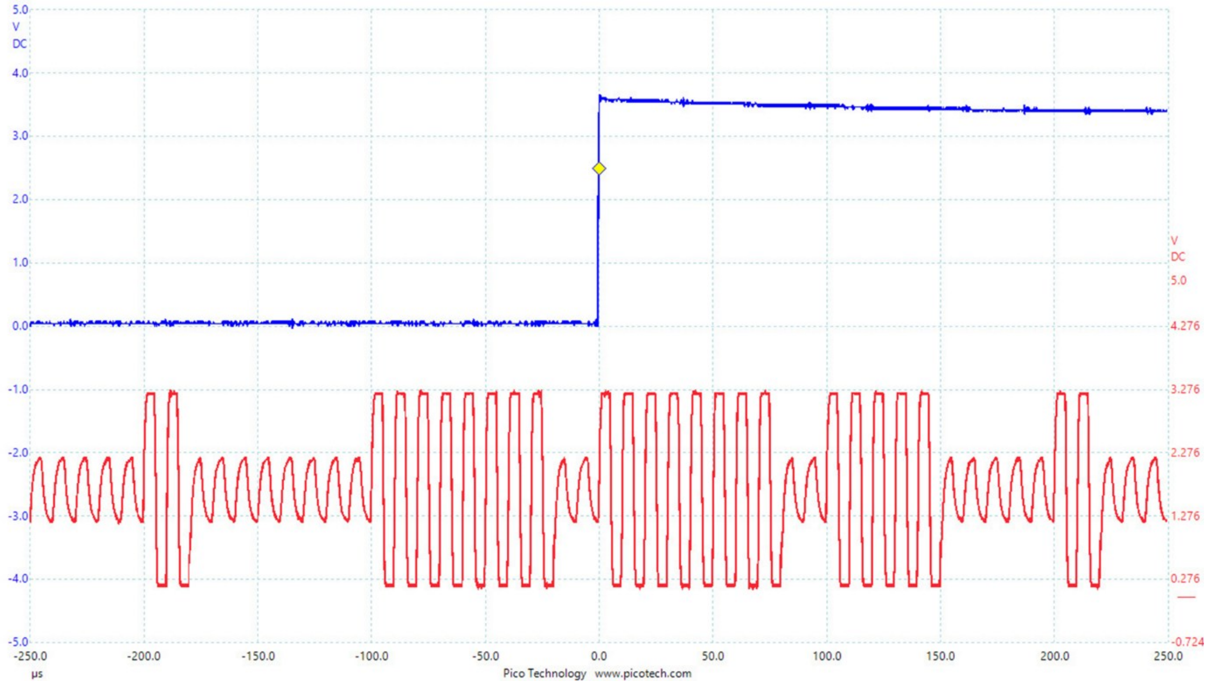


IRIG-G extension to IRIG core

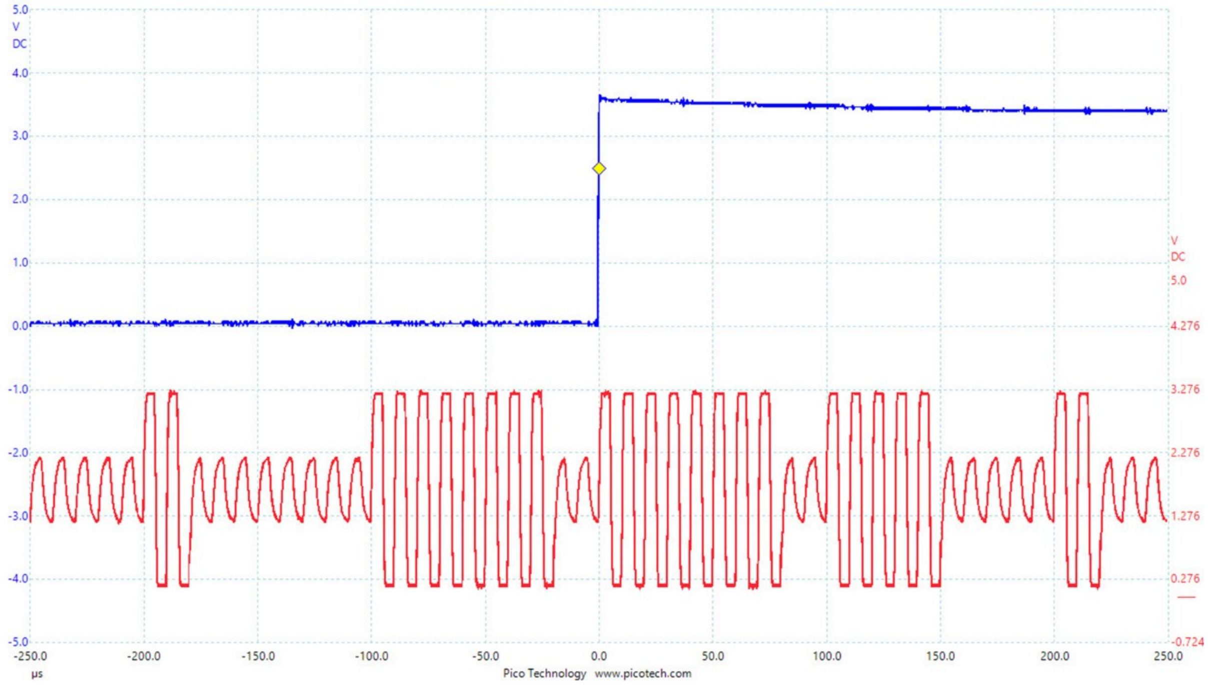
17. February 2022

NetTimeLogic recently added IRIG-G support to our IRIG core.

IRIG-G is very similar to IRIG-B but rather than transferring 1 frame per second as for IRIG-B, IRIG-G transfers 100 frames per second (so every 10ms a new frame) and has a slightly different frame format since now the concept of milliseconds is added to the frame.

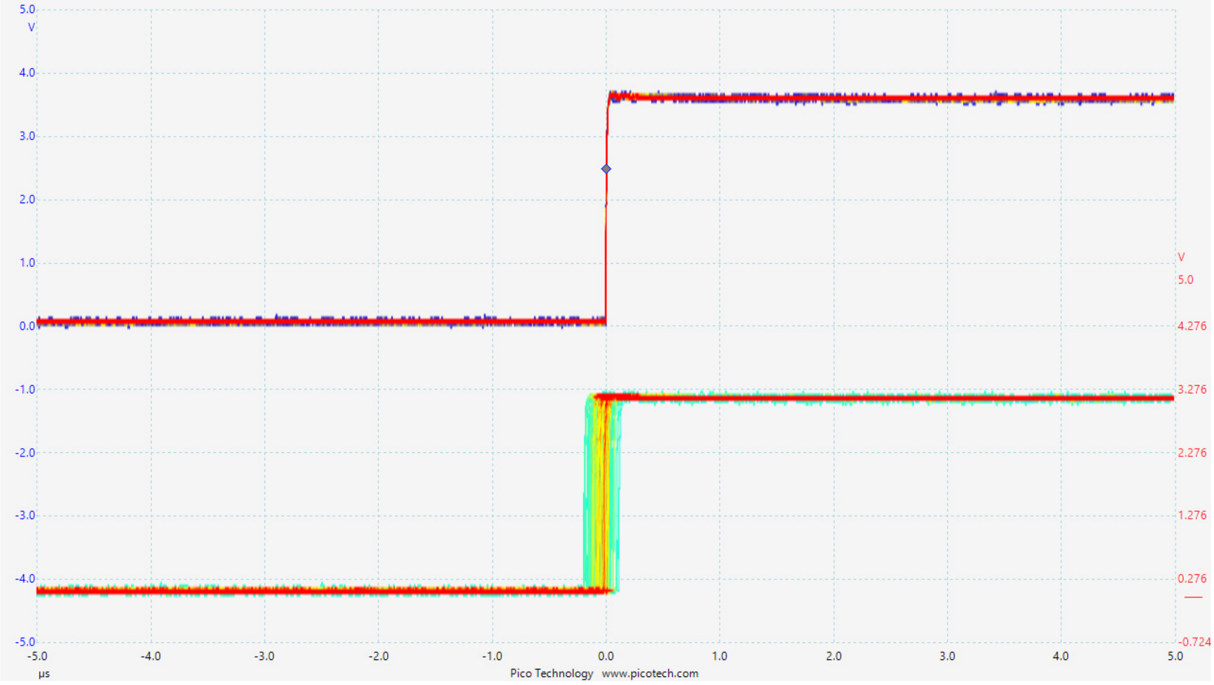
For DCLS this was not really a big challenge since this just means we have now a bit rate of 10kpps rather than 100pps and we send the frame with the different Format with the millisecond information (which is only a fraction of the nanosecond resolution we have in the FPGA). On the receive side we still stick to the 1/s calculation rate for adjustments to reduce the jitter and have a more accurate drift measurement (error/s vs error/10ms). This is simply done by just using every 100th frame where the milliseconds are 0 (only 10s and 100s of milliseconds are transferred every 10ms, so for sure a frame with 0ms will be there every second).

For AM modulated IRIG-G it gets interesting since the carrier frequency for IRIG-G is 100kHz. That doesn't sound like a lot since it is still a 10us period. Well, in our previous post we described how to do high accuracy IRIG AM with DACs and ADCs using some cheap ADCs and DACs with a sampling rate of max 1MSPS. This turned out to be rather tricky. If we have a sampling time of 1us (1MSPS) and a period to generate of 10us this will end up with 10 samples per period, which is not a lot. First we created a "nice" sine wave on the DAC with rather big steps in between since we only have 10 points to draw a sine wave. This looked nice, almost like a real sine wave, but on the receive side our algorithm with detecting the zero crossing and ¼ of the sine wave check for the amplitude turned out to be a bit optimistic. Between the zero crossing and the high peak we only have 1-2 samples if we are lucky and this is not good by means of oversampling, so we often ended up to sample already the falling edge of the sine wave after the first quarter resulting in the case that the receiver interpreted this as a 0 rather than a 1. So we decided to make the sine wave less a sine wave for these low oversampling rates (up to 10) but more a square wave, so going to +1 at the crossing and after half a period to -1 (or 0.3 and -0.3 for a 0). This gives us more time to still sample the high period after the zero crossing detection. Due to the DACs behavior (settling time) this looks more or less still like a sine wave but with a bit longer peak time and shorter rise/fall time.



IRIG-G Sine wave

With this little tweak we got IRIG-G AM running with the same cheap ADC/DAC we used before for IRIG-B. And with an even higher accuracy than with IRIG-B, since the 0 crossing is now much more narrow.



Synchronization accuracy

So the conclusion is that IRIG-G is easy to do in an FPGA when it comes to DCLS and it is still possible to do also IRIG AM with cheap ADC/DACs with a "low" (1MSPS) sampling rate . However if you get your hands on DACs and especially ADCs with a bit higher sampling rates (e.g. 2 or better 5MSPS) you are always on the save side.