This memo describes the operation of the interrupt system for the Mulan VME card.

The Mulan VME card can generate an interrupt on any one of the seven possible VME interrupt levels, the particular level selected by a DIP switch on the card. Each of the four channels of A/D logic can generate this interrupt under three conditions; its phase-locked loop (DCM) being unlocked, its FIFO being full or its FIFO storage limit being exceeded. Three byte-wide parameters in USER_CSR space control the enabling/disabling/clearing of these interrupt conditions, CSRint1 for the FIFO limit condition, CSRint2 for the FIFO full condition and CSR3 for the unlocked condition. These parameters can be both written and read.

The bits in each of the parameters have the following significance:

1  CSRint1

| IL4 | IL3 | IL2 | IL1 | CL4 | CL3 | CL2 | CL1 |

A “1” for any of IL4 through IL1 indicates that the FIFO limit has been exceeded and that an interrupt has been generated. CL4 through CL1 are the respective enabling bits. An IL bit can only be set as a result of a true interrupt condition if its corresponding CL bit is also true.

2  CSRint2

| IF4 | IF3 | IF2 | IF1 | CF4 | CF3 | CF2 | CF1 |
A “1” for any of IF4 through IF1 indicates that the FIFO is full and that an interrupt has been generated. CF4 through CF1 are the respective enabling bits. An IF bit can only be set as a result of a true interrupt condition if its corresponding CL bit is also true.

3 CSRint3

| ID4 | ID3 | ID2 | ID1 | CD4 | CD3 | CD2 | CD1 |

A “1” for any of ID4 through ID1 indicates that the DCM is unlocked and that an interrupt has been generated. CD4 through CD1 are the respective enabling bits. An ID bit can only be set as a result of a true interrupt condition if its corresponding CD bit is also true.

4 Detailed Operation

The following discussion applies to all three of the CSRint registers.

The Mulan A/D VME card has a single interrupt latch that can signal an interrupt to the VME interrupt handler on any one of the seven interrupt levels, the particular level selected by a seven position DIP switch. This latch is set when any I bit transitions from a “0” to a “1”. The latch is reset when an interrupt acknowledge occurs at the selected interrupt level.

A write to a CSRint register can be used to set or clear any of the I and C bits, except that corresponding I and C bits cannot be both set to “1”. If an I bit and its corresponding C bit are both written high, then only the I bit will actually be set to a “1”, the corresponding C bit will be a “0”. This is because the setting of an interrupt bit (I bit) always disables the enabling bit that allowed it to be set, thereby preventing the occurrence of an endless stream of interrupts from a single interrupt condition.

I bits are cleared either by writing “0’s” to them or by reading the appropriate CSRint register. Reading a CSRint register always clears the I bits after reading them but has no effect upon the C bits.

An I bit may also be set when a true interrupt condition occurs while the corresponding C (enable) bit is high.
5 Typical Interrupt Sequence

The typical sequence for an interrupt starts by writing “1’s” to the desired C (enable) bits of the three CSRint registers. When one or more of the enabled interrupt conditions occurs, the appropriate I bits are set, the corresponding C bits are cleared and the interrupt latch is set. Setting of the interrupt latch causes an interrupt to be sent to the interrupt handler, which then responds by generating an interrupt acknowledge cycle, thereby clearing the interrupt latch. The interrupt handler then reads the three CSRint registers to determine the particular condition(s) responsible for the interrupt. After acting on the interrupt condition, the interrupt handler could, if desired, re-enable the same condition.

6 A Few Subtleties

Once an interrupt has occurred, it must be acknowledged before another interrupt can occur. During this “dead time” no IL, IF or ID bits can become set from a true interrupt condition, but these bits can be set by writing “1’s” to the I bits.

Reading a CSRint register clears any “set” interrupt bits in that register.

Once an interrupt has been acknowledged, another interrupt can occur, but only for an IL, IF or ID bit that is not already set. Reading a CSR register clears all of the corresponding IL, IF or ID bits but does not re-enable those bits because the corresponding CL, CF or CD bits were automatically cleared when the interrupt bits were set. A new setting of one or more of these enable bits is required.

If a CSRint register is written with one or more “1’s” for the I bits, the I bits are set, the corresponding C bits are cleared and an interrupt may occur even though the actual interrupt condition does not exist. This provides a mechanism for testing interrupts without the existence of an actual interrupt situation.