Delay line memory with a fatal flaw

John Dunn - April 03, 2019

This delay line device was an early form of computer memory. It's pretty much obsolete stuff these days, but I did once get involved with one. The project was ultimately a failure, but the reasons why it failed might be instructive, even today.

Super-simplified, the thing looked like this.

![Figure 1 Delay line memory](image_url)

The plan was to have a really, really, really long quartz delay line carrying data through a closed loop. Data would go in at one end of the loop, come out at the other, be regenerated, and then sent
back in again ad infinitum. System synchronization was to be accomplished using a special sync code in front of the data block, which the controller would detect to keep track of where that data block was.

Unfortunately, the plan was fatally flawed.

Apart from imperfections in the data refreshment process itself, which might perhaps have been overcome someday, there was almost always some random point in the data block where the assigned synchronization code would be accidentally replicated. When the controller would find that accidental replication, it would misinterpret where the data block was and the whole storage scheme fell apart.

I was still bright-eyed, young, and skinny back when this was happening and although I recognized this inherent flaw, I could NOT convince the very senior project engineer in charge of it all that the flaw was there. It was "Oh, no. This will work just fine. All we have to do now is ... ".

You've heard of the phrases "proof-reader's error" and "blind spot"? That's what this was.

So, how's it going with that whiz-bang project you're working on these days?

*John Dunn* is an electronics consultant, and a graduate of The Polytechnic Institute of Brooklyn (BSEE) and of New York University (MSEE).

**Related articles:**
- Delay line eases Spice dead-time generation
- Delay line upgrades vintage scope
- One small step for synchronization