MX26C1000A 1M-Bit (128Kx8) CMOS Multiple-Time-Programmable ROM™

Application Note

R.T. Culver

Macronix America, Inc.

May 1997

MACRONIX INTERNATIONAL CO., LTD

On-Board Programming of the MX26C1000A 1M-Bit (128Kx8) CMOS Multiple-Time-Programmable ROM

Application Note 05/19/97

Introduction

The MX26C1000A, **1M-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 MX26C1000A 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 MX26C512A (512K-bit MTP ROM™) and the MX26C1000A 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 MX26C1002 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 MX26C1004 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 MX26C1000A, **1M-bit MTP ROM™ (Multiple Time Programmable Read Only Memory)**. The designer should recognized that there is a significant difference between *on-board programming* (OBP) and *in-circuit programming* (ICP). OBP requires an external board 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 electrically erased and programmed by the external programmer while still on-board. The MX26C1000A 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 using the MX26C1002 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 this memory device 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 Vt 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 Vt may only be raised to 5.5-volts. This low threshold (5.5-volts) does not provide sufficient memory cell margin for the memory device 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 user is planning to use the MX26C1000A to replace 27Cxxx series EPROMs or 28Cxxx series flash memory in an existing OBP design, there should be no hardware changes required. If a new OBP design is being contemplated, 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 board 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 second consideration should be how to provide a method for the external programmer to supply 12.5-volts to the memory V_{PP} pin during program/erase operations without affecting the other system circuits or components. This can be accomplished rather easily by

connecting the V_{PP} pin to the V_{CC} supply via series resistor and blocking diode as shown in Figure 3. The V_{PP} pin itself can then be routed directly the external programmer connector so that the programmer can raise V_{PP} to the required programming voltage of 12.5-volts. The series blocking diode will isolate the programmer V_{PP} voltage from the system 5.0-volt V_{CC} supply line. Note that if the 5.0-volt V_{CC} supply can sink as well as source current to achieve regulation, the series blocking diode may be omitted.

The PGM/ pin needs to be active and controlled by the external programmer during programming. At all other times, it is not used and may remain high. Using a pull-up resistor to V_{CC} and then routing PGM/ directly the external programmer connector is the simplest design approach to satisfy this requirement.

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

Design Rule 1: Position 0.1uF ceramic capacitors, one each for V_{CC} and V_{PP} and as close to these pins and GND as possible.

Design Rule 2: Position one 4.7uF electrolytic capacitor between V_{CC} and GND as well as V_{PP} and GND for each set of eight memory devices. This bulk capacitor will maintain even voltage to the memory array

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

Design Rule 4: Use short and wide traces for V_{CC} and V_{PP} 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.

Design Rule 6: Never use flat ribbon cable between the external programmer and the system board; use a PC board type extender with edge connectors or design and build your own connector cable using 18-AWG (or larger) wire and keep it as short as possible. Flat ribbon cable has too much inductance and can cause unpredictable programming results.

Software Considerations

The manufacturer of your external programmer will need to supply programming software compatible with your board design and the MX26C1000A. 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 MX26C1000A.

Byte Programming

Refer to Figure 1 (Byte Programming Algorithm) and Figure 3 (Design Example) while following the 10 steps listed below:

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', 2', and 3'.

Program Step 2: (Read ID Codes)

The second step in the programming 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 A9', force 0' low, force EMRD/' 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 (D2H). Restore A9' to a normal V_{IL}/V_{IH} level and return EMRD/' to a high level.

Program Step 3: (Setup for Programming)

Apply 12.5-volts to the PP' pin and 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: (Program Byte)

Apply data to 0-D7' (must force EMRD/' high first) and pulse GM' low for 100us while holding all addresses constant then decrement etries'.

Program Step 6: (Verify Byte)

Read and verify the just programmed data byte (force EMRD/' low to read data). If it verifies, then one additional programming cycle should be executed (Program Step 7) 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 and 6 up to twenty times. If all retries have been exhausted, the MX26C1000A memory device has failed, and you must restore V_{PP} to 5-volts and proceed to Program Step 10 to terminate the programming attempt.

Program Step 7: (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 8: (Next Address)
Was this the last address to be programmed? If no then increment the address and return to Program Step 4.

Program Step 9: (Verify All Bytes)
Restore VPP to 5-volts, force EMRD/' low and verify all programmed bytes.
If no errors, device passed else device failed.

Program Step 10: (Return Control)
Programming is complete. Force EMRD/' high, EMWR' low, 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', 2', and 3'.

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 A9', force 0' low, force EMRD/' 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 (D2H). Restore 9' to a normal VIL/VIH level and return EMRD/' to a high 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 A9' and the PP' pin of the memory.

Erase Step 6: (Erase Pulse)
Pulse GM/' low for 1 second; wait 500 milliseconds; then restore VPP to 5.0-volts and A9' to a normal VIL/VIH level in preparation for Erase Step 7 (Erase Verify).

Erase Step 7: (Erase Verify)

After A9' and V_{PP} have been restored to normal operating levels, read every memory location 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 MX26C1000A 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 MX26C1000A 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 the additional programming and erase cycles provided for by the recommended algorithms, memory cell threshold voltage (Vt) margins will be adequate even though the memory device 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 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	CE/	OE/	PGM/	A0	A9	VPP	VCC	OUTPUTS
Read	VIL	VIL	X	X	X	VCC	VCC	DOUT
Output HiZ	VIL	VIH	X	X	X	VCC	VCC	High Z
Standby (TTL)	VIH	X	X	X	X	VCC	VCC	High Z
Standby (CMOS)	VCC	X	X	X	X	VCC	VCC	High Z
Program	VIL	VIH	VIL	X	X	VPP	VCC	DIN
Program Verify	VIL	VIL	VIH	X	X	VPP	VCC	DOUT
Erase	VIL	VIH	VIL	X	VPP	VPP	VCC	High Z
Erase Verify	VIL	VIL	VIH	X	X	VPP	VCC	DOUT
Program Inhibit	VIH	X	X	X	X	VPP	VCC	High Z
Manufacturer ID	VIL	VIL	X	VIL	VPP	VCC	VCC	C2H
Device ID Code	VIL	VIL	X	VIH	VPP	VCC	VCC	D2H

Notes:
1. VCC = 5.0 volts
2. VPP = 12.5 volts
3. X = Either VIL or VIH
4. A1 - A8 & A10 - A16 = VIL (during auto select)

Figure 1. BYTE PROGRAMMING ALGORITHM

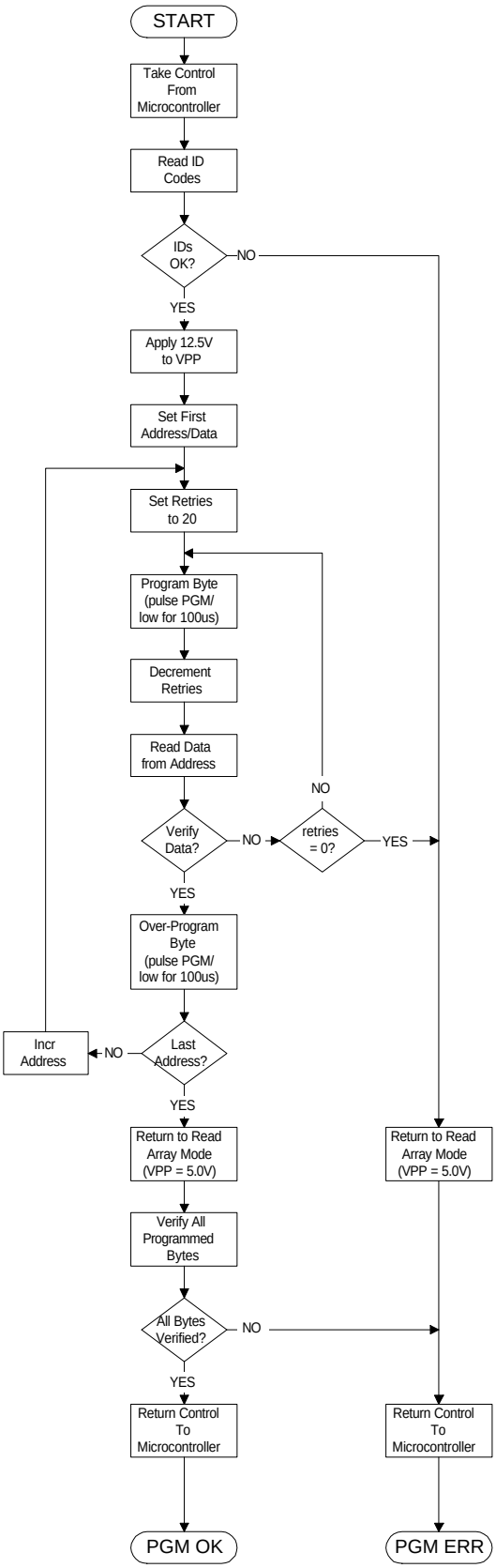


Figure 2. ERASE ALGORITHM

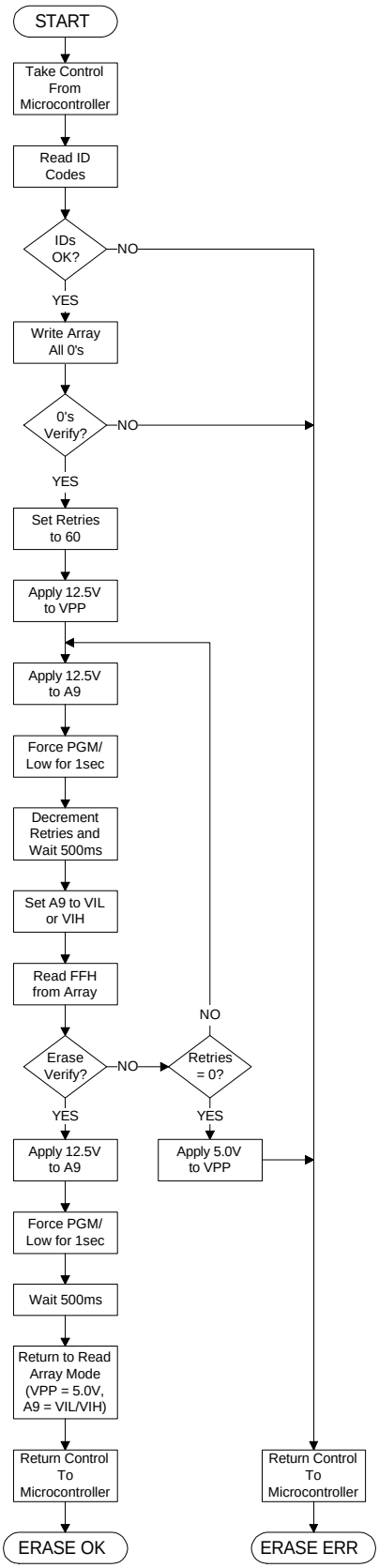
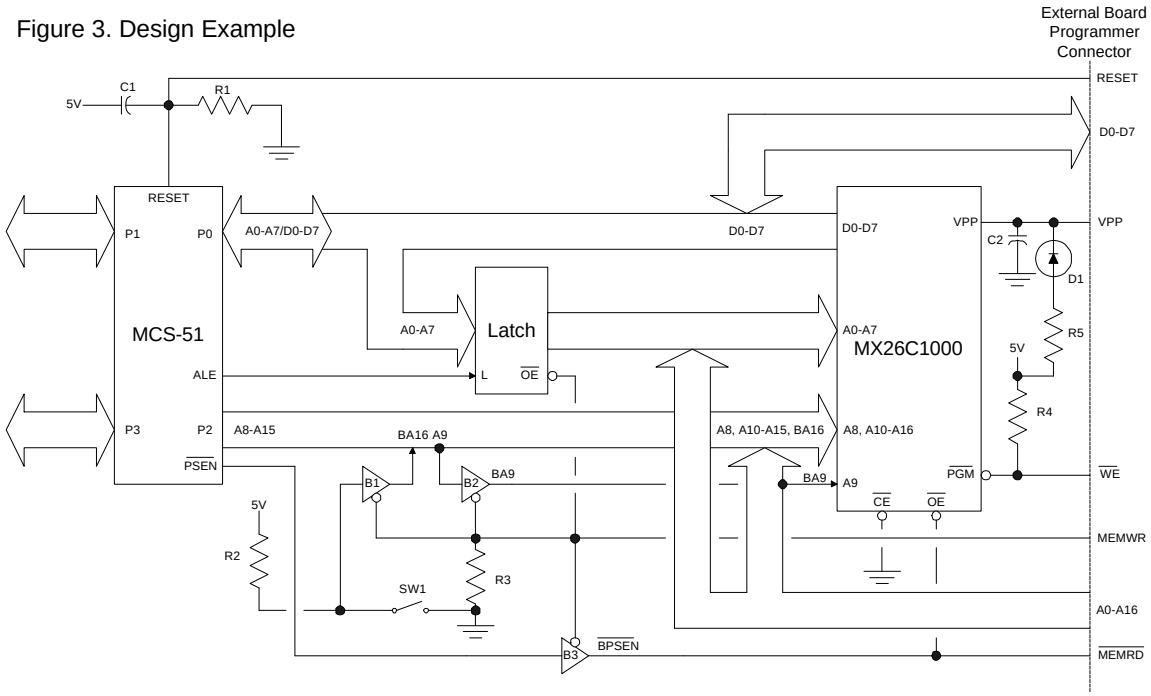


Figure 3. Design Example



This design example utilizes an Intel 8051 microcontroller. It can only address 64K of external program memory, but the MX26C1000A provides 128K of program memory. This application utilizes switch W1' to select between two applications stored in the MX26C1000A memory device by forcing 16' to be either high or low. The 8051 will then execute application code from either the high memory bank or low memory bank depending on the position of W1'. Buffers 1', 2', and 3' are used to isolate W1', 9', and SEN/' when the external programmer asserts EMRD/'.