The Global Event System

John Winans

1.0 Overview

The support for the global event system has been designed to allow an application developer to control the APS event generator and receiver boards. This is done by the use of four new record types. These records are customized and are only supported by the device support modules for the APS event generator and receiver boards.

The use of the global event system and its associated records should not be confused with the vanilla EPICS events and the associated event records. They are very different.

2.0 The Event Generator

The Event Generator is used to generate global event codes and send them out to one or more Event Receivers. A group of interconnected event generators and receivers is referred to as an 'event circuit.' There may be more than one event generator on the same event circuit. And it is possible for a single IOC to be part of multiple event circuits.

2.1 EG Records

The EG record type is used to select the options of a specific event generator card. In order to properly configure it, you should first be familiar with its operating modes. This is specified in the document "Event System" by Frank Lenksus.

2.1.1 Field descriptions

OUT (Output Link) Specifies the link number of the event generator board. Only the 'Card' value of the link specification is used.

MOD1 (Mode Select for RAM 1) Used to select the operating mode of event sequence RAM1. It is important to know that the configuration of the events in the RAM may not be altered unless it is either 'Off' or in 'Alternate Mode'. Should a configuration attempt be made when the RAM is not in one of these modes, it will be deferred until the RAM mode is changed to either 'Off' or 'Alternate'. When MOD1 is set to alternate from any other mode, MOD2 will also be set to alternate. If MOD1 is changed from alternate to any other mode, MOD2 will be set to off.

MOD2 (Mode Select for RAM 2) The same as MOD1 except it represents the mode for sequence RAM2.

R1SP (RAM 1 Speed) Event clock 1 rate in Hz. This must be set to the clock rate of the signal source on the Event CLK 1 input. It is only used to calculate the 'desired position' value of events that are placed into the sequence RAM. These events are specified by the use of 'egevent' records.

If all 'egevent' record types that use the generator being configured will be using 'Clock Ticks' as their 'Delay Units' the value placed into R1SP is not used and may be left as zero.

R2SP (Ram 2 Speed) Same ar R1SP for the second sequence RAM.

FIFO (FIFO Enable) Used to enable or disable the input-fifo on the generator board. The fifo is used to allow more than one event generator to exist on the same fiber-optic line.

CLR1 (Clear RAM 1) Performing a 'put' operation on this field causes sequence RAM1 to be cleared. The use of this field is undefined (and will cause great problems) if there are any 'egevent' records configured in the database. This is provided for testing purposes.

CLR2 (Clear RAM 2) Same as CLR1, but for sequence RAM2.

TRG1 (Manual Trigger RAM 1) If a 'put' operation is performed on this field, a one-time trigger on sequence RAM1 will be initiated. The result would be that same as if there were a hardware trigger applied to the 'Event TRG1' input on the card.

TRG2 (Manual Trigger RAM 2) Same as TRG1, but for sequence RAM2.

ENAB (Master Enable) Master card enable. No events are generated unless the card is enabled. In general, there should never be any reason to disable an event generator. This is provided for testing purposes.

TAXI (Taxi Violation Flag) This is set to a non-zero value when there has been a taxi violation. It simply reflects that state of the violation signal on the taxi receiver module. Taxi violations can not occur when FIFO is set to 'Off.'

VME (Manual Event Generation via VME Access) Used to send out a one-shot event code. A put to this field will cause the event to be sent. It appears to be zero if it is ever read. This can be 'OUT-LINKed' to by an other record in order to generate an arbitrary event when it is processed.

ETEn (Event Trigger Enable) These are the enables for the trigger event inputs on the card. They must be set to 'On' in order to send the trigger event codes.

ETn (Event Trigger) These are used to program the event codes that correspond to the trigger event inputs on the card.

VAL Placed into the record to make EPICS happy and is completely worthless.

2.1.2 Record Processing

It is intended that EG records be set to passive processing only. They are not altered by the device support code in response to being processed. Their purpose is only to specify the operating modes of the event generator and, as such, are processed if and when a value in it is altered by a put operation. To start things going, however, they should have their 'process at init' flag set to YES.

As an observation, it is not advisable to alter the R1SP or R2SP fields. You may, but then all related egevent records must be processed again in order to recalculate their desired position values. The MOD1 and MOD2 fields can also cause some nasty side effects if they are altered between any two non-off modes. In general, the operating mode should be set to off and then to some other mode if it is desired to switch between two modes.

2.2 EGEVENT Records

The EGEVENT record is used in conjunction with an EG record in order to specify a single event that is to be placed into a sequence RAM. The event code and its time displacement from the trigger are specified in this record.

2.2.1 Field Descriptions

OUT	(Output Link) Used to specify what event generator link that this event is related to. Only the Card number is used
ENM	(Event Number) The event number that is to be placed into the sequence RAM.
RAM	(Sequence RAM Specifier) Which RAM the event is to be placed into. (Ignored when the generator is in ‘Alternate’ mode.)
DELY	(Desired Delay) The desired time delay between the trigger that starts the RAM sequence and when this event should be sent. This field must be expressed in the units selected in the UNIT field described below.
ADLY	(Actual Delay) This is a read-only field that is set to the actual delay value after accounting for rounding caused by the clock resolution as well as collisions that can occur if more than one event is placed into the same sequence RAM location.
DPOS	(Desired Position) This is a read-only field that represents desired position in the sequence RAM that the event should be placed. It is expressed in clock ticks.
APOS	(Actual Position) This is a read-only field that represents the actual position in the sequence RAM that the event is placed. It is expressed in clock ticks.
UNIT	(Delay Specifier Units) The time units used to express the delay value in the DELY and ADLY fields.
VAL	Placed into the record to make EPICS happy and is completely worthless.

2.2.2 Record Processing

It is intended that EGEVENT records be set to passive processing only. They are not altered by the device support code in response to being processed. Their purpose is only to specify the desired position and code of an event in a sequence RAM. The read-only fields will be updated as necessary when ever the sequence RAM is reloaded. To start things going, however, they should have their ‘process at init’ flag set to YES.

Sequence RAMs are reloaded when ever any of the EGEVENT records related to it has its DELY, ENM or UNIT values changed. It is not advisable to alter the UNIT field unless the associated sequence RAM mode is set to ‘Off’.

2.3 Event Generator Device Support

The device support module for the event generator may be used by EG and EGEVENT record types.

In order to configure the event generator device support, a call must be made to set the address for each of the event generator cards present in the IOC. This configuration call is as follows:

EgConfigure(<card number>, <Base address in A16>)

The <card number> field may be 0-4 and is used to specify which card is to be configured. This is the card number that is referenced in the EG and EGEVENT records when building the database. The <Base address in A16> field is a 16-bit number that represents the address of the card in the A16 memory space.

Database records that specify card numbers that are not configured will generate 'bad field' errors when they are initialized by iocInit. And will then be ignored by the event generator device support if ever processed.

3.0 The APS Event Receiver

3.1 ER Records

The ER record type is used to select the options of a specific event receiver card. In order to properly configure it, you should first be familiar with its operating modes. This is specified in the document "Event System" by Frank Lenksus.

3.1.1 Field Descriptions

OUT	(Output Link) Used to specify which event receiver board is represented by this record.
ENAB	(Master Enable) Master card enable. No events will be received if the card is disabled.
TAXI	(Taxi Violation Flag) Set to a non-zero value if there is a taxi violation on the event receiver board.
TRGn	(Trigger Enable) Trigger event enables. Setting these allows the corresponding bit to be set on the event receiver.
OTPn	(One Time Pulse Enable) Setting these allows the corresponding bit to be set in the event receiver.
OTLn	(Output Level Enable) Output level enables. Setting these allows the corresponding bit to be set in the event receiver.
DGnE	(Delay Generator Enable) Programmable pulse delay enables.

DGnD (Delay Generator Delay Value) Delay value used for the programmable pulse delay outputs. These values must be expressed in 10-mHz clock periods and has no other selectable resolution.

DGnW (Delay generator Width Value) Width of the programmable pulse. These values must be expressed in 10-mHz clock periods.

VAL Placed into the record to make EPICS happy and is completely worthless.

3.1.2 Record Processing

It is intended that ER records be set to passive processing only. They are not altered by the device support code in response to being processed. Their purpose is only to specify the operating modes of the event receiver and, as such, are processed if and when a value in it is altered by a put operation. To start things going, however, they should have their ‘process at init’ flag set to YES.

3.2 EREVENT Records

The EREVENT records are used to specify what bits are to be set in the event receiver mapping RAM. The use of these bits depend on the which outputs are enabled on the event receiver card (specified in the ER record.) Additionally, this record type is used to select the VME interrupt and time-latch option.

3.2.1 Field Descriptions

OUT (Output Link) Specifies the card containing the mapping RAM to be programmed.

ENAB (Event Enable) Enables the operation of this record. If this is set to ‘Disabled’, then the values in the record are ignored.

ENM (Event Number) The event number to be described. This indicates the position in the mapping RAM that is to be programmed.

OUTn (Output Enable) May be set to ‘Enable’ or ‘Disable’ in order to set or clear the corresponding bit in the mapping RAM. The meanings of these bits depend on what outputs are enabled on the event receiver board (selected in the related ER record) and are described in the document “Event System” by Frank Lenksus.

VME (VME Interrupt Enable) May be set to ‘Enable’ or ‘Disable’ in order to select a VME interrupt upon the occurrence of event ENM. The time is automatically latched when the VME interrupt is generated.

VAL Placed into the record to make EPICS happy and is completely worthless.

3.2.2 Record Processing

It is intended that EREVENT records be set to passive processing only. They are not altered by the device support code in response to being processed. Their purpose is only to specify the desired actions to be performed upon receipt of a specific event code. To start things going, however, they should have their ‘process at init’ flag set to YES.

3.3 EVENT Records

The regular EPICS event records may be used when it is desired to cause a record to process upon a given event code by the global event system. This is accomplished by configuring an EVENT record (the regular EPICS event record) and selecting the APS event receiver for the DTYP field. The card and signal are then used to select the event receiver card and event number, respectively. Any time the event number specified is received by the event receiver, that event record will be processed (pretty cool huh?)

This is the only relationship between the vanilla EPICS event codes (and their associated records) and the global APS event system.

3.4 Event Receiver Device Support

The device support for the event receiver may be used by ER, EREVENT and EVENT record types.

In order to configure the event receiver device support, a call must be made to set the address for each of the event receiver cards present in the IOC. This configuration call is as follows:

ErConfigure(<card>, <A16 board address>, <IRQ Vector>, <IRQ Level>)

Where <card> is the card to be configured, <A16 board address> is the 16-bit address of the board in A16 space, <IRQ Vector> is the vector number to use when generating VME interrupts, and <IRQ Level> is the VME backplane

4.0 Event System Observations

This section describes a few observations that would otherwise be left to experimentation for the user figure out. Much of the annoyances described here have ben left in the system because there are simple work arounds, or they represent situations that should never be encountered on a running system.

4.1 Event Generator Sequence RAM Modes

It is intended that the modes of the EG never be altered. It was considered that the MOD1 and MOD2 fields be made SPC_NOMOD fields. But, during the system testing of the event hardware itself, it became useful to be able to make adjustments to the operating mode. Thus the ability to change the mode was put into the record and device support. However, exactly what is done when the mode is changed is probably not useful to the database designer.

First of all, remember that the sequence RAMs can not be updated unless they are either in ALT mode or OFF. This is due to the hardware constraints. In order to alter a sequence RAM that is not set to ALT or OFF, the RAM must be changed to one of those modes, altered, and then reset back to the desired mode. (No it is not reasonable to do this automatically.) Should the mode be carelessly altered, the EG card will have the mode updated, but the sequence RAM(s) will not be updated again until the mode is set to ALT or OFF. (In an actual application program, it is not reasonable to think that the operating modes of the sequence RAMs will be changed.)

Unless you have a strong need to use more than one sequence RAM at the same time, it is strongly recommended that the ALT mode be used. This is so that you may alter the event positions on the fly when debugging.

4.2 EGevent Records

The delay selected in an EGevent record is done by specifying the desired period of delay. The period is converted to click ticks by the use of the R1SP and R2SP values specified in the EG record. Since only one event code may be in any single sequence RAM position at any given time, any collisions are resolved at RAM load time by scanning for the ‘next’ unused position. Thus it is possible that the same database end up loading the RAMs in two different images depending on the order that the records get processed (that earlier records get higher priority.) If this causes problems, it is recommended that the units of delay be specified in ‘clock ticks’ and that the same delay values not be used in multiple records.

The use of ‘clock ticks’ as the delay units specification will eliminate the rounding caused by the conversion from alternate units into clock ticks.

4.3 ER Interrupts

It should be obvious that the event system is capable of generating VME interrupts at a rate that far exceeds the CPUs ability to process them. Much care should be put into the design of the databases that control the event system so that this does not happen. It has been observed that when such a problem does occur, vxWorks dies and prints “Work queue overflow” on its console.

5.0 Example Database for Global Time Synchronization

This section describes a few example database records that are used to set up the event system. The records shown here are those used by the global synchronous timing system. And provide a good example of event generation from database records as well as from trigger inputs. It also includes a heartbeat generator that is required by the event system itself.

5.1 Timing System Overview

The example timing system uses a free running 1000 hertz clock. This clock is input on the TRG0 line of the EG card on the master timing IOC. Each time the TRG0 input is pulsed, an Increment Time Stamp (0x7C) event should be sent out so that the ER cards can update their notion of the time.

Additionally, the timing system has to take care of high order counter truncation and slave resynchronization. This is handled by the use of the Reset Time Stamp Counters (0x7D) event (the processing of the time stamp information is described in more detail in the document on the global timing system.)

5.1.1 Event Generator Database Records

The records related to the event generator card are used to initialize and generate events. The EG record used on the timing system master is:

```
record(eg, "$(prefix)_eg")
    {
        field(DESC, "")
        field(ASG, "")
        field(SCAN, "Passive")
        field(PINI, "YES")
        field(PHAS, "0")
        field(EVNT, "0")
        field(TSE, "0")
        field(TSEL, "0")
        field(DTYP, "APS event generator G")
        field(DISV, "1")
        field(SDIS, "0")
        field(DISS, "NO_ALARM")
        field(PRIO, "LOW")
        field(FLNK, "0")
        field(OUT, "#C0 S0 @")
        field(MOD1, "Off")
        field(R1SP, "0")
        field(MOD2, "Off")
        field(R2SP, "0")
        field(FIFO, "NO")
        field(ENAB, "YES")
        field(ETE0, "YES")
        field(ET0, "0x7c")
        field(ETE1, "NO")
        field(ET1, "0x0")
        field(ETE2, "NO")
        field(ET2, "0x0")
        field(ETE3, "NO")
        field(ET3, "0x0")
        field(ETE4, "NO")
        field(ET4, "0x0")
        field(ETE5, "NO")
        field(ET5, "0x0")
        field(ETE6, "NO")
        field(ET6, "0x0")
        field(ETE7, "NO")
        field(ET7, "0x0")
    }
```

The important items of note are that ENAB is YES, ETE0 is YES, ET0 is set to 0x7C, and that the record be set to process at init time. The scan should always be set to passive since it only makes sense to process the record when the field values change.

In order to take care of the heart beat (0x7A) and time stamp reset/resync (0x7D) events, longout records are used that have their output links pointed to the VME field on the above EG record. When they are processed, the VAL field of the longout record is sent out on the event system.

```
record(longout, "$(prefix)_hbeat")
    {
        field(DESC, "")
        field(ASG, "")
        field(SCAN, "1 second")
        field(PINI, "NO")
        field(PHAS, "0")
        field(EVNT, "0")
        field(TSE, "0")
        field(TSEL, "0")
        field(DTYP, "Soft Channel")
        field(DISV, "1")
        field(SDIS, "0")
        field(DISS, "NO_ALARM")
        field(PRIO, "LOW")
        field(FLNK, "0")
        field(OUT, "$(prefix)_eg.VME  PP NMS")
        field(DOL, "122")
        field(OMSL, "supervisory")
        field(EGU, "rocks")
        field(HOPR, "0")
        field(LOPR, "0")
        field(HIHI, "0")
        field(LOLO, "0")
        field(HIGH, "0")
        field(LOW, "0")
        field(HHSV, "NO_ALARM")
        field(LLSV, "NO_ALARM")
        field(HSV, "NO_ALARM")
        field(LSV, "NO_ALARM")
        field(HYST, "0")
        field(ADEL, "0")
        field(MDEL, "0")
        field(SIOL, "0")
        field(SIML, "0")
        field(SIMS, "NO_ALARM")
        field(IVOA, "Continue normally")
        field(IVOV, "0")
    }

record(longout, "$(prefix)_resync")
    {
        field(DESC, "")
        field(ASG, "")
        field(SCAN, "10 second")
```

```
        field(PINI, "NO")
        field(PHAS, "0")
        field(EVNT, "0")
        field(TSE, "0")
        field(TSEL, "0")
        field(DTYP, "Soft Channel")
        field(DISV, "1")
        field(SDIS, "0")
        field(DISS, "NO_ALARM")
        field(PRIO, "LOW")
        field(FLNK, "0")
        field(OUT, "$(prefix)_eg.VME  PP MS")
        field(DOL, "125")
        field(OMSL, "supervisory")
        field(EGU, "rocks")
        field(HOPR, "0")
        field(LOPR, "0")
        field(HIHI, "0")
        field(LOLO, "0")
        field(HIGH, "0")
        field(LOW, "0")
        field(HHSV, "NO_ALARM")
        field(LLSV, "NO_ALARM")
        field(HSV, "NO_ALARM")
        field(LSV, "NO_ALARM")
        field(HYST, "0")
        field(ADEL, "0")
        field(MDEL, "0")
        field(SIOL, "0")
        field(SIML, "0")
        field(SIMS, "NO_ALARM")
        field(IVOA, "Continue normally")
        field(IVOV, "0")
    }
```

There should be nothing interesting about the longout records described above. The only important thing is that they properly point to the VME field of the EG record.

5.1.2 Event Receiver Database Records

The records used in the event receiver database are used to initialize the event receiver card. The ER record used in the receiver database is:

```
record(er, "$(prefix)_ER")
    {
        field(DESC, "")
        field(ASG, "")
        field(SCAN, "Passive")
        field(PINI, "YES")
        field(PHAS, "0")
        field(EVNT, "0")
        field(TSE, "0")
        field(TSEL, "0")
        field(DTYP, "APS event receiver")
```

```
    field(DISV, "1")
    field(SDIS, "0")
    field(DISS, "NO_ALARM")
    field(PRIO, "LOW")
    field(FLNK, "0")
    field(OUT, "#C0 S0 @")
    field(ENAB, "YES")
    field(TRG0, "Disabled")
    field(TRG1, "Disabled")
    field(TRG2, "Disabled")
    field(TRG3, "Disabled")
    field(TRG4, "Disabled")
    field(TRG5, "Disabled")
    field(TRG6, "Disabled")
    field(OTP0, "Disabled")
    field(OTP1, "Disabled")
    field(OTP2, "Disabled")
    field(OTP3, "Disabled")
    field(OTP4, "Disabled")
    field(OTP5, "Disabled")
    field(OTP6, "Disabled")
    field(OTP7, "Disabled")
    field(OTP8, "Disabled")
    field(OTP9, "Disabled")
    field(OTPA, "Disabled")
    field(OTPB, "Disabled")
    field(OTPC, "Disabled")
    field(OTPD, "Disabled")
    field(OTL0, "Disabled")
    field(OTL1, "Disabled")
    field(OTL2, "Disabled")
    field(OTL3, "Disabled")
    field(OTL4, "Disabled")
    field(OTL5, "Disabled")
    field(OTL6, "Disabled")
    field(DG0E, "Disabled")
    field(DG0D, "0")
    field(DG0W, "0")
    field(DG1E, "Disabled")
    field(DG1D, "0")
    field(DG1W, "0")
    field(DG2E, "Disabled")
    field(DG2D, "0")
    field(DG2W, "0")
    field(DG3E, "Disabled")
    field(DG3D, "0")
    field(DG3W, "0")
}
```

Much like the EG record, the only interesting to note is that this record is passive and processed at init time.

Now, in order to cause an IRQ to occur when the reset/resync time stamp event is received, we use the following erevent record:

```
record(erevent, "$(prefix)_erevent7d")
    {
        field(DESC, "")
        field(ASG, "")
        field(SCAN, "Passive")
        field(PINI, "YES")
        field(PHAS, "0")
        field(EVNT, "0")
        field(TSE, "0")
        field(TSEL, "0")
        field(DTYP, "APS event receiver")
        field(DISV, "1")
        field(SDIS, "0")
        field(DISS, "NO_ALARM")
        field(PRIO, "LOW")
        field(FLNK, "0")
        field(OUT, "#C0 S0 @")
        field(ENAB, "Enabled")
        field(ENM, "0x7d")
        field(OUT0, "Disabled")
        field(OUT1, "Disabled")
        field(OUT2, "Disabled")
        field(OUT3, "Disabled")
        field(OUT4, "Disabled")
        field(OUT5, "Disabled")
        field(OUT6, "Disabled")
        field(OUT7, "Disabled")
        field(OUT8, "Disabled")
        field(OUT9, "Disabled")
        field(OUTA, "Disabled")
        field(OUTB, "Disabled")
        field(OUTC, "Disabled")
        field(OUTD, "Disabled")
        field(VME, "Enabled")
    }
```

Interesting points here are that the output link field points to the same ER card as the above er record. The event number specified in the ENM field is the reset/resync time stamp event, and we can see that the VME field is set to ENABLED. This does nothing more than to tell the ER card that we want an IRQ on event number 0x7D. Note that we could also have turned on any of the output pulse/level outputs as well.

We need not include a record to enable anything on the increment time stamp or heart beat events as they are handled by the ER card automatically.

Exactly what happens when the IRQ arrives for event 0x7D is described in detail in the global timing documentation. Suffice it to say that the timing system registers a callback with the event receiver driver that gets called upon receipt of the event.

Should you desire to process a database record upon the receipt of an event (in this case event number 0x7D) you may use a regular epics event record and set it up like this:

```
record(event, "$(prefix)_event")
```

```
{
    field(DESC, "")
    field(ASG, "")
    field(SCAN, "Passive")
    field(PINI, "NO")
    field(PHAS, "0")
    field(EVNT, "0")
    field(TSE, "0")
    field(TSEL, "0")
    field(DTYP, "APS event receiver")
    field(DISV, "1")
    field(SDIS, "0")
    field(DISS, "NO_ALARM")
    field(PRIO, "LOW")
    field(FLNK, "$(prefix)_calc1.PROC PP MS")
    field(INP, "#C0 S125 @")
    field(SIOL, "0")
    field(SIML, "0")
    field(SIMS, "NO_ALARM")
}
```

Interesting tidbits here are that the record's INP link is set to the ER card, the signal number is set to the event number of interest, and that the forward link field be set to the record you wish to process upon receipt of the event code. Remember also that the VME interrupt must be enabled for the desired event code (in this case, 125 (0x7D)) by the use of an EREVENT record type for the same event number, that has the VME field set to ENABLED.

6.0 Event System Observations

This section describes those items that might otherwise be overlooked by the overwhelming detail of the record support fields. Here we provide a simple overview of the ways events can be generated by the EG card and what can be done with them by the ER card.

6.1 Event Generator Input Sources

6.1.1 50 ohm Trigger Inputs

The Event Generator hardware can generate event codes from 50 ohm input sources. The event codes generated are configured in the ET0-ETn fields and enabled by the ETE0-ETEn fields in the event generator record. The trigger inputs are edge sensitive and generate the event code placed in the corresponding ETn field of the EG record.

6.1.2 Software (EG records)

It is possible to generate any event code at any time by writing it to the VME field of the EG record. The VME field has a 'write-only' kind-of operation. Reading it will always return the value zero and not cause any events to be transmitted.

6.1.3 Sequence RAMs

The sequence RAMs are programmed by the use of the EGEVENT records. Each record represents a single event code that is placed into a sequence RAM. The record describes the event code number and its position in the RAM (in terms of time offset from trigger.) If the RAM is enabled in the EG record, and a trigger is present (either 50 ohm input or by writing a value to the EG records TRGn field) the sequence RAM will be cycled thru and the present events will be sent out.

6.2 Event Receiver Outputs

The event receiver has many output configurations available. This section provides an overview of each one.

6.2.1 One Time Output Pulse

Any given event can generate a one-time one microsecond output pule if the output pulse enable is set for the desired signal in the ER record and the related trigger bits are set in the mapping RAM via an EREVENT record.

6.2.2 Programmable Delay and Width Pulse

Any given event that is received can cause a pulse to be generated after a specified delay, and last for a specified width. The delay and width values are specified for the desired signal by the ER record's DGnD and DGnW fields. It has to be enabled by the use of the DGnE field as well and the corresponding bits have to be set in the mapping RAM by the use of EREVENT records.

6.2.3 Level Outputs

Any pair of events may be used to toggle an output signal by enabling it in the ER record and by setting the corresponding bits in the mapping RAM by use of a EREVENT records.

6.2.4 Special One Time Output Pulse

These outputs are designed such that if the event code has its high order bit set, the seven low order bits are presented on these output lines as 1 microsecond pulses. The idea here is that you can have up to seven pulses generated simultaneously. This mode is NOT configurable, but can be enabled on a per-bit basis.

6.2.5 VME Interrupt and Record Processing

The ER board is capable of generating a VME interrupt upon receipt of an event code. This is enabled via an EREVENT record that has the VME field enabled. When this is done, an regular EPICS EVENT record can be processed when the IRQs are received. The

EVENT record to be processed has to have its scan rate set to “I/O Intr” mode. The card and signal fields in the EVENT record’s link are used to specify the ER card number and the event number that is to cause the record to be processed.