Nintendo MMC1 info for 8-bit NES carts

Matthew J. Richey
mr6v@andrew.cmu.edu

INTRODUCTION:

This text provides additional information regarding Nintendo's MMC1 chip, beyond that which is contained in all other publicly available documents known to the author. All information was gathered from the examination of the MMC1 patent documents and from experimentation using Pascal Felber's IO-56 hardware. The information contained in this document appears to correctly describe the operation of several MMC1 cartridges; however, this document is merely a set of results gathered through experimentation, and therefore should not be considered to be 100% accurate.

It is known that the MMC1 chip has four registers, to which data can be written by sending values to certain memory locations in the NES CPU address space. Each register is 5 bits in width. Additionally, writing a byte value whose highest bit is a 1 will "reset" certain states of the MMC. Refer to Marat Fayzullin's NES tech document or to Y0SHi's NES tech document for more information.

This document will reference the four MMC1 registers as follows:

```
        Register 0 (reg0) - written to via $8000-$9FFF
        Register 1 (reg1) - written to via $A000-$BFFF
        Register 2 (reg2) - written to via $C000-$DFFF
        Register 3 (reg3) - written to via $E000-$FFFF
```

REGISTER 0 BITS:

It is known that reg1 and reg2 are commonly used to switch CHRROM pages and that reg3 is used to switch PRGROM pages. reg0 is used to switch between various MMC states. Each of the five lowest bits of this register control a specific state of operation of the MMC. Some of these states work in combination, and in some cases, a state will "override" another state. Some of these states are affected by the MMC "reset" signal, and others are not.

The following reg0 bits and their corresponding MMC states have been previously documented:

```
        bit 0 - toggles between "horizontal" and "vertical" mirroring
                0 = vertical, 1 = horizontal

        bit 4 - sets 8KB or 4KB CHRROM switching mode
                0 = 8KB CHRROM banks, 1 = 4KB CHRROM banks
```

The other reg0 bits control the following states:

bit 1 - toggles between H/V and "one-screen" mirroring
0 = one-screen mirroring, 1 = H/V mirroring

bit 2 - toggles between low PRGROM area switching and high PRGROM area switching
0 = high PRGROM switching, 1 = low PRGROM switching

bit 3 - toggles between 16KB and 32KB PRGROM bank switching
0 = 32KB PRGROM switching, 1 = 16KB PRGROM switching

There appear to be two bits in reg0 which define the screen mirroring. Bit 0 is used to toggle between two mirroring states, horizontal mirroring and vertical mirroring, as defined by Marat Fayzullin in his NES tech document. Bit 1 controls a third mirroring state. When a logic 0 is written to bit 1, the mirroring is similar to the one-screen mirroring that is used in mapper #7 carts such as the "Wizards and Warriors" series. This state occurs when the A10 cart line is held low (logic 0). The operation of this mirroring state is described further in Yoshi's tech document in the mapper #7 section. Note that the one-screen mirroring mode overrides the horizontal/vertical mode. In other words, when bit 1 of reg0 is set to 0, bit 0 is ignored by the MMC.

The PRGROM address space extends from $8000 to $FFFF and is often divided into two equal sections, the low PRGROM space ($8000-$BFFF) and the high PRGROM space ($C000-$FFFF). Bit 2 of reg0 determines which of these spaces (low or high) is switched via writes to reg3. When bit 2 is set to logic 0, the high PRGROM space is switched. Writes to reg3 determine the 16KB bank that is switched into the high PRG space, and the low PRG space is "hard-wired" to the first bank in the cart (bank 0). When bit 2 is set to logic 1, the low PRGROM space is switched. Writes to reg3 determine the 16KB bank that is switched into the low PRG space, and the high PRG space is "hard-wired" to the last (highest numbered) bank in the cart. Both states number the banks in the same way, i.e. bank 0 is always the first bank in the cart.

Bit 3 of reg0 is used to set a MMC state in which the entire 32KB of PRGROM space is switched (using reg3). Each of the 32KB banks consists of two adjacent 16KB banks. For example:

32KB bank number	16KB bank numbers
0	0,1
1	2,3
2	4,5
...	...

In 32KB switching mode, the 16KB bank which appears at the low PRGROM space is always the lower numbered bank. For example, if 32KB bank 1 is selected, 16KB bank 2 is switched into $8000 and 16KB bank 3 is switched into $C000. These combinations of 16KB banks are the only ones which are possible in 32KB mode, e.g. it is not possible to have bank 3 at $8000 and bank 4 at $C000. In this mode, even-numbered 16KB banks always appear at $8000 and odd-numbered banks at $C000. Another fact is that this state overrides the 16KB switching state; in other words, when bit 3 of reg0 is set to logic 0, the MMC ignores bit 2.

AN IMPORTANT NOTE: In this state (32KB switching), the MMC appears to ignore the lowest bit of reg3. In other words, it is necessary for whoever is doing the switching to shift the desired 32KB bank number one bit to the left before writing to reg3. Thus in the case of an emulator, in this MMC state, a value written to reg3 by the executing code should be shifted one bit to the right before using the value to index into a ROM dump.

No bits of reg0 other than bit 4 appear to affect CHRROM switching in any way. Further experimentation will be performed to confirm this.

RESET SIGNAL:

The "reset" signal is generated by writing a byte value whose high bit is a 1 to any of the four MMC1 registers. This signal affects the bits of reg0 as follows:

```
        bit 0 - unknown
        bit 1 - unknown
        bit 2 - reset to logic 1
        bit 3 - reset to logic 1
        bit 4 - unaffected
```

FURTHER SWITCHING INFORMATION:

It seems that, often, an MMC1 cartridge will write a value to reg0 only once, at the very beginning of code execution. However, it is possible that some cartridges may write to this register at other times during game play. In this case, it is useful to know exactly which banks will be located at the address space after a write to reg0 is completed, based on which banks occupied the space before the write. This section will discuss the topic for PRGROM switching, since CHRROM switching seems to always be either 8KB or 4KB, depending on the cart, and this does not change at any point throughout game play.

When transitioning between the low PRGROM switch and high PRGROM switch (via a toggle of reg0 bit 2), the MMC will shift the bank number that is in the switched space before the write into the switched space following the write. The non-switched space in each of these cases is "hard-wired" as described above. For example, the MMC is initially in the low PRGROM switching state with bank 5 located in its low PRGROM space. Logic 0 is then written to bit 2 of reg0. The resulting state will contain bank 5 in the high PRGROM space. Issuing the "reset" signal has the same effect as writing a logic 1 into reg0 bit 2.

The transition between 16KB and 32KB switching states is slightly more complicated. On a 16KB->32KB transition (writing logic 0 to reg0 bit 3), the

32KB bank to be switched into the address space is that which contains the 16KB bank that lies in either the low PRGROM or high PRGROM space BEFORE the transition, depending on the value of reg0 bit 2. Here is an example to illustrate. Say that reg0 bit 2 is currently set to 0 and reg0 bit 3 is currently set to 1, i.e. the MMC is in 16KB, high PRGROM switching mode. Bank 2 is currently being addressed at the high PRGROM space. A logic 0 is then written into reg0 bit 3, switching the MMC into 32KB mode regardless of the new value written to bit 2 (since bit 3 overrides). 16KB bank 2 is "paired" with 16KB bank 3 in 32KB mode (see chart above), and this combination corresponds to 32KB bank 1. Therefore, after the transition, 32KB bank 1 will be addressed at the PRGROM space.

The transition from 32KB->16KB switching mode is a little bit tricky. The rule is simple, though. In almost all cases, the MMC will switch into the PRGROM space, either high or low depending on the new value written into reg0 bit 2, the even 16KB bank number of the pair corresponding to the currently addressed 32KB bank (see chart above). For example, 32KB bank 1 is addressed before the transition. Logic 1 is written into both bits 2 and 3 of reg0, selecting 16KB low PRGROM switching mode. 16KB banks 2 and 3 correspond to 32KB bank 1, and therefore the even bank, bank 2, will be located at the low PRGROM space following the transition. There is one exception to this rule however. In a rare case, it is the odd-numbered bank that is switched into the newly selected 16KB space (high or low). This will happen only when an odd-numbered bank is located in a 16KB space immediately before a 16->32 transition and reg3 is not written to before the transition back to 16. This may be difficult to follow, but consider an example. In 16KB switching mode, say low PRGROM switching, bank 5 is located at the low PRGROM space. A transition is made to 32KB switching mode. Then the transition is made back to 16KB low PRG mode, either through the reset signal or via a write to reg0. If not a single value was written to reg3 during the time period in which the MMC was in 32KB switching mode, the odd-numbered bank (bank 5) will again appear at the low PRG space following the 32->16 transition. Writes to any other register (reg0-reg2) during this time period do not have the same effect, i.e. they can be written to any number of times and the odd bank will still be switched back, provided that reg3 is not touched. The bank to be switched into the 16KB space on this 32->16 transition, regardless of whether the newly selected space is high or low PRGROM, always follows these rules. In other words, even when going from 16KB low PRG -> 32KB -> 16KB high PRG, these same rules apply.

AN IMPORTANT NOTE: It may be seem logical, when attempting to determine which bank is currently located at the PRGROM space, to use the value most recently written to reg3. This is not a good idea, since it will not always give the correct bank number! Consider the following. The MMC is in 16KB switching mode, and bank 2 is currently at the switched location (either high or low PRG, it doesn't matter). But is bank 2 there because it was switched in with reg3? Not necessarily. It may have been switched into the space via a 32->16 transition, in which case most recent value that was written into reg3 is NOT the value $02 but in fact $01, since that is the corresponding 32KB bank. It is wise to be aware of this fact.

CREDITS:

Thanks to D (d@animal.blarg.net) for studying the MMC1 patent documents and providing the direction for my experiments. Without his help this document would have taken weeks to produce. Additional thanks to D and also to Sardu at bloodlust software (bldlust@southwind.net) for information on the one-screen mirroring and the operation of reg0 bit 1.

Thanks also to Marat Fayzullin and Y0SHi for their invaluable NES technical documents, and to Pascal Felber for his excellent hardware, the IO-56.

Thank you for reading; I hope that this information was helpful to you. Please feel free to send any comments, corrections, and/or additions to me via email at the above address.