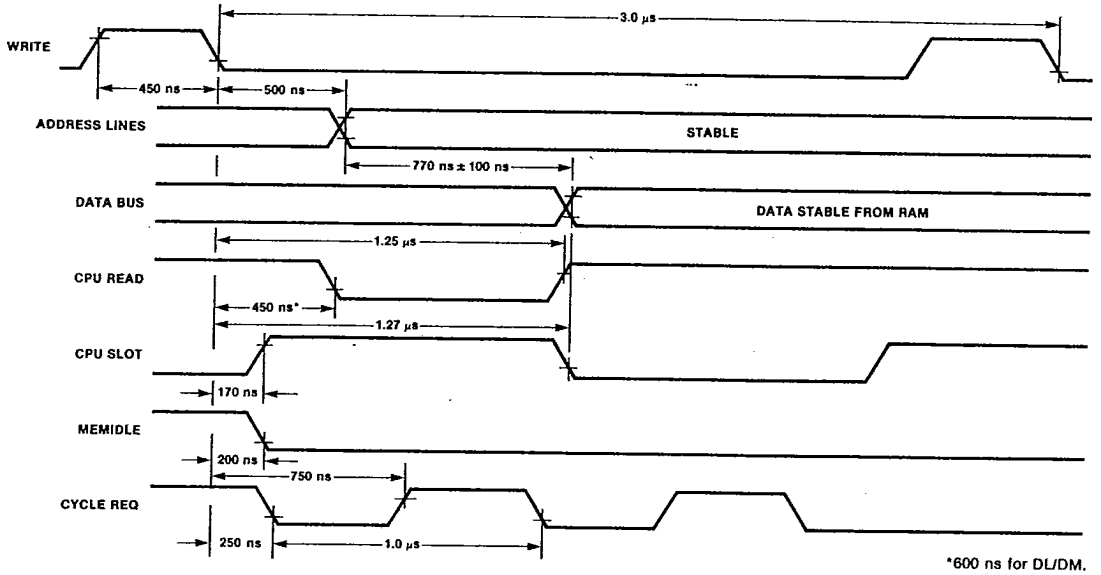
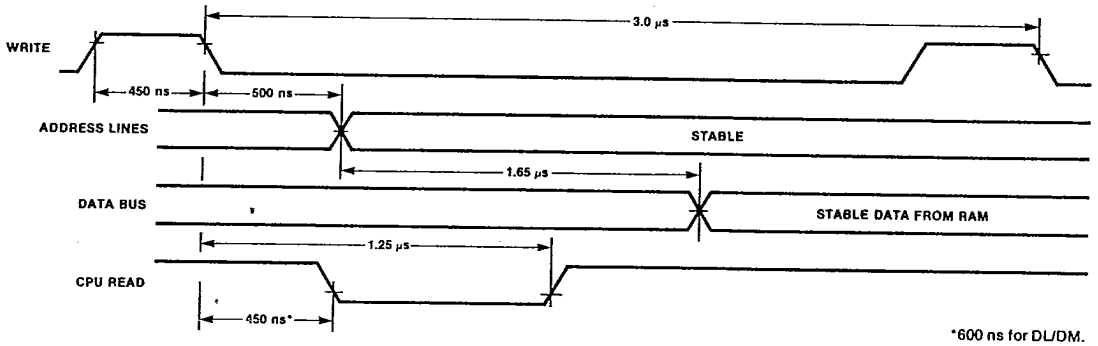


Fig. 13 DMI Timing Signals Output During Long-Cycle Memory Read, with Address Out of Program Counter



*600 ns for DL/DM.

Fig. 14 SMI Timing Signals Output During Long-Cycle Memory Read, with Address Out of Program Counter



*600 ns for DL/DM.

Fig. 15 DMI Timing Signals Output During Long-Cycle Memory Read, with Address Out of Data Counter

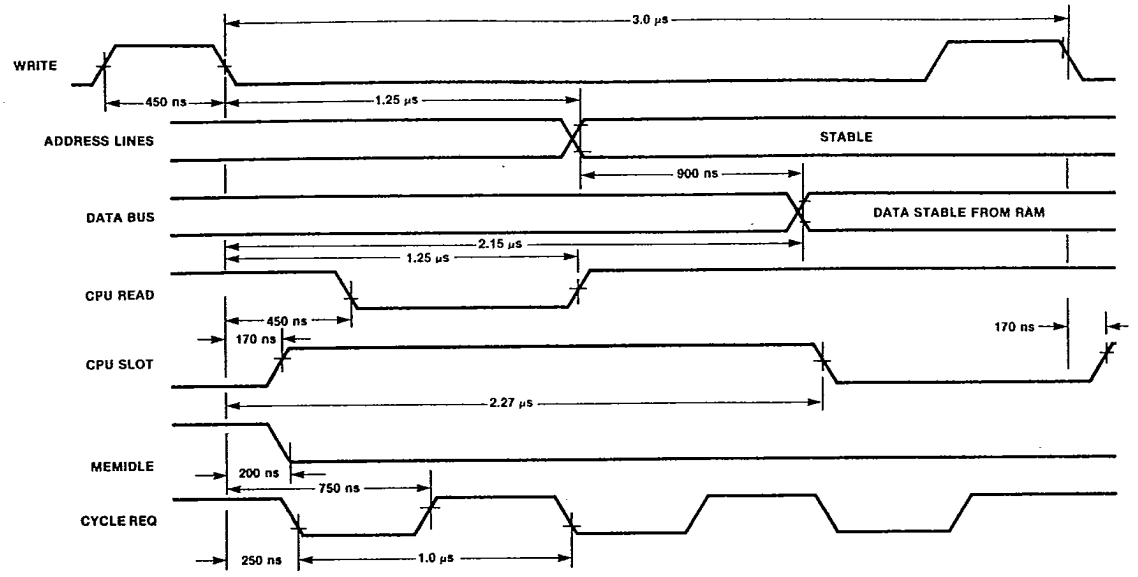


Fig. 16 SMI Timing Signals Output During Long-Cycle Memory Read, with Address Out of Data Counter

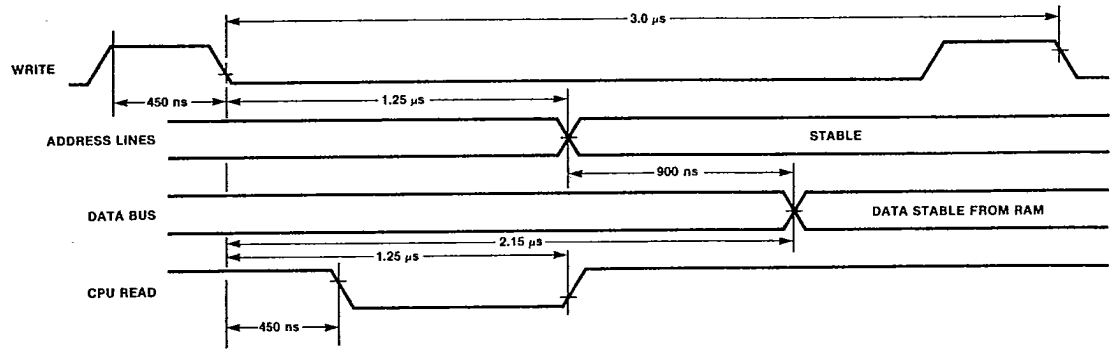


Fig. 17 DMI Timing Signals Output During a Write-to-Memory

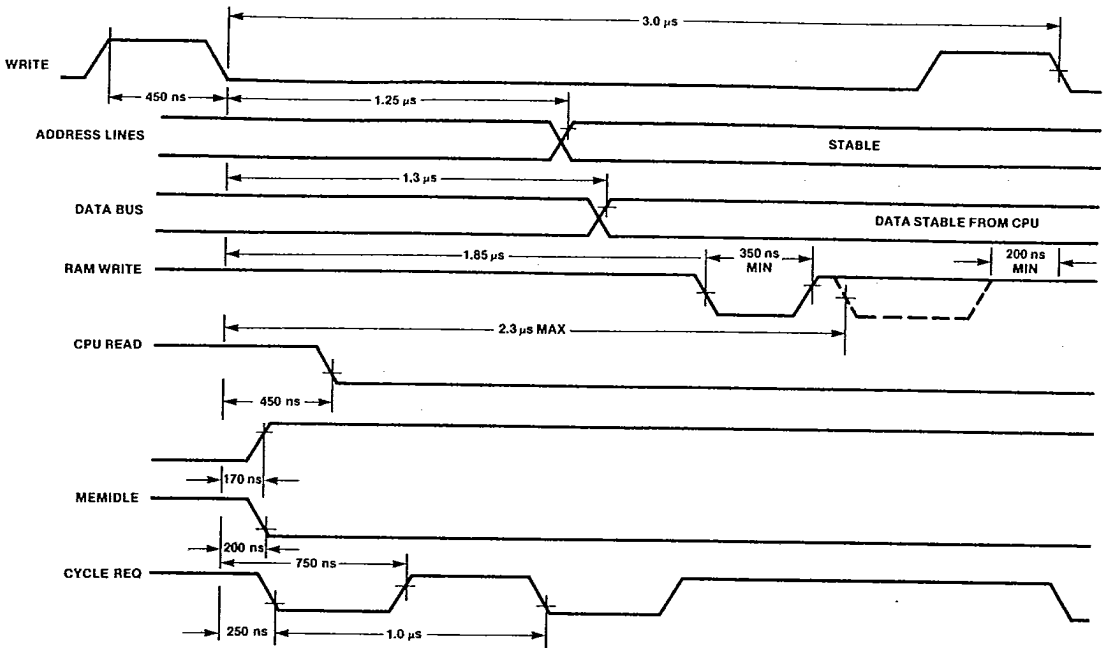
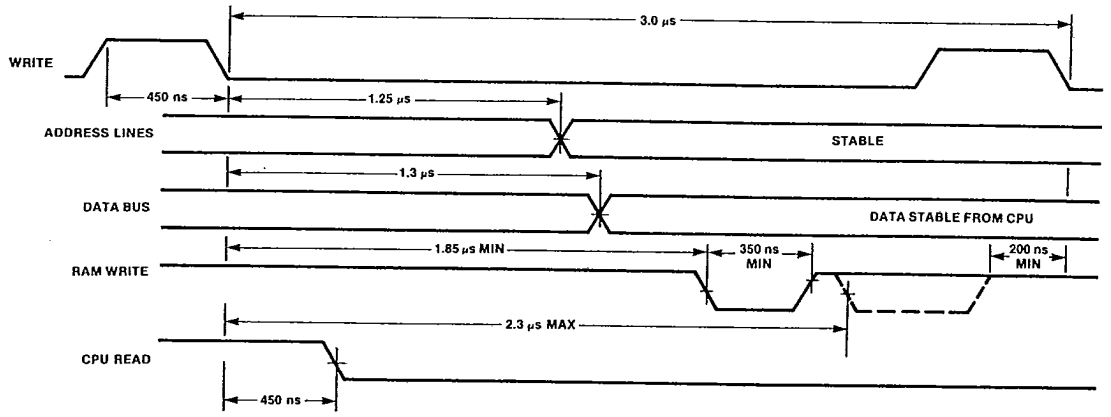


Fig. 18 SMI Timing Signals Output During a Write-to-Memory



2483 B-14

Timing Characteristics

The timing characteristics of the DMI and SMI are described in *Tables 3* and *4*, respectively.

Table 3 DMI Output Signals Timing Characteristics
$V_{SS}=0$ V, $V_{DD}=+5$ V $\pm 5\%$, $V_{GG}=+12$ V $\pm 5\%$, $T_A=0°C$ to $+70°C$.

Symbol	Characteristic	Min	Typ	Max	Unit	Notes
$P\phi$	ϕ Clock Period	0.5		10	μs	
tad_1	Address Delay if PC0	50	300	500	ns	3
tad_2	Address Delay to High Z (Short Cycle With DMA On)	tcs_2+50		tcs_2+200	ns	3
tad_3	Address Delay to Refresh (Short Cycle With REF On)	tcs_2+50		tcs_2+400	ns	3
tad_4	Address Delay if DC	$2P\phi+50-td_2$		$2P\phi+400-td_2$	ns	3
tad_5	Address Delay to High Z (Long Cycle With DMA On)	tcs_3+50		tcs_3+200	ns	3
tad_6	Address Delay to Refresh (Long Cycle With REF On)	tcs_3+50		tcs_3+400	ns	3
tcr_1	CPU READ – Delay	50	250	450	ns	1, 7
tcr_2	CPU READ + Delay	$2P\phi+50-td_2$		$2P\phi+400-td_2$	ns	1
tcs_1	CPU SLOT + Delay	$80-td_2$		$320-td_2$	ns	1
tcs_2	CPU SLOT – Delay (PC0 Access)	$2P\phi+60-td_2$		$2P\phi+420-td_2$	ns	1
tcs_3	CPU SLOT – Delay (DC Access)	$4P\phi+60-td_2$		$2P\phi+420-td_2$	ns	1
tm_1	MEMIDLE + Delay (PC0 Access)	$2P\phi+50-td_2$		$4P\phi+400-td_2$	ns	1
tm_2	MEMIDLE – Delay (PC0 Access)	$4P\phi+50-td_2$		$4P\phi+350-td_2$	ns	1
tm_3	MEMIDLE + Delay (DC Access)	$4P\phi+50-td_2$		$4P\phi+400-td_2$	ns	1
tm_4	MEMIDLE – Delay (DC Access)	$6P\phi+50-td_2$		$6P\phi+350-td_2$	ns	1
tcy_1	WRITE to CYCLE REQ – Delay	$80-td_2$		$400-td_2$	ns	1, 4
tcy_2	WRITE to CYCLE REQ + Delay	$P\phi+80-td_2$		$P\phi+400-td_2$	ns	1, 4
tcy_3	CYCLE REQ + to + Edge Delay		$2P\phi$			1, 4
tcy_4	CYCLE REQ – to – Edge Delay		$2P\phi$			1, 4
twr_1	$\overline{\text{RAM WRITE}}$ – Delay	$4P\phi+50-td_2$		$4P\phi+450-td_2$	ns	3
twr_2	$\overline{\text{RAM WRITE}}$ + Delay	$5P\phi+50-td_2$		$5P\phi+300-td_2$	ns	3
twr_3	$\overline{\text{RAM WRITE}}$ Pulse Width	350		$P\phi$	ns	3
twr_4	$\overline{\text{RAM WRITE}}$ to High-Z Delay	tcs_2+40		tcs_2+200	ns	3
trg_1	REGDR – Delay	70	300	500	ns	1
trg_2	REGDR + Delay	$2P\phi+80-td_2$		$2P\phi+500-td_2$	ns	1
td_4	WRITE to Data Bus Input Delay			$2P\phi+1000$	ns	
td_7	WRITE to Data Bus Output Delay	$2P\phi+100-td_2$		$2P\phi+850-td_2$	ns	2

Notes

1. $C_L=50$ pF
2. $C_L=100$ pF
3. $C_L=500$ pF
4. CYCLE REQ is a divide-by-2 of ϕ for all instructions except the STORE instruction.
5. On a given chip, the timing for all signals tends to track. For example, if CPU SLOT for a particular chip is fairly slow and its timing falls near the Max delay value specified, the timing for all signals on that chip tends to be near the Max delay values. This is a result of processing parameters (which affect device speed), which are quite uniform over small physical areas on the surface of a wafer.
6. Input and output capacitance is 3 pF to 5 pF, typical, on all pins except V_{DD}, V_{GG}, and V_{SS}.
7. 600 ns max for DL/DM.

Table 4 SMI Output Signals Timing Characteristics $V_{SS} = 0$ V, $V_{DD} = +5$ V ±5%, $V_{GG} = +12$ V ±5%, $T_A = 0°C$ to $+70°C$.

Symbol	Characteristic	Min	Typ	Max	Unit	Notes
$P\phi$	ϕ Clock Period	0.5		10	μs	
tad_1	Address Delay if PC0	50	300	500	ns	3
tad_4	Address Delay if DC0	$2P\phi + 50 - td_2$		$2P\phi + 400 - td_2$	ns	3
tcr_1	CPU READ – Delay	50	250	450	ns	1, 8
tcr_2	CPU READ + Delay	$2P\phi + 50 - td_2$		$2P\phi + 400 - td_2$	ns	1
twr_1	$\overline{RAM}$ $\overline{WRITE}$ – Delay	$4P\phi + 50 - td_2$		$4P\phi + 450 - td_2$	ns	3
twr_2	$\overline{RAM}$ $\overline{WRITE}$ + Delay	$5P\phi + 50 - td_2$		$5P\phi + 300 - td_2$	ns	3
twr_3	$\overline{RAM}$ $\overline{WRITE}$ Pulse	350		$P\phi$	ns	3
trg_1	REGDR – Delay	70	300	500	ns	1
trg_2	REGDR + Delay	$2P\phi + 80 - td_2$		$2P\phi + 500 - td_2$	ns	1
td_4	WRITE to Data Bus Input Delay			$2P\phi + 1000$	ns	
td_7	WRITE to Data Bus Output Delay	$2P\phi + 100 - td_2$		$2P\phi + 850 - td_2$	ns	2
tr_1	WRITE to $\overline{INT}$ $\overline{REQ}$ – Delay			430	ns	2, 6
tpr_1	$\overline{PRI}$ $\overline{IN}$ to $\overline{INT}$ $\overline{REQ}$ – Delay		200	240	ns	2, 7
t_{ex}	$\overline{EXT}$ $\overline{INT}$ Setup Time	400			ns	

Notes

1. $C_L = 50$ pF
2. $C_L = 100$ pF
3. $C_L = 500$ pF
4. On a given chip, the timing for all signals tends to track. For example, if CPU SLOT for a particular chip is fairly slow and its timing falls near the Max delay value specified, the timing for all signals on that chip tends to be near the Max delay values. This is a result of processing parameters (which affect device speed), which are quite uniform over small physical areas on the surface of a wafer.
5. Input and output capacitance is 3 pF to 5 pF, typical, on all pins except V_{DD}, V_{GG}, and V_{SS}.
6. Assume Priority In was enabled (PRI IN = 0) in previous F8 cycle before interrupt is detected in the PSU.
7. The PSU has interrupt pending before Priority In is enabled.
8. 600 ns max for DL/DM.

DMI System RAM Characteristics

The ac characteristics of static and dynamic RAMs suitable for use with the DMI can be derived from the worst-case timing waveforms presented in *Figures 3, 4, 5, 11, 13, 15,* and *17*. Three distinct cases arise.

1. Static RAM with no DMA or refresh. The timing characteristics recommended are identical for the DMI and SMI.

2. The DMI used with dynamic RAM and no DMA. The recommended RAM characteristics are as follows:

Access Time	500 ns max
Address Setup Time to Write	600 ns max
Data Setup Time to Write	550 ns max
Write Pulse Width	350 ns max
Data and Address Hold Times	200 ns max
Read Cycle Time	900 ns max
Write Cycle Time	3 μs max

3. A system using either static or dynamic RAM with a DMI and DMA. The DMA access dominates the timing requirements, with resulting recommended RAM characteristics as follows:

Access Time	550 ns max
Address/Data Stable Time*	580 ns max

SMI System RAM Characteristics

The following RAM characteristics are recommended for use with the SMI, based on the waveforms shown in *Figures 12, 14, 16,* and *18*.

Access Time	900 ns max
Address Setup Time to Write	600 ns max
Data Setup Time to Write	550 ns max
Write Pulse Width	350 ns max
Data and Address Hold Times	200 ns max

The above times must also allow for any buffer delays that may be present on the data bus.

*In most memory specifications this is the cycle enable width during Read or Write.

DC Characteristics

The dc characteristics of the F3852 DMI and F3853 SMI are provided in *Table 5*.

Supply Currents

Symbol	Characteristic	Min	Typ	Max	Unit	Test Conditions
I_{DD}	V_{DD} Current		35	70	mA	f = 2 MHz, Outputs Unloaded
I_{GG}	V_{GG} Current		13	30	mA	f = 2 MHz, Outputs Unloaded

Absolute Maximum Ratings

These are stress ratings only, and functional operation at these ratings, or under any conditions above those indicated in this data sheet, is not implied. Exposure to the absolute maximum rating conditions for extended periods of time may affect device reliability, and exposure to stresses greater than those listed may cause permanent damage to the device.

V_{GG}	−0.3 V, +15 V
V_{DD}	−0.3 V, +7 V
All Other Inputs and Outputs	−0.3 V, +7 V
Storage Temperature	−55°C, +150°C
Operating Temperature	0°C, +70°C

Note
All voltages with respect to V_{SS}.

Recommended Operating Ranges

Part Number	Supply Voltage (V_{DD})			V_{GG}			V_{SS}
	Min	Typ	Max	Min	Typ	Max	
F3852/F3853	+4.75 V	+5 V	+5.25 V	+11.4 V	+12 V	+12.6 V	0 V

Ordering Information

Order Code	Package	Temp. Range*
F3852PC	Plastic	C
F3852PL	Plastic	L
F3852DC	Ceramic	C
F3852DL	Ceramic	L
F3852DM	Ceramic	M
F3853PC	Plastic	C
F3853PL	Plastic	L
F3853DC	Ceramic	C
F3853DL	Ceramic	L
F3853DM	Ceramic	M

*C = Commercial Temperature Range 0°C to +70°C
L = Limited Temperature Range −40°C to +85°C
M = Military Temperature Range −55°C to +125°C

Table 5 DMI and SMI DC Characteristics
V_{SS} = 0 V, V_{DD} = +5 V ±5%, V_{GG} = +12 V ±5%, T_A = 0°C to +70°C, unless otherwise noted.

Signal	Symbol	Characteristic	Min	Max	Unit	Test Conditions
Data Bus (DB_0-DB_7)	V_{IH}	Input High Voltage	2.9	V_{DD}	V	
	V_{IL}	Input Low Voltage	V_{SS}	0.8	V	
	V_{OH}	Output High Voltage	3.9	V_{DD}	V	I_{OH} = −100 μA
	V_{OL}	Output Low Voltage	V_{SS}	0.4	V	I_{OL} = 1.6 mA
	I_{IH}	Input High Current		3	μA	V_{IN} = V_{DD}, 3-State Mode
	I_{IL}	Input Low Current		-3	μA	V_{IN} = V_{SS}, 3-State Mode
Address Lines ($ADDR_0$-$ADDR_{15}$) and $\overline{RAM\ WRITE}$	V_{OH}	Output High Voltage	4.0	V_{DD}	V	I_{OH} = −1 mA
	V_{OL}	Output Low Voltage	V_{SS}	0.4	V	I_{OL} = 3.2 mA
	I_L	Leakage Current		3	μA	V_{IN} = V_{DD}, 3-State Mode
	I_L	Leakage Current		-3	μA	V_{IN} = V_{SS}, 3-State Mode
Clock (φ, WRITE)	V_{IH}	Input High Voltage	4.0	V_{DD}	V	
	V_{IL}	Input Low Voltage	V_{SS}	0.8	V	
	I_L	Leakage Current		3	μA	V_{IN} = V_{DD}
MEMIDLE, CYCLE REQ, CPU READ	V_{OH}	Output High Voltage	3.9	V_{DD}	V	I_{OH} = −1 mA
	V_{OL}	Output Low Voltage	V_{SS}	0.4	V	I_{OL} = 2 mA
Control Lines ($ROMC_0$-$ROMC_4$), $\overline{PRI\ IN}$	V_{IH}	Input High Voltage	3.5	V_{DD}	V	
	V_{IL}	Input Low Voltage	V_{SS}	0.8	V	
	I_L	Leakage Current		3	μA	V_{IN} = 6 V
REGDR, CPU SLOT	V_{OH}	Output High Voltage	3.9	V_{DD}	V	I_{OH} = −300 μA
	V_{OL}	Output Low Voltage	V_{SS}	0.4	V	I_{OL} = 2 mA
	V_{IH}	Input High Voltage	3.5	V_{DD}	V	Internal Pull-Up
	V_{IL}	Input Low Voltage	V_{SS}	0.8	V	
	I_{IL}	Input Low Current (REGDR)	-3.5	-14.0	mA	V_{IN} = 0.4 V & Device Outputting a Logic "1"
	I_L	Leakage Current		3	μA	V_{IN} = 6 V
Interrupt Request ($\overline{INT\ REQ}$)	V_{OH}	Output High Voltage			V	Open-Drain Output [1]
	V_{OL}	Output Low Voltage	V_{SS}	0.4	V	I_{OL} = 1 mA
	I_L	Leakage Current		3	μA	V_{IN} = V_{DD}
External Interrupt ($\overline{EXT\ INT}$)	V_{IH}	Input High Voltage	3.5		V	
	V_{IL}	Input Low Voltage		0.8	V	
	V_{IC}	Input Clamp Voltage		15	V	I_{IH} = 185 μA
	I_{IH}	Input High Current		10	μA	V_{IN} = V_{DD}
	I_{IL}	Input Low Current		-255	μA	V_{IN} = 2 V
	I_{IL}	Input Low Current	-150	-500	μA	V_{IN} = V_{SS} [1]

Notes
1. The values are −100 μA and −750 μA for DL/DM extended temperature ratings.
2. Positive current is defined as conventional current flowing into the pin referenced.