Correlator control and implementation meeting, July 26, 2012, Arpad's room

Present: Jonathan, Jintao, Harro, Arpad

action items:

Jonathan: test fix of Erlang performance problems reading UDP packets
not done at full blast, remains, presumably fixed

Jonathan + Salvatore: figure out how to make packet spacing adjustable
not done, low priority right now but may become high, remains

Jonathan: ask Paul to order a few 8GB DDR3 modules
not done

Des + Harro: come up with list of the parts of software that are done, need work, need writing, and an indication of time involved
partly done, Des made a list, Harro has not yet taken a good look. Action remains, after the holidays a more detailed plan will have to result from this list

Jonathan explains the ongoing work on the delay module, change so that fractional delay in inserted once per sample. Fractional delay (using lookup table) works well in modelsim, but not in hardware! Inserting different delays changes nothing. Currently trying to pin down where it goes wrong, most likely a problem with timing. Hope is that it will be sorted before the end of the week.

Next step will be feeding it data continuously, 8 seconds worth of data. Hopefully done before vacation (two weeks).

Jintao is simulating delay module, looking at the synchronisation of data and coefficients.

Harro is working on how to send data to the UB. Lengthy discussion follows about timing of data sources, correlator, sending of data and coefficients, recorded, real time, gaps in data and how to handle them.

The conclusions, as I (Arpad) see it.

The correlator keeps track of the amount of data left in the memory of the front node, and when there is not enough left will wait for the memory to

fill up again. This could be used as a trigger to warn the control system that new data has to be sent.
Harro points out that one single signal sent from UB to control system could get lost, and that directly polling the fullness of the memory buffers would be better than repeatedly sending signals and would give additional useful info (for example on what data sources are lagging). Jonathan sees no problem in inserting these values into registers for the control system to poll.

The cornerturning routine right now assumes all data to be valid. Here a check will have to be built in to drop VDIF packets with (partly) invalid data.

Probably the best way to feed recorded data to the UB will be to do cornerturning on the Mark5s, send this data to an AriBox, where a reader program sends data of a specific timerange to the UB, after getting a command from the control system. This reader program presumably will need to be able to extract the time information from the headers. Sending data to the correlator and correlating has to start at a certain time, so one way or another this time must be known and associated with a chunk of data. After that, in principle, there should be only dead reckoning. The advantage of having the reader functionality separate from the cornerturning is that it can be used when we finally get to process real VDIF data.

Harro and Des have been contemplating an Erlang rotclock, to serve as a kind of pps. This could be used by the control system to synchronise sending data and coefficients.

It is clear that something is needed to feed data to the UB for testing leading up to "operability". Clearly this something will at first assume that no packets are missing or invalid and in time order and whatnot. However, it seems to me quite important that the feedback mechanism gets built in as soon as possible.

After all this we discussed the little document I have been preparing (which Harro did not see yet), trying to outline various options for the correlator architecture. After discussions with Jonathan this will be re-written and distributed among the whole group. The intention of the document is to have something to show for example to the other JMT members, who by and large have no idea what is going on. With the UniBoard I mean!