9.5.2 Accessing Non-Volatile Memory

The nv memory, if implemented, can be accessed through the PCI interface or the Add-On interface. Accesses from both the PCI side and the Add-On side must be synchronous with the PCI clock (BPCLK for the Add-On). Accesses to the nv memory from the PCI interface are through the Bus Master Control/Status Register (MCSR) PCI Operation Register. Accesses to the nv memory from the Add-On interface are through the Add-On General Control/ Status Register (AGCSTS) Add-On Operation Register.

Accesses to the MCSR register are from the PCI bus and are, therefore, automatically synchronous to the PCI clock. Accesses to the AGCSTS register from the Add-On side must be synchronous with respect to BPCLK (as described in Section 9.1.4).

Some nv memories may contain Expansion ROM BIOS code for use by the host software. During initialization, the Expansion BIOS is located within system memory. The starting location of the nv memory is stored in the Expansion ROM Base Address Register in the S5933 PCI Configuration Registers. A PCI read from this region results in the S5933 performing four consecutive byte access to the nv memory device. Writes to the nv memory are not allowed by writing to this region. Writes to the nv memory must be performed as described below.

The S5933 contains two latches within the MCSR register to control and access the NVRAM. One is an 8 bit latch called the NVRAM Address/Data Register which is used to hold NVRAM address and data information. The other is a 3 bit latch called the NVRAM Access Control Register which is used to direct the address and data information and to control the NVRAM itself. Reading or writing to the NVRAM is performed through bits D31:29 of this register. These bits are enable and decode controls rather than a command or instruction to be executed. D31 of this register is the primary enable bit which allows all accesses to occur. When written to a '1', D31 enables the decode bits D30 and D29 to direct the data contained in the address/data latch, D23:16, to the low address, high address or data latches. D31 should be thought of as "opening a door" where as long as D31 = 1, then the door is open for address or data information to be altered. The table on page 5-16 of the S5933 data book shows the D31:29 bit combinations for reading, writing, and loading address/data information. Additionally, D31 doubles as an S5933 status bit. A '1' indicates that the S5933 is currently busy reading or writing to the NVRAM. A '0' indicates a complete or inactive state.

For the examples below, we will assume the S5933 is I/O mapped with a base address of FC00h. These examples will read one byte of the Vendor ID and write one byte to the Vendor ID.

This example will write 1 byte from NVRAM location 0040h and read it back:

In	**FC00h + 3Fh** (offset of NVRAM Access Control Register) until **D31 = 0** (not busy).
Out	**FC00h + 3FH** an **80h** (CMD to load the low address byte). This sets decode bits and opens door for low address latch.
Out	**FC00h + 3Eh** (offset of Address/Data Register) **40h** (the low byte of the address desired) 40h goes into latch but is not latched yet.
Out	**FC00h + 3Fh** an **A0h** (CMD to load the high address byte). This latches the low address through changing the decode bits and opens the door for the high address latch.
Out	**FC00h + 3Eh** a **00h** (the high byte of the address desired). 00h goes into the latch but is not latched yet.
Out	**FC00h + 3Fh** an **00h** (inactive CMD). This latches the high address through the disabling D31, 'closes the door'.
Out	**FC00h + 3Eh DATA** (the data byte to be written). DATA byte goes into the latch but is not latched yet.
Out	**FC00h + 3Fh** a **C0h** (CMD to write the data byte). This latches the data byte through changing the decode bits and begins to write NVRAM data operation.
In	**FC00h + 3Fh** until **D31 = 0** (not busy).
Out	**FC00h + 3Fh** an **E0h** (CMD to read the address latched).
In	**FC00h + 3Fh** until **D31 = 0** (not busy).
In	**FC00h + 3Eh** the **data.**

This example will read 1 byte from NVRAM location 0040h:

In **FC00h + 3Fh** (offset of NVRAM Access Control Register) until **D31 = 0** (not busy).

Out **FC00h + 3Fh** an **80h** (CMD to load the low address byte). This sets decode bits and opens door for low address latch.

Out **FC00h + 3Eh** (offset of Address/Data Register) **40h** (the low byte of the address desired) 40h goes into latch but is not latched yet.

Out **FC00h + 3Fh** an **A0h** (CMD to load the high address byte). This latches the low address through changing the decode bits and opens the door for the high address latch.

Out **FC00h + 3Eh** a **00h** (the high byte of the address desired) 00h goes into latch but is not latched yet.

Out **FC00h + 3Fh** an **E0h** (CMD to read NVRAM data). This latches the high address through changing the decode bits and begins to read the NVRAM data operation.

In **FC00h + 3Fh** until **D31 = 0** (not busy).

In **FC00h + 3Eh** the **data.**

This example will read 1 byte from NVRAM location 0041h and contains an extra step to demonstrate D31 operation:

In **FC00h + 3Fh** (offset of NVRAM Access Control Register) until **D31 = 0** (not busy).

Out **FC00h + 3Fh** an **80h** (CMD to load the low address byte). This sets decode bits and opens the door for low address latch.

Out **FC00h + 3Eh** (offset of Address/Data Register) **40h** (the low byte of the address desired) 40h goes into latch but is not latched yet.

Out **FC00h + 3Eh** (offset of Address/Data Register) **41h** (the low byte of the address desired) 41h goes into latch but is not latched yet.

Out **FC00h + 3Fh** an **A0h** (CMD to load the high address byte). This latches the low address through changing the decode bits and opens the door for the high address latch.

Out **FC00h + 3Eh 00h** (the high byte of the address desired) 00h goes into latch but is not latched yet.

Out **FC00h + 3Fh** an **E0h** (CMD to read the address latched). This latches the high address through changing the decode bits and begins the read NVRAM data operation.

In **FC00h + 3Fh** until **D31 = 0** (not busy).

In **FC00h + 3Eh** the **data.**

Notes:

1. Latched addresses do not automatically increment after a read or write. They must be loaded with new values.
2. Latched addresses remain after reads and writes. It is allowable to only update one address byte for the next access.
3. A processor may perform a one word write to load an address byte and control command simultaneously.