

The serial stream is converted into 10-bit words by means of the serial-to-parallel converter. The clock for this word-stream is generated by means of a divide-by-10. The word-rate is 66.5 kHz, of course.

The framer recognises the synchronisation words and works in close relation with the serial-to-parallel converter. For example, the divide-by-10 has to be synchronised to make sure that the 10 output bits all belong to the same word. (See also "Transmission of Digital Data".) The framer also drives the in-sync/out-sync LED's.

This completes the block diagram of the actual decoder.

For HRPT and CHRPT, a channel selector is added. Selecting channels by hardware instead of software has the advantage that slower PC's can be used to record at least a few channels. Note that with CHRPT, the word rate is close to what most parallel ports can handle (regardless how fast the PC is).

With the current software at least two channels have to be de-selected to make the recording of CHRPT possible. With HRPT, recording of all channels is no problem.

The speed of the data stream coming from the decoder is dictated by the satellite, but now and then the PC can't read data because it has to do other things (it has to save the collected data on disc from time to time). This problem is solved by adding a so-called elastic store between the decoder and the parallel port. The decoder writes data continuously into the store, which the PC reads in chunks. As long as the average read-speed is equal to the write-speed no data are lost. This is shown in

Fig.6. The circle visualises the actual store and the two arrows point to the write and read locations.

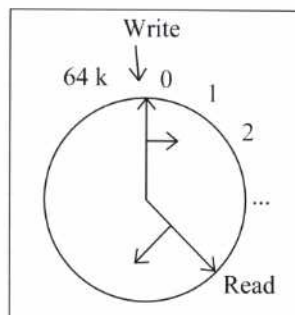


Fig.6 - The Elastic Store

The decoder controls the "write-arrow" and starts by writing data at address 0 of the store, filling it until the maximum address. Then it jumps to address 0 again, and so on.

The PC controls the "read-arrow" in such a way that read and write arrows never pass each other. If, for example, the PC reads too fast, the read arrow will come closer to the write arrow. Just before the arrows cross each other the elastic store sends a "buffer empty" signal to the PC, which stops the software reading until there is new data available. If the PC is too slow then the write arrow will pass the read arrow, and data will be lost.