MX26C512 512K-Bit (64Kx8) CMOS Multiple-Time-Programmable ROM™

Application Note

R.T. Culver

May 1997

MACRONIX INTERNATIONAL CO., LTD

On-Board Programming Design Considerations for the MX26C512 (64Kx8) CMOS Multiple-Time-Programmable ROM

Application Note 05/19/97

Introduction

The MX26C512, **512K-bit MTP ROM™ (Multiple Time Programmable Read Only Memory)** is the first in a series of non-volatile memory devices to be offered by Macronix that are positioned between Flash and EPROM technologies. Although the MX26C512 is electrically erasable, the programming algorithm makes it appear to be more like an EPROM than flash because of its requirement to hold address and data lines static while externally generating program pulses and supplying high voltage (12.5-volts) during programming. The MX26C512 and the MX26C1000 (1M-bit MTP ROM™) are the only two devices in the MTP ROM™ family that are EPROM-like from an in-circuit programming viewpoint. All the newer members of the family will be flash-like for in-circuit programming - in which case you will not need to use this application note in the future. The flash-like parts will also program on an EPROM programmer like an EPROM or flash device. The next device in the series will be the MX26C514 that will internally latch the address and data lines during programming. High voltage and external timing control of the program pulses will still be required, but it will be much easier to implement in applications requiring *in-circuit programming* (ICP). The MX26C516 will be the third device in the planned series and will offer 5-volt only programming and operation.

This application note describes design considerations and guidelines for an **on-board programming** implementation using the MX26C512, 512K-bit MTP ROM™. The designer should recognized that there is a significant difference between *on-board programming* (OBP) and *in-circuit programming* (ICP). OBP requires an external programmer to be connected to the system to perform the actual programming; ICP can be done by the application hardware alone. Unlike EPROMs, that often need to be removed from the board and exposed to UV light in order to erase them, the **MTP ROM™** can be erased and programmed by the external programmer while still on-board. The MX26C512 memory device is not a good candidate for ICP programming, however, because of the expense and complexity of the additional on-board hardware required to perform this function. This application note will focus on OBP only. If ICP is required, please consider the MX26C514 memory device that has built-in features making it more suitable for ICP. OBP has been used successfully by system manufacturers for many years, but the design requirements to implement it may not be well known. The remainder of this application note will cover those design requirements.

On-Board Programming (OBP) Considerations with V_{CC} = 5.0 Volts

Since these memory devices will normally be installed in applications where V_{CC} is set to 5-volts only, a brief discussion regarding programming margins will be discussed first.

When using a commercially available programmer for programming, V_{CC} is usually raised to 6.25-volts for the program and verify algorithms. The programmed threshold voltage (Vt) of previously erased cells will be raised to a value greater than V_{CC} (6.25-volts), and will typically reach values from 6.5 to 9.0-volts thus providing 1.5 to 4.0-volts of operating margin after V_{CC} is lowered to its normal operating voltage of 5.0-volts. Unless a dual power bus is used to isolate the MTP ROM™ power supply from the rest of the circuits used on-board, it may not be possible to raise V_{CC} to 6.25-volts without harming other system components. For this reason, a programming algorithm with the V_{CC} supply set to 5.0-volts only is usually selected for OBP. With V_{CC} left at 5.0-volts, the programmed threshold voltage (Vt) may only be raised to 5.5-volts. This low threshold (5.5-volts) does not provide sufficient cell Vt margin for the memory to work reliably after programming. A modified algorithm is recommended and shown in Figure 1 that provides one additional programming pulse (over-programming pulse) to boost programmed cell Vt levels above the 6.5-volt range. This extra programming pulse will ensure that normal Vt margins will be achieved thus providing reliable in-circuit operation. A similar extra erase pulse is used during the erase algorithm as shown in Figure 2 to ensure that erased cells have their Vt levels reduced to a nominal value of 1.0 to 2.0-volts.

Hardware Considerations

If the designer is planning to use the MX26C512 to replace 27Cxxx series EPROMs or 28Cxxx series flash memory in an existing OBP design, there should be no hardware changes required. If they are going to be used in a new OBP design, the following considerations should be given to the hardware design:

The first design step is to provide the programmer a means of taking control of the CPU address bus, data bus, and control lines during programming without disrupting or harming the other system components. Many CPUs provide a means to directly tri-state these buses and control lines with an appropriate signal such as Reset. All that needs to be done is run a trace between this CPU signal and the external programmer connector so that the programmer may activate it and take control during programming. Some CPUs may not tri-state all lines in which case additional isolation buffers between the CPU and the memory must be used and controlled by the programmer. Special consideration must be given to the A9 address line because it must be raised to 12.5-volts when reading the identification codes and during the erase algorithm. Make sure that no other devices using A9 will be damaged by (or clamp) the 12.5-volts required for these operations. This usually requires the use of an isolation buffer between A9 and its source, the CPU. Figure 3 shows a complete design example for the designer reference.

The next design consideration is to provide a method for the external programmer to supply 12.5-volts to the memory OE/V_{PP} pin during programming without affecting the other system circuits or components. This task is somewhat complicated by the fact that V_{PP} and OE/ share the same device pin. This requires an isolation buffer between the OE/V_{PP} pin and any other circuits or memory devices connected to this line when it is raised to 12.5-volts by the external programmer. The programmer must be able to disconnect or disable this buffer while programming the device. Additionally, parallel 0.1uF ceramic and 4.7uF

electrolytic capacitors must shunt the OE/VPP pin to ground during programming only (see C2 and C3 in Figure 3). These capacitors should be located as close the OE/VPP pin as possible, but they must be disconnected during normal operation to prevent OE/ signal rise/fall time degradation. Figure 3 shows a very simple circuit using a single form ' relay, rather than a buffer, to implement this feature. The relay is activated when the external programmer takes control of the CPU buses and control lines.

The following five PC board layout and design rules should be followed to provide optimum performance and ensure on-board program/erase reliability:

Design Rule 1: Position a 0.1uF ceramic capacitor between the VCC pin and as close to system GND as possible. A similar capacitor must be connected to the OE/VPP pin and system GND through a switch or relay controlled by the external programmer. The path and lead lengths of these capacitors should be as short as possible to minimize their inductance.

Design Rule 2: Position one 4.7uF electrolytic capacitor between VCC and system GND for each set of eight memory devices. This bulk capacitor will maintain even voltage to the memory array. A similar capacitor must be connected to the OE/VPP pin and system GND through a switch or relay controlled by the external programmer. The path and lead lengths of these capacitors should be as short as possible to minimize their inductance.

Design Rule 3: Use a single ground plane to eliminate potential ground current loops.

Design Rule 4: Use short and wide traces for VCC and VPP power paths (0.08" minimum width preferred).

Design Rule 5: Keep the traces for Data lines (D7-D0) as wide as possible (0.012" minimum width) to reduce their impedance. This will reduce the harmful switching spikes that occur when driving heavily loaded data buses.

Software Considerations

The manufacturer of your external programmer will need to supply programming software compatible with your board design and the MX26C512. This will include signals to take control of the system buses and signal lines in an orderly fashion during programming and implementing the program/erase algorithms for the MX26C512. When the manufacturer of the external programmer writes the programming algorithms for the MX26C512 they must consider the timing affects of the additional capacitance installed on-board and connected to the OE/VPP pin. OE/VPP is pulsed for every program/verify cycle and the additional capacitance will slow down the rising and falling edges of these pulses.

Byte Programming

Refer to Figure 1 (Byte Programming Algorithm) and Figure 3 (Design Example) while following the 14 steps:

Program Step 1: (Take Control)

The first step in the programming algorithm is to take control of the microcontroller address and data busses. Assert ESET' to disable the 8051 microcontroller and tri-state its 0' and 2' port lines then assert EMWR' to tri-state buffers 1' and 2', and close relay 1'.

Program Step 2: (Read ID Codes)

The second step in the programming algorithm should be to read the manufacturer ID and device type to make sure the proper devices have been installed in the application (see Mode Select Table for the various command modes). Apply 12.5-volts to 9', force 0', GM/', and PP' low then read the manufacturer ID code (C2H) from the data lines 0-D7'. Next force 0' high and read the device ID code (D1H). Restore 9' to a normal V_{IL}/V_{IH} level and return GM/' and PP' to a V_{IH} level.

Program Step 3: (Setup for Programming)

Set the initial address and data.

Program Step 4: (Initialize Retry Count)

Set etries' to 20 (the variable etries' is used to count attempts remaining).

Program Step 5: (Set High Voltage)

Apply 12.5-volts to PP'.

Program Step 6: (Program Byte)

Apply data to 0-D7' and pulse GM/' low for 100us while holding all addresses constant then decrement etries'.

Program Step 7: (Return to Read Array Mode)

Force PP' low.

Program Step 8: (Verify Byte)

Read and verify the just programmed byte (force GM/' low). If it verifies, then one additional programming cycle should be executed (Program Steps 9, 10, and 11) to provide additional threshold voltage (Vt) margin for the programmed cells. This is necessary because programming is being done with V_{CC} set at 5.0-volts (see On-Board Programming (OBP) Considerations with V_{CC} = 5.0 Volts above). If the data fails to verify, repeat steps 5 through 8 up to twenty times. If all retries have been exhausted, the MX26C512 memory device has failed, and you must restore PP' to a V_{IL} or V_{IH} level and proceed to Program Step 14 to terminate the programming attempt.

Program Step 9: (Set High Voltage)
Apply 12.5-volts to PP'.
Program Step 10: (Over-Program Pulse)
The last byte programmed verified successfully. Now issue one more programming pulse (over-program pulse). Apply data to 0-D7' and pulse GM/' low for 100us while holding all addresses constant.

Program Step 11: (Return to Read Array Mode)
Force PP' low.

Program Step 12: (Next Address)
Was this the last address to be programmed? If not then increment the address and return to Program Step 4.

Program Step 13: (Verify All Bytes)
Force PP' and GM/' low and verify all programmed bytes.
If no errors, device passed else device failed.

Program Step 14: (Return Control)
Programming is complete. Restore PP' to a normal V_{IL} or V_{IH} level and force EMWR' and ESET' low

Erasing
Refer to Figure 2 (Erase Algorithm) while following the 9 steps listed below:

Erase Step 1: (Take Control)
The first step in the erase algorithm is to take control of the microcontroller address and data busses. Assert ESET' to disable the 8051 microcontroller and tri-state its 0' and 2' port lines then assert EMWR' to tri-state buffers 1' and 2', and close relay 1'.

Erase Step 2: (Read ID Codes)
The second step in the erase algorithm should be to read the manufacturer ID code and device type to make sure the proper devices have been installed in the application (see Mode Select Table for the various command modes). Apply 12.5-volts to 9', force 0' low, force PP' and GM/' low and then read the manufacturer ID code (C2H) from the data lines 0-D7'. Next force 0' high and read the device ID code (D1H). Restore 9' to a normal V_{IL}/V_{IH} level and return GM/' and PP' to a V_{IH} level.

Erase Step 3. (Program/Verify All 0)
Prior to erasing, all memory locations must be written to 00 hex first (see Byte Programming above). The erase function is a chip erase and all locations will be returned to FF hex when done.

Erase Step 4: (Initialize Retry Count)
Set etries' to 60 (the variable etries' is used to count erase attempts remaining).
Erase Step 5: (Setup Erase Mode)
Apply 12.5-volts to 9' and PP'.

Erase Step 6: (Erase Pulse)
Pulse GM/' low for 1 second; wait 500 milliseconds; then restore 9' to a normal V_{IL} or V_{IH} level and force PP' low in preparation for Erase Step 7 (Erase Verify).

Erase Step 7: (Erase Verify)
After 9' has been restored to normal operating levels and PP' has been forced low, read every memory location (force PGM/ low to read) and check for FF hex data. If any location fails to read FF hex data, repeat steps 5 through 7 up to sixty times. If erased successfully, continue with step 8 else the MX26C512 memory device failed and you must abort further erase attempts and proceed to Erase Step 9.

Erase Step 8. (One Additional Erase Cycle)
Repeat steps 5 and 6 to provide one additional erase cycle for threshold voltage (Vt) margin purposes, and the erase function is complete.

Erase 9 (Return Control)
At this point the MX26C1000 memory device contains no valid program instructions for the 8051 microcontroller. The user may now proceed to the byte programming algorithm or shut the system down to wait for future programming attempts.

Summary

With V_{PP} set to 5.0-volts, the MTP™ memory is protected against any accidental programming or erasure even if valid program or erase commands are executed. The programming algorithms are simple to implement in software or firmware. With the additional programming and erase cycles provided for by the recommended algorithms, cell threshold voltage (Vt) margins will be adequate even though the memory was programmed with V_{CC} set to only 5.0-volts. Hardware changes will be required to allow the external programmer to take control of the system busses and supply 12.5-volts supply to the OE/V_{PP} pin and A9 pin during erase/programming. These changes should be relatively easy to implement and they give the user the flexibility of loading or modifying data held by the MTP™ memory while still on-board.

Mode Select Table

MODE	PINS				
	CE/	OE/VPP	A0	A9	OUTPUTS
Read	VIL	VIL	X	X	DOUT
Output HiZ	VIL	VIH	X	X	High Z
Standby (TTL)	VIH	X	X	X	High Z
Standby (CMOS)	VCC	X	X	X	High Z
Program	VIL	VPP	X	X	DIN
Program Verify	VIL	VIL	X	X	DOUT
Erase	VIL	VPP	X	VH	High Z
Erase Verify	VIL	VIL	X	X	DOUT
Program Inhibit	VIH	X	X	X	High Z
Manufacturer ID	VIL	VIL	VIL	VH	C2H
Device ID Code	VIL	VIL	VIH	VH	D1H

Notes:

1. VH = 12.0 V +/- 0.5 V
2. X = Either VIL or VIH
3. A1 - A8 & A10 - A15 = VIL (during auto select)
4. VPP = 12.5 V (min) to 13.0 V (max)
5. VCC = 5.0 V

Figure 1. BYTE PROGRAMMING ALGORITHM

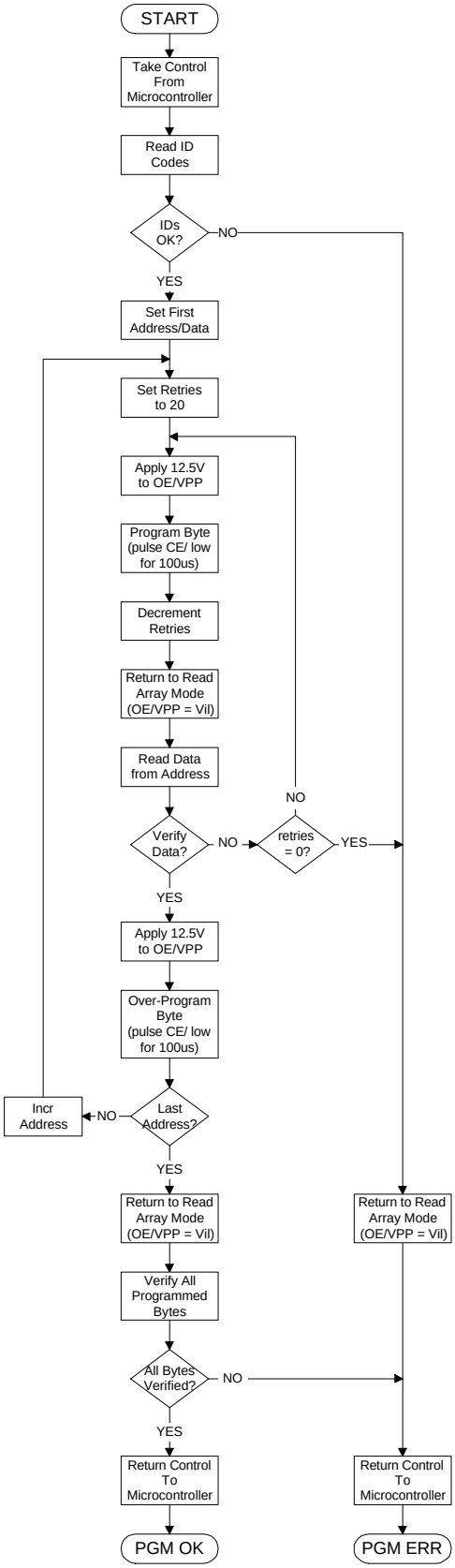

Figure 2. ERASE ALGORITHM

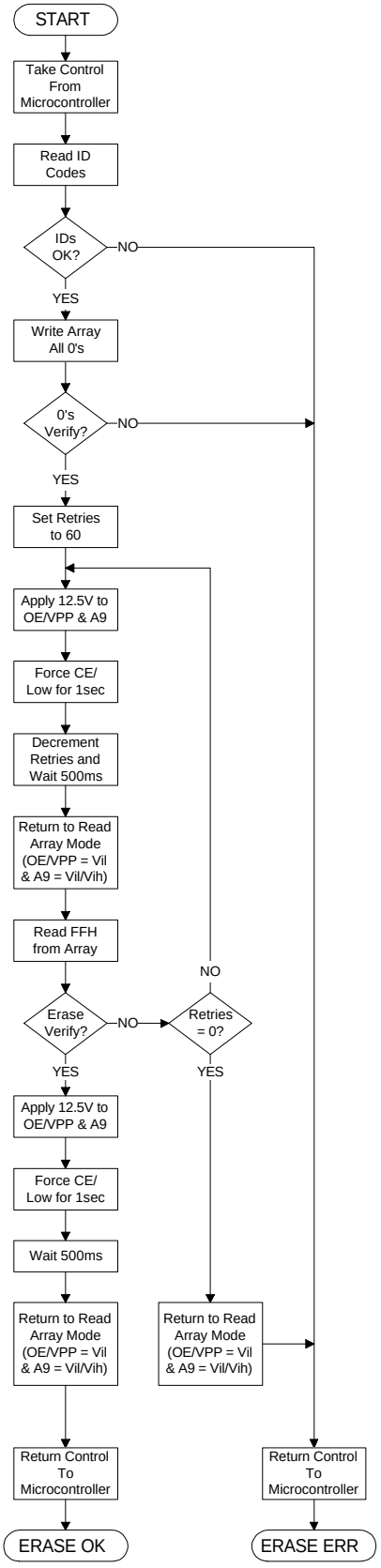

Figure 3. Design Example