Tech Note: FIFO Centering

W Duckwall, Zayante, Inc.

1.0 Introduction

I can take only marginal credit for any of this. The train of thought was begun by an engineer new to 1394, who asked simply why should the dribble bits be treated any differently in retransmission from the data bits. I couldn’t come up with much of a reason. If I may pedantically elaborate –

2.0 The Infamous Dribble Bits

Packets are followed by 20 ns worth of dribble bits – 2 bits at S100, 4 bits at S200, 8 bits at S400. The last two bits are logically required because -

Packet reception starts with data prefix AB=10
Assume bits are numbered 1, 2, 3 ...
For odd numberd bits,   Strobe = Data   (thus A xor B = 0)
For even numbered bits, Strobe != Data   (thus A xor B = 1)
All packets have an even number of bits
After transmission of data bits, line state will be either AB=01 or AB=10, depending on last data bit; thus 50% chance it ends up in wrong state.
Adding 2 dribble bits allows transition to desired state (data prefix or data end)

The other dribble bits (2 additional at S200, or 6 additional at S400) are a historical implementation artifact. They have no particular importance, and they aren’t checked on reception. It is difficult to envision a scenario in which they would be used for any embedded signalling scheme in the future, since they are specific to S200/S400 packets.

3.0 Retransmission vs. Regeneration

The standard outlines a particular method for data reception and repetition - see P1394A Draft 1.2 Tables 7-16, 7-17, 7-21. Omitting most of the hard stuff, it all boils down to:

Retransmit all the data bits
(Leave the received dribble bits in the resynchronizer FIFO)
Generate the appropriate number of new dribble bits, and tack them onto the retransmitted packet

An alternate and perhaps non-compliant approach would be be simply:

Retransmit all the data bits, including the dribble bits

What’s the difference?

3.1 “Extra” dribble bits at S400/S200

At S400/S200, there are those extra 6/2 dribble bits, all ones, that are never checked by any receiver. Furthermore, as mentioned above, they are not present in S100 packets, and so are not likely to be useful for signalling. If they are received correctly as ones, then there is no way to tell whether they are retransmitted or regenerated - either way they come out as a string of ones. If they are received as something other than all ones, then what is the correct behavior? Should a phy “correct” them and send out all ones, or should it pass them on as they were received. If you can be convinced that these bits will not be used in some fashion in P1394G (coming in 2023), then it doesn’t matter, and we don’t have to worry about “correct” behavior. And even if you do think there may be some future use, then there’s at least a 50% chance that your particular future use would require that the bits be retransmitted as they were received, rather than “corrected” and sent out as all ones. Conclusion: There’s probably no penalty for retransmitting these bits as received.

3.2 The last two dribble bits

The standard refers to the next-to-last dribble bit as the penultimate dribble bit. By inference, the last dribble bit must be the ultimate dribble bit. (John Cleese of Monty Python has shown some interest in this discussion.) It is well concealed by Table 7-15, but the standard calls for both the penultimate and the ultimate dribble bits to be set to the same value - either 00 for data_end, or 11 for data_prefix.

The ultimate dribble bit may be sensed by a phy directly from the receiving stage flip-flop, as a quick means of determining the packet ending status (whether data_prefix or data_end). This is generally faster than waiting for the arbitration receivers to register the ending line state. But regardless of whether the final line state comes from the receiver flip-flop, or from the arb comparator, the phy will retransmit it verbatim, unless fly-by arbitration is involved, in which case a data_end may be switched to data_prefix.

The penultimate dribble bit is generally ignored by phys. To comply with table7-15, it should be the same as the ultimate dribble bit. However, it isn’t checked currently, and I don’t believe there would be any detectable difference in operation if a phy output the “wrong” value for the penultimate bit. If it were enrolled in some clever signalling scheme at a later time, then probably it would be desired to retransmit it as received, rather than “correct” it, but it’s getting a little late to enroll this bit into some new invention, given the number of phy devices already in production.

4.0 Effect on FIFO Centering

The current mechanism for FIFO centering requires that N bits, where N = number of dribble bits (always 20 ns, regardless of packet speed) plus an extra ~20 ns of data (40 ns total delay) be stored in the FIFO before readout begins. In normal operation, the FIFO is drained down to only the dribble bits when end-of-packet is reached (and not before). Clock slip accounts for ± 17 ns.

If you buy the arguments above, then it is OK to unload at least the extra dribble bits (for S200/S400 operation) from the FIFO for retransmission. Thus it would be permissable to begin unloading the FIFO when 2 bits (corresponding to the last 2 dribble bits) plus ~20 ns of data are stored. FIFO unloading could begin for S200 operation after 2 bit times (10 ns) plus an extra ~20 ns, or 30 ns total delay (a 10 ns savings). Similarly, FIFO unloading could begin for S400 operation after 2 bit times (5 ns) plus an extra ~20 ns, or 25 ns total delay (a 15 ns savings). These savings are not large, but they make sense as parts of a larger strategy to minimize phy propagation delay. Fifteen ns is ~10% of a typical phy propagation delay, or the amount of delay of 3 meters of cable or fiber.

5.0 Compliance

Finally, whether a simple retransmission of the extra dribble bits (2 at S200, 6 at S400) is compliant to the standard or not is a fine point. There are two cases:

Phy receives a correct packet (all extra dribble bits are ones) -
Retransmitted packet will also be correct, and compliant.

Phy receives an incorrect packet (one or more of the extra dribble bits are zeros) -
Retransmitted packet will also be incorrect (extra dribble bits retransmitted as received)
A “compliant” phy would correct the extra dribble bits to all ones, but -

This “compliance” to a historical implementation artifact:

- has no logical basis
- incurs a propagation delay timing penalty for S200 and S400
- inadvertantly forces a choice of implementation

Unless there is some consideration I’ve totally overlooked, I’d like to suggest that we explicitly allow simple retransmission of the dribble bits, or at least retransmission of the “extra” (all but the last two) dribble bits.