UPDATES 6 JANUARY 2026

OPULENT VOICE

pluto_msk repository bug hunts and development since 12 December 2025

TL;DR

- Since the last ORI report, we've made significant progress on the Opulent Voice FPGA implementation. The major achievement was identifying and fixing a critical encoder bug that caused corrupt encoder data, which created a corrupted spectrum.
- The full transmit chain now produces a clean MSK spectrum.
- Receiver progress has also happened, but this work is ongoing. Full receive function is not yet complete, but it's very close.

LIBRESDR LVDS CLOCK DOMAIN RESOLUTION (DEC 13-15)

- Successfully ported the MSK modem from PlutoSDR (CMOS interface, Zynq 7010) to LibreSDR (LVDS interface, Zynq 7020).
- The Challenge: In LVDS mode, the AD9361's l_clk runs at 245.76 MHz (4× the 61.44 MHz sample rate), but our MSK modem was designed to operate at the sample rate directly.
- The Solution? We implemented a counter-based clock divider using BUFG to derive a 61.44 MHz clock from the 245.76 MHz interface clock. Added synchronization registers between clock domains.
- Result: Correct bit rate (54,200 bits/second) achieved on LibreSDR hardware.

THE GREAT SPUR HUNT (DEC 16 - JAN 1)

- **Extensive debugging of mysterious spectral spurs appearing at 3375 Hz intervals in the transmitted signal.**
- **A methodical approach confirmed the problem was in the encoder chain, with periodicity corresponding to byte boundaries after rate-1/2 FEC expansion (54,200 / 3,375 ≈ 16 bits = 2 bytes)**
- **Root cause was in ov_frame_encoder.vhd. The OUTPUT state had a variable range declaration bug.**

Configuration	Spur Frequency	Periodicity
No Encoder (bypass)	clean	-
Bit Doubling with Fake FEC, no interleaver	13,550 Hz	4 bits
Real FEC, no interleaver	6,600 Hz	8 bits
Real FEC, interleaver	3,375 Hz	16 bits

THE BUG THAT CAUSED THE SPUR

- **Broken:**

```
VARIABLE out_bit_idx : NATURAL RANGE 0 TO 7;
```

- **The calculation exceeded this range:**

```
out_bit_idx := out_idx*8 + j;  -- Goes up to 2143!
```

- **What Happened? Vivado synthesized *exactly* 3 bits for RANGE 0 TO 7. When out_idx >= 1, the index calculation wrapped around due to 3-bit truncation. Every output byte was reading from interleaved_buffer(0..7), which was the first byte only, and not the entire address width. All 268 output bytes were copies of byte 0 (which happened to be 0xFF after randomization).**
- **Why did simulation pass? Because it did. Perfectly. VHDL simulators typically use full integer range internally and may warn but don't truncate like hardware synthesis. That is indeed what happened to us.**

AD9361 Q CHANNEL ISSUE IDENTIFIED

- We discovered that the Q channel (quadrature) DAC was disabled in hardware.
- I channel control register: 0x1E (good!)
- Q channel control register: 0x00 (what?)
- This caused transmission to appear as a real signal with two-sided spectrum instead of proper complex IQ with single-sided spectrum. Image was 3dB down, so it looked like *something* was happening, but not enough to get to at least 40dB down, which is what should happen with complex modulation.
- Real: double sided. Complex: single sided.
- We confirmed that the modulator was putting out complex signals, that the radio was configured for complex signals, even with all of the registers correctly configured.

THE BUG BEHIND THE REAL VS COMPLEX

- After all other possibilities were exhausted, we went back to the basic settings for Libre SDR operation. We'd gotten a big fat failure when attempting to do LVDS tuning during the first week or so of porting to the Libre SDR, which has an LVDS interface. Tuning is important, but could it really cause this much damage to our signal?
- It turns out that the answer is "yes, it can".
- We turned LVDS tuning back on. This time, it worked. Whatever failure we had when we first ported had been fixed, possibly by all the other good work we've done over the past few weeks. LVDS tuning succeeded, confirmed by dmesg logging, and a true complex signal was immediately confirmed on the spectrum analyzer.

NEWSLETTER: SYNC WORD SOFT DECISIONS

- We created technical content for ORI communications covering:
- Sync Word Optimization: Exhaustive search of 24-bit sequences revealed 6,864 patterns achieve optimal 8:1 Peak Sidelobe-to-Mainlobe Ratio (PSLR), significantly better than traditional concatenated Barker codes at 3:1 PSLR. We selected sync word: 0x02B8DB (mnemonic: "oh to be eight dB").
- Soft-Decision Implementation: We documented the transition from hard-decision to soft-decision correlation in frame_sync_detector.vhd, including threshold calibration procedure.
- This threshold calibration was carried out, and sync word detection has been confirmed over the air in the lab. Our sync detector, with soft decision instead of hard decision detection, is optimized not only for very good autocorrelation properties, but excels in multi path conditions. This is professional quality work and meets or exceeds the standards set by any other digital voice protocol in use on VHF or UHF.

PULL REQUEST APPROVED

- Dozens of commits were folded back into main a few days ago. A very large amount of work from multiple people has been consolidated back into the main checkout for the repository. It's still named pluto_msk, despite porting to the Libre SDR.
- https://github.com/OpenResearchInstitute/pluto_msk/pull/27

AMSAT-UK MDT IMPACT

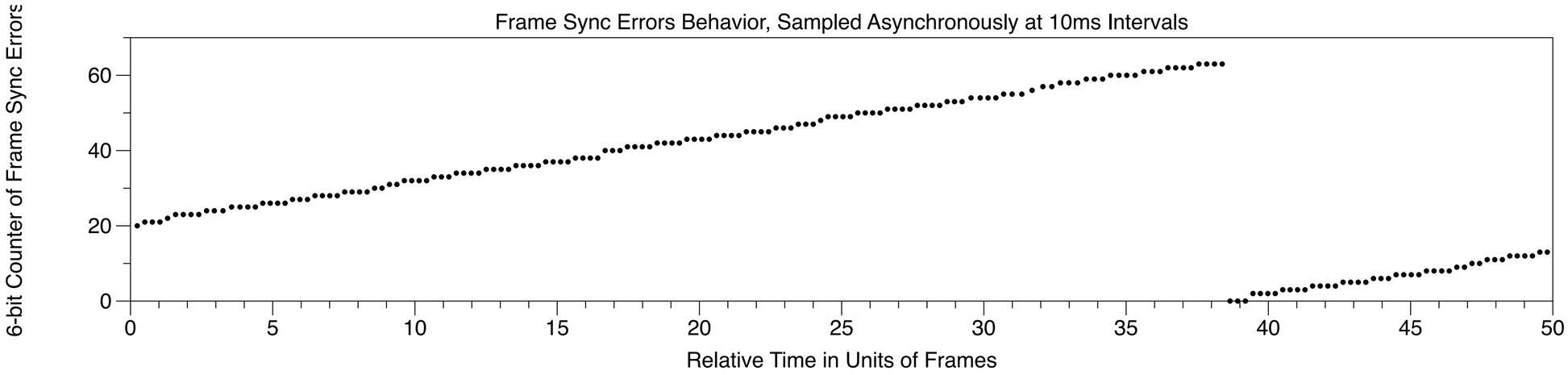
- While the bare Opulent Voice modem can fit within the gates available on the ICE FPGA selected for the development of FunCube+, the framing for the full protocol pushes it well over that limit.
- The weak signal detector can be done, with some limitations. This is where development should most likely focus. The basics of that digital signal processing procedure have been summarized in email, and will be brought out to the public repository in the next week or so.
- https://github.com/OpenResearchInstitute/Mode-Dynamic-Transponder

DECODER STALL

- This is our current challenge. We're stalled out in FEC_DECODE state in the state machine in the decoder. We're not sure why. We've added ILA, confirmed the stall, and have bypassed the FEC with "fake" FEC. This resulted in even stranger behavior!
- We believe that the frame sync word detection is working, that we're in locked state, but that something is going wrong in the forward error correction decoder itself, or interface to it. Receive FIFO isn't moving at all, so no data is getting through despite very good detection and calibration of the thresholds.

FRAME SYNC ERRORS BEHAVIOR

- This is what the frame sync errors counter is doing, sampled asynchronously every ~10ms. It's only a 6-bit counter, so it wraps around. Grossly, the behavior can be described as "incrementing by 1 for every single frame, or thereabouts". You can't see it here, but the time-averaged behavior is more like "incrementing by 1 for every 35ms of elapsed time". What you can see here is that sometimes it increments by 2, and sometimes it increments in the middle of what looks like it ought to be a frame.



MAIA SPECTRUM DISPLAY