Hi Tony,

The 20 code step still turned out too low and I switched to 256_{10} = 100h steps.
The picture below is a one signal period with related codes marks.

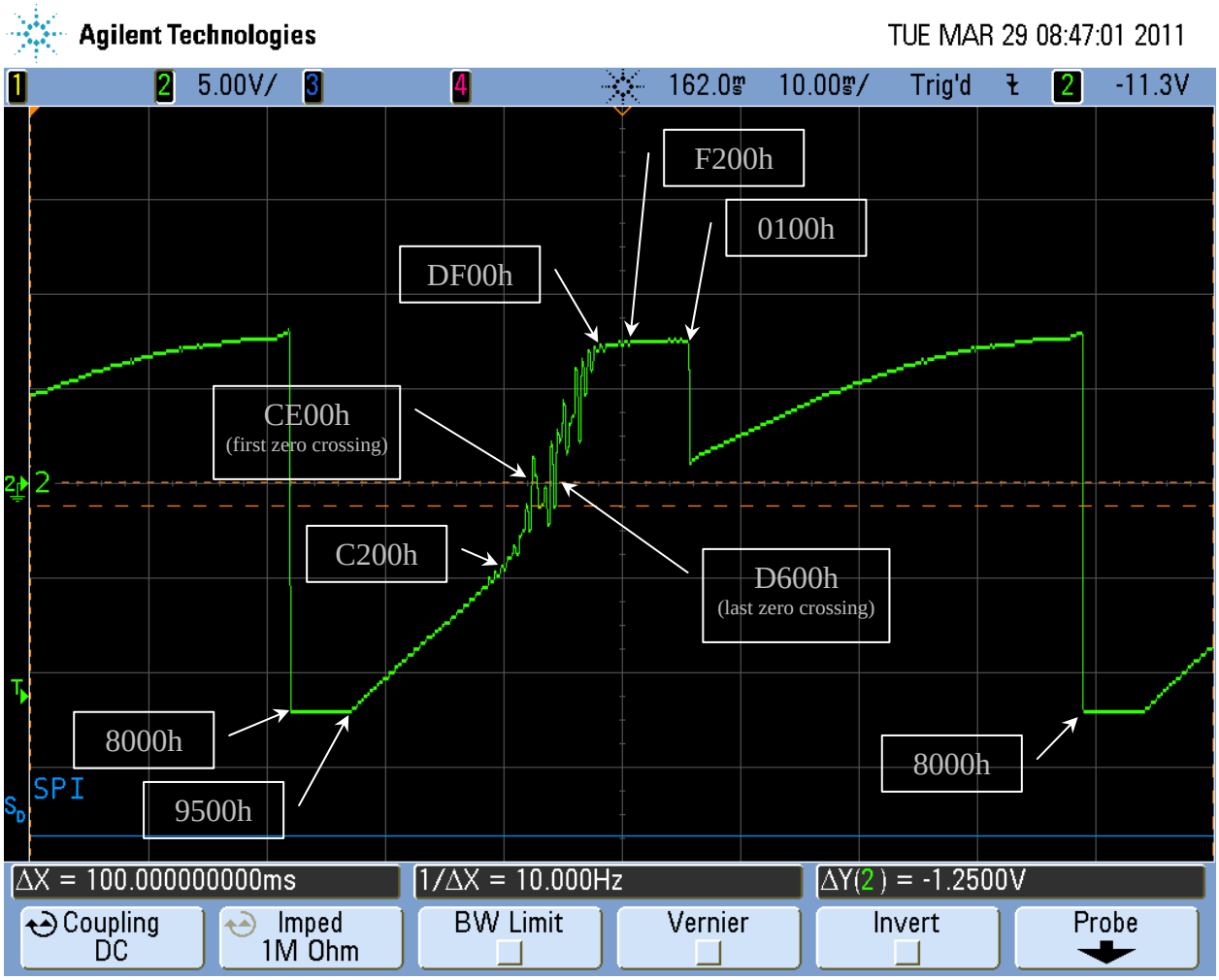


Figure 1. Output waveform.

Figure 2. Reference voltage (yellow). AC mode, 20mV/div

Figure 2 shows reference voltage noise. It looks like there is no relationship between the output signal bump and reference.
Next picture shows reference voltage on the full output voltage period. It looks stable.

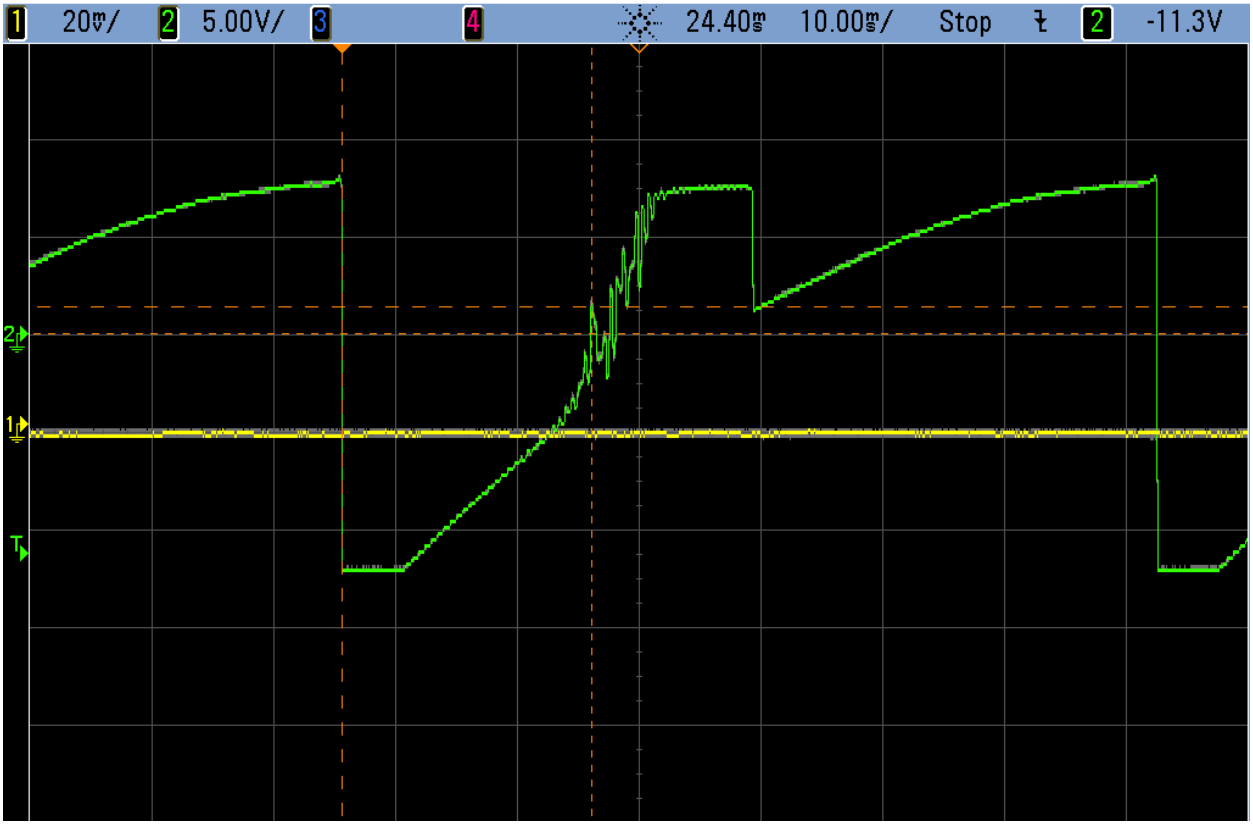


Figure 3. Reference voltage (yellow) during output signal period. AC mode 20mV/div. Infinity persistence.

I have found the source of the bumps in the next two pictures. The bumps are slowly moving over the OffsetA and OffsetB outputs. I have not found such noise in the ±12V sources (figure 5).

The figure 7 shows matching bumps on the offset DAC and on the output. Figures 6 and 7 show the same time position in the output. This position is in the screen center on the figure 3.

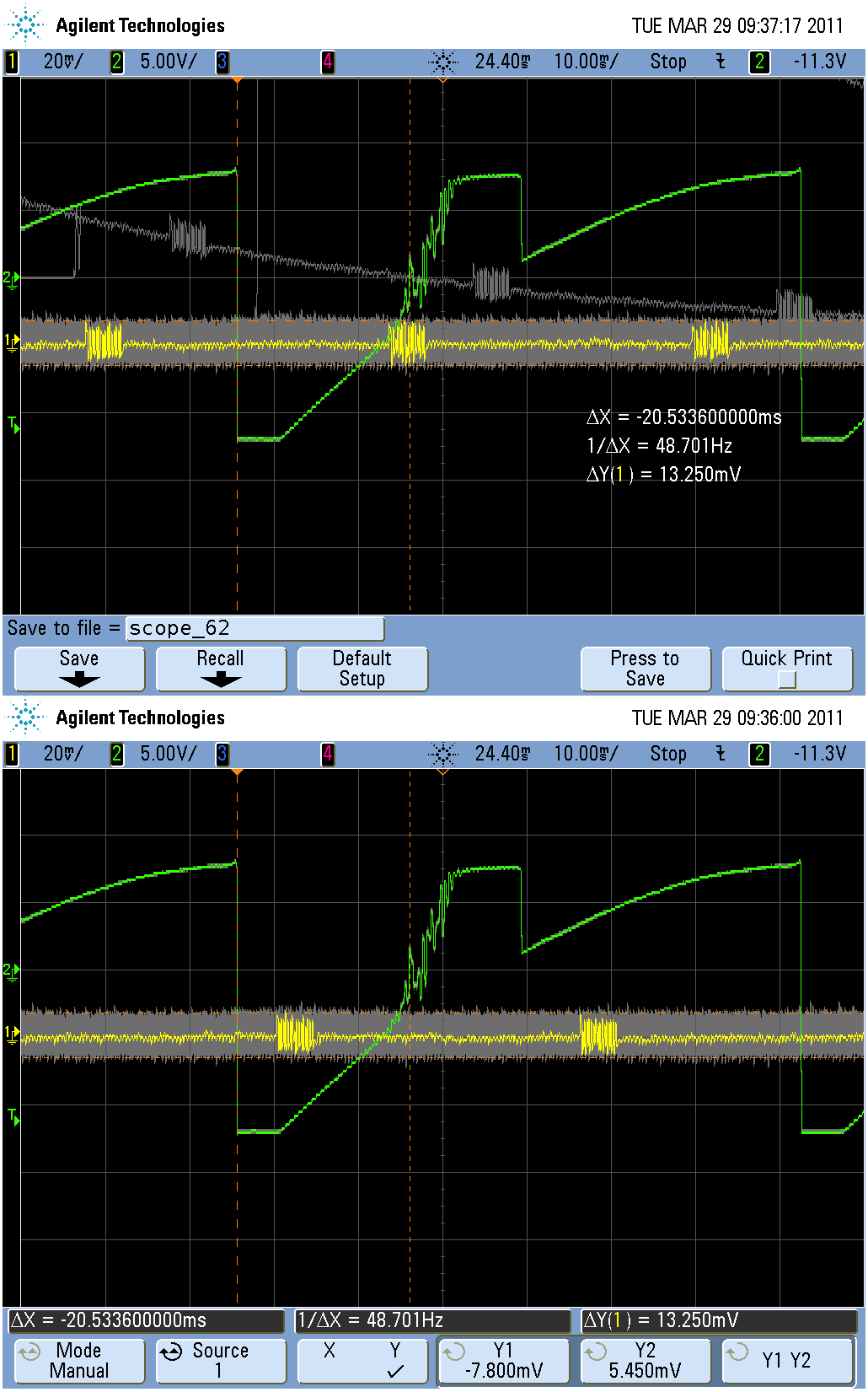


Figure 4. OffsetA (upper) and OffsetB (lower) during output signal period. AC mode 20mV/div. Infinity persistence.

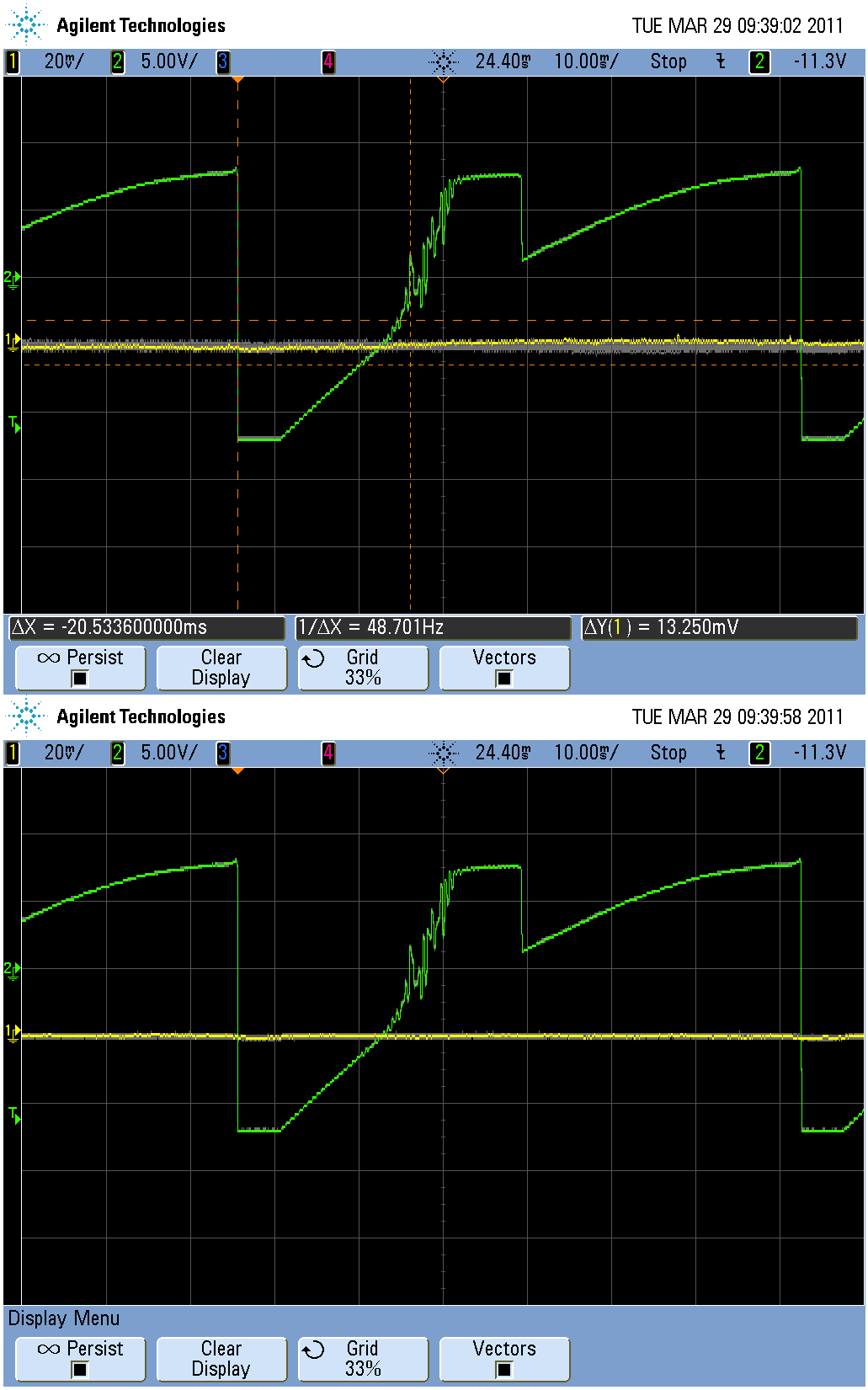


Figure 5. +12V (upper) and -12V (lower) during output signal period. AC mode 20mV/div. Infinity persistence.

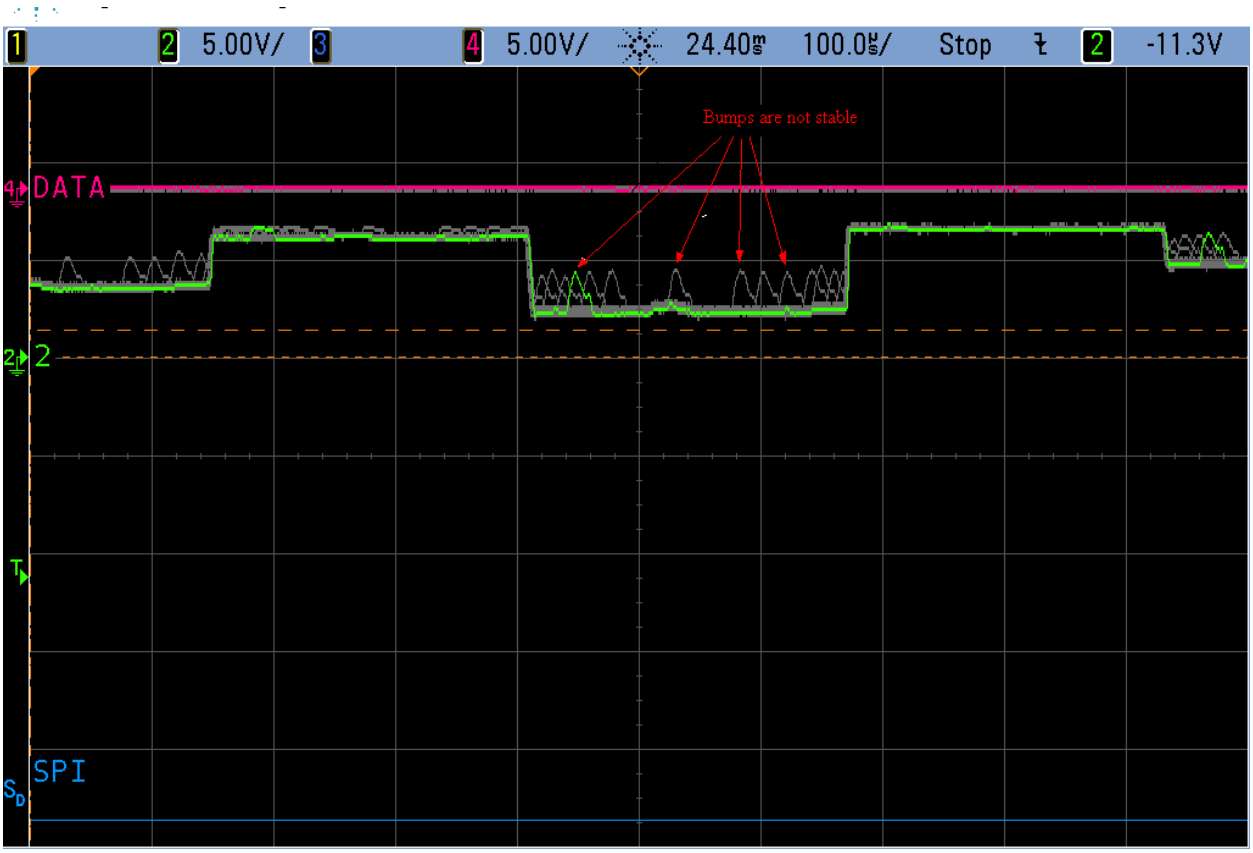


Figure 6. The bumps are moves

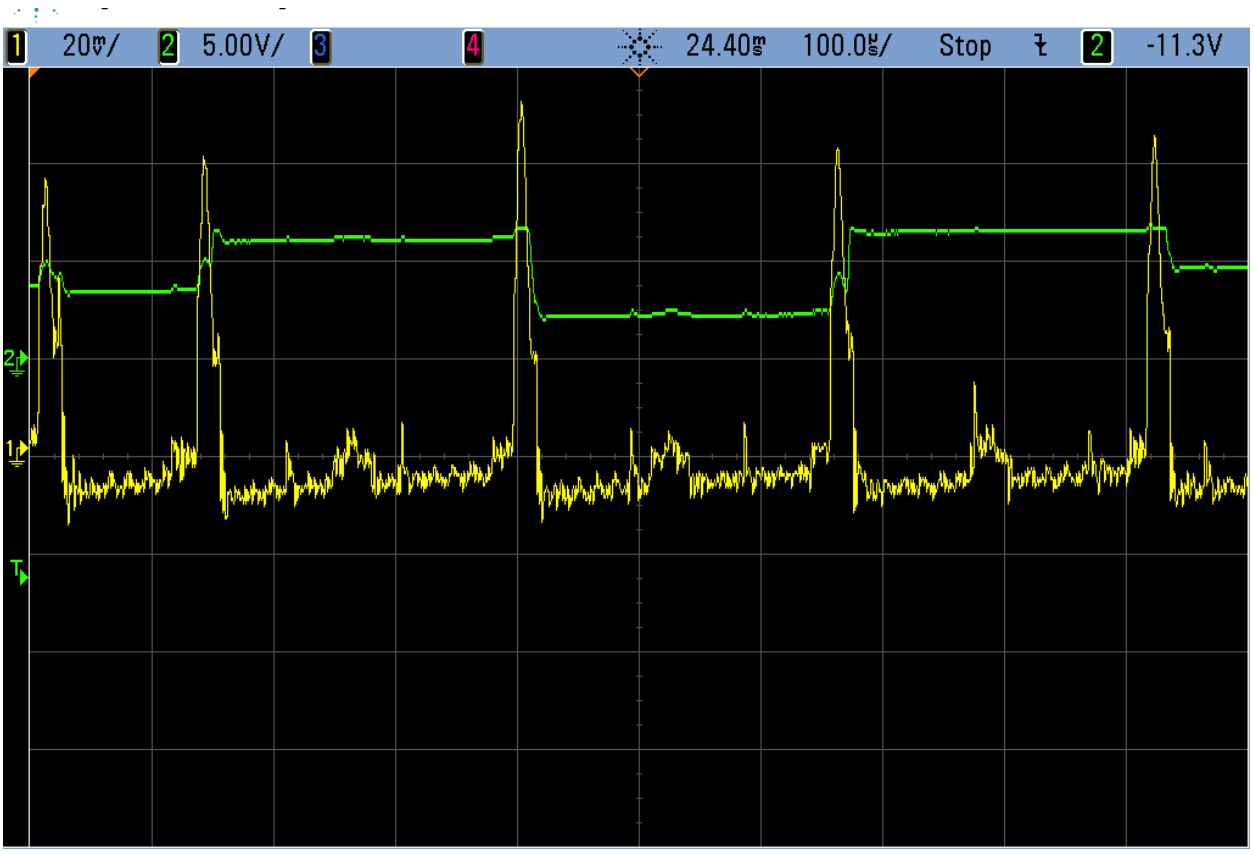


Figure 7. The bumps in the output (green) and offset (yellow)

Next picture shows outputs 2, 5, 6, 7 in broadcast mode.

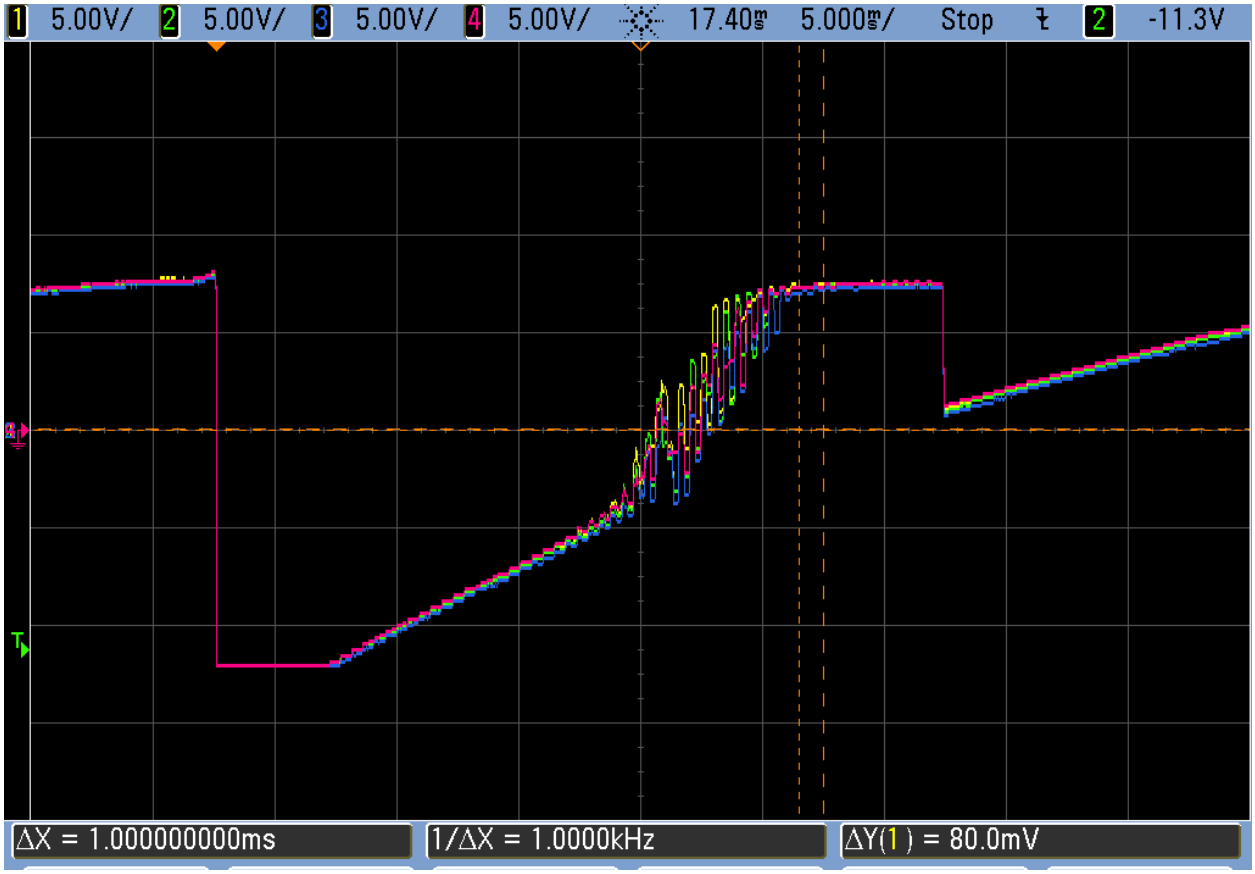


Figure 8. Outputs 2, 5, 6 and 7

And finally I go back to the 16 code step and reduce codes range to the 0000h … 9000h.

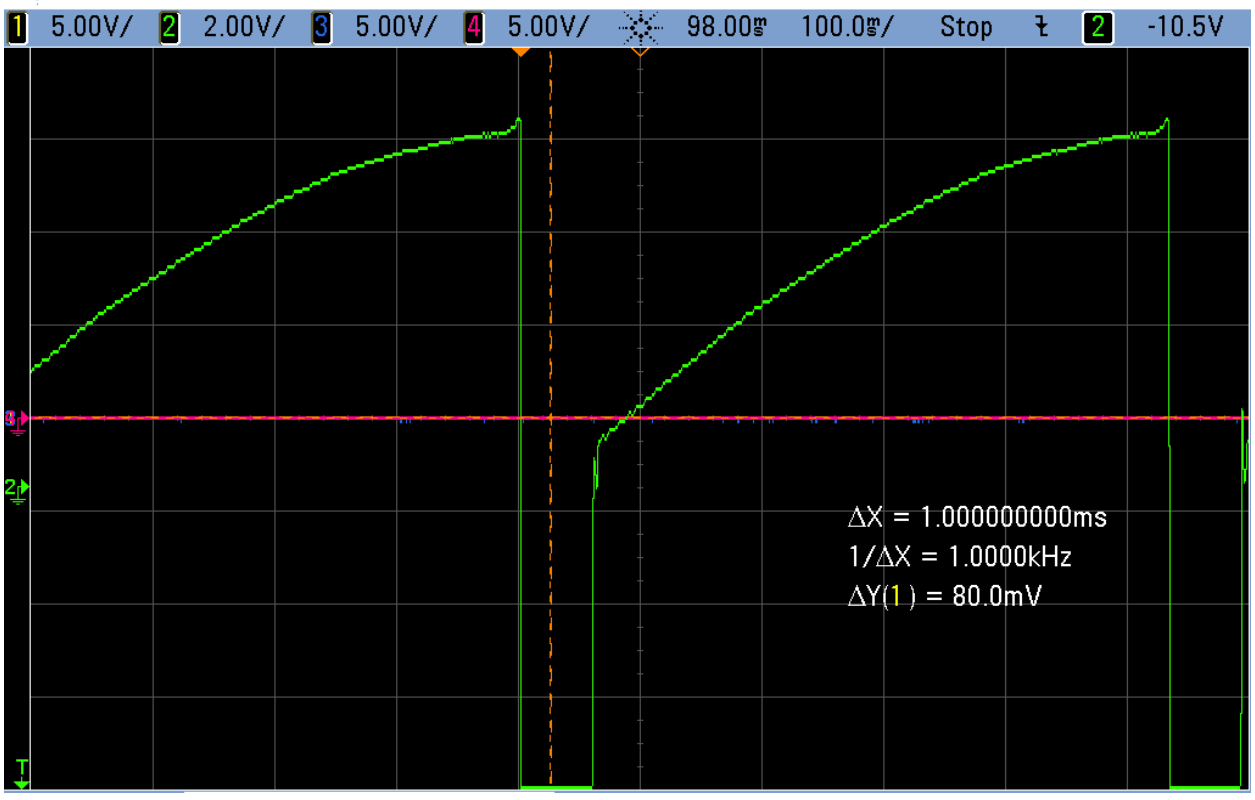


Figure 9. Output waveform for codes 0000h … 9000h

I have found an interesting behavior in the rising area of the signal shape.

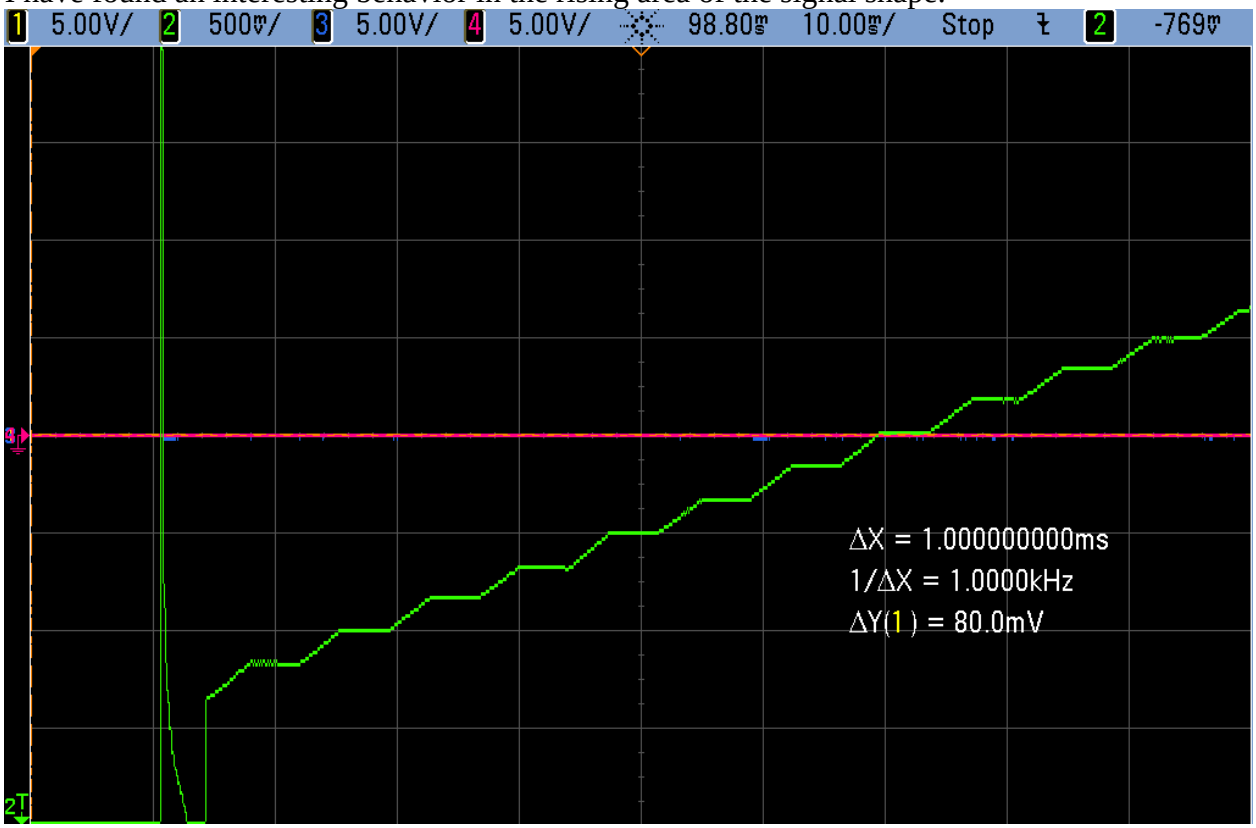


Figure 10.

The output is increased linear when the high byte is odd and does not change (or changes very slightly) when the high byte is even.

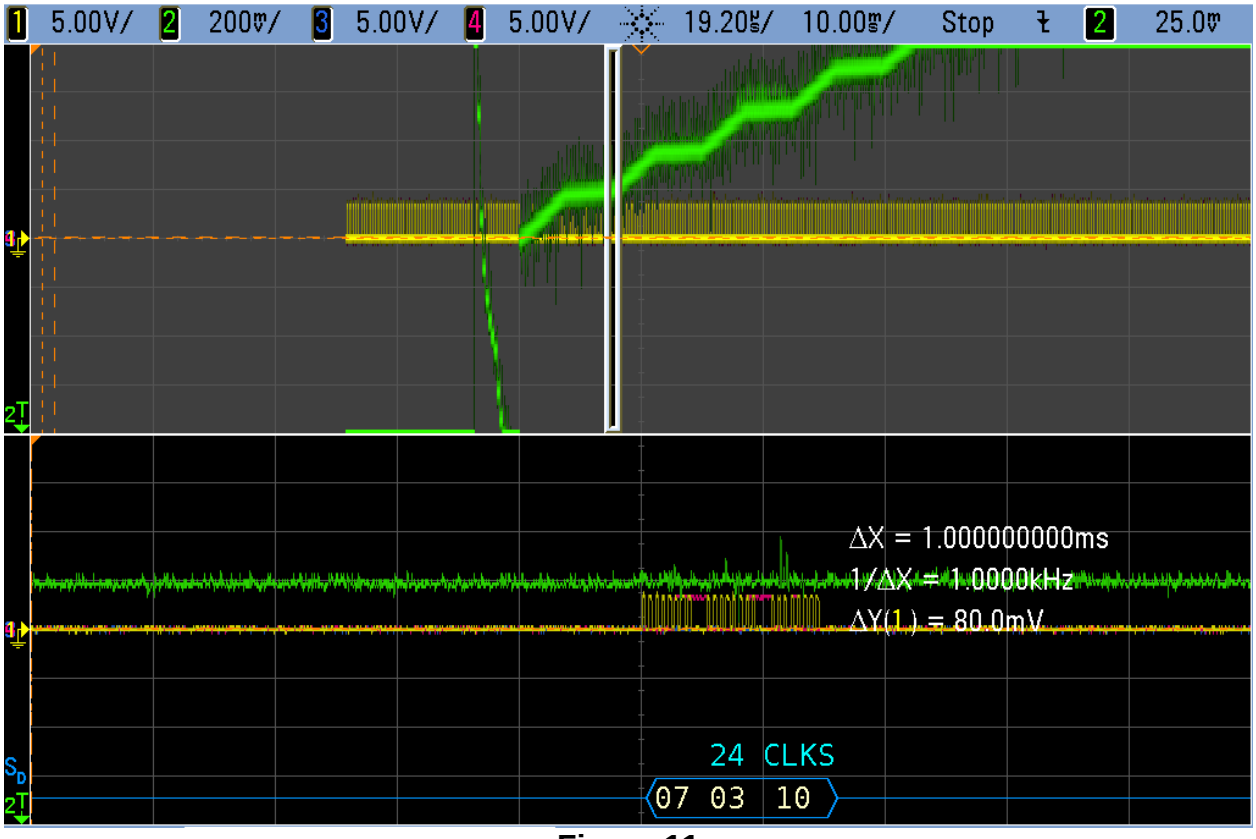
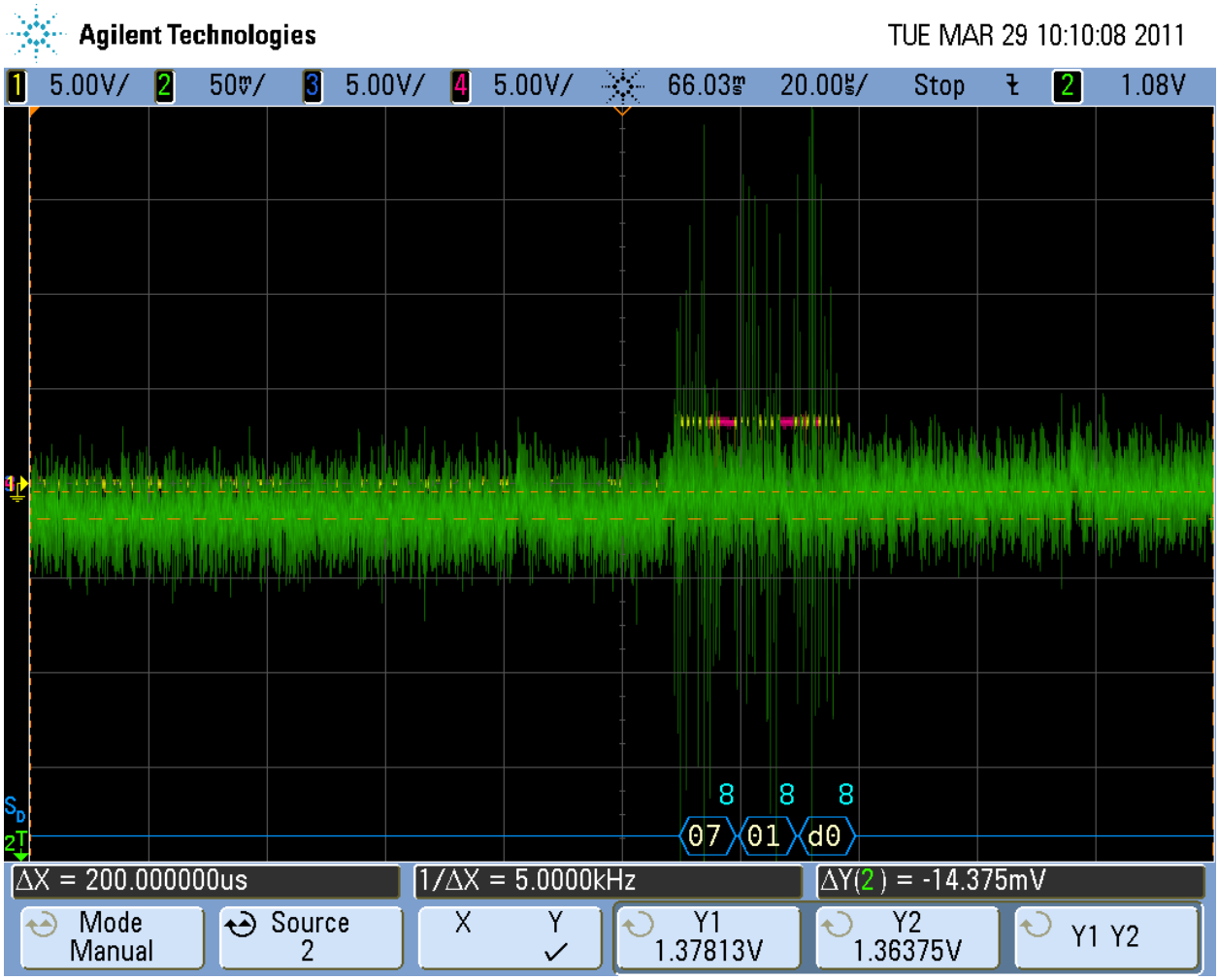


Figure 11.

The last picture shows a 16 code step. It is invisible due to the noise.



I hope this information will be useful and help solve the problem.

Regards,

Roman Ermakov.