

Known malfunctions to fix.

1. The readout hangs waiting for the first event from the CPLD and also after a multiple of 28560 triggers. It continues by itself after “some” minutes. This time is about the same in both cases.
2. If an MCM is empty the system hangs. Empty MCM may happen when running with zero suppression. It appears like the SRU waits for data that is never coming. One solution could be to always send the channel header word.
3. We can read only one MCM and then two channels at a time if full time sweep of 1000 samples, or 135 samples of 16 channels. This should be changed to read one full MCM of: 1000 time samples of 10 bit words for 16 channels and 8 SALTRO chips = 1280000 bits. These words are rearranged in the CPLD to be 3 10 bit words and 2 status bit = a 32 bit word.
4. It happens sometimes that the first channel read is lost. Do not know why. Probably is it read by the CPLD from the SALTRO, but then lost on the way to the computer.
5. The system seems to be sensitive to external disturbances. Then the readout hangs waiting for an event to be sent to the SRU from the CPLD. If one sends a command to resync the data line one can restart data taking. In some cases one has to reset the SRU. In this case one can still send a command to the CPLD (e.g. write a register) but no reply is sent by the CPLD. On the new LV board that we will test soon we have introduced LVDS transceivers in the CPLD end of the four DTC lines. Previously the LVDS signals went directly into the CPLD. Let us see if this solves the instability problem. **Update aug17: this problem may be solved after introducing transievers and RJ45 connector for shielded cable on the LV card . Need long operation experience under different conditions to verify but so far we have not been able to provoke it to happen..**

New implementations:

6. The data transfer UDP driver to the computer on the SRU is a black box (the control UDP communication is part of the SRU firmware from ALICE). This is a binary module from the TOTEM experiment at CERN. We were not allowed to get the source code. This we want to be replaced by our own UDP driver.
7. We can read only one MCM, one SRU can handle up 40 MCM. (maybe 36 when we use the LVDS coax signals). We need at least to have 32 MCM and preferentially 40 (36). The master student planned to use the onboard SRU 2Gb memory module for this. But this was never implemented.
8. The SALTRO sampling clock of 20 MHz is derived from the clock on the DTC link.

In our discussions we have said that a phase difference between sampling clocks on an MCM and between MCMs on the same SRU of 1 ns can be tolerated as long as the phase differences are stable. If longer we have to think deeper.

9. When running up to three SRU they must be operating with the same 40MHz clock and trigger. We have decided to feed these signals via the LVDS coax IO. **Update aug17: We have decided to use the TLU for clock, trigger, busy and event synchronization (event number) according the scheme named "Synchronous Mode With Trigger Number on page 24-25 in the TLU documentation.**
10. One of the lemo NIM inputs to the SRU shall receive a long signal which controls whether the preamp-shaper shall be powered up or not. Timing is not critical for this signal It will only change state on a millisecond time scale. When this signal change state two bits in a register on the CPLD should be changed, the other bits may not be changed in this register. One bit controls the preamp operation (shutdown pin) and one the ADC operation (biasctrl pin). This register exists and the action works, (the current drops from 3.2A to 0.4A when switching off preamp and ADC, running with 20 MHz sampling clock). The other bits in this register control other functions in the SALTRO. They are common to all SALTRO on the MCM.
11. **Further current reduction can be done by switching off the ADC clocks and the readout clocks. The switching of clocks is currently not implemented. Note that it takes at least 100 microseconds for the SALTRO to become fully operational after a switch on, i.e. no triggers may be issued during this time.**

NB.

Item 7 can not be tested with many MCMs until we have fabricated more of them. To do so we must test the chips and this can not be done until we can handle one MCM in a routine way without hangups. To produce more MCMs will take quite long, probably not less than 6 months.

Priorities.

There are several possible reasons for not being able to finish all the items in this list or maybe to postpone some. It is important to plan the work with intermediate goals where the system has reached an improved level of functionality/operability. A tentative list of such intermediate steps are.

- A. Fix the known malfunctions, 1-5. Task 6, a new UDP, may be part of this task
Expected operability – We can read out one MCM (and the protoMCM) running internal clock and trigger

With the system in this shape we could go ahead and test the chips and assemble all MCMs means that the system is not available for further work (we may be able to borrow an extra SRU)

- B. Add item 7,
Operational status – We can run many (40) MCMs on one SRU with internal clock and trigger.
- C. Add item 9, TLU
Operational status – We can run several SRUs. With external clock, trigger, busy and event numbers. Event number has lowest prio.
- D. Add 10 pwr on/off
Operational status – we reduce average pwr a lot. By pwr pulsing on millisecond timescale
- E. Add 11. Stop sampling and readout clocks.
Operational status – we can reduce average power further. Now we are probably beyond what we can reasonably achieve.

Item 8 the spread of clock phases may enter in several of the goals.