

Re: Two questions about USB CRC

Source: <http://sci.tech-archive.net/Archive/sci.math/2008-08/msg01183.html>

- *From:* longjiang_yu@xxxxxxxxxxxxx
 - *Date:* Sun, 10 Aug 2008 18:56:47 -0700 (PDT)
-

On 8A0å, H4ö11 , riderofgiraffes <mathforum..org...@xxxxxxxxxxxxxxx> wrote:

"Without the remainder bit-wise inverted, trailing zeros at the end of a packet could not be detected as data transmission errors."

As you say, two streams with trailing zeros at the end could have the same encoding results. For example, two streams, one is 10000000, and one is 100000000000000000. The shift register is preloaded with all 1s. The result of the former is 10000. The result of the latter is 11100. I've got the different results. Could you please share your idea about it with me?

When the transmitter computes the CRC it is appended to the data stream. If you use an uninverted remainder then the result of the check on reception should be 0. You don't compute the CRC of only the payload and then compare, you simply compute the CRC of the entire data stream, including the CRC. That way you can compare against a constant, which is hardware more efficient.

If on transmission there are trailing 0s (after the appended CRC) these won't be detected by the decoding.

If you invert the remainder before appending then the final answer will no longer be 0, but should be the same on all transmissions. You can precalculate this as described in the paper, or you can simply run a few blocks of data through the system and see that the result is always the same.

Try it. Try several examples. Notice how with the

Re: Two questions about USB CRC

remainder uninverted, computing the CRC of the entire stream is zero. Then append the inverted CRC and see that the result is always the same.

Thank you for your answer. I'm very appreciated for your detailed explanations.

Let's go back to the first topic.

As you say, the shift register is preloaded with 0s and then encode 00000000 00000000 00000001, the same answer is got as encoding 00000001.

But even if the shift register is preloaded with 1s and then encode 00000000 00000000 00000000 00000000 00000001, the same answer is got as encoding 00000001, because the period of the CRC5 shift register (x^5+x^2+1) is 32.

Could you please share your idea about it with me?

.