SMXR Diagnostic Utilities User's Guide

J.Schlereth

1.0 Overview

The SMXR Diagnostic Utilities are a set of programs which test the functions of the SMXR module. The tests are organized into categories which separate as much as possible the functions of the SMXR module. These categories are the Register test, ID PROM test, Memory test, Flash Memory test, Diagnostic modes test, Serial Trigger Sum test and SMD Data test. For each category, there is an interactive exerciser utility as well as a certification tester. The certification test is generally run as background task for a fixed number of iterations in order to identify problems within the module. The interactive exerciser may then be used to further study the problem areas.

2.0 Exerciser Utilities

This section gives a brief description of the exerciser utilities. Each of these utilities may be run directly from the operating system shell or from the menu driven ***smxr*** utility. The menu driven utility is invoked from the shell with the following syntax

smxr -g<slot>

where slot is the geographical address of the module being tested. A simple menu, shown below, will appear from which the exerciser utilities may be selected.

```
1-SMXR Registers
2-SMXR IDPROM
3-SMXR Memory
4-SMXR FlashMeM
5-SMXR DiagMem
6-SMXR Serial Cap
7-SMXR Data Cap

Q - Quit
```

.

2.1 Register test - smxre1

This utility tests the four sets of registers on the SMXR module. It can be run directly from the shell with the command:

smxre1 -g<slot>

where slot is the crate slot number. It can also be run from the ***smxr*** menu, item 1

```
1-SMXR Registers
```

The program tests each of the four sets of registers selected from the menu shown below:

```
1-Status
2-Flash memory
3-Prog. Delay
4-Trigger Sums

Q - Quit
```

Item 1, Status, directs the program to read the control/status register and display the result both as a 32-bit hex number and as a bit field descriptor. The values of the read/write bits can be changed by typing the new hex value followed by return <CR>.

Item 2, Flash memory, directs the program to test the registers which are used to erase, download and upload the flash memory. the 32 bit register is divided into eight 4-bit nibbles with each nibble controlling one of the SMXR channels. The value of each nibble may be one of the following:

4- to erase, 2 - to donwload, 1 - to upload

Each bit should clear after a few seconds as the operation completes. The register is polled by the program and the updated value displayed once per second. Type <CTRL-C> to terminate the polling.

Item 3, Prog delay, tests the programmable delay registers. The registers are displayed as both 32bit values and as bit filed descriptors. They may be changed by entering the new hex value followed by <CR>. The test only verifies the read/write capability of the registers and does not check if the correct delays are set.

Item 4, Trigger Sums, tests the trigger sum registers. There are eight 32-bit registers which correspond to 16bit low and high thresholds for each of the SMXR channels. The values may be changed by typing the new value followed by <CR>.

2.2 ID PROM test - smxre2

This utility simply reads the ID PROM and displays the contents.It may be run directly from the shell with the command

smxre2 -g<slot>

where slot is the slot number of the board. It may also be run from the ***smxr*** menu item 2

```
2-SMXR IDPROM
```

There are no options associated with this program. The ID PROM contents are just read and displayed for a few seconds.

2.3 Memory test - smxre3

This utility tests the lookup table and diagnostic memory by writing and reading the RAM from the VMEbus. It may run directly from the shell with the command

smxre3 -g<slot>

It may also be run from the ***smxr*** menu item 3

```
3-SMXR memory
```

The top level menu of this utility offers the selections shown below:

```
1-Memory            00400000
2-Data Pattern      Random
3-BLT Write
4-BLT Read
5-LW Write
6-LW Read
7-Clear
8-BLT Loop

Q - Quit
```

Item 1 is used to select the memory section to test. The lookup table is divided into four sections as viewed from VME address space and is organized as pairs of 16-bit words with each word containing the data for an SMXR channel. The diagnostic memory can also be tested with this utility and appears as another address range from the VMEbus. The address offset of the currently selected memory section is shown on the right hand column next to item 1.

```
0-1 Lookup
2-3 Lookup
4-5 Lookup
6-7 Lookup
Diagnostic
```

Selecting item 1 brings up a second menu which contain the choices for the memory section to test. The choices are the test the lookup table memory for SMXR channels 0 and 1, 2 and 3, 4 and 5, 6 and 7 or the diagnostic memory. Type the first character of the menu item in order to make the selection.

```
10000000
7FFFFFFF
AAAAAAAA
Sequence
Random
```

Selecting item 2 brings up a menu which presents the choices for the data patterns to use in the memory tests. A copy of what to write to the selected is stored in local RAM. This buffer is used to compare with the data eventually read back from the SMXR memory. As above, type the first character of the item to make the selection.

Items 3 and 4 of the top level menu direct the program to perform VME block transfers (BLT), write and read respectively, on the selected memory. After the read is done, a comparison is made of the local buffer and the data read back. If any errors are detected, both the local buffer and the returned data are displayed side by side on a scrolled page. The local, or expected, data are shown of the left hand side and the read back data on the right hand side.

Items 5 and 6 direct the program to do word by word VME 32 bit transfers and then execute the same tests on the data as described in the preceding paragraph.

Item 7 is used to clear the selected memory, i.e. set each word to zero.

Item 8 starts a loop of BLT wrote followed BLT read and data comparison which runs until an error is detected or interrupted by <CTRL-C>.

2.4 Flash Memory test - smxre4

This utility test the functionality of the flash EPROM. It may be run directly from the shell with the command

smxre4 -g<slot>

It may also be run from the ***smxr*** menu item 4

```
4-SMXR FlashMem
```

The flash memory is used to save the contests of each lookup table so that it can be reloaded automatically upon power up. This utility can be used to test the operation of each lookup table one at a time with one of five data patterns. The top level menu is shown below:

```
1-Data Pattern     Random
2-Write Lookup     0-1 Lookup
3-Program Flash    00000000
4-Clear Lookup
5-Read Lookup
6-Loop Test

Q - Quit
```

Item 1 is used to choose the data pattern as described in section 2.3 and save a local copy of the data buffer which will eventually be written to the lookup table.

Item 2 is used to choose the lookup to test and directs the program to write the buffer filled with the selected data pattern to the lookup table memory.

Item 3 can be used to directly program the flash memory by writing to the 32-bit flash register. This register contains eight 4-bit nibbles which control the eight SMXR channels. Each nibble may contain the value 4 to erase the flash, 2 to download the flash or 1 to upload the flash. After writing the value to the register, the program polls the register once per second and each bit will be cleared once the operation is completed.

Item 4 is used to clear the lookup table and is normally executed after erasing and downloading the flash memory.

Item 5 is used to read the lookup table and compare its contents to the local copy. Any errors are reported in the same manner as described in the memory test, section 2.3.

Item 6 is used to loop over the cycle of programming the flash and checking it. It directs the program to write the selected data pattern to the selected lookup table, erase the flash, download the flash, clear the lookup table, upload the flash and compare the lookup table memory to the expected values. Failure of the flash memory to erase, download or upload properly are reported as program errors.

2.5 Diagnostic Mode test - smxre5

This utility test the data processing functions of the SMXR module by cycling test events from the card's diagnostic memory. it may be run directly from the shell with the command:

smxre5 -g<slot>

It may also be run from the ***smxr*** menu item 5

```
5-SMXR DiagMem
```

The program is controlled with the top level menu shown below:

```
1-Start Address    00000000
2-Select Data      Random
3-Select Mode      VME
4-Clear
5-Strobe Data
6-Loop Test

Q - Quit
```

Item 1 is used to select the starting address of the diagnostic memory from which to strobe the data. This address is automatically increment to start an event boundary as events are strobed through the SMXR.

Item 2 is used to select the data pattern and item 3 to select the mode with which the SMXR will process the incoming data. Each of the four diagnostic modes, Data-32, Data-48, Calibrate-32 and Calibrate-48 can be tested with one of the with one of three data patterns loaded into diagnostic memory: Random, 32-word or 48-word. For the latter two modes, the capacitor ID and parity are set correctly for the data generated. The random data pattern does not set these bits correctly and the error detection logic of the SMXR should set one or more error bits for most words.

Item 4 is used to set the SMXR to a know state.

For the selected data pattern and mode, data may be strobed one event at a time or allowed to loop.The SMXR processes the event strobed from diagnostic memory and stores the result in dual port memory. Each of the eight SMXR channels gets the same data. The data read from the dual port memory is displayed on a scrolled page with the expected words on the left column and the data for each channel in the right columns. Any discrepancies are highlighted. A sample page is shown below:

```
 Buf0  Simulated      Ch0  Ch1    Ch2  Ch3    Ch4  Ch5    Ch6  Ch7
0000      0000      0000  0000  0000  0000  0000  0000  0000  0000
0001      0001      0001  0001  0001  0001  0001  0001  0001  0001
0002      0002      0002  0002  0002  0002  0002  0002  0002  0002
0003      0003      0003  0003  0003  0003  0003  0003  0003  0003
0004      0104      0104  0104  0104  0104  0104  0104  0104  0104
0005      0105      0105  0105  0105  0105  0105  0105  0105  0105
0006      0106      0106  0106  0106  0106  0106  0106  0106  0106
0007      0107      0107  0107  0107  0107  0107  0107  0107  0107
0008      0208      0208  0208  0208  0208  0208  0208  0208  0208
0009      0209      0209  0209  0209  0209  0209  0209  0209  0209
0010      020a      020a  020a  020a  020a  020a  020a  020a  020a
0011      020b      020b  020b  020b  020b  020b  020b  020b  020b
0012      030c      030c  030c  030c  030c  030c  030c  030c  030c
0013      030d      030d  030d  030d  030d  030d  030d  030d  030d
0014      030e      030e  030e  030e  030e  030e  030e  030e  030e
0015      030f      030f  030f  030f  030f  030f  030f  030f  030f
0016      0010      0010  0010  0010  0010  0010  0010  0010  0010
0017      0011      0011  0011  0011  0011  0011  0011  0011  0011
0018      0012      0012  0012  0012  0012  0012  0012  0012  0012
0019      0013      0013  0013  0013  0013  0013  0013  0013  0013
 Home H  Middle M  End E  Next Page >  Prev Page <  Continue C  Quit Q
```

The display can be scrolled one page at a time or moved to the middle or end of the buffer by typing the letter indicated on the last line of the display. Typing C for continue displays the next event. Typing Q for quit ends the test.

Item 5 is used to strobe the data one event at a time. The program reads the dual port memory after each is processed and displays the results.

Item 6 is used to set the SMXR looping continuously over all the events in the diagnostic memory. After each event is processed, the dual port memory is read and compared with the expected result. The first four events in error are displayed as described above. After that, the program keeps running so that signals can be diagnosed with the scope.

2.6 Serial Trigger Sum test - smxre6

This utility tests the trigger sum logic and the serial output of the result trigger masks. An auxiliary module is required to capture the serial output. it may be run directly from the shell with the command

smxre6 -g<slot>

It can also be run form the ***smxr*** menu item 6

```
6-SMXR Serial Cap
```

The program is controlled with the top level menu shown below:

```
1-Trigger Sums     Ch 2,5
2-Select Data      Random
3-Select Mode      DiagD 32
4-Strobe Data
5-Loop Test

Q - Quit
```

Item 1 is used to set the trigger sum thresholds. The trigger sum logic is only implemented on Channels 2 and 5 so that four registers have to set, namely the low and high thresholds for each of these channels.

Item 2 is used to select the data pattern to be loaded into diagnostic memory. This can be either random or Data-32 with the capacitor ID values set correctly.

Item 3 selects the data mode which must be Data-32 since this is the only mode in which the trigger sum logic operates.

Item 4 is used to strobe data through the SMXR and capture the serial trigger sum masks in the fibre optic receiver auxiliary module.The data are received as shown below:

```
Buf0 Trig Sums
     Ch2    Ch5
    0848    0848
    09ec    09ec
    0d38    0d38
    0d61    0d61
    0d2c    0d2c
    10c2    10c2
    0f60    0f60
    1040    1040

    Trig Sum Regs                  Expected Captured
High2345    aaaa                        00 00
Low 1234    5555                        00 00
                                       00 00
                                       00 00
                                       00 00


 C-Continue  Q-Quit
```

The display shows the eight trigger sums expected from the data strobed through the two channels, the high and low trigger sum registers and the expected and captured 5-byte stream of data for the event. The expected values for the five bytes are as follows:

TABLE 1.

Byte 0	the L1A buffer number
Byte 1	channel 5 trigger mask, sums 0-3
Byte 2	channel 5 trigger mask, sums 4-7
Byte 3	channel 2 trigger mask, sums 0-3
Byte 4	channel 2 trigger mask, sums 4-7

For each trigger mask byte, a bit is set to 1 if the trigger sum exceeds the threshold. The bits alternate from low to high threshold starting with sum 0 and running to sum 7. Any errors in the captured data are highlighted. Typing C for continue strobes another event. Typing Q for quit stops the test.

To allow the program to loop continuously, item 5 can be selected. The first four errors are reported after which the program continues to loop.

2.7 SMD Data Capture test - smxre7

This utility test the inputs connected to the SMD module. An auxiliary module may be used to emulate the SMD data source. the program can be run directly from the shell with the command

smxre7 -g<slot>

It can also be invoked from the ***smxr*** menu item 7

```
7-SMXR Data Cap
```

The difference between this test and the Diagnostic mode test is that data for an event is stored in the memory of the SMC auxiliary module or in the digital SQUIDS and strobed into the front panel connectors of the SMXT in response to the generation of one or four L1A signals by a Test Clk module. The program is controlled with a top level menu shown below:

Item 1 is used to set the Test Clk module to a known state with the backplane clock enabled. The value of the Test Clk CSR is displayed on the same line

Item 2 is used to program the Test Clk FIFO with a set of L1A pulses and to set the module running with the selected pattern. The L1As can be set to one of the four buffers, 0-3, or a set of 4 L1As, each with a different buffer number can be chosen to test the queuing in the SMXR.

Item 3 is used to select the data pattern of the events loaded into the SMC emulator module.

```
1-Reset TestClk    0200ffff
2-Test TSIE FIFO   1068ffff
3-Select Data      32 word
4-Select Mode      Data  32
5-SMC DAC          00000000
6-Reset SMXR
7-Strobe Data
8-Loop Test
9-Channel Mask

Q - Quit
```

Item 4 is used to select the SMXR mode, Data-32, Data-48, Calibrate-32 or Calibrate-48.

Item 5 is used to set the SMC DAC register. The register is read back from the emulator and displayed on the right.

Item 6 is used to reset the SMXR and take the Test Clk module out of continuous mode.

Item 7 is used to strobe a single event through the SMXR and display the data read from the dual port buffers. The events are displayed on a scrolled screen as described in section 2.5, Diagnostic Mode tests.

Item 8 is used to loop continuously over events and will stop and display any errors. Only the first four events with errors are displayed after which the program continues to loop.

Item 9 is used to set a mask of live input channels which should be used to check for errors.

3.0 Verification Utilities

This section gives a brief description of the verification or certification utilities. Unlike the exerciser utilities, these programs are invoked with arguments that specify the number of iterations to perform, the channels to test and other options. They are generally executed from a shell script which specifies a fixed number of iterations of each test and saves the results to a log file.

3.1 Register test - smxrv1

The register test performs simple write/read check of four sets of SMXR registers. Random data words are generated and used as test patterns. Unused bits are masked off. The registers tested are the Control/Status register, the programmable delay register, the Bunch Number Latency register and the Trigger Sum registers. The syntax of the command is as follows:

```
Syntax: smxrv1 [<options>]
Function:       Test Memory of SMXR
Options:
       -g<slot>        specify geo32 slot
        -i<iter>        number of patterns for each test
```

The requested number of iterations is performed on each register and the printed as in the sample below:

```
Register Tests started on Thu Mar  9 16:41:00 2000

Status register Test                    100 patterns, 0 errors

Programmable Delay register Test        100 patterns, 0 errors

Bunch Number latency register Test      100 patterns, 0 errors

Trigger Sums register Test              100 patterns, 0 errors

Register Tests done - Thu Mar  9 16:41:02 2000
```

3.2 ID PROM test - smxrv2

This utility reads the IPRPOM and prints the string. The syntax of the command is as follows:

```
Syntax: smxrv2 [<options>]
Function:       Report ID of SMXR
Options:
   -g<slot>        specify geo32 slot
```

3.3 Memory test - smxrv3

This utility does a write/read check of the lookup table and diagnostic memory using the VMEbus. There is a 128 Kword lookup table for each of the SMXR channels and a 128 Kword diagnostic memory. each of five data patterns, marching 1, marching 0, alternating 1 and 0, sequential and random, are used for the number of iterations specified. the syntax of the command is as follows:

```
Syntax: smxrv3 [<options>]
Function:       Test Memory of SMXR
Options:
       -g<slot>        specify geo32 slot
        -i<iter>        number of passes for each pattern
```

The results for each data pattern are printed as in the sample below:

```
Memory Tests started on Mon Jul 26 08:26:02 1999


Data pattern: 10000000
    1 passes, Memory: Ch0 Lookup, 0 errors
    1 passes, Memory: Ch1 Lookup, 0 errors
    1 passes, Memory: Ch2 Lookup, 0 errors
    1 passes, Memory: Ch3 Lookup, 0 errors
    1 passes, Memory: Ch4 Lookup, 0 errors
    1 passes, Memory: Ch5 Lookup, 0 errors
    1 passes, Memory: Ch6 Lookup, 0 errors
    1 passes, Memory: Ch7 Lookup, 0 errors
    1 passes, Memory: Diagnostic, 0 errors
```

3.4 Flash Memory test - smxrv4

This utility performs the complete cycle of erasing, downloading, uploading and checking the flash memory for each SMXr channel. The same five data patterns are used as in the memory test describes in section 3.3. The syntax of the command is as follows:

```
Syntax: smxrv4 [<options>]
Function:       Test Flash memory of SMXR
Options:
       -g<slot>        specify geo32 slot
       -i<iter>        number of passes for each pattern
```

The results of the test are printed for each pattern as in the sample shown below:

```
Flash Memory Tests started on Mon Jul 26 08:27:11 1999


Lookup Table data pattern: 10000000
    1 passes, Chan0, 0 program errors, 0 data errors
    1 passes, Chan1, 0 program errors, 0 data errors
    1 passes, Chan2, 0 program errors, 0 data errors
    1 passes, Chan3, 0 program errors, 0 data errors
    1 passes, Chan4, 0 program errors, 0 data errors
    1 passes, Chan5, 0 program errors, 0 data errors
    1 passes, Chan6, 0 program errors, 0 data errors
    1 passes, Chan7, 0 program errors, 0 data errors
```

The number of program errors reports the number of times that the bit which instructs the flash memory to erase, download or upload fails to clear within a 10 second time-out period indicating that the operation failed.

3.5 Diagnostic Mode test - smxrv5

The diagnostic mode utility runs three different data patterns through each SMXr channel in four different modes, Data-32, Data-48, Calibrate-32 and Calibrate-48. A random data pattern tests the error detection logic of the SMXR since the capacitor ID and parity bits are filled with random data. The 32-word and 48-word patterns have the parity and capacitor ID fields set correctly in the test data.For the data modes, each of the four buffers corresponding to the L1 buffer number is tested while for the calibrate modes, data is always saved in buffer 0. This is a rather complex test, especially in the data modes where the lookup tables are used to form the output data.

The syntax of the command is as follows:

```
Syntax: smxrv5 [<options>]
Function:       Test Diagnostic Modes of SMXR
Options:
       -g<slot>        specify geo32 slot
       -c<mask>        specify channel mask
       -i<iter>        number of events for each test
```

The channel mask is a hex number whose range is 1:ff which indicates the channels to test, one bit for each SMXR channel. The least significant bit represents channel 0.

he number of errors in each SMXR channel is reported. In addition, an error log file is kept which saves the entire event for the first 100 events which give errors. This log file can be used to isolate rarely occurring errors. It may be viewed with the ***analerr*** utility.

The results of the test for each data pattern are presented as in the sample below.

```
Diagnostic Mode Tests started on Mon Jul 26 08:29:06 1999


Diagnostic data pattern:  Random

 Diagnostic Mode:   Data  32
   Dual port memory buffer: 0     100 diagnostic events
   Dual port memory buffer: 1     100 diagnostic events
   Dual port memory buffer: 2     100 diagnostic events
   Dual port memory buffer: 3     100 diagnostic events

 Diagnostic Mode:   Data  48
   Dual port memory buffer: 0     100 diagnostic events
   Dual port memory buffer: 1     100 diagnostic events
   Dual port memory buffer: 2     100 diagnostic events
   Dual port memory buffer: 3     100 diagnostic events

 Diagnostic Mode:   Calib 32
   Dual port memory buffer: 0     100 diagnostic events

 Diagnostic Mode:   Calib 48
   Dual port memory buffer: 0     100 diagnostic events
```

3.6 Trigger Sum Serial Data test - smxrv6

This utility requires the auxiliary fibre optic receiver module. It tests the trigger sum logic in a systematic way by setting the thresholds just above and below the values of the expected trigger sums. A random data pattern is loaded into diagnostic memory and Data-32 mode is used for the tests. The syntax of the command is as follows:

```
Syntax: smxrv6 [<options>]
Function:       Test Trigger sums of SMXR
Options:
        -g<slot>        specify geo32 slot
       -i<iter>        number of events for each test
```

The program prints a summary like the sample shown below.

```
 Serial Data Capture Tests started on Thu Mar  9 16:44:18 2000

    10 diagnostic events      , 0 errors

 Serial Data Capture Tests done -  Thu Mar  9 16:44:23 2000
```

In addition, a log file is kept of the first 100 events which give errors and may be viewed with the ***analerrm*** utility.

3.7 SMD Data Capture test - smxrv7

This utility is similar to the diagnostic mode test described in section 3.5, except that the data source is the auxiliary SMC/SMD emulator module which strobes data into the front panel connectors f the SMXR in response to signals originating from the SMXR J3 connector. The Test Clk module is also needed as it is programmed to generate the L1A signals.

The syntax of the command is as follows:

```
Syntax: smxrv7 [<options>]
Function:       Test Data Capture of SMXR
Options:
        -g<slot>        specify geo32 slot
        -c<mask>        specify channel mask
        -i<iter>        number of events for each test
```

The channel mask has the same meaning as in the ***smxrv5*** utility of section 3.5**.** The program runs for the specified number of iterations in each of the four modes, Data-32, Data-48, Calibrate-32 and Calibrate-48. Only a random pattern is used to load the SMC/SMD emulator.

A printed summary as well as an error log file is produced. The error log file contains the first 100 events which show errors and can be read with the ***analerr*** utility.