Digital Signal Processing RTL Modules

CASE WESTERN RESERVE UNIVERSITY

Hardware-Efficient DSP Solutions for ASIC design

Ishika Kanakath BS +MS Computer Engineering
Case Western Reserve University

MACOM Partners from RF to Light

🤓 🎯

Project Overview

Challenge: Implement efficient DSP algorithms in hardware while maintaining precision and minimizing resource usage for the digital design IP library.

Solution: Four optimized RTL modules that leverage hardware-specific optimizations for real-time signal processing.

SystemVerilogCadence XceliumFSM DesignFixed-Point

📚 Project # 1 : DC offset removal

Function: Removes DC bias from streaming data using moving average over 2^N samples **Key Features:**

- FSM-based control (IDLE→ACCUMULATE→UPDATE_DC)
- Bit-shift division for hardware efficiency
- Dual outputs: raw (dout) and corrected (dout_correct)
- Continuous operation during offset updates

Version 2 Improvements: Separated output logic from FSM, eliminating state-dependent glitches and providing consistent, predictable outputs every clock cycle

🖥️Project # 1 : Results

Background: DC offset is a constant bias that shifts signals away from zero, reducing dynamic range and causing saturation in ADC/DAC systems. Removing it is critical for audio processing, sensor conditioning, and communication systems.

Version 1 Results:

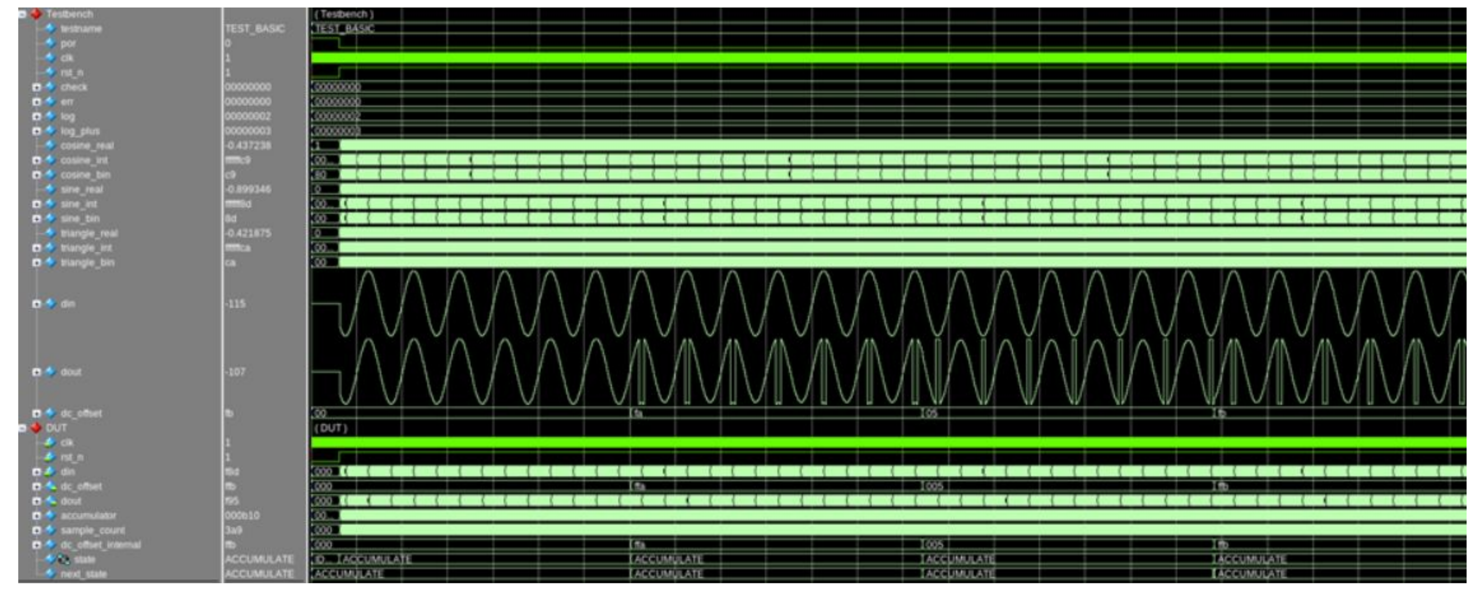
- Successfully removes DC offset using 1024-sample moving average
- FSM transitions visible at correct intervals (IDLE→ACCUMULATE→UPDATE_DC)
- Critical Issue: Output behavior inconsistent across FSM states, causing dout to alternate between raw and corrected values
- Waveform shows functional DC removal but unreliable output delivery

Version 2 Results:

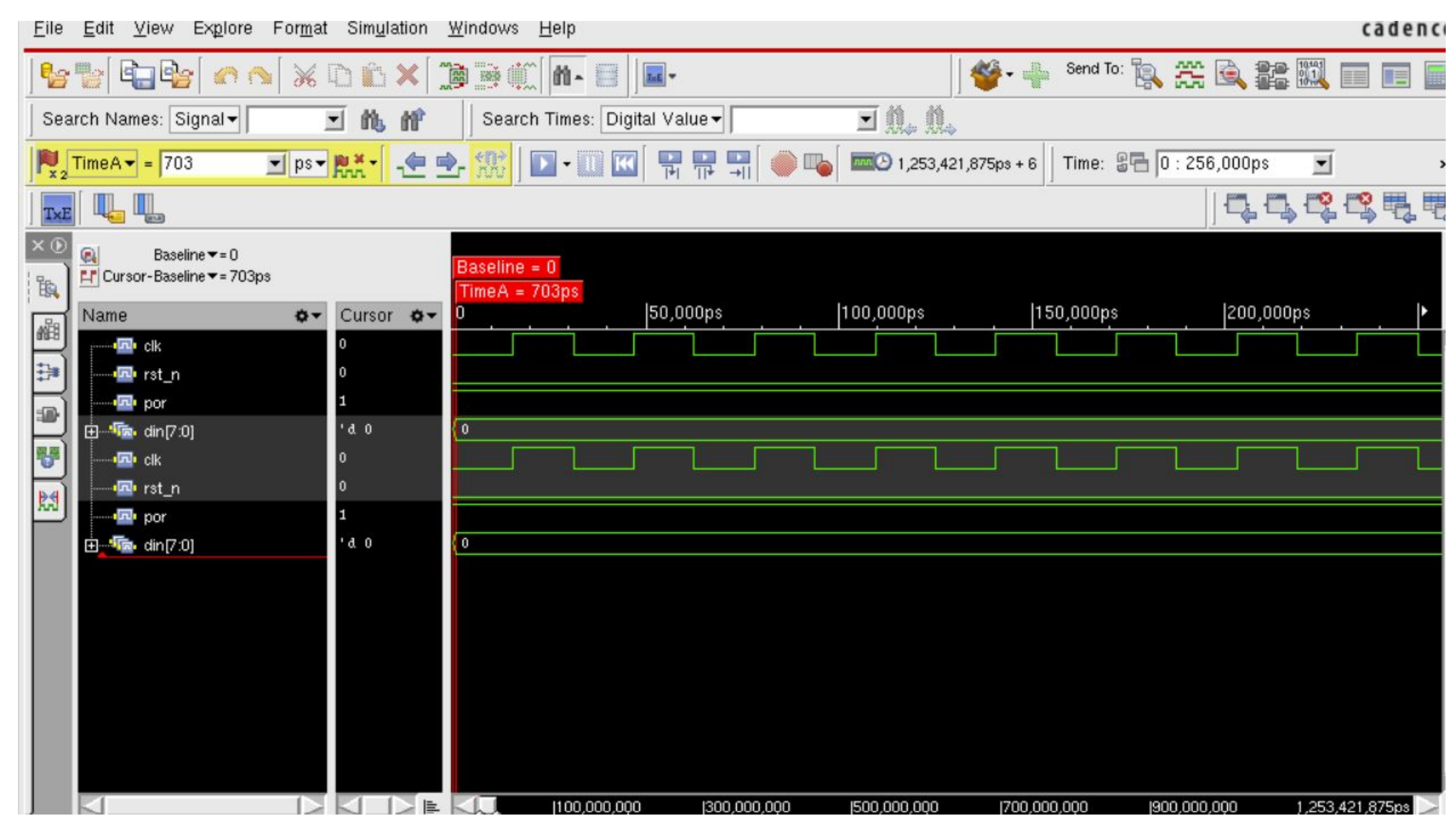
- Maintains same DC removal algorithm but with architectural improvements
- **Dual Output Solution:** Separated output logic from FSM control
  - dout: Always provides raw input data
  - dout_correct: Always provides DC-corrected signal
- **Performance:** Clean, glitch-free transitions with no output interruptions
- **Waveform Verification:** Both outputs remain consistent across all FSM states, eliminating Version 1's state-dependent behavior

Key Achievement: Version 2 transforms an academically correct but practically problematic implementation into a production-ready module. The consistent dual-output architecture enables reliable integration into larger DSP systems while maintaining continuous operation during offset updates.

DC Offset Waveform # 1



DC offset waveform # 2



📚 Project # 2 : Convergent Rounding

Module Function: Implements unbiased rounding that eliminates systematic errors in DSP calculations by rounding 0.5 cases to nearest even number

Key Features:

- Width-parametrized function handles any input/output bit widths
- Automatic binary point calculation for fractional bit trimming
- Saturation protection prevents overflow/underflow
- Round-to-even for 0.5 cases: 50% round up, 50% round down

Implementation:

- Extracts integer and fractional parts using bit masks
- Compares fraction against half-bit (0.5 threshold)
- At exactly 0.5: checks LSB of integer part to determine even/odd
- Embedded in DC offset module to replace simple truncation

Results: Eliminates accumulation bias in iterative calculations, crucial for long-term numerical stability in filters and accumulators

Verification: Exhaustively tested all 65,536 possible 16-bit inputs to verify correct rounding behavior and saturation limits

Project # 3 : Absolute Magnitude DC Offset

Block-Based Implementation (Average Module):

- **Dual Mode:** ABS=0 for DC offset, ABS=1 for magnitude averaging
- **FSM Architecture:** IDLE → ACCUMULATE → UPDATE_DC over 1024 samples
- **Features:** Convergent rounding, valid flag pulses when ready
- **Results:** Extracts DC offset (~308) with clean corrected output

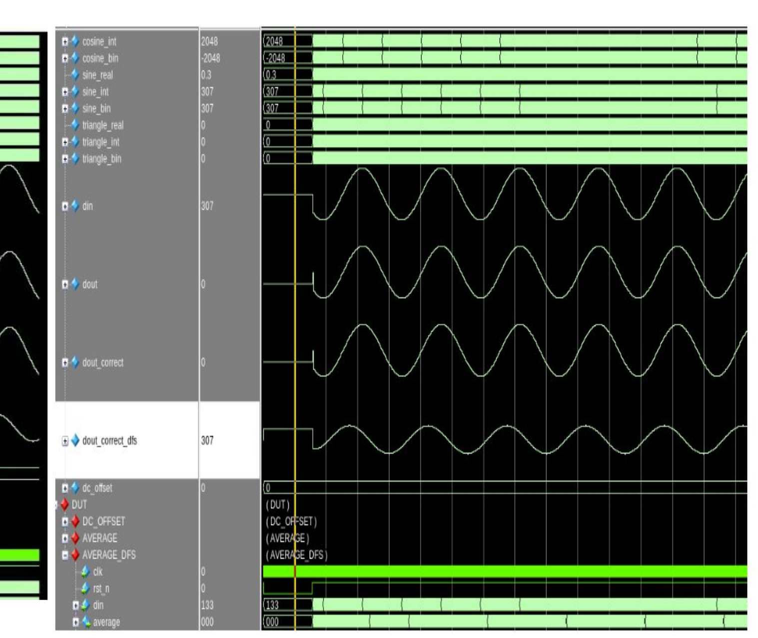
IIR Filter Implementation (Digital Filter Signals):

- **Algorithm:** First-order recursive filter: dc_estimate += (input - dc_estimate) >> SHIFT_AMOUNT
- **Operation:** Each sample updates estimate by a fraction of the error (1/16 for SHIFT=4)
- **Internal Scaling:** Uses 20-bit precision internally, scales input up by 8 bits to prevent quantization loss
- **Convergence:** Exponentially approaches true DC value with time constant τ = 2^SHIFT samples
- **Observed Issues:** Output amplitude ~50% reduced, phase shift present (inherent to IIR filters)

Comparison:

- Block: Discrete steps every 1024 samples, full amplitude preserved
- IIR: Smooth continuous tracking, some attenuation, instant response from reset

Average Block **IIR Block**



📜 Project # 4 : Digital Amplifier

Module Function: Hardware-efficient gain control using custom s1.PRECISION fixed-point format (1 sign + 1 integer + 15 fractional bits)

Implementation:

- Gain range: [-2.0, +1.999969]
- 33-bit multiplication with convergent rounding
- Output: DATA_WIDTH+1 bits prevents overflow
- Single-cycle operation with registered output

Verification: 21 test cases validate unity/fractional/negative gains, saturation limits, and cross-sign operations

Results: All tests passed (±1 LSB tolerance), proving correct fixed-point arithmetic implementation essential for DSP hardware