

SRS users group

proposal for activities

please join

use our WEB page

<https://espace.cern.ch/rd51-wg5/default.aspx>

February 24, 2010, RD51 Miniweek

SRS User's Group

Chips and hybrids

The 1st RD51 frontend hybrid is based on the APV25 chip with analogue readout, powered by HDMI cables. This subgroup consists of the ones who think of new applications and they should recommend new or different RO chips for more RD51 applications. Related issues are: digital versus analogue readout and trigger or no trigger.

SRS-UG

Architectures

- The 1st RD51 systems use an external hardware trigger signal to copy chip data into an intermediate event buffer from where formatted events are transmitted via IP packets to the DAQ computer. However SRS can be used for different trigger and readout architectures. This subgroup should therefore address more optimized architectures, like N-level trigger, trigger-less, shared memory, etc

SRS-UG

Packaging and Commissioning

The 1st SRS applications are based on chip carriers that are connected via long, powered HDMI cables to digitizer/FECs which are housed in a 6Ux220 Euro-chassis and get powered via standard ATX supplies. Up to 36 FEC's are connected via CAT6 network cables to a single Scalable Readout Unit (SRU). The D.T.C. link protocol between FECs and SRU consists of two phases: 1.) Data & Trigger 2.) Controls. The SRU is mastered by control and DAQ software running on the DAQ computer(s) and the Trigger signals are to be provided to LVDS or NIM inputs on the SRU.

Users of the pilot SRS applications should participate in setting up a first template for reproducing SRS systems. This includes component procurement, packaging, powering, connectivity, reliability and qualification test procedures.

SRS-UG

Slow Controls

The slow controls is a tree of addresses that are mastered by the controls software on the DAQ computer. Each controls branch is associated with a dedicated port which controls a group of Ethernet addresses. The control targets are local read-write addresses which implemented in the hardware units, like SRU, FEC, Digitizer or Extension card, Chips. The intermediate transport of control data is transparent to the user and to be handled by the card resident firmware. The mapping between physical addresses and logical addresses inside each port is based on a configuration file. Some configuration files are static, like for SRU or FEC. Others depend on the choices taken, in particular choice of the readout chip. This subgroup should provide the first basic configuration files as templates for future users.

SRS-UG

Triggering

- The 1st SRS applications work with a precisely timed trigger signal that is provided by external trigger logic to one of the signal inputs of the SRU, or directly to the corresponding FEC card input. This trigger signal initiates event-synchronized transfer of data from the frontend chip to the FEC card buffer and defines an event number to associated with the chip data. In principle it is possible to allow for a second-level trigger signal to arrive at some latency after the first level. A second level signal allows to enable or discard transfer of accumulated events that can be processed at the FEC buffer level. In case of a single level trigger, all data received from the readout chips get transferred to the DAQ computer. A second level trigger allows for online processing and discarding of (empty) events. The choice depends on the bandwidth limit into the DAQ computer which is targeted as 10 Gbit/s per SRU (but only 1 Gbit/s for the first SRU prototypes in 2010). This subgroup should work on dataflow and bandwidth with simulations based on measured SRS parameters.

SRS-UG

Data formats

The event data that are received from frontend chips must be uniquely and safely formatted before being wrapped in IP systems. Initially, we should “lend” a format from some detector that uses DATE for readout. This subgroup should however identify a common SRS data format that works with all packets that are transmitted and unpacked in a destination buffer on the DAQ computer. Ideally all data belonging to the same event-number should get routed and unpacked into a the same destination buffer (implicit event-building). In order to get first applications bootstrapped the first SRS systems will use a format that already exists. The APV data will be digitized into event blocks of 12 bit samples, preceeded by a header with wordcount, timestamp etc . However in order to support different chips for the final system, SRS needs a suitable format that can be transmitted over IP packets without overhead penalty, hence it should include Multi-event packing for small payloads a la LHCb. Further it would be very desirable that data include a data type identifier in their header such that the unpacker software can automatically switch to the correct unpacker mode (the alternative is to have 1 unpacker code per chip and per detector) This subgroup should consist of persons having online and offline experience in experiments and recommend a Data format that eventually allows to combine different detectors and different chips without changing the unpacker.

SRS-WG

Data Acquisition and offline

- The default Data Acquisition system for SRS is the ALICE DATE software <http://ph-dep-aid.web.cern.ch/ph-dep-aid/> over Gigabit Ethernet. As presented in previous WG5 meetings, the porting of DATE to the SRS is well advanced. The use of DATE by RD51 is free however requires to sign and comply to the MoU rules of the Alice DAQ team. DATE works with raw and with Root file formats, the latter being normally preferred for offline data evaluation based on standard methods. The DATE control panel is easy to handle and to learn also by non software experts. The stability of DATE has proven to be excellent and does not require an on call expert, hence users of SRS with DATE profit from a stable software environment. In the simplest case, detectors up to 10K channels require no SRU . Up to 4 FEC cards can get directly connected to 4 GBE Ethernet plugs of a the DATE computer. Systems up to 50 k channels can use a single server PC if one SRU is used to connect up to 36 FEC cards, however the CPU load situation may call for the use of several PC's like it is the case in the ALICE experiment. This subgroup should be composed of both online and offline experts that are concerned about SRS applications, but also of future users and shifters

SRS-UG

Firmware algorithms

The programmable hardware (Virtex-5 FPGA) in each FEC card consists of both system-defined and user-defined modules, written by firmware developers, i.e. typically students with VHDL or Verilog expertise. In the simplest case, the chip data assembled in an event buffer of the FEC is merely formatted by a header and shipped to the DATE computer. However if online processing (peak amplitude, pedestal averaging/ subtraction etc) is needed, a user-defined firmware algorithm is needed. This algorithm must comply with the allowed trigger latency. Before embarking however on such algorithms, one should carefully trade off available bandwidth and CPU power against hardware algorithms. If enough bandwidth and CPU is available, it is preferable to transfer the data to the DAQ computer(s) and perform the algorithms there. Zero-suppression algorithms are user-defined firmware that depend on the readout chip. Once developed for one particular chip, it can and should be shared with all Rd51 users of the SRS. Other parts of the firmware are serial line protocol conversions like I2C or SPI, JTAG etc. This subgroup should consist of a mix of physicists, hardware and software engineers in order to decide which algorithms are needed and to write the proper Firmware code.