EDOC222
CDOC222

EUROGAM DATA ACQUISITION SYSTEM
CHARISSA DATA ACQUISITION SYSTEM

Register Server implementation for VME access

Edition 1.0

Oct 1995

Nuclear Physics Support Group

Central Laboratory of the Research Councils
Daresbury Laboratory

VME Register Server

The **VME** register server is an implementation of the generalised **Eurogam** register server concept which contains register classes specific to the requirements of access to **VME** address spaces. For details of the general **Eurogam Register Server** the **Eurogam** documents **EDOC024**, **EDOC034**, **EDOC087** and **EDOC088** should be consulted.

Definitions

format of numbers

decimal	1234
hexadecimal	0x1234 or @1234
binary	%1011

All register values are text strings unless otherwise specified.

data word bit identification

Data word bit identification uses bit 15 (for 16 bit data words) and bit 31 (for 32 bit data words) as the most significant bit and bit 0 as the least significant bit.

For a definition of the allowed forms of **register names** see **EDOC210** (**CAMAC** register server).

Initial register class

The **VME** register server contains on startup an inbuilt set of registers. These inbuilt registers are intended primarily for test and debugging and applications using the **VME** register server should normally define specific registers for the modules being used.

register **VME.Address** access permitted - read and write

defines the VME hardware address (**a**); the VME address modifier (**m**); and the data width (**d**).

write register defines values for **a,m** & **d**

format parameter1 [parameter2 [parameter3 ....]]

example -a 0x28000000 -m 0x09 -d 32

selected parameters may be defined or redefined

example -m 0x0d

read register returns the current values of the parameters as a text string in the same format as a complete write register

Note - this register will aways return the values of **a**,**m**,& **d** used by the last **VME** access and is thus useful as a diagnostic tool for debugging actions of defined specfic registers.

initialise register set parameters to default values (A32 D32 non privilged data access).

(-a 0 -m 0x09 -d 32)

register **VME.Execute** access permitted - read and write

executes a VME cycle using parameters currently defined via the **VME.address** register

write register the data supplied is taken and a VME (write) cycle executed

read register a VME (read) cycle is executed and the data obtained is returned

register **VME.Status** access permitted - read only

Status (**s**) = 0 if last VME access was successful and status = 1 if last VME access caused a VME bus error.

read register returned data %**s** ie %1

register **VME.Data** access permitted - read only

holds data used by the last **VME** operation - either supplied (write) or returned (read)

read register returned data 0xabcd (D16) or 0xabcdefgh (D32)

Note - this register will always return the data used by the last **VME** access.

register **VME.Debug** access permitted - read and write

controls the server debug level

write register	set the server debug level
read register	return the server debug level
initialise register	set the server debug level to 0 (ie off)

Debugging messages are controlled by a set of bit flags which can be Ored together to create a debug “level”. The following flags have been defined. If a flag bit is set then debug messages from the specified procedures will be produced.

0x01	interface procedures
0x02	intrinsic register procedures
0x04	xVME class procedures

Configuration register class

When the register server task starts it creates a single instance of the configuration register class and initialises it using the name of the configuration file given when spawing the register server. The records of the configuration file have exactly the same form as the command line interface to the register server (see EDOC088) except that the -f and -c parameters are omitted and the data format is assumed to be a (ascii characters).

```
ie.   ersdefine   register-name   class-name
      erswrite    register-name   data

eg.   ersdefine   hsm.control  xVME
      erswta      hsm.control  -a 0x81000000  -m 0x9  -w 32
      erswrite    hsm.control  0xc0000000
```

These three lines would define a register hsm.control and write the value 0xc0000000 to it using the **VME** access mode A32, D32 to the module at **VME** address 0x81000000.

xVME register class

The xVME register class is used for all single shot **VME** accesses. The register value is a data item of length (in bits) given by the -l attribute which occupies a field in the **VME** data word with least significant bit given by the -b attribute. Whole words can be accessed by setting the length and bit attributes to zero. The class accepts data values in decimal, hexadecimal or binary format.

Register values for the xVME class are text strings.

register attributes

-a	VME base address
-o	VME address offset
-w	VME address width (16,24,32 bits)
-d	width of VME data access (16,32 bits)
-u	use - data (d); program (p)
-m	mode- non-privileged (n); supervisory (s)
-m	VME address modifier
-p	access capability to VME (rw, ro, wo)
-l	length of bit field
-b	bit position
-i	initial value
-z	data format (d (decimal), x (hexadecimal), b (binary))
-s	if = 1 returns access status as a second data item for read-register requests in the form %s

-m if a numeric value is supplied this is taken to be a VME address modifier and -u and -w parameters are ignored. Otherwise -m, -u and -w parameters are taken to define the VME address modifier.

default attributes : -a 0x0 -o 0x0 -w 32 -d 32 -u d -m n -p rw -l 0 -b 0
-z x -s 0

default address modifier is thus 0x09 (A32 data access non-privileged)

The default attribute value is used for any attribute not defined by a write-attributes command.

data format - data for write operations may be supplied in any format but the data format attribute will determine the format used when returning data from read operations.

The initialise-register operation is permitted only for registers having access attribute rw or wo and will only write data to **VME** if an initial value has been explicitly defined with the -i attribute.

bVME register class

The bVME register class is used for block mode **VME** accesses. The register value is a sequence of data items. The first data item corresponds to the VME address obtained by adding the -a and -o parameters. Subsequent data items are obtained by incrementing the VME address by 2 or 4 bytes as determined by the -d parameter. The class accepts data values in decimal, hexadecimal or binary format.

Register values for the bVME class are text strings.

register attributes

-a	VME base address
-o	VME address offset
-w	VME address width (16,24,32 bits)
-d	width of VME data access (16,32 bits)
-u	use - data (d); program (p)
-m	mode- non-privileged (n); supervisory (s)
-m	VME address modifier
-p	access capability to VME (rw, ro, wo)
-l	number of data items
-z	data format (d (decimal), x (hexadecimal), b (binary))

-m if a numeric value is supplied this is taken to be a VME address modifier and -u and -w parameters are ignored. Otherwise -m, -u and -w parameters are taken to define the VME address modifier.

default attributes : -a 0x0 -o 0x0 -w 32 -d 32 -u d -m n -p rw -l 1 -z x

default address modifier is thus 0x09 (A32 data access non-privileged)

The default attribute value is used for any attribute not defined by a write-attributes command.

data format - data for write operations may be supplied in any format but the data format attribute will determine the format used when returning data from read operations.