TECHNICAL NOTE

TN246

Z180-Based Boards and Changes to the Serial EEPROM

The slower write speed of serial EEPROMs (manufacturer part # 24LC04B) used on Z180-based boards may cause problems if you use an external EPROM burner instead of directly compiling your application to the board. However, not all Z180 board customers who use an external EPROM burner will experience problems. The determining factor is if your BIOS bin file has a patch that compensates for the slower write speed of the 24LC04B EEPROM. Some Z180-based boards are unaffected; they are: the BL10xxx series, the BL11xx series, the BL12xx series and the BL13xx series.

There is no problem if you are using the BIOS bin file that came with your new board. The problem occurs when older bin files are used with newer boards. The symptoms of the problem are vague because the failed write will be silent and its result depends on the application.

BIOS .bin Files

If your application is not running correctly, and it ran fine prior to moving to 24LC04B, check to make sure you have the BIOS patch. In Table 1 there is a list of bin files and the addresses for the required patch. The patch is to change “0A” to “00” at the specified address. The changed value is a loop counter used by a DJNZ instruction, which decrements the value before looking at it. Using “00” means it will loop 256 times. After a .bin file is patched, it must be programmed into the development system’s EPROM or flash memory device. Then, the patched BIOS will be used when the application .bin file is created.

Table 1. BIOS bin Files Requiring a Patch

BIOS bin File	Patch Address	Board Type
2421.bin	0x1423	PK21xx
2608.bin	0x140e	PK22xx
2611.bin	0x069e	PK2240
2658.bin	0x084a	PK227x
2707.bin	0x13df	BL16xx
2900A.bin	0x14A8	CM71xx
2902.bin	0x086D	CM71xx
2903.bin	0x06a1	CM71xx
2951.bin	0x14e0	CM7110

CM71xx Series

The CM71xx series of boards is a special case because new boards are shipped without an EPROM. The BIOS .bin file burned onto the EPROM of a RoHS compliant CM71xx board must be patched for any application using that board to be able to write to its EEPROM.

The .bin file is typically created from an application using the "Compile | Compile to File" option or the "Compile | Compile to File with *.RTI File" option. The BIOS on the EPROM of the board from which the .bin file was created must be patched and typically is one of the BIOS files listed in Table 1. If the .bin file was created from an RTI file, the BIOS in use when the RTI file was created must be patched.

You can patch an RTI file yourself, but since the location varies depending on the BIOS version used when the RTI file was created, we recommend finding the patch address by searching with a binary editor such as the freeware XVI32.exe. The following location lists links to a number of available binary editors for free download:

`www.chmaas.handshake.de/`

The hex string to search for is “7A E6 07 5F 16 00 19 5E EB 16 0A”. The last character, 0A, should be changed to “00”. The file can then be saved and used in Dynamic C 32 with the “Compile to File with *.RTI File” option to create patched .bin files.

If the search hex string is not found, search with the last character changed to “00” to verify that the file is already patched. If the search hex string is still not found, please contact Rabbit Semiconductor and provide the .bin and/or .rti files which require patching.

BIOS Files on EPROMS

The patch is also required for the BIOS files programmed onto EPROMs that have the part numbers listed in Table 2. The patch is the same, namely changing “0A” to “00” at the specified address.

EPROMs listed in column 4 (“BIOS unpatched”) of Table 2 are never paired with 24LC04B EEPROMs on factory-shipped boards, so a BIOS patch is not needed. If you are using one of the EPROMs listed in column 5, the patch should have been made at the factory.

If you are experiencing problems that started after moving to the 24LC04B, check the EPROM content to make sure the BIOS is patched. The EPROM must be erased and reprogrammed using the patched BIOS.

Table 2. Programmed EPROMs and the BIOS Patch

Board Type	EPROM Type	BIOS File Part #	Part # of EPROM* (BIOS unpatched)	Part # of EPROM† (BIOS patched)	Patch Address
BL1600	UVEPROM	691-0167	693-0002	693-0026	0x13DF
BL16xx	UVEPROM	691-0055	693-0014	693-0028	0x13DF
BL16xx	Flash	691-0040	694-0019	694-0049	0x13DF
BL1600/10	Flash	691-0021	694-0008	694-0050	0x06AF
BL16xx	UVEPROM	691-0173	693-0022	693-0029	0x13DF
PK2100/10/20/30	UVEPROM	691-0051	693-0011	693-0030	0x1423
PK2100/10/20/30	UVEPROM	691-0060	693-0019	693-0031	0x1423
PK2100	UVEPROM	691-0178	693-0021	693-0032	0x1423
PK2100	Flash	691-0070	694-0027	694-0051	0x1423
PK2100/10/20/30	Flash	691-0076	694-0035	694-0052	0x1423
PK2210/30	UVEPROM	691-0052	693-0012	693-0024	0x140E
PK2200/20	UVEPROM	691-0059	693-0018	693-0025	0x140E
PK2200/20	Flash	691-0071	694-0030	694-0048	0x140E
PK2210/30	Flash	691-0078	694-0037	694-0047	0x140E
PK2200/20	Flash	691-0022	694-0009	694-0045	0x140E
PK2240	Flash	691-0069	694-0029	694-0046	0x069E

* These EPROMs are not paired with 24LC04B EEPROMs.

† These EPROMs are paired with 24LC04B EEPROMs.

WatchDog Timeout

The slower speed of 24LC04B can cause another problem to occur if a large amount of data is written. If the amount of time it takes to write the data is too long the watchdog times out. The solution is to add a call to `hitwd()` within the loop that is writing the data. This function hits the watchdog timer, postponing a hardware reset for 2 seconds. The slower speed of 24LC04B can only cause a watchdog timeout if there is no call to either `VdInit()` or `uplc_init()` or if interrupts are turned off.

Write-Protection Scheme

Several years ago, before the change to 24LC04B, Z180-based boards used serial EEPROMs that allowed the upper half to be write-protected while the lower half was write-enabled. Boards purchased several years ago used such EEPROMs. Lack of availability, however, forced a change to a new part. The new serial EEPROM had a write protection scheme that was all or nothing. It was no longer possible to write-enable the lower half while write-protecting the upper half. Serial EEPROMs with part # 24LC04B use this same all or nothing write protection scheme.

All Z180-based boards using serial EEPROMs with the all or nothing write protection scheme are shipped with the EEPROM write-enabled; this includes the boards listed on page 1 as unaffected by the slower write speed of 24LC04B (the BL10xx series, the BL11xx series, the BL12xx series and the BL13xx series). When the write-protect jumper is moved from its factory-shipped position (see your board's manual for exact jumper settings) it will write-protect the entire chip. Earlier revisions of board manuals describe the write-protect jumper as being able to write protect the upper half of the flash, leaving the lower half write-enabled. This is no longer true.

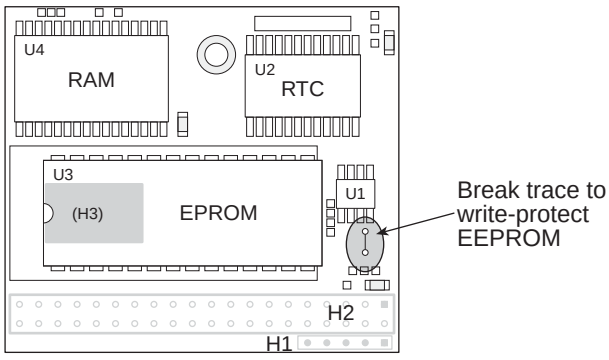
Table 3 gives the write-protect and write-enable jumper positions for the relevant boards.

Table 3. Jumper Positions

Board Family	Jumper Location	Protect	Enable
BL10xx	J16	1-2	2-3
BL11xx	J21	1-2	2-3
BL12xx	J11	1-2	2-3
BL13xx	J2	1-2	2-3
CM71xx	Trace	Cut	Uncut
BL16xx	J1	17-19	19-21
PK21xx	J3	1-2	2-3
PK22xx	JP5	1-2	2-3
SmartBlock	J2	1-2	2-3

For CM71xx boards, there is a trace that controls the write-protect status. See Figure 1 for the location of the trace.

Figure 1. Trace Location on CM71xx Boards



Summary

The information given in this document applies only to Z180-based boards. If your application is not running correctly and you suspect it has something to do with the serial EEPROM part # 24LC04B, first make sure the write-protect jumper is set correctly. If that checks out and you are using a BIOS bin file from Table 1 or a programmed EPROM from Table 2, make sure that the BIOS contains the appropriate patch.

If you are still experiencing problems and need some help, contact our technical support team online at:

www.rabbitsemiconductor.com/support/questionSubmit.shtml