MCRUSH Test Tool User Manual

v0.4 (02 Mar 2000)
Henk Boterenbrood

♦ sndr_cr + (SHASLINK) recv_sl and recv_cr + (SHASLINK) sndr_sl

- Test one link at a time sending data from (M)CRUSH to SHASLINK, or from SHASLINK to (M)CRUSH; includes check of every word transferred.
- **sndr_cr/recv_cr** should be run on a (M)CRUSH module, **recv_sl/sndr_sl** should be run on a SHASLINK module.
- 20 Mbyte (DMA size <20 words) or 40 Mbyte (DMA size >= 20 words) of data is transferred when data check is enabled.
- 100 Mbyte (DMA size <20 words) or 200 Mbyte (DMA size >= 20 words) of data is transferred when data check is disabled.
- Data check disable is optional (see options below): either a nibble-pattern (0x0f781e69 and 0x2d5a3c4b) or random data (with known seed) is sent (this is a compile option).
- Link 4 and 5 can not be tested like this because link 4 is needed for booting the (M)CRUSH, and link 4 and 5 form a cable pair).

Options:

-l<*link*>	use Link Port <*link*> (default: 0)
-s<*size*>	use <*size*> number of words for each DMA (default: 256)
-c<*speed*>	set link speed:<*speed*> =1: 20 MB/s, <*speed*> =2: 40 MB/s (default: 40 MB/s)
-n	disable data check (link transfer rate should now approach 20 or 40 MB/s)
-d	(only if checking enabled) report each individual error in the data, otherwise print a table of errors after completion of the test.

♦ ppcrush

- 'Peek' and 'poke' in SHARC memory map
- (Single) blockmoves.
- SHARC Flag read/write:
  Flag 0 (output): S-LINK reset: use SLIDAD to observe reset status
  Flag 1 (output): FPGA reset: put something in registers/buffer, apply reset and observe changes.
  Flag 2 (output): red LED
  Flag 3 (input): Output FIFO empty status: use test mode to download data and observe flag value
- Some SHARC addresses (word addresses):

`0x00000000-0x000000FF`	SHARC IOP registers (see SHARC documentation)
`0x00020000-0x0003FFFF`	SHARC internal memory (normal word addressing)
`0x00400000-0x00400015`	MCRUSH registers in FPGA
`0x00480000`	Output FIFO
`0x00500000-0x00580000`	ZBT buffer memory (256 Kword, twice)

Options:

-p <*foo*>	download the contents of file <foo> to the SHARC in the form of ***POKE*** operations (so 'write' operations only) to the MCRUSH registers at start-up; <foo> must be an ASCII file containing lines with the following syntax: ***<RegAddr> <RegValue> ...comments can go here....*** with <*RegAddr*> and <*RegValue*> hexadecimal numbers with 0<=RegAddr<=0x14 (maximum number of such presets currently is: 128).

mcreg

- (Write/)read test of selected registers
- All values (consecutively) in registers with < 21 significant bits (each value 10 times), about 1 million random values for larger registers. (A write-cycle with inverted databits to a (dummy) register is inserted between every write- and read-cycle to be able to detect possibly tri-stated data lines).

mcmem

Series of test on the ZBT buffer memory, using:

- random data: memory is written 10x (A write-cycle with inverted databits to a (dummy) register is inserted between every write- and read-cycle to be able to detect tri-stated data lines).
- patterns: `0xAAAAAAAA,0x55555555,0xA5A5A5A5,0x5A5A5A5A,0xAAAA5555,0x5555AAAA,0x55AA55AA,0xAA55AA55,0xFFFFFFFF,0x00000000.`
- walking-1 and walking-zero: every bit in every memory location is tested individually.
- overlay: walking-1/0 and pattern in address (data follows some patterns too), check of all other locations:
  1st round: data at unwritten addresses unequal to `0x00000000`? → error
  2nd round: data at unwritten addresses unequal to `0xFFFFFFFF`? → error

Options:

-o	skip the overlay test (which typically takes a few minutes…)

mcregbit

- Checks the functioning of bit patterns and individual bits in the following registers:
  - Separator and Separator Mask registers
  - Separator Control-or-Data and Separator Control-or-Data Mask registers
  - TDC Trailer and TDC Trailer Mask registers
  - TDC Trailer Control-or-Data and TDC Trailer Control-or-Data Mask registers
  - MCRUSH Header register
  - TDC Input Mask register
  - TDC Readout Mask register
- Different bit patterns are written to the registers (the same 'patterns' as listed above for **mcmem (**except 0x00000000) and a walking zero) and correspondingly formatted S-Link input TDC data is injected and Output FIFO data is read out and checked to see if the settings work as they should.
- Options below apply to the data downloaded in each subtest with one particular pattern; options '-t' and '-m' (see below) do not apply in the test of the TDC Input Mask and TDC Readout Mask registers.

Options:

-t<*tdcs*>	<*tdcs*> is the number of TDC time slots in the input data between *Separators* (default: 5); should lie between 1 and 32 (but remember that although more than 18 slots can be present only up to 18 slots of data is handled by the MCRUSH hardware).
-m<*mask*>	<*mask*> is a hexadecimal bit mask denoting which TDC slots carry data (default: 0x5); the tool checks whether <*mask*> matches the number of TDC time slots <*tdcs*> (see option '-t' above).
-s<*size*>	the (average) size per TDC 'event' (default: 4); must be >= 2; includes the TDC header and trailer
-n<*evts*>	the number of 'events' per TDC generated (default: 3)
-d	display S-Link input data too (default: no display)

- **mcintr**
  - Tests all possible MCRUSH interrupts and interrupt sources, using the test mode to download input data:
    - IRQ0: I2O-FIFO full
    - IRQ0: Buffer memory partition full (check enabled partitions)
      (**NB**: the MCRUSH can NOT recover from this without resetting the ALTERA hardware, because getting rid of the partition-full would require read-out of the TDC's data to get the read/write pointers out of the partition-full state, but as soon as read-out is enabled partition-full causes the read-out to be disabled again…)
    - IRQ0: Maximum TDC eventsize reached (sizes in test: maxsize = 6, evtsize=8; check of enabled/disabled partitions)
    - IRQ1: S-Link LDERR:
      - address (bits 0-17)
      - control-word bit (18)
      - overrun bit (19)
    - IRQ2: Event-ID outside window (EARLY)
      - event-id (bits 0-11)
      - TDC slot number (bits 12-16)
      - early bit (17)
      - late bit (18)
      - overrun bit (19)
    - IRQ2: Event-ID outside window (LATE)
      - event-id (bits 0-11)
      - TDC slot number (bits 12-16)
      - early bit (17)
      - late bit (18)
      - overrun bit (19)

Options:

-s	stop after every subtest, show intermediate result
-r<*rep*>	repeat every interrupt test <*rep*> times (default 1x); display summary per interrupt source

♦ mcgen

- Tool for injecting S-Link TDC test data into the MCRUSH using the MCRUSH test mode.
- The data that is read from the output FIFO is displayed, with MCRUSH event headers and TDC trailers marked, and a clear separation between consecutive events.
- If FIFO output data does not match the expected number of words or events this is reported; words read from the Output-FIFO are NOT checked individually for correctness.
- Interrupts detected are reported after every test run.
- Output FIFO readout is optionally done through DMA (see 'options' below).
- The total maximum number of data words that can be injected in one test run is 8192.
- If the number of events generated is greater than or equal to 8 and the event size is greater than 4, optionally (option '-v') some variation in individual TDC event sizes is made (up to +3 and -3 datawords per event extra) and an occasional NODATA word is inserted. Use option '-d' to see the generated input data displayed.
- Other settings used are:
  - ❖ Separator: `0xBBBBBBBB`
  - ❖ Separator mask: `0xFFFFFFFF`
  - ❖ TDC Header: `0xA`*`tnnn`*`000` `(`*`t`*`=TDCnumber%16,` *`nnn`*`=EventId)`
  - ❖ TDC Trailer: `0xC0`*`nnn`*`000` `(`*`nnn`*`=EventId)`
  - ❖ TDC Trailer mask: `0xFF000FFF`
  - ❖ TDC data word: `0x3`*`tTTwwww`* `(`*`t`*`=TDCnumber%16,` *`T`*`=TDCnumber,` *`wwww`*`=WordCounter)`
  - ❖ MCRUSH header: `0x990`*`mmmmm`* `(`*`mmmmm`*`=bitmask denoting which TDCs have data present in the Output-FIFO for the current event)`
  - ❖ MCRUSH maximum eventsize setting: `1024`

Options:

-t<*tdcs*>	<*tdcs*> is the number of TDC time slots in the input data between *Separators* (default: 5); should lie between 1 and 32 (but remember that although more than 18 slots can be present only up to 18 slots of data is handled by the MCRUSH hardware).
-m<*mask*>	<*mask*> is a hexadecimal bit mask denoting which TDC slots carry data (default: 0x5); the tool checks whether <*mask*> matches the number of TDC time slots <*tdcs*> (see option '-t' above).
-s<*size*>	the (average) size per TDC 'event' (default: 4); must be >= 2; includes the TDC header and trailer
-n<*evts*>	the number of 'events' per TDC generated (default: 3)
-i<*evtid*>	use <*evtid*> as the event-id of the first event (default: 0x000); for every subsequent event the event-id is incremented by 1
-D	the Output FIFO is read out by a DMA (default: no DMA)
-v	create some variation in individual TDC eventsizes (default: no variation)
-r	receive only (default: send-and-receive); S-LINK input data is NOT locally generated and downloaded; to be used in combination with DMA (option '-D'); **mcgen** can be used in this way to check incoming *real* S-Link input data, but the amount of data is limited to a total of 8192 words; to be used in combination with e.g. the **SHASLINK csmsim** CSM/TDC data (ATLAS Muon Chamber data) generator application
-o	do NOT display MCRUSH FIFO output (default: display)
-d	display S-Link input data (default: no display)

- **mcgenf**
  - Tool for injecting S-Link TDC test data into the MCRUSH using the MCRUSH test mode.
  - The test data is read from a binary file in blocks of at maximum 1000 words, one block at a time after which it is downloaded in the MCRUSH; however, before reading and downloading starts the file is read once to determine the first event-id, the number of TDC slots between separators –reporting differing distances between separators–, and the number of events per TDCs (depending on the number of TDC-trailers seen).
  - The data that is read from the output FIFO is displayed (or not, using option '-o'), with MCRUSH event headers and TDC trailers marked, and a clear separation between consecutive events.
  - Words read from the Output-FIFO are NOT checked individually for correctness.
  - Interrupts detected are reported after every block download; if an interrupt occurred the next block of data read from the Output-FIFO (if any data is available) is displayed even when displaying has been switched off.
  - The total maximum number of data words that can be injected in the MCRUSH in one test run is unlimited, in principle (although the reading of the file and the displaying of data lasts prohibitively long when the file is big…)
  - Expected header types are:
    - ❖ Separator: `0xDD000000`
    - ❖ Separator mask: `0xFF000000`
    - ❖ TDC Header: `0xA4xxxxxx` (*t*=TDCnumber%16, *nnn*=EventId)
    - ❖ TDC Trailer: `0xC4xxxxxx`
    - ❖ TDC Trailer mask: `0xFF000000`
    - ❖ MCRUSH header: 0x990*mmmmm* (*mmmmm*=bitmask denoting which TDCs have data present in the Output-FIFO for the current event)

Options:

-f<*file*>	<*file*> contains the CSM/TDC data to be downloaded
-o	do NOT display MCRUSH FIFO output (default: display)
-d	display S-Link input data (default: no display)

- **mccheck**
  - Tool for checking CSM/TDC input data received in the MCRUSH Output FIFO word-for-word.
  - Individual errors are reported and a 'freeze' of the FPGA stops further input…
  - To be used in combination with SHASLINK data generator tool **csmsim** (it is important to use matching options!).
  - The TDC data is expected in the form of blocks of data containing a configurable number of 'events' for a configurable number of TDCs; blocks are read from the MCRUSH Output-FIFO by one DMA each; the number of blocks to receive can be set; the data in the blocks is expected to follow a range of patters and is expected to be the same for every block; a check on properly incrementing event-identifiers in TDC-headers and -trailers is performed.
  - Data values assumed are:
    - ❖ Separator: 0xBBBBBBBB
    - ❖ TDC Header: 0xA*tnnn*000 (*t*=TDCnumber%16, *nnn*=EventId)
    - ❖ TDC Trailer: 0xC0*nnn*000 (*nnn*=EventId)
    - ❖ TDC data words: 0x6aaaaaaa, 0x55555555, 0x73fcf3fc, 0x3fcf3fcf, 0x65a5a5a5, 0x5a5a5a5a, 0x2aaa5555, 0x5555aaaa, 0x55aa55aa, 0x2a55aa55, 0x7fffffff, 0x10000000, 0x70f0f0f0, 0x0f0f0f0f
    - ❖ TDC data word: 0x3*tTTwwww* (*t*=TDCnumber%16, *T*=TDCnumber, *wwww*=WordCounter)

Options:

-t<*tdcs*>	<*tdcs*> is the number of TDC time slots in the input data between *Separators* (default: 5); should lie between 1 and 32 (but remember that although more than 18 slots can be present only up to 18 slots of data is handled by the MCRUSH hardware).
-m<*mask*>	<*mask*> is a hexadecimal bit mask denoting which TDC slots carry data (default: 0x5); the tool checks whether <*mask*> matches the number of TDC time slots <*tdcs*> (see option '-t' above).
-s<*size*>	the (average) size per TDC 'event' (default: 4); must be >= 2; includes the TDC header and trailer
-n<*evts*>	the number of 'events' per TDC, per block, generated (default: 3)
-i<*evtid*>	use <*evtid*> as the event-id of the first event (default: 0x000); for every subsequent event the event-id is incremented by 1
-b<*blocks*>	the number of blocks of TDC-events generated (default: 1)
-c	data words in the datastream are **NOT** checked (default: checked)
-D	the Output FIFO is **NOT** read out by a DMA (default: DMA)
-v	there is some variation in individual TDC eventsizes (default: no variation)
-o	display MCRUSH FIFO output (default: not displayed)

- The application reads (by DMA) and checks data blocks in parallel and can sustain this at a rate of approximately 7 Mbyte/s (9 Mbyte/s when datawords are not checked). Parameters for the **csmsim** generator should be set such that this rate is not exceeded, enabling **mccheck** to run for long periods at a time ('duration tests').

- **csmsim**
  - Application for **SHASLINK** module: CSM/TDC-data generator application
  - Can for instance be used in combination with tool **mcgen** (using **mcgen** with options '-r -D', see above)
  - This application generates TDC data as a CSM would (or at least, tries to approximate this) and sends it out across the SHASLINK's S-LINK output interface, using DMA.
  - The TDC data is generated in the form of blocks of data containing a configurable number of 'events' for a configurable number of TDCs; blocks are sent by one DMA each; the number of blocks to be sent and the frequency are also configurable; datawords in a block follow a particular pattern and is the same for each block sent; properly incrementing event-identifiers are inserted in TDC headers and trailers.
  - Data values used are:
    - Separator: 0xBBBBBBBB
    - TDC Header: 0xA*tnnn*000 (*t*=TDCnumber%16, *nnn*=EventId)
    - TDC Trailer: 0xC0*nnn*000 (*nnn*=EventId)
    - TDC data words: 0x6aaaaaaa, 0x55555555, 0x73fcf3fc, 0x3fcf3fcf, 0x65a5a5a5, 0x5a5a5a5a, 0x2aaa5555, 0x5555aaaa, 0x55aa55aa, 0x2a55aa55, 0x7fffffff, 0x10000000, 0x70f0f0f0, 0x0f0f0f0f
    - TDC data word: 0x3*tTTwwww* (*t*=TDCnumber%16, *T*=TDCnumber, *wwww*=WordCounter)

Options:

-t<*tdcs*>	<*tdcs*> is the number of TDC time slots in the input data between *Separators* (default: 5); should lie between 1 and 32 (but remember that although more than 18 slots can be present only up to 18 slots of data is handled by the MCRUSH hardware).
-m<*mask*>	<*mask*> is a hexadecimal bit mask denoting which TDC slots carry data (default: 0x5); the tool checks whether <*mask*> matches the number of TDC time slots <*tdcs*> (see option '-t' above).
-s<*size*>	the (average) size per TDC 'event' (default: 4); must be >= 2; includes the TDC header and trailer
-n<*evts*>	the number of 'events' per TDC, per block, generated (default: 3)
-i<*evtid*>	use <*evtid*> as the event-id of the first event (default: 0x000); for every subsequent event the event-id is incremented by 1
-b<*blocks*>	the number of blocks of TDC-events generated (default: 1)
-f<*freq*>	the frequency with which blocks of TDC-events are sent (default: 1)
-v	create some variation in individual TDC eventsizes (default: no variation)
-d	display S-Link input data (default: no display)

- Examples of measured maximum S-LINK output rate [Mbytes/s] (depending on SHASLINK hardware and software limitations; actual maximum rates may be limited by the S-LINK physical implementation):

Options used	Rate [MB/s]
-t18 -m3 -n20 -s10 -b1000:	155.5
-t18 -m3 -n10 -s10 -b1000:	153.5
-t18 -m3 -n9 -s10 -b1000:	152.7
-t18 -m3 -n8 -s10 -b1000:	151.3
-t18 -m3 -n7 -s10 -b1000:	135.4
-t18 -m3 -n6 -s10 -b1000:	123.3
-t18 -m3 -n5 -s10 -b1000:	109.6
-t18 -m3 -n4 -s10 -b1000:	93.9
-t18 -m3 -n3 -s10 -b1000:	75.8
-t18 -m3 -n2 -s10 -b1000:	54.7
-t18 -m3 -n1 -s10 -b1000:	29.8