SERIAL BUS COMMANDS FOR TAB chips 0-9

Modifications to this File:

03-Aug-05 corrected swap of “raw delay” and “scl delay”
13-May-05 original version

Signals used:

to the TAB:
serfrm seradr serdata
from the TAB
serfrmout ser_dat_out

Protocol format:

```
|->Frame bit
|  |->Chip address
|  |   |->File address (0- test memory
|  |   |                1- em thresholds
|  |   |                2- jet thresholds
|  |   |                3- status/mode
|  |   |                4- TAB file memory
|  |   |                5- init pulse for PLL
|  |   |                6- delay parameters
|  |   |                7- raw file memory
|  |   |                8- read par count
|  |   |                9- write lfsr seed cable 1
|  |   |                10-write lfsr seed cable 2
|  |   |                11-write lfsr seed cable 3
|  |   |                12-enable lfsr
|  |   |                13-not used
|  |   |                14-not used
|  |   |                15-not used
|  |   |                )
|  |   |
|  |   |            address space          data
---------          ----------------   ----------------
1000101010000000   0000000000000000   0000000000000000
      serfrm            seradr            serdat
```

COMMANDS:

```
--read the firmware version register
1000100110000000   0000000000000010    0000000000000000
--read TAB status register
1000100110000000   0000000000000000    0000000000000000
--generate init pulse for PLL
1000101010000000   0000000000000000    0000000000000000
--set the TAB mode (0-FIR 1-test memory)
1000100110000000   0000000000000001    0XXXXXXXXXXXXXX
--load delay parameters:
1000101100000000   0000000000000000    0000000000101111 scl delay
1000101100000000   0000000000000001    0000000000101111 raw delay
--load em parameters:
1000100010000000   0000000000000000    0000000000000001 threshold 1
1000100010000000   0000000000000001    0000000000000011 threshold 2
1000100010000000   0000000000000010    0000000000000101 threshold 3
1000100010000000   0000000000000011    0000000000001001 threshold 4
1000100010000000   0000000000000100    0000000000010001 threshold 5
1000100010000000   0000000000000101    0000000000100001 threshold 6
1000100010000000   0000000000000110    0000000001000001 threshold 7
1000100010000000   0000000000000111    0000001111111111 em iso thr
1000100010000000   0000000000001000    0000000000111111
                                                       -- a parameter (bits 0,1)
                                                     --   b parameter (bits 2,3)
                                                    -     iso enable  (bit 4)
                                                   -      had ebable  (bit 5)
1000100010000000   0000000000001001    0000000000000111
                                                      ---
                                                       L2 em_mask_bit_threshold
--load jet parameters:
1000100100000000   0000000000000000    0000000000000011 threshold 1
1000100100000000   0000000000000001    0000000000000110 threshold 2
1000100100000000   0000000000000010    0000000000001100 threshold 3
1000100100000000   0000000000000011    0000000000011000 threshold 4
1000100100000000   0000000000000100    0000000000110000 threshold 5
1000100100000000   0000000000000101    0000000001100000 threshold 6
1000100100000000   0000000000000110    0000000011000000 threshold 7
1000100100000000   0000000000000111    0000000000000000
                                                      ---
                                                      jet_mask_bit_threshold
                                                   ---
                                                      tau_mask_bit_threshold
```

```
--ADF TAB pseudo random number test
--test starts by the first buf_out(33)=1 coming from ADF
--random number generators and the test is enabled
1000111000000000   0000000000000000    0000000000000001
--disabled
1000111000000000   0000000000000000    0000000000000000

--load the seeds for RNG
--cable 1
1000110010000000   0000000000000000    0000000000000000
                             ------    ----------------
                               gen#      seed H  seed L
--cable 2
1000110100000000   0000000000000000    0000000000000000
                             ------    ----------------
                               gen#      seed H  seed L
--cable 3
1000110110000000   0000000000000000    0000000000000000
                             ------    ----------------
                               gen#      seed H  seed L

--generator to energy mapping in each cable
-- qe_i(i)      em energy coming for ADF
-- qh_i(i)      hd energy coming for ADF
-- lsfr_out(i)  energy generated by pseudo random generator
----check them against the data coming from the ADF cable
----SIGNAL lfsr_out      : STD_LOGIC_VECTOR (31 downto 0);
----t1:
----FOR i IN 0 to 15 GENERATE
----       lfsr_xor(i*2)     <= lfsr_out(i*2)     xor qe_i(i);
----       lfsr_xor((i*2)+1) <= lfsr_out((i*2)+1) xor qh_i(i);
----END GENERATE;
```

```
--load test memory content
--event#0
--energy                event tower       had     em
1000100000000000   0000000000000010    0000000100000001 tower 2
1000100000000000   0000000000001100    0000000100000001 tower 12
1000100000000000   0000000000010110    0000000100000111 tower 22
1000100000000000   0000000000100011    0000000100000111 tower 35
--control P/F           event   adr     Parity  Frame
1000100000000000   1000000000000001    0000000100000001 cable1
1000100000000000   1000000000000011    0000000100000001 cable2
1000100000000000   1000000000000101    0000000100000001 cable3
--control S/B           event   adr     Spare     Bx
1000100000000000   1000000000000000    0000000000000000 cable1
1000100000000000   1000000000000010    0000000000000000 cable2
1000100000000000   1000000000000100    0000000000000000 cable3


--event#1
--energy                event tower       had     em
1000100000000000   0000000000001000010    0000000100000001 tower 2
1000100000000000   0000000000001001100    0000000100000001 tower 12
1000100000000000   0000000000001010110    0000000100000111 tower 22
1000100000000000   0000000000001100011    0000000100000111 tower 35
--control P/F           event   adr     Parity  Frame
1000100000000000   1000000000001000001    0000000100000001 cable1
1000100000000000   1000000000001000011    0000000100000001 cable2
1000100000000000   1000000000001000101    0000000100000001 cable3
--control S/B           event   adr     Spare     Bx
1000100000000000   1000000000001000000    0000000000000000 cable1
1000100000000000   1000000000001000010    0000000000000000 cable2
1000100000000000   1000000000001000100    0000000000000000 cable3
```

```
--read back the TAB file memory
event#0
                        event addr (of the memory word)
1000101000000000   0000000000000000    0000000000000000
1000101000000000   0000000000000001    0000000000000000
1000101000000000   0000000000000010    0000000000000000
1000101000000000   0000000000000011    0000000000000000
1000101000000000   0000000000000100    0000000000000000
1000101000000000   0000000000000101    0000000000000000
1000101000000000   0000000000000110    0000000000000000
1000101000000000   0000000000000111    0000000000000000
1000101000000000   0000000000001000    0000000000000000
1000101000000000   0000000000001001    0000000000000000
1000101000000000   0000000000001010    0000000000000000
1000101000000000   0000000000001011    0000000000000000
1000101000000000   0000000000001100    0000000000000000
1000101000000000   0000000000001101    0000000000000000
1000101000000000   0000000000001110    0000000000000000
1000101000000000   0000000000001111    0000000000000000
1000101000000000   0000000000010000    0000000000000000
1000101000000000   0000000000010001    0000000000000000
1000101000000000   0000000000010010    0000000000000000
1000101000000000   0000000000010011    0000000000000000

event#1
                        event addr
1000101000000000   0000000000100000    0000000000000000
1000101000000000   0000000000100001    0000000000000000
1000101000000000   0000000000100010    0000000000000000
1000101000000000   0000000000100011    0000000000000000
1000101000000000   0000000000100100    0000000000000000
1000101000000000   0000000000100101    0000000000000000
1000101000000000   0000000000100110    0000000000000000
1000101000000000   0000000000100111    0000000000000000
1000101000000000   0000000000101000    0000000000000000
1000101000000000   0000000000101001    0000000000000000
1000101000000000   0000000000101010    0000000000000000
1000101000000000   0000000000101011    0000000000000000
1000101000000000   0000000000101100    0000000000000000
1000101000000000   0000000000101101    0000000000000000
1000101000000000   0000000000101110    0000000000000000
1000101000000000   0000000000101111    0000000000000000
1000101000000000   0000000000110000    0000000000000000
1000101000000000   0000000000110001    0000000000000000
1000101000000000   0000000000110010    0000000000000000
1000101000000000   0000000000110011    0000000000000000



--read back the raw file memory
event#0
                        event addr
1000111100000000   0000000000000000    0000000000000000
```

```
INPUT TEST MEMORY DATA FORMAT:
-- serial address shift register
-- (address is shifted left - MSB comes first)
-- structure of the seradr
-- bit 15  0:energy data             1:control data
--     14
--     13
--     12
--     11
--     10   event#(4)
--      9   event#(3)
--      8   event#(2)
--      7   event#(1)
--      6   event#(0)
--      5   tower_address(5)
--      4   tower_address(4)
--      3   tower_address(3)
--      2   tower_address(2)       control_address(2)
--      1   tower_address(1)       control_address(1)
--      0   tower_address(0)       control_address(0)
--
--
-- control data addresses
-- cable #1
-- control_address="000":   Bx          low  byte
-- control_address="000":   spare       high byte
-- control_address="001":   Frame       low  byte
-- control_address="001":   parity      high byte
-- cable #2
-- control_address="010":   Bx          low  byte
-- control_address="010":   spare       high byte
-- control_address="011":   Frame       low  byte
-- control_address="011":   parity      high byte
-- cable #3
-- control_address="100":   Bx          low  byte
-- control_address="100":   spare       high byte
-- control_address="101":   Frame       low  byte
-- control_address="101":   parity      high byte
```

```
-- serial data shift register
-- (data is shifted right - LSB comes first!!!!)
-- structure of the serdat
--
--            energy data                  contol data
--
-- bit 15  hd_tower_energy(7)                          parity(7)
--     14  hd_tower_energy(6)                          parity(6)
--     13  hd_tower_energy(5)                          parity(5)
--     12  hd_tower_energy(4)                          parity(4)
--     11  hd_tower_energy(3)                          parity(3)
--     10  hd_tower_energy(2)                          parity(2)
--      9  hd_tower_energy(1)                          parity(1)
--      8  hd_tower_energy(0)                          parity(0)
--      7  em_tower_energy(7)        bunch_crosing(7)  frame(7)=0
--      6  em_tower_energy(6)        bunch_crosing(6)  frame(6)=0
--      5  em_tower_energy(5)        bunch_crosing(5)  frame(5)=0
--      4  em_tower_energy(4)        bunch_crosing(4)  frame(4)=0
--      3  em_tower_energy(3)        bunch_crosing(3)  frame(3)=0
--      2  em_tower_energy(2)        bunch_crosing(2)  frame(2)=0
--      1  em_tower_energy(1)        bunch_crosing(1)  frame(1)=0
--      0  em_tower_energy(0)        bunch_crosing(0)  frame(0)=1
```

```
---INPUT ENERGY MAPPING vs test memory channel number
--                    (energy)
--
--     8      8 17  26  35  44  53 62 71 80
--     7      7 16  25  34  43  52 61 70 79
--     6      6 15  24  33  42  51 60 69 78
--     5    * 5 14  23  32  41  50 59 68 77
--phi  4    * 4 13  22  31  40  49 58 67 76
--     3    * 3 12  21  30  39  48 57 66 75
--     2    * 2 11  20  29  38  47 56 65 74
--     1      1 10  19  28  37  46 55 64 73
--     0      0  9  18  27  36  45 54 63 72
--                  *   *   *   *
--                       eta
--           0  1  2   3   4   5  6  7  8
--                       eta
--                  *   *   *   *
--     8      8 17 M8  M17 M26 M35 62 71 80
--     7      7 16 M7  M16 M25 M34 61 70 79
--     6      6 15 M6  M15 M24 M33 60 69 78
--     5    * 5 14 M5  M14 M23 M32 59 68 77
--phi  4    * 4 13 M4  M13 M22 M31 58 67 76
--     3    * 3 12 M3  M12 M21 M30 57 66 75
--     2    * 2 11 M2  M11 M20 M29 56 65 74
--     1      1 10 M1  M10 M19 M28 55 64 73
--     0      0  9 M0  M9  M18 M27 54 63 72



  ---INPUT ENERGY MAPPING vs test memory channel number
--              (control information)
--
--     8      8 17  26  35  44  53 62 71 80
--     7      7 16  25  34  43  52 61 70 79
--     6      6 15  24  33  42  51 60 69 78
--     5    * 5 14  23  32  41  50 59 68 77
--phi  4    * 4 13  22  31  40  49 58 67 76
--     3    * 3 12  21  30  39  48 57 66 75
--     2    * 2 11  20  29  38  47 56 65 74
--     1      1 10  19  28  37  46 55 64 73
--     0      0  9  18  27  36  45 54 63 72
--                  *   *   *   *
--                       eta
--           0  1  2   3   4   5  6  7  8
--                       eta
--                  *   *   *   *
--     8      8 17 C45 C45 C45 C45 62 71 80
--     7      7 16 C45 C45 C45 C45 61 70 79
--     6      6 15 C45 C45 C45 C45 60 69 78
--     5    * 5 14 C23 C23 C23 C23 59 68 77
--phi  4    * 4 13 C23 C23 C23 C23 58 67 76
--     3    * 3 12 C23 C23 C23 C23 57 66 75
--     2    * 2 11 C23 C23 C23 C23 56 65 74
--     1      1 10 C01 C01 C01 C01 55 64 73
--     0      0  9 C01 C01 C01 C01 54 63 72
--
```

```
--TAB status register
                          data(0)   => parity_error(0),
                          data(1)   => parity_error(1),
                          data(2)   => parity_error(2),
                          data(3)   => sync_error(0),
                          data(4)   => sync_error(1),
                          data(5)   => sync_error(2),
                          data(6)   => bc_error(0),
                          data(7)   => bc_error(1),
                          data(8)   => bc_error(2),
                          data(9)   => pll_lock_error,
                          data(10)  => mode_ff,
                          data(11)  => '0',
                          data(12)  => '0',
                          data(13)  => '0',
                          data(14)  => sync_alignment_error,
                          data(15)  => pll_locked,
```