

# Using Fairchild's MICROWIRE™ EEPROM

Fairchild  
Application Note 758



Fairchild Semiconductor manufactures a wide range of low density serial EEPROMs that use the MICROWIRE interface as a means of communication. Although all of these devices use the MICROWIRE interface, there are slight variations in interfacing due to differences in memory sizes, features, and technology used to implement the device. Additionally, the MICROWIRE interface does not specifically define any protocol, it only defines a basic set of signal lines to interconnect two or more devices. Due to these reasons, additional information is necessary to fully understand how to best interface to Fairchild's family of MICROWIRE EEPROM.

The goal of this application guide is to cover a diversity of information in regard to basic timing, interfacing options, and functionality of different EEPROMs. I will use an outline approach, so the appropriate heading can be located easily. Each section attempts to be stand alone so the information can be easily extracted. The outline appears below:

## OUTLINE

1. 1.0 Description of EEPROM Families
  - A. 1.1 CMOS EEPROM
  - B. 1.2 NM93C Family
  - C. 1.3 NM93CS Family
  - D. 1.4 NM93CxxA Family
2. 2.0 HARDWARE CONNECTIONS
  - A. 2.1. INTERFACE PIN DESCRIPTIONS
    - a. 2.1.1 Chip Select (CS)
    - b. 2.1.2 Serial Clock (SK)
    - c. 2.1.3 Data-In (DI)
    - d. 2.1.4 Data-Out (DO)
    - e. 2.1.5 Program Enable (PE)
    - f. 2.1.6 Protect Register Enable (PRE)
    - g. 2.1.7 Organization (ORG)
    - h. 2.1.8 Status (RDY/BUSY)
  - B. 2.2. FOUR WIRE BUS
  - C. 2.3. THREE WIRE BUS
3. 3.0 TIMING CONSIDERATIONS
  - A. 3.1 BUS TIMING
  - B. 3.2 INSTRUCTION SEQUENCE DESCRIPTIONS
    - a. 3.2.1 Read Cycle
    - b. 3.2.2 Sequential Read
    - c. 3.2.3 Erase and Erase All
    - d. 3.2.4 Write and Write All
    - e. 3.2.5 Program Enable and Program Disable
    - f. 3.2.6 Protect Register Read
    - g. 3.2.7 Protect Register Enable
    - h. 3.2.8 Protect Register Disable
    - i. 3.2.9 Protect Register Clear
    - j. 3.2.10 Protect Register Write
  - C. 3.3 INTERFACING SOLUTIONS
4. 4.0 CONCLUSION

## 1.0 Description of EEPROM Families

### 1.1 CMOS EEPROM

Fairchild builds a range of MICROWIRE CMOS EEPROMs in memory sizes ranging from 256-bit to 16,384-bit. The NM93C family is the base family and the NM93CS is a similar family with additional features, there are also other devices with slight variations on the interface. All these devices are available with certain "standard" options such as operating temperature ranges and operating voltage ranges, packaging options and test options. These options being fairly standard variations for semiconductor devices, will not be addressed beyond this. The purpose of this article is to address basic functionality and interfacing, including various tricks to simplify or modify the interface.

### 1.2 NM93C Family

The NM93C family of EEPROM is available in 256-, 1024-, 2048-, 4096-bit, and 16,384-bit sizes. All of these are internally organized in 16-bit words, therefore all data transactions deal with 16 bits. This family of EEPROMs has 7 instructions that deal with read, write, and a basic level of data protection. The instructions are listed in Table 1. It is important to note that there is a basic difference in length of the instruction between the NM93C06 or NM93C46 and the NM93C56 or NM93C66. This is due to the larger devices needing additional address bits.

The NM93C family of EEPROM, like all of Fairchild's serial EEPROMs have a basic level of write protection that can be turned on or off by the use of the ERASE/WRITE DISABLE (EWDS) and ERASE/WRITE ENABLE (EWEN) instructions. Although there are two erase instructions included in the NM93C family, these are included only for compatibility with older EEPROMs that require erase before write. These EEPROMs don't require erase before write and it is recommended that in application the erase not be used as this adversely affects endurance.

### 1.3 NM93CS Family

The NM93CS EEPROMs are available in 256-, 1024-, 2048-, and 4096-bit sizes. Making them different, they have two additional functions, sequential read and user configurable write protection, and don't have either of the erase functions, ERASE and ERASE-ALL as they are not needed. Like all of the CMOS EEPROMs, these have self timed programming cycles and operate from a single external supply of either 4.5V to 5.5V or 2.7V to 5.5V. In these devices it is necessary to eliminate the erase cycles from the code as they may adversely affect the performance of the device.

As these have additional functions, the instruction set includes a total of 10 instructions, 3 that operate on the memory array, 2 that deal with the basic write protection and 5 that deal with the user configurable write protection. Refer to the NM93CS instruction set table Table 2 for definitions of these instructions. As with the NM93C family, there is a basic difference in instruction length depending on memory size.

To further increase data security in these EEPROMs there are also two additional input signals defined, Program Enable (PE) and Protect Register Enable (PRE). These signals are on pins that

are unused on the NM93C family providing upward compatibility to the NM93CS devices.

**TABLE 1. NM93C Family Instruction Set Table**

Instruction	SB	Op Code	Address	Data	Comments
READ	1	10	A7/A5–A0		Reads data stored in memory.
EWEN	1	00	11XXXX		Write enable must precede all programming modes.
ERASE	1	11	A5–A0		Erase register A5A4A3A2A1A0.
WRITE	1	01	A5–A0	D15–D0	Writes register.
ERAL	1	00	10XXXX		Erase all registers.
WRAL	1	00	01XXXX	D15–D0	Writes all registers.
EWDS	1	00	00XXXX		Disables all programming instructions.

**TABLE 2. NM93CS Family Instruction Set Table**

Instruction	SB	Op Code	Address	Data	PRE	PE	Comments
READ	1	10	A5–A0		0	X	Reads data stored in memory, starting at specified address.
WEN	1	00	11XXXX		0	1	Write enable must precede all programming modes.
WRITE	1	01	A5–A0	D15–D0	0	1	Writes register if address is unprotected.
WRALL	1	00	01XXXX	D15–D0	0	1	Writes all registers. Valid only when Protect Register is cleared.
WDS	1	00	00XXXX		0	X	Disables all programming instructions.
PRREAD	1	10	XXXXXX		1	X	Reads address stored in Protect Register.
PREN	1	00	11XXXX		1	1	Must immediately precede PRCLEAR, PRWRITE, and PRDS instructions.
PRCLEAR	1	11	111111		1	1	Clears the Protect Register so that no registers are protected from WRITE.
PRWRITE	1	01	A5–A0		1	1	Program address into Protect Register. Thereafter, memory addresses $\geq$ the address in Protect Register are protected from WRITE.
PRDS	1	00	000000		1	1	One time only instruction after which the address in the Protect Register cannot be altered.

## 1.4 NM93CxxA Family

There are two variations on the standard implementation of the Microwire bus. Both variations can be viewed as enhancements. The first enhancement is a Organization (ORG) input that allows the user to select the internal configuration of the memory as either 8 bits wide or 16 bits wide. When the input is high or unconnected, the device is configured as 16 bits wide, when the ORG input is at a low level, the memory is configured as 8 bits wide, but twice as deep. The feature is present on the NM93C46A, NM93C56A, NM93C66A, and NM93C86A.

The second variation is the STATUS output. This is the Busy/Ready polling to indicate programming status. All other devices have this feature on the Data-Out (DO) output, the NM59C11 alone has status available as a separate output and not on the Data-Out output. This can simplify interfacing to a bidirectional data bus.

## 2.0 Hardware Connection

### 2.1 INTERFACE PIN DESCRIPTIONS

In this section, each possible input or output will be described followed by the most popular variations of bus connections. Not all devices have all of the described I/Os. The I/Os are available according to Table 3, I/O Functionality.

#### 2.1.1 CHIP SELECT (CS)

Chip Select is used to differentiate between various devices on the same Microwire bus. In the case of EEPROM it cannot be tied high even if it is the only device on the bus as it performs several additional functions. As it applies to any of the Microwire EEPROMs, the rising edge resets the internal circuitry of the device, a function necessary prior to initiating

any new cycle. As shown in the functional block diagram (Figure 1) chip select also gates the data input and clock input, thus disabling these functions.

During the course of clocking in the start bit, op-code address and data-in or data-out, chip select must be held high continuously, otherwise the internal circuits will be reset and the cycle will have to be started again with a new start bit.

During programming cycles chip select initiates the internal programming cycle. The falling edge of chip select will start the internal programming cycle when a programming op-code has been entered (Erase, Write, Erase All, Write All) and then, in conjunction with Data-Out (DO), will indicate if programming is complete. If programming is complete, Data-Out will drive high, if incomplete it will drive low.

Several additional notes in regard to chip select:

If a programming cycle is partially clocked in and then chip select dropped, the EEPROM may enter into a programming mode. This

is determined by how many bits have been clocked in when chip select is dropped. If the start bit, op-code, and all of the address has been clocked in, a programming cycle will be initiated with no or partial data. If less than a complete address has been clocked in, the programming cycle will not be initiated. Refer to Figure 2, reference line 1.

Chip select hold time at the end of a cycle is referenced to the last rising edge of clock (SK). The hold time from the rising edge is the same as the minimum SK high time for the particular device. This is stated in the datasheets as 0 ns hold time from the falling edge of SK which assumes that SK high time is always minimum. In this case SK can be left in the high state or taken low at a later time. Internally chip select gates SK, therefore SK is not critical.

**TABLE 3. I/O Functionality by Device**

	<b>CS</b>	<b>SK</b>	<b>DI</b>	<b>DO</b>	<b>PE</b>	<b>PRE</b>	<b>ORG</b>	<b>STAT</b>
NM93C Family	X	X	X	X				
NM93CS Family	X	X	X	X	X	X		
NM93C46A	X	X	X	X			X	

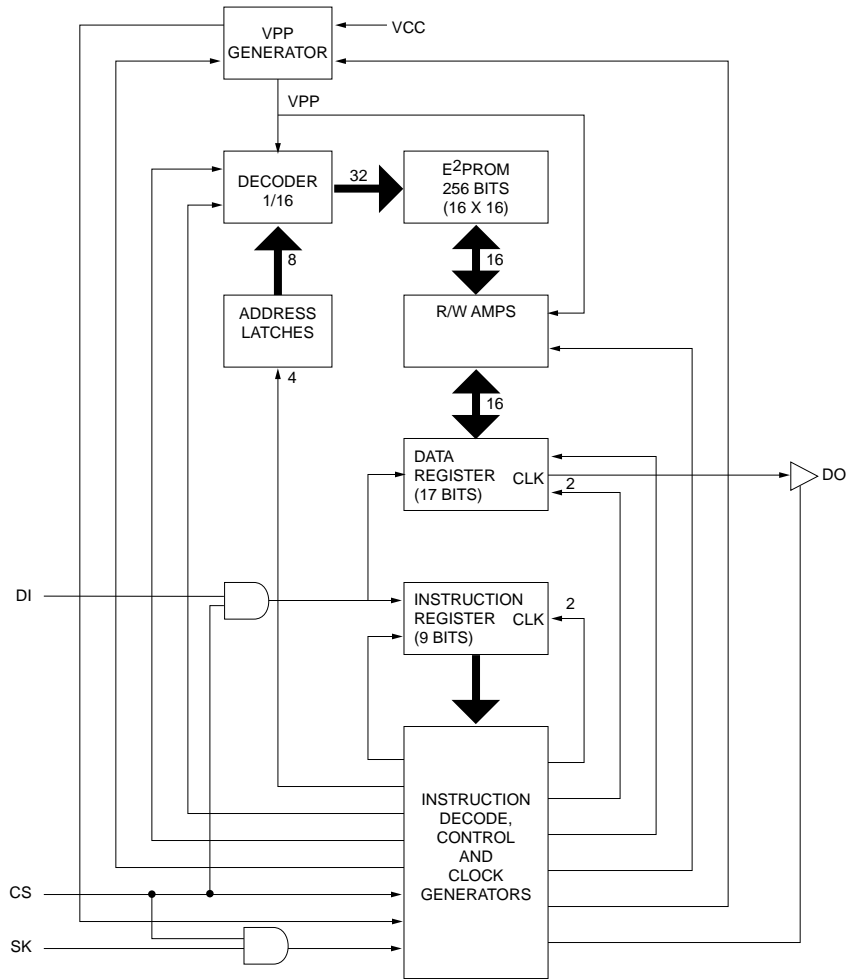


FIGURE 1. Block Diagram

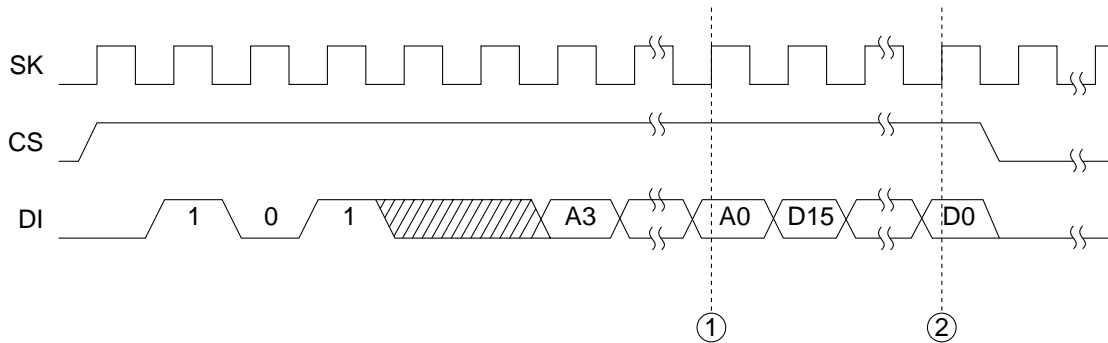


FIGURE 2. Programming Cycle Point of No Return

### 2.1.2 SERIAL CLOCK (SK)

The clock input is used to clock all data, address, op-code, and start bits into or out of the EEPROMs. SK clocks both input and output on the rising edge only, the falling edge has no effect on the devices. The only function it is not necessary for is the Busy/Ready Polling which is an asynchronous function.

Since SK is gated by chip select, it is a "Don't Care" any time chip select is low. It is also don't care prior to a start bit being clocked in and during Busy/Ready Polling. During these conditions Data-In (DI) must be held at a low level, otherwise a start bit will be interpreted.

If it is desirable to insert additional clock cycles during a instruction sequence for the purpose of byte aligning the data,

there are several places in the data stream they may be inserted as described below:

- On any instruction, zeros can be clocked into the DI input before the start bit. Any number of clock cycles may be added if Data-In (DI) is held at zero. The first 1 clocked in will be interpreted as the start bit. This requires special precautions if a bidirectional data bus is used (Data-In tied to Data-Out) as the Busy/Ready Polling will interfere with the Data-In if it is not cleared out at the end of each programming cycle. See Section 2.3, THREE WIRE BUS, for more information.
- During a Read instruction, it is allowable to continue to clock the device after the 16 bits of data has been clocked out. In the case of the NM93CS family this will cause the memory to increment to the next register and present its contents on the Data-Out pin. In the case of all other devices, whatever was present on the Data-In pin will become present on the Data-Out pin (Fall thru). Refer to Figure 1, Block Diagram.
- During a Write or Write-All, additional clock cycles may be added after address A0 and before the valid data. The EEPROM will write into the memory the most recent 16 bits, or in the case of the NM93C46A, the most recent 8 bits or 16 bits depending on the status of the ORG input. Adding additional clocks after the valid data will cause the data to be misaligned.
- During the EWEN, Erase, Erase All, EWDS, WEN, WDS cycles, it is not necessary to clock in a data field, although it is mandatory to clock in a complete address field, even if the addresses are "Don't Care". Additional clocks can be added after the address field.

### 2.1.3 DATA-IN (DI)

The Data-In input receives the Start-Bit, Address, and input data in a serial stream, each bit clocked in on the rising edge of SK. DI is gated by the chip select to provide a high degree of noise immunity. As shown in the block diagram, Data-In is routed to both the instruction shift register and the data shift register. When the start bit is clocked into the last bit of the instruction register, the clock is switched to the data register to receive input data and clock data out simultaneous. The Data-Out remains in high impedance unless a read cycle or Busy/Ready status is being done. The safest state is to keep the Data-In pin in a low level as a start bit is a high level.

### 2.1.4 DATA-OUT (DO)

The Data-Out (DO) output sends read data onto the microwire bus and is clocked out on the rising edge of SK. It also carries the programming status after a programming cycle which is an asynchronous function that does not require the clock. At all other times the Data-Out is in the high impedance state. During a Read cycle, the Data-Out output begins to drive actively after the last address bit (A0) is clocked in. During the Busy/Ready polling it begins to drive active after chip select is raised to a high level.

During the Busy/Ready Polling, the Data-Out output drives low while the device is still in the internal programming cycle. After the EEPROM has completed the internal programming cycle, the Data-Out pin will drive high when chip select is high. Subsequently, if chip select is brought high again, Data-Out will again drive high indicating it has completed the programming cycle. To clear the Busy/Ready Polling it is necessary to raise chip select and clock in a start bit. Once the start bit is clocked in, Data-Out will return to the high impedance state. It is not necessary to continue with a cycle after this start bit has been clocked in, although it is permissible to start a new cycle with this start bit. This clearing of the Busy/Ready status may be necessary if a bidirectional data bus is used (Data-In tied to Data-Out) as the Data-Out output will interfere with the new data being presented on the Data-In input.

### 2.1.5 PROGRAM ENABLE (PE)

The program enable (PE) input will enable all programming cycles when it is held at a high level during the duration of a programming cycle. Conversely, it will disable all programming, including programming of the protect register, while it is held low. This input has no affect on any other cycle, so it may be permanently tied high or low, or may be used in an active mode. This input is available on the NM93CS family only.

### 2.1.6 PROTECT REGISTER ENABLE (PRE)

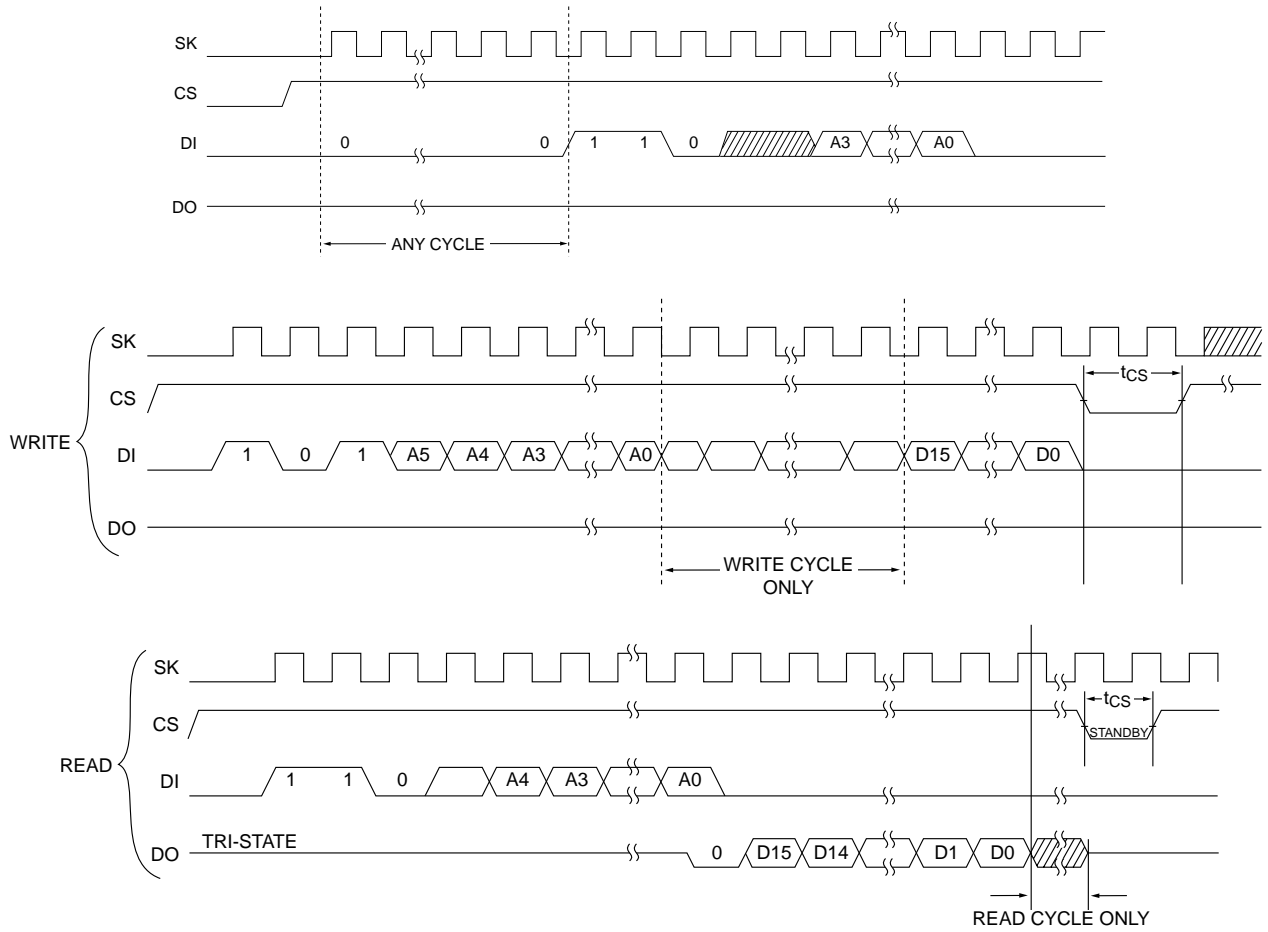
The protect register enable (PRE) input is used to switch between memory operations and protect register operations since the same op-codes are used for both. With the PRE input high, the op-codes define operations in the protect register, with the PRE input low, the op-codes define operations in the memory. This pin may be tied high or low, or used in the active mode. This input is available on the NM93CS family only.

### 2.1.7 ORGANIZATION (ORG)

The Organization input (ORG) is used to control the internal organization of the memory. The two selectable organizations are 16-bit words and 8-bit words. Simply by holding the ORG pin at a high level, 16-bit words are selected, by holding the input at a low level 8-bit words are selected. When in the 8-bit mode, one additional address bit is required in the instruction sequence since the depth of the memory is doubled. This input is available only on certain device types, refer to the individual datasheets.

### 2.1.8 STATUS (RDY/BUSY)

The status output indicates the programming cycle status after a programming cycle. When the device is in the programming mode and therefore cannot accept any other cycles, this pin will be low. After completion of the cycle the STATUS pin will be driven high. When this function is present, the Busy/Ready Polling is not available on the Data-Out



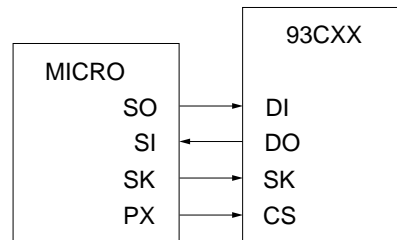
**FIGURE 3. Possible Locations for Additional SK Cycles**

output. In some systems, particularly those using a bi-directional data bus, this can simplify interfacing by eliminating the possible contention between the Ready indication and the incoming data from the host device. This output is available only on certain device types, refer to the individual datasheets.

**2.2 FOUR WIRE BUS**

The 4 wire bus is the simplest interconnection between the EEPROM and the host device. In most cases the only signals necessary to provide are clock, chip select, Data-In and Data-Out as shown in Figure 4. The PRE, PE, ORG, and STATUS pins are not shown as they are variations on this and the 3 wire bus connection. Multiple devices can be connected to the microwire

bus, the only limitations being loading and available chip select means. In some systems it is necessary to have a bi-directional data line as described below in 3 wire bus.



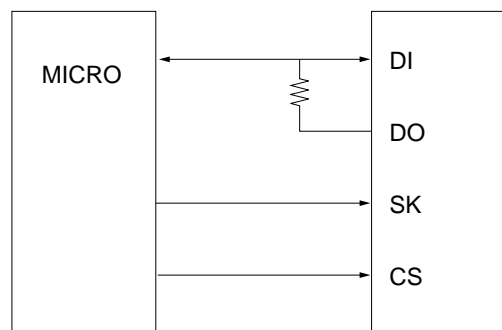
**FIGURE 4. Four Wire Connection**

## 2.3 THREE WIRE BUS

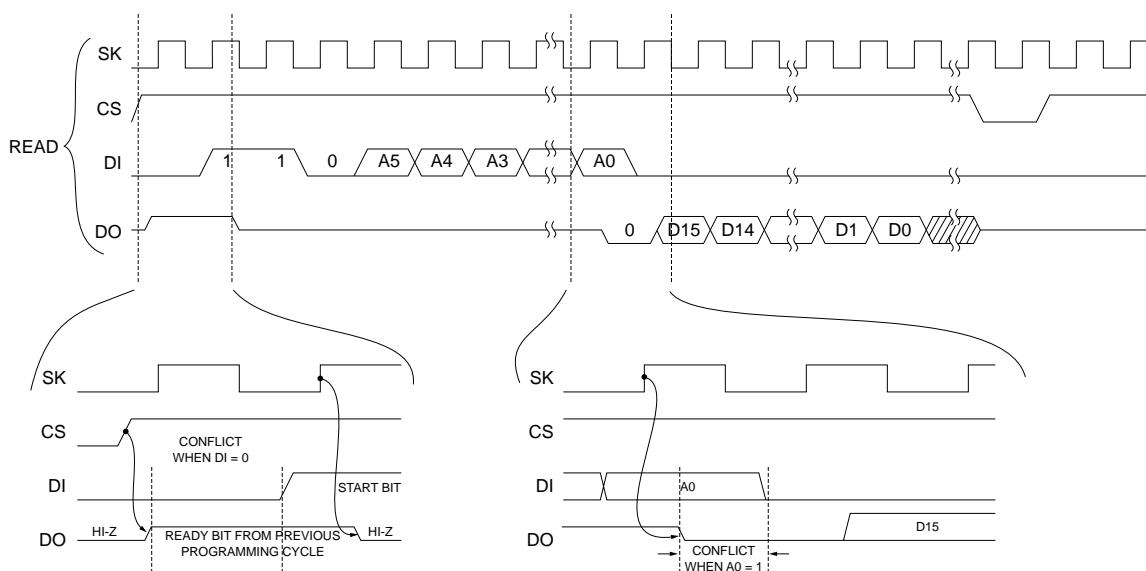
The 3 wire bus operates in the same mode as the 4 wire bus with the exception that the Data-In and Data-Out pins on the EEPROM are tied together. When using this connection, there are two precautions that need to be observed.

- When Data-In is tied to Data-Out, there is a possible conflict between address A0 in the instruction sequence and the dummy bit. This only occurs during a READ cycle. This is not harmful to the device and the internal circuitry of the EEPROM guarantees that the device will function properly under this condition. To decrease the noise created by the condition, a resistor may be placed in the locations indicated in Figure 5. The timing diagram in Figure 6 shows the bus conflict.
- The second possible area of conflict occurs when the Busy/Ready status is on the Data-Out output. Since the device will continue to indicate a Ready status indefinitely after a programming cycle (until a start-bit is clocked in), this can conflict with the beginning of the next cycle if leading zeros

are clocked in (See Figure 6). The solution is to either use a separate cycle to clear the Ready bit or to eliminate any leading zeros from the instruction sequence.



**FIGURE 5. Three Wire Connection Showing Optional Resistor**



**FIGURE 6. Three Wire Connection Bus Conflict Areas**

## 3.0 Timing Considerations

The following information describing the Microwire bus timing must be used in conjunction with the datasheet as it is an expansion and clarification of the datasheet. First, the basic timings with respect to the clock (SK) will be described, followed by instruction sequence timing, and finally, specific information in each instruction sequence.

### 3.1. BUS TIMING

The synchronous data timing shown in Figure 7 is similar to that shown in the various datasheets. There is one significant modification to the timing specification though, the chip select (CS) hold time is referenced to the rising edge of the clock rather than the falling edge. With this modification, the hold time specification

must be changed to be the same as the minimum clock (SK) high time. Other significant points are:

- The only active edge of the clock is the rising edge.
- The only time the clock is necessary is when clocking data into or out of the EEPROM. It is not necessary during Busy/Ready Polling.
- The clock may be left in either the high state or low state between cycles. It is safer to leave the clock in the low state.
- When chip select (CS) is high, clock (SK) is a critical signal. With the exceptions noted in Section 2.1.2 tilted SERIAL CLOCK (SK), no additional clock cycles or noise that crosses the  $V_{IH}$  or  $V_{IL}$  thresholds can be tolerated.



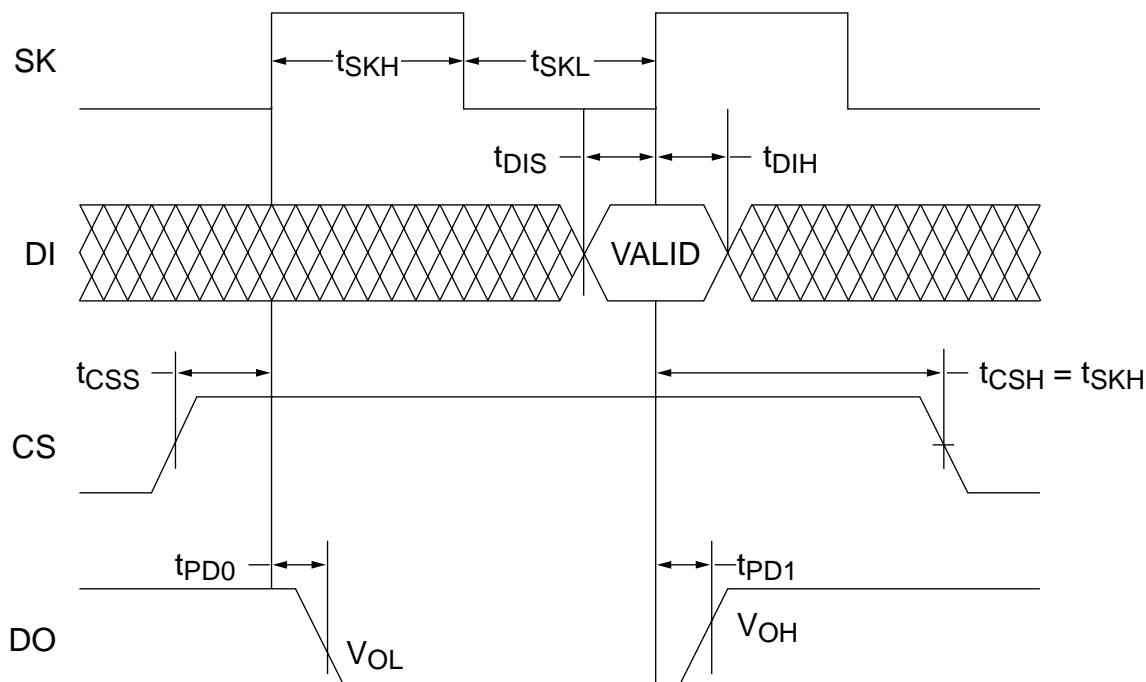


FIGURE 7. Synchronous Timing

## 3.2 INSTRUCTION SEQUENCE DESCRIPTIONS

### 3.2.1 READ CYCLE

The READ cycle requires the host to raise chip select (CS) and then clock in thru the Data-In (DI) pin a start-bit, op-code, and address. Following clocking in the last address bit, the Data-Out (DO) output comes out of the high impedance state and drives a low level on the output. This is referred to as the dummy bit and is a good indication that a READ mode has been successfully entered if difficulty is encountered during initial debug of a system. The dummy bit is clocked out of the EEPROM on the same rising edge of SK that clocks in the last address bit, A0. This is shown in Figure 8.

### 3.2.2 SEQUENTIAL READ

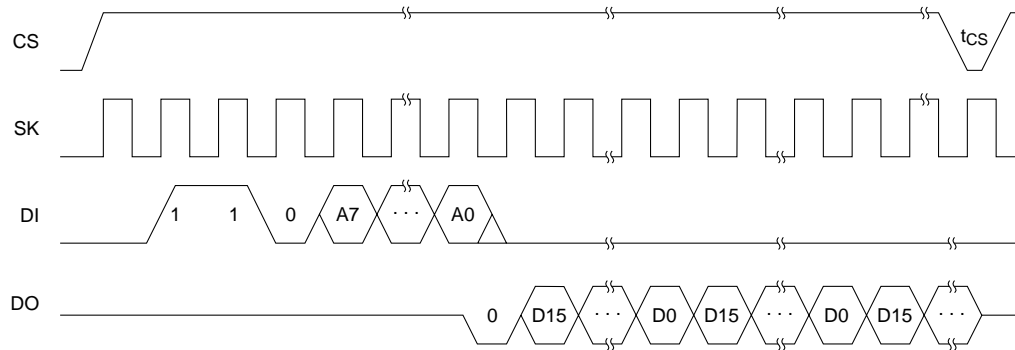
Sequential read is a read mode available only on the NM93CS family. It is entered by entering a READ cycle and clocking out the first 16-bit word. After reading the first 16-bit word if chip select (CS) is kept high, address A + 1 may be clocked out followed by address A + 2 and so on. When the maximum address is reached, the memory continues in the sequential read mode at address 0. In this manner, the host may operate the memory in a continuous loop read. When initiating a SEQUENTIAL READ, the first data

word is preceded by a dummy bit as in a standard READ, although the dummy bit is suppressed in all subsequent data words as shown in Figure 8.

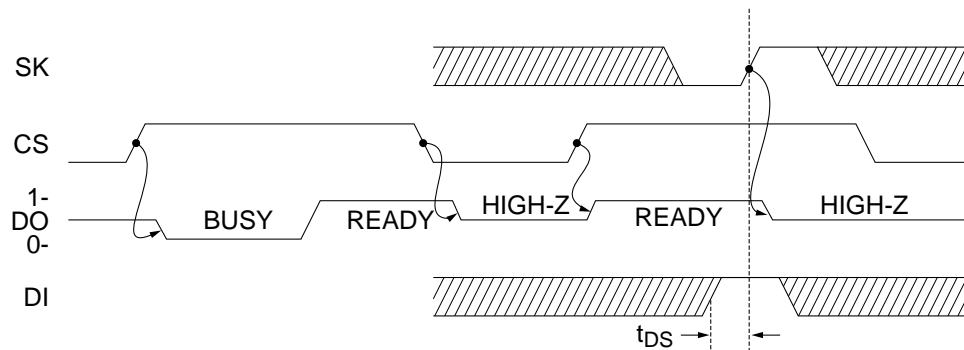
### 3.2.3 ERASE AND ERASE ALL

The ERASE cycles return the contents of the EEPROM to a clear state which is read as 1's. It is not necessary for any of the CMOS EEPROM described in this article, and is included in the NM93C family and NM93C46A only for compatibility with older devices that require erasing. It is recommended that the erase cycles be eliminated from the instructions to simplify the code, speed up writing and to improve the endurance obtained in the application. These modes are entered by clocking in a start-bit, op-code, and address. It is not necessary to clock in the data field as it is assumed to be all 1's. It is necessary to clock in the address, even in the case of ERASE-ALL where it is "don't care" in all except the first two bits of the address field which is used as additional op-code bits. After the full address field has been clocked in, chip select must be returned to a low level in initiate the erase cycle. In all devices, except the NMC9306, programming completion can be determined by Polling as shown in Figure 9, or a simple 10 ms timeout will guarantee programming is complete if polling is not used.





**FIGURE 8. Sequential Read Sequence**



**FIGURE 9. Busy/Ready Polling Sequence**

### 3.2.4 WRITE AND WRITE ALL

The Write and Write All cycles will write a specified data word into the specified address, or in the case of Write All, the same data pattern will be written into all locations. In all devices a new data pattern may be directly written over an existing data pattern without erasing the first data pattern. The write mode is entered by clocking in a start-bit, op-code, address, and data. The full address field must be clocked in for the Write All even though it doesn't care in all but the first 2 bits. It is also necessary to clock in a full data field to assure correct alignment of data. The write cycle will be initiated after 8- or 16-bit have been clocked into the device in some of the devices and in other devices after chip select is brought low regardless of how many data bits have been clocked in. Refer to the specific datasheets to determine which method is used.

### 3.2.5 PROGRAM ENABLE AND PROGRAM DISABLE

Program enable and program disable are the instructions that enable or disable writing and, where included, erasing. The instruction name varies depending on the specific device but includes EWEN, EWDS, WEN, and WDS. These instructions enable or disable the entire memory array with a single instruction. All devices power up in the disable mode and once placed in the enabled mode remain enabled until a disable instruction is performed or  $V_{CC}$  is cycled. These instructions provide the most basic level of data protection. Although since most lost data is the result of the host device becoming uncontrolled and performing the

“Program Subroutine” it may be helpful to structure the software such that the enable command is not included in the “Program Subroutine” but is in a separate subroutine. If a greater degree of data security is needed, a NM93CS family device is recommended, or other more elaborate schemes involving redundant data storage and polling.

### 3.2.6 PROTECT REGISTER READ

The protect register read (PRREAD) command is the same as a word read command except the input PRE must be held at a high level and the address is don't care. In spite of the address being don't care, the entire address field must be clocked in. On the Data-Out pin the contents of the protect register will be clocked out MSB first descending to LSB.

### 3.2.7 PROTECT REGISTER ENABLE

Similar to the programming enable instructions described above, the PREN instruction is necessary to perform any programming instruction that affects the Protect Register. Unlike the enable instructions described above, a PREN must immediately precede each programming instruction that involves the protect register. The Protect Register programming instructions are PRCLEAR, PRWRITE, and PRDS.

### 3.2.8 PROTECT REGISTER DISABLE

The protect register disable instruction permanently disables any further programming instructions to the protect register. Therefore it can only be performed once in the lifetime of a

NM93CS device. The purpose of it is to permanently configure a portion of the EEPROM as true ROM and a portion as Read/Write EEPROM. Great caution should be exercised prior to executing this instruction as there is no second chance. It is performed by sending a start-bit, opcode and an address field of all 0's while both the PRE and PE inputs are at a high level. This instruction must be immediately proceeded by a PREN instruction.

### 3.2.9 PROTECT REGISTER CLEAR

The protect register clear instruction will clear the contents of the Protect Register making the entire contents of the EEPROM alterable only if the PRDS instruction has not previously been executed. This is done by clocking in a start-bit, op-code, and address field of all ones. This instruction must be immediately proceeded by PREN instruction and requires that both PRE and PE inputs be held at a high level.

### 3.2.10 PROTECT REGISTER WRITE

The Protect Register write command (PRWRITE) allows the host to write the protect register with the address where the memory is to be segmented into ROM and EEPROM. The defined address is the first ROM address and the ROM field then continues to the top of memory. To execute this command a start-bit, op-code, and address must be clocked in, the address field containing the memory address that defines the ROM/EEPROM boundary. The PRE and PE inputs must be held at a high level.

## 3.3 INTERFACING SOLUTIONS

When interfacing serial microwire EEPROMs to microcontrollers there is an apparent conflict that occurs when selecting clock polarity and phase. This can be easily overcome in most situations, although when using some microcontrollers that do not allow selection of either clock polarity or clock phase, the only solution may be to resort to bit set and bit re-set instructions to interface to the EEPROM rather than use of the serial interface provided on the microcontroller.

In the instance where there is a dedicated serial interface provided, the conflict typically occurs as follows. Figure 10 demonstrates an EEPROM READ as this involves data being transferred from the micro to the EEPROM (Start bit, op-code, and address) and data transferred from the EEPROM to the micro (address contents). The conflict occurs in this example when the micros

clock sets data up on the falling edge of SK and expects the EEPROM to accept it on the rising edge, but then expects the EEPROM to do the same when it sends data back to the micro.

1. The micro sets up a data bit. A propagation delay after the falling edge the data bit is valid at the EEPROM DI pin.
2. The EEPROM uses the rising edge of SK to clock the data bit into its internal register.
3. When the data direction changes the EEPROM sets the data up starting at the rising edge of SK.
4. The micro attempts to clock the data bit in that was set up on clock edge 3.

This example will work if the micro requires 20 ns or less data hold time after edge 4. If greater than 20 ns is required, an alternate strategy is needed.

- 1a. The micro sets up the data bit on the rising edge and a propagation delay later it is valid at the EEPROM.
- 2a. The EEPROM clocks the data into its internal register. The EEPROM requires only 10 ns data hold time, which can normally be guaranteed.
- 3a. The EEPROM sets the Data-Out up on the rising edge.
- 4a. The micro clocks the data into its internal registers on the falling edge of the clock and a minimum data setup and hold time is guaranteed for the micro based on the minimum high and low time of the SK clock used in the application.

It should be noted that in the second example, CS (chip select) is asserted when SK is low. If this cannot be done, the DI input should be low when CS is asserted. If both DI and SK are high when CS is asserted the EEPROM will recognize this as a rising edge of SK. To accommodate this in a design, it is allowable to clock in any number of logic zeros prior to the start bit.

## 4.0 Conclusion

The serial EEPROM offered by Fairchild all share a common structure. Separating them are various features that give benefit to various applications such as the need for a bi-directional data bus or need for one byte word width. There are a number of "tricks" that may simplify interfacing to these which can easily be understood with the help of a functional block diagram. Given this information the overall job of using a serial interface EEPROM will be simpler.

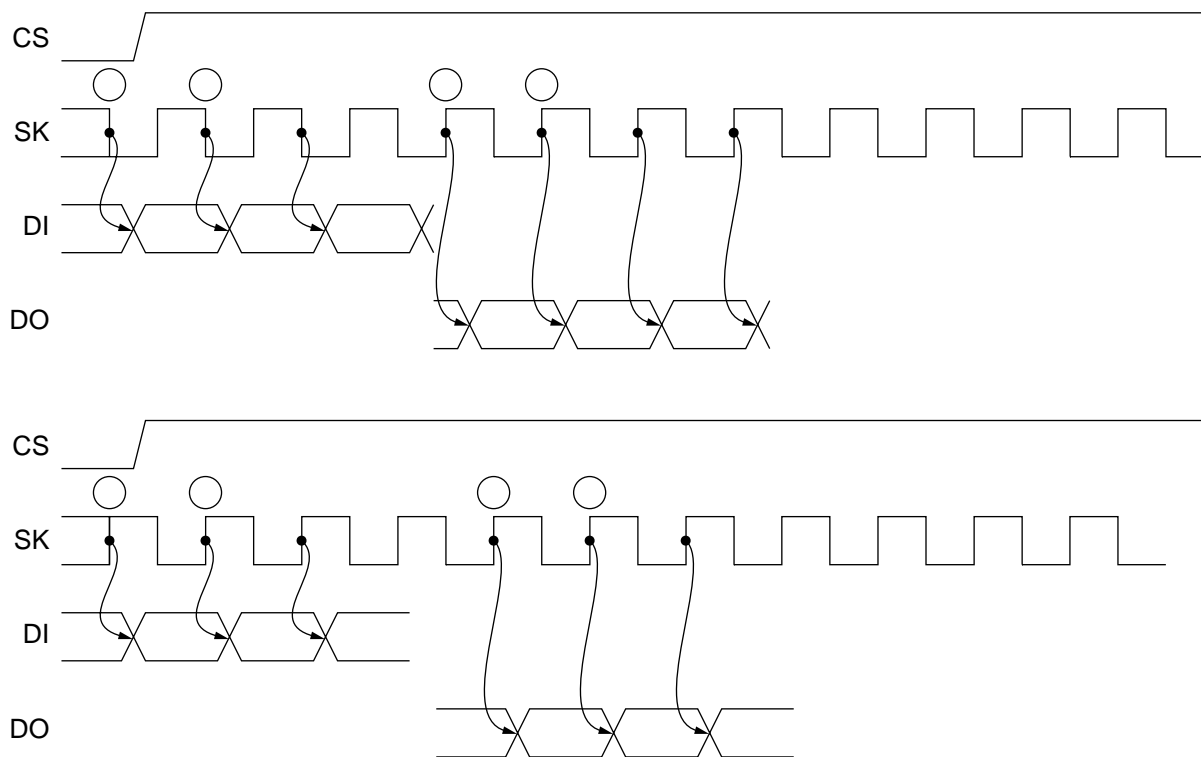


FIGURE 10.

## Life Support Policy

Fairchild's products are not authorized for use as critical components in life support devices or systems without the express written approval of the President of Fairchild Semiconductor Corporation. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

**Fairchild Semiconductor  
Americas  
Customer Response Center**  
Tel: 1-888-522-5372

**Fairchild Semiconductor  
Europe**  
Fax: +44 (0) 1793-856858  
Deutsch Tel: +49 (0) 8141-6102-0  
English Tel: +44 (0) 1793-856856  
Français Tel: +33 (0) 1-6930-3696  
Italiano Tel: +39 (0) 2-249111-1

**Fairchild Semiconductor  
Hong Kong**  
8/F, Room 808, Empire Centre  
68 Mody Road, Tsimshatsui East  
Kowloon, Hong Kong  
Tel: +852-2722-8338  
Fax: +852-2722-8383

**Fairchild Semiconductor  
Japan Ltd.**  
4F, Natsume Bldg.  
2-18-6, Yushima, Bunkyo-ku  
Tokyo, 113-0034 Japan  
Tel: 81-3-3818-8840  
Fax: 81-3-3818-8841