Requirements for a multi event time stamp module (TITRIS-2)

General:
The aim with respect to the currently available time stamp VME module TITRIS is to implement a multi event capable time stamp module based on the TRIVA-5 hardware. In the "old" TITRIS module, only a single time stamp register is available, which is always overwritten with the next time stamp input signal. In contrast, the new TITRIS-2 shall be able to collect a predefined number of time stamps internally. The time stamp synchronization and calibration mechanism shall be identical to the old TITRIS. The width of a time stamp will be also 48 bits and the least significant bit shall be 20ns (10ns).

Requirements:

Input parameters:
To use the TITRIS-2 in a flexible way an input parameter (N_EVENT) shall be loaded via VME before enabling time stamping. N_EVENT shall be between 1 and 32 (64, more?).

VME memory / registers:
There will be two banks of memory for time stamps accessible from VME for reading, a control/status register for each bank and a bank switch register. All registers shall be accessible from VME for read and writing.
Each bank of time stamps is a memory space of N_EVENT * 64 bits. Although a time stamp is 48 bit wide, 64 bits should be reserved, to enable the readout of a time stamp with two single cycle a32d32 VME accesses. In addition the two time stamp banks shall be readout in BLT, MBLT (and as test case 2eSST) VME block mode. The upper unused 8 bits could be used as cycling event counter. This would allow also data integrity checks for the block readout modes in data analysis.

Control and status register (see protocol below), exists twice, one for each time stamp bank:
bit 0: bank id: 0 for bank0, 1 for bank1
bit 1: module calibrated: 1, power on: 0
bit 2: module enabled to accept time stamps: 1, power on: 0
bit 3: bank completely empty: 0, not empty: 1, power on: 0
bit 4: error: time stamp signal occurred with full bank before bank switch
bits 5-7: more errors..?
bits 8-15 (23): actual number of time stamps in bank (1-N_EVENT)
bits 24-31 free?, more ideas

Bank switch register (see protocol below):
bit 0: 1: switch bank, 0: do nothing and power on.
bit 1: 1: switch to bank 1, 0: switch to bank 0 and power on. This bit gets only meaningful if bit 0 is 1. Only used because of safety (paranoia)

If required, the described bits can be split and distributed in more registers.

**Protocol:**
After the TITRIS-2 has been calibrated for cable delays and was enabled like the old TITRIS, it accepts time stamp input signals and begins filling bank0 at bank offset 0. At the occurrence of the first time stamp signal, after the module was enabled or after bank switch, the FPGA sets bit 3 of the control – and status register to 1. If the current bank is full and the bank switch signal is 0, the FPGA sets the error bit 4 if a further time stamp signal is detected.

For each time stamp signal the time stamp counter (bits 8-15 (23)) shall be incremented by one. This incrementing shall go on, even if a bank is full and no bank switch has occurred. With this method it is easy find out how many faulty time stamp signals occurred. The VME processor is responsible for setting these time stamp counter to 0 before bank switch.

On the occurrence of the bank switch bit the FPGA switches to bank1 and sets bit 0 of the bank switch register to 0. Then the FPGA starts accepting timestamps at offset 0 of bank1. Before filling bank1, the FPGA can check if bit 3 of bank1 is 0 (but this has already been checked by the VME processor.) On the occurrence of the first time stamp signal after bank switch the FPGA sets bit 3 of the control - and status register of bank1 to 1.

After the next bank switch signal, the protocol goes on in the same way and time stamps are now filled again into bank0.

A bank switch is always forced by the VME processor by setting bit 0 to 1 and bit 1 (to either 0 or 1, depending on the bank to switch to) of the bank switch register simultaneously. The current bank needs not to be full, the bank switch can occur at any time. This is useful, if multi event buffers need to be emptied just after the spill, in order to have empty buffers at begin of the spill.

As mentioned the VME processor is responsible to make the bank switch. This will be done in a state, were dead time is kept on and no further triggers or time stamps shall occur. Just before the VME processor enforces the bank switch, it checks that bit 3 of the bank to switch to is 0 (next bank completely empty). Just before the bank switch the VME processors resets the time stamp counter (bits 8-15 (23)). After bank switching, the VME processor releases the local dead time and then reads the previous, still full (or partially filled) bank of time stamps. After the readout is completed, the VME processor sets bit 3 to 0. After a certain number of triggers/time stamps the dead time will be raised by the trigger system and the next bank switch - and readout sequence will be started.