UNIC-CASS Project

Digital Timing Recovery System with ML-based TED and Polyphase Filters

Juan Tiago Ruiz Rodriguez

Agustin Arese

Juliana Denise Maidana

Juan Ignacio Chirino Bohorquez

Fundación FULGOR

CAS IEEE CIRCUITS AND SYSTEMS SOCIETY

Outline

- Introduction & Team Composition
- Design Overview & Objectives
- Timing Recovery Principles
- Block Diagram & Architecture
- Expected Outcome & Specifications
- Simulation Results
- Physical Implementation

Project Design

- Design Title:
  - Digital Timing Recovery System with ML-based TED and Polyphase Filters
- Design Acronym:
  - DCDR-ML-PF

Team Composition

- Mentor:
  - Ariel Pola
- Members
  - Juan Tiago Ruiz Rodriguez
  - Agustin Arese
  - Juliana Denise Maidana
  - Juan Ignacio Chirino Bohorquez

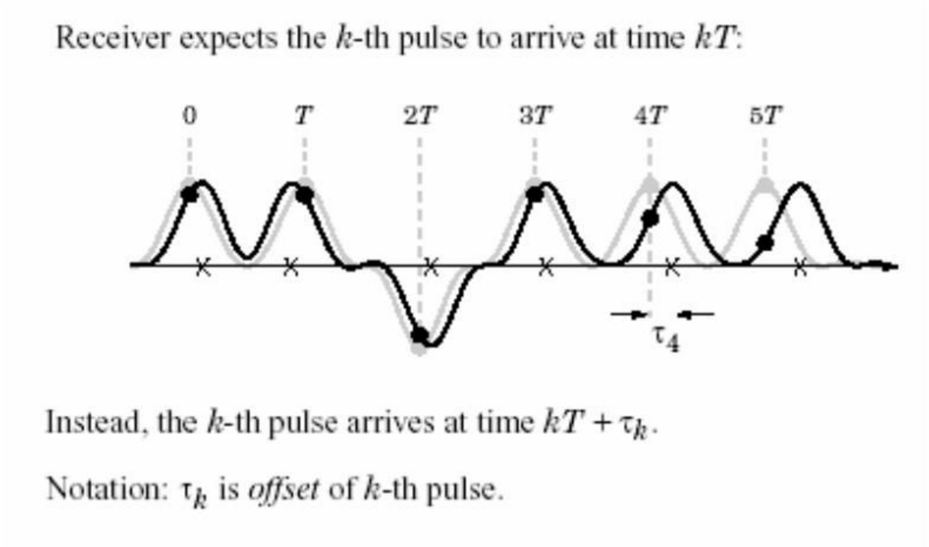
Design Details

- Description of the idea
  - A resource-efficient digital timing-recovery core is proposed for serial communications: it reconstructs optimal sampling instants by combining a 16-phase polyphase interpolator (SRRC coefficients) with a Maximum-Likelihood Timing Error Detector (ML-TED), a PI loop filter and a phase accumulator. The interpolator is implemented via hardware folding — a single 12-tap FIR whose phase-specific coefficients are time-multiplexed by a counter — minimizing area while providing the 16 polyphase responses. Three consecutive reconstructed samples (selected phase plus adjacent phases) are fed to the ML-TED, which computes a difference (an estimate of the derivative) and multiplies it by the remaining sample to produce a timing error metric; that error is filtered by a PI stage and integrated in an accumulator whose most significant bits select the interpolator phase corresponding to the optimal sampling instant. The core accepts serial input that carries two samples per symbol (unknown phase), corrects timing offsets and jitter digitally, and outputs the selected reconstructed sample for thresholding by a slicer. Control and observability are provided by an FSM, two control pins and a diagnostics/debug unit; the whole design emphasizes serial processing and polyphase integration (interpolation + matched filtering) to reduce resource usage, following the polyphase timing-recovery ideas popularized by Fred Harris.

Timing Recovery - Overview

- Purpose
  - Synchronize the sampling of symbols at the receiver, mitigating the effects of clock drift and channel delay.
- Key Parameters
  - Sampling Frequency – Estimates the symbol period (generally known).
  - Sampling Phase – Defines the optimal instant to sample (center of the symbol → best SNR).

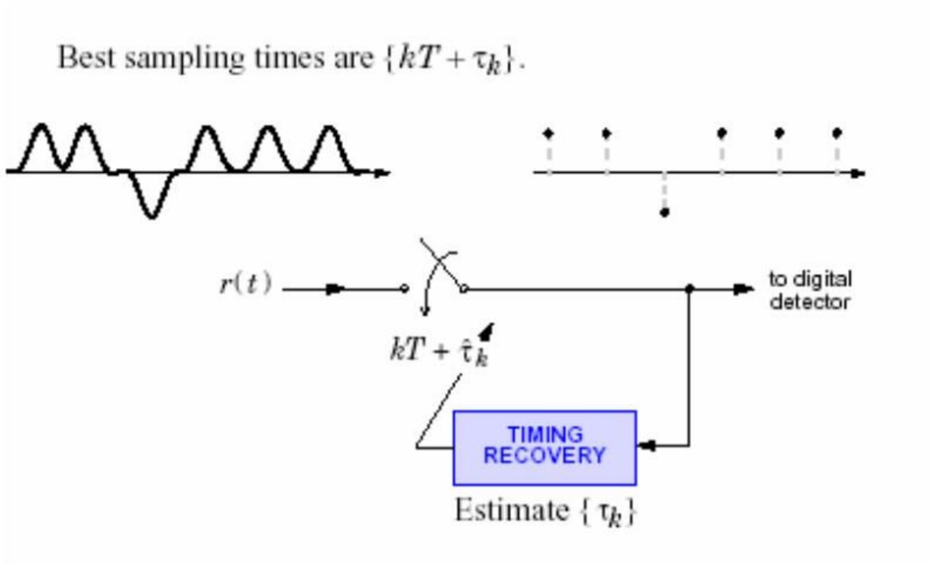
Problem Statement



Receiver expects the k-th pulse to arrive at time kT:

Instead, the k-th pulse arrives at time $kT + \tau_k$.

Notation: τ_k is *offset* of k-th pulse.



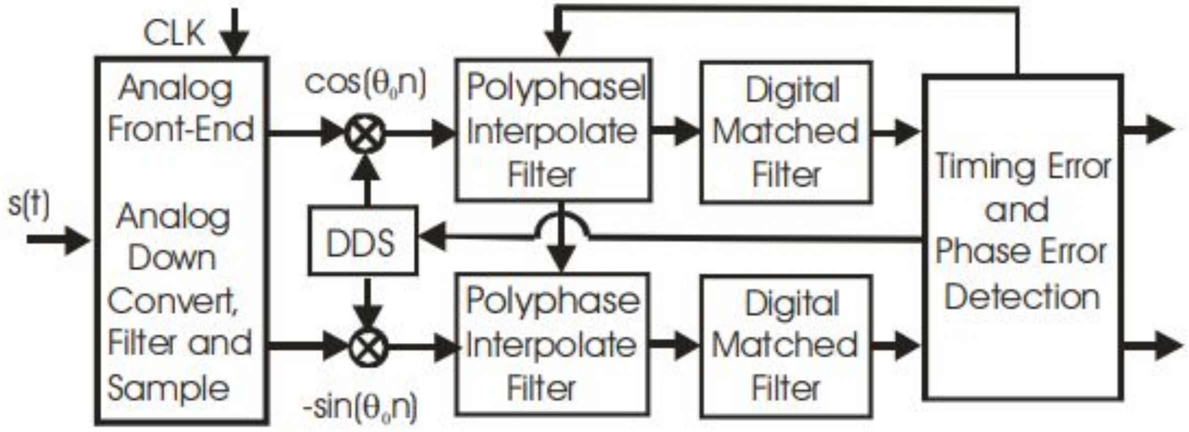
Best sampling times are $\{kT + \tau_k\}$.

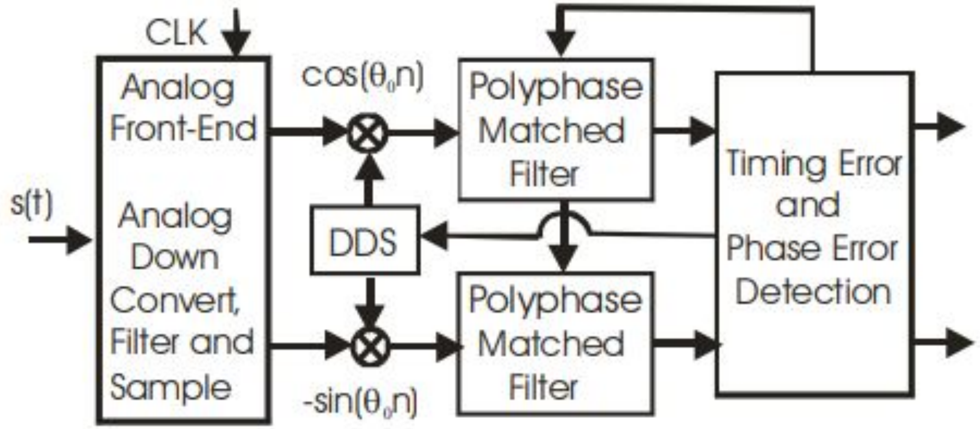
The receiver does not know the exact arrival time of pulses. Timing recovery must:

- Estimate the timing error for each symbol.
- Correct the timing error by one of the following strategies:
  - Adjusting a VCO's phase.
  - Adjusting the ADC sampling instant.
  - Using an interpolator to select the correct phase.

Of these options, we apply the last one: a polyphase interpolator + ML-based TED to reconstruct optimal sampling instants.

Strategies for Interpolation

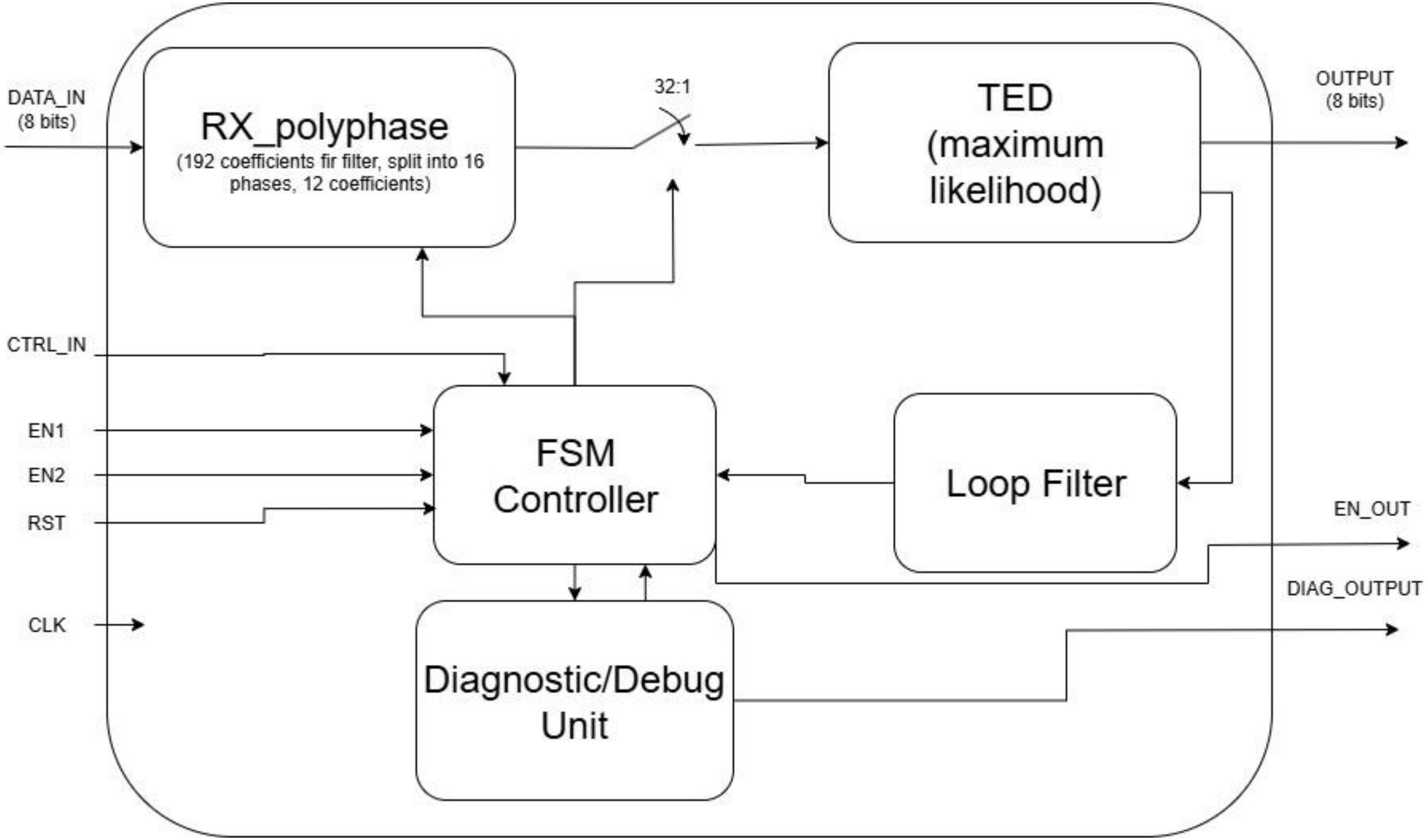

CLK
s(t)
Analog Front-End
Analog Down Convert, Filter and Sample
cos(θ_0n)
-sin(θ_0n)
DDS
Polyphase Interpolate Filter
Polyphase Interpolate Filter
Digital Matched Filter
Digital Matched Filter
Timing Error and Phase Error Detection



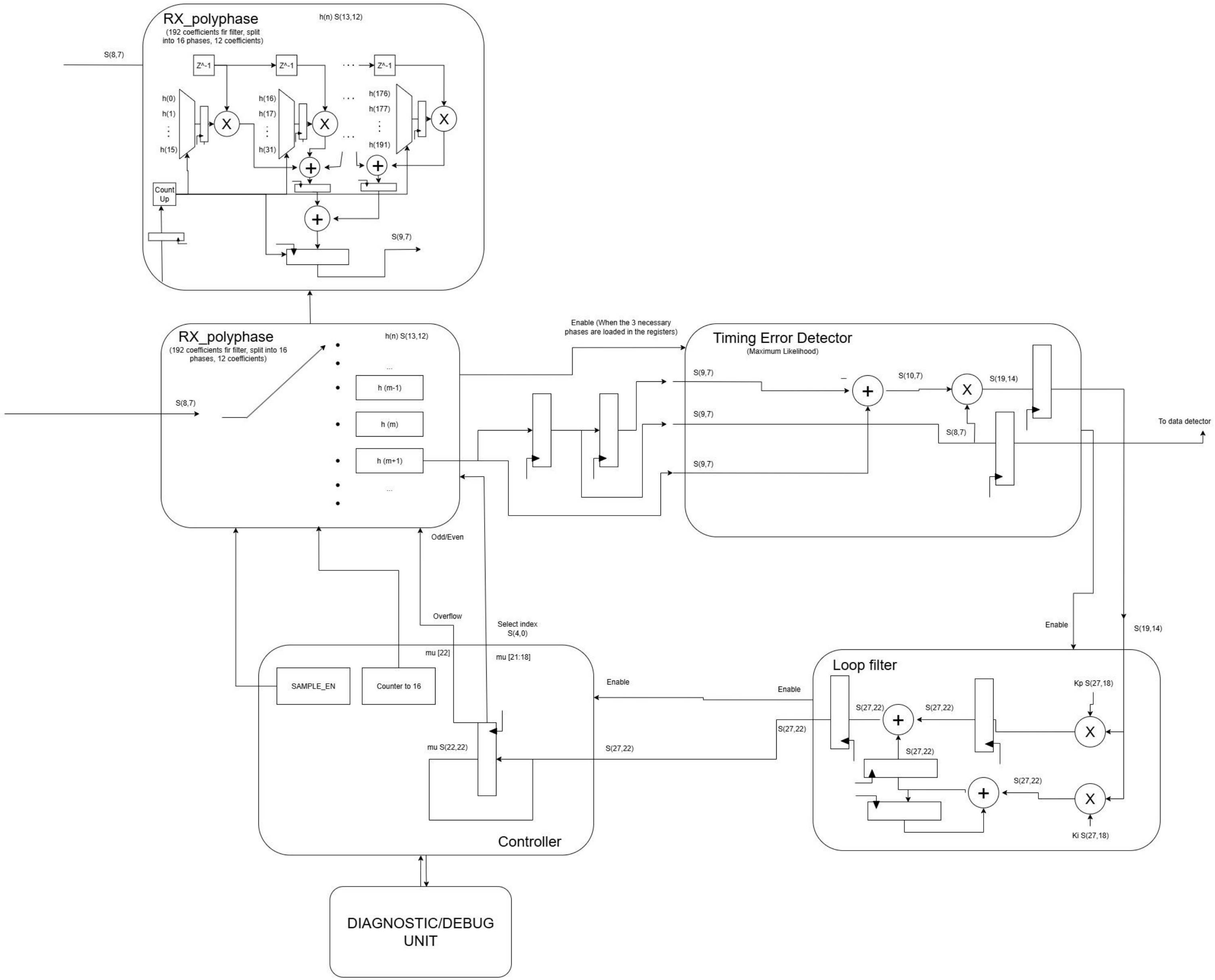
Two Strategies for Interpolation

1. Separate Interpolation & Filtering
   - Interpolate received samples first.
   - Then process them with the matched filter.
2. Integrated Polyphase Matched Filter (implemented)
   - Embeds the interpolation process within a polyphase matched filter.
   - A single filter bank provides both functions: interpolation and matched filtering.

Block Diagram



Block Diagram (detailed)



Design Details

- ## Expected Outcome

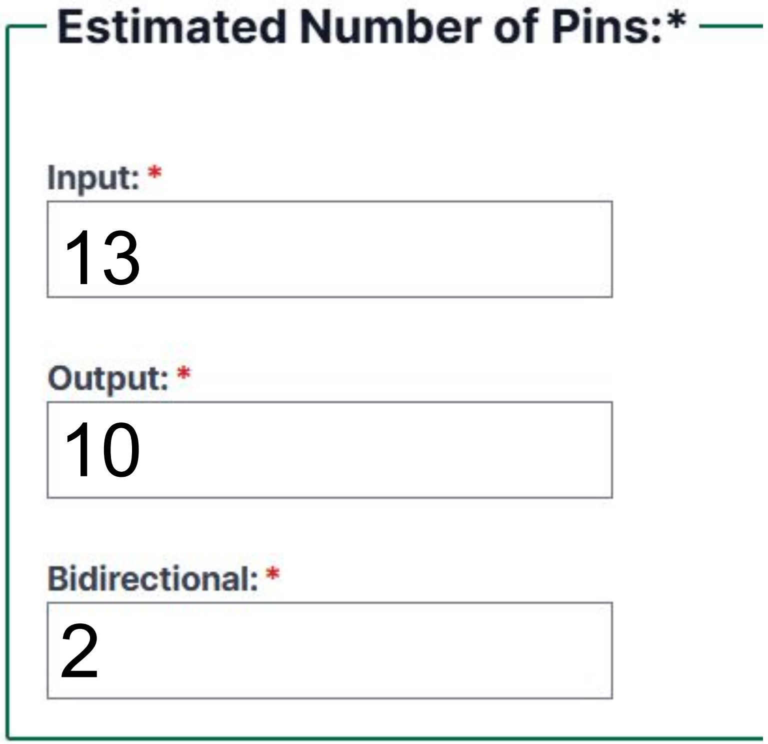
  - Our project delivers a fully functional serial timing recovery integrated circuit that will be able to demonstrate a successful clock and data recovery for digital communication systems, implemented using open-source tools and PDK. We expect to gain comprehensive hands-on experience in advanced digital signal processing (DSP) implementation, starting with a mathematical algorithm development, RTL design, and final IC tapeout, while learning about the complete open-source EDA toolchain including synthesis, place-and-route, and physical verification flows. The designed chip will serve as a practical communication system component suitable for software defined radios (SDR) and SERDES applications and also as an educational platform, providing understanding of polyphase filter architectures, timing recovery principles, and feedback-loop design. Thanks to this project, we aim to develop crucial skills in hardware design through a resource-efficient approach, floating-point and fixed-point system simulation, and system-level timing analysis. All of these skills provide a valuable foundation for careers in IC design. Additionally, our open-source reference design demonstrates that advanced communication circuit design is accesible using open-source methodologies, making it valuable for educational purposes.

Design Details

IEEE CIRCUITS AND SYSTEMS SOCIETY

- Number of Pins
  - VDD: Bidirectional
  - GND: Bidirectional
  - CLK: 1 input
  - RST: 1 input
  - EN1: 1 input
  - EN2: 1 input
  - DATA_IN (8 BITS): 8 inputs
  - CTRL_IN: 1 input
  - OUTPUT (8 BITS): 8 outputs
  - DIAG_OUTPUT: 1 output
  - EN_OUT: 1 output
- Design Type:

  Digital



Estimated Number of Pins:*

Input: *

13

Output: *

10

Bidirectional: *

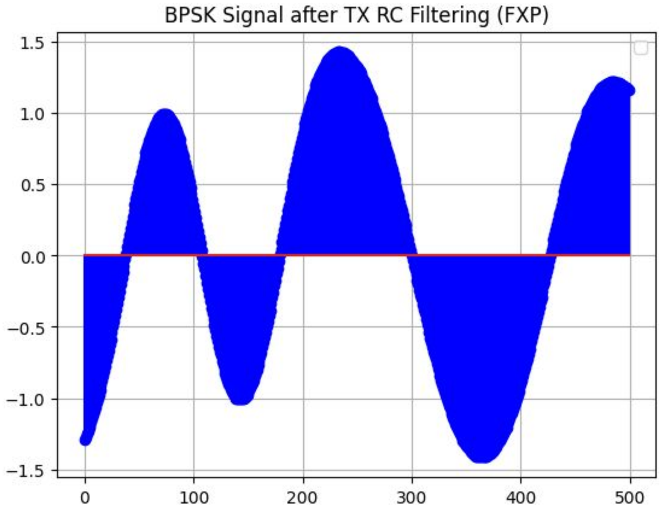
2

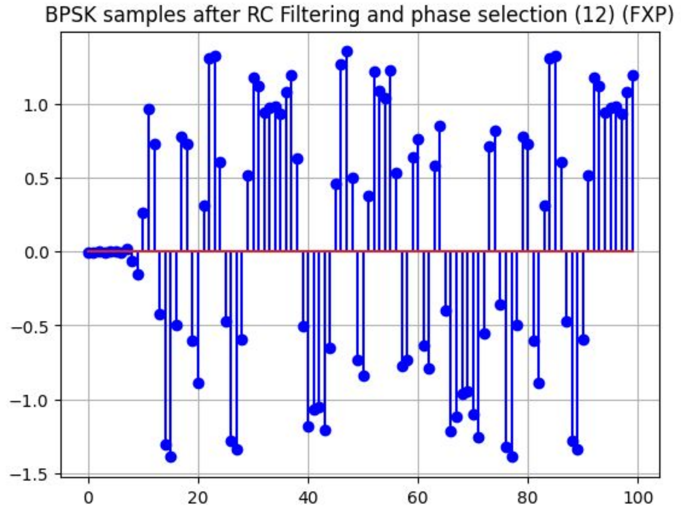
ihp 130 nm

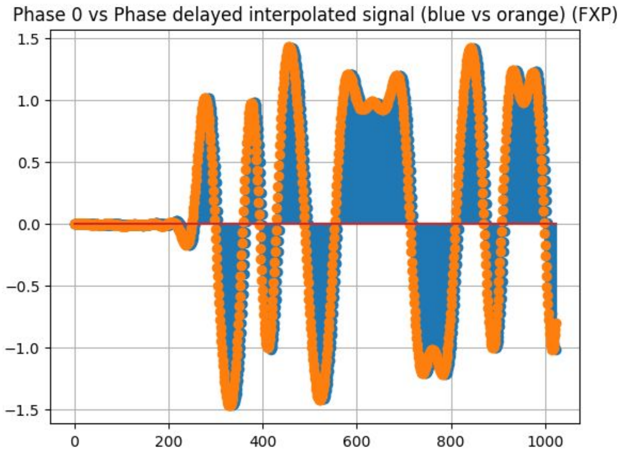
Simulations:

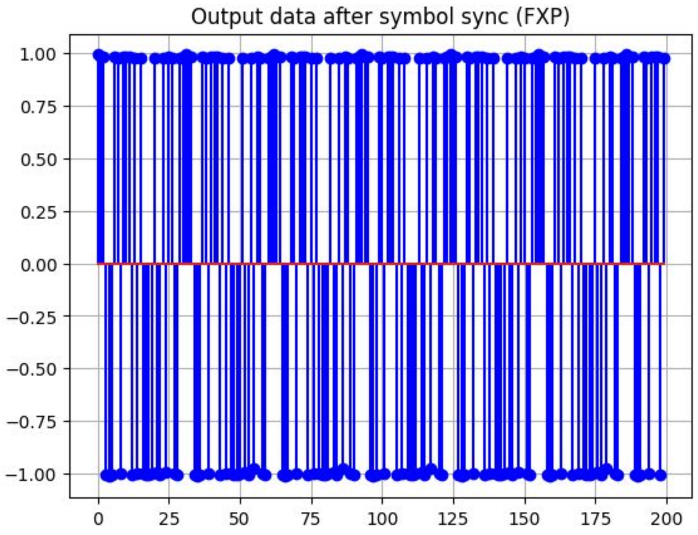
- Floating-Point Python Simulation (available in timing_recovery_simulation_FP_FXP.ipynb uploaded to github).
- Fixed-Point Python Simulation.
- Vivado software synthesis and functional simulation.
- FPGA Implementation (100 MHz) with fixed-point simulation vector matching of signals extracted from ILA.
- Physical Design Run for area estimation.

Simulations: Python (fixed-point)


BPSK Signal after TX RC Filtering (FXP)

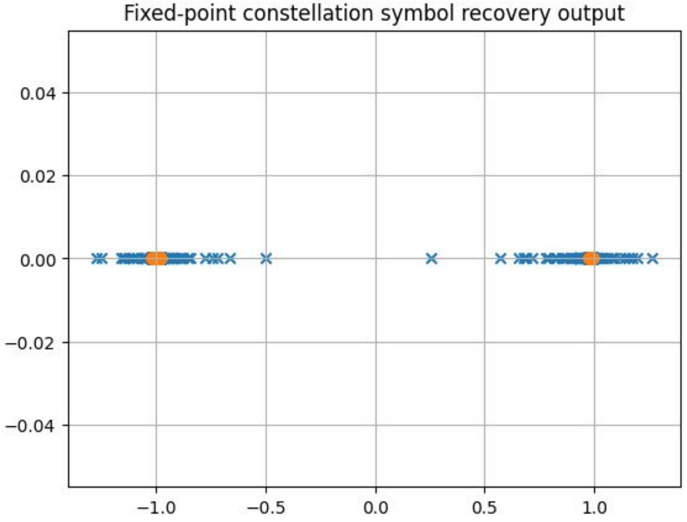

BPSK samples after RC Filtering and phase selection (12) (FXP)


Phase 0 vs Phase delayed interpolated signal (blue vs orange) (FXP)

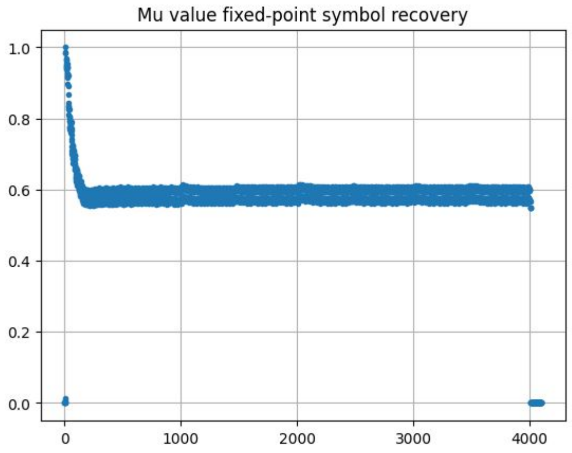

Output data after symbol sync (FXP)

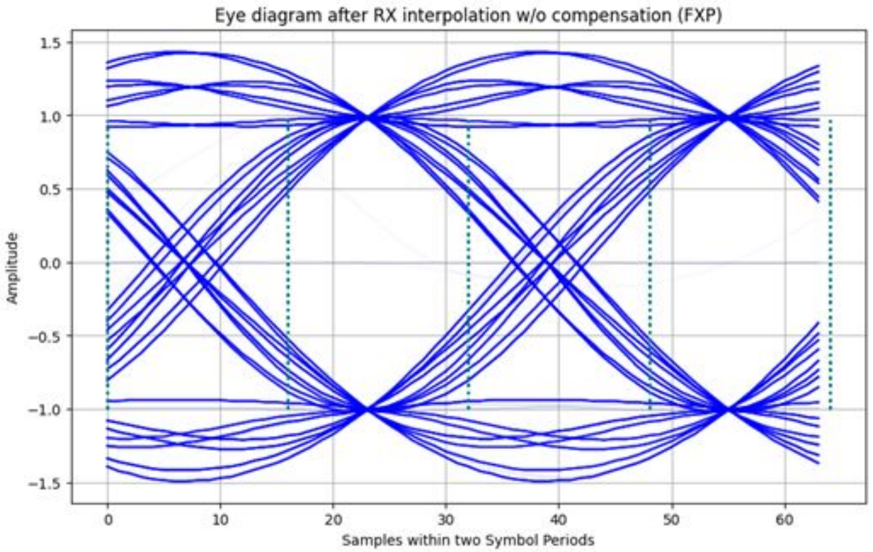
Simulations: Python (fixed-point)

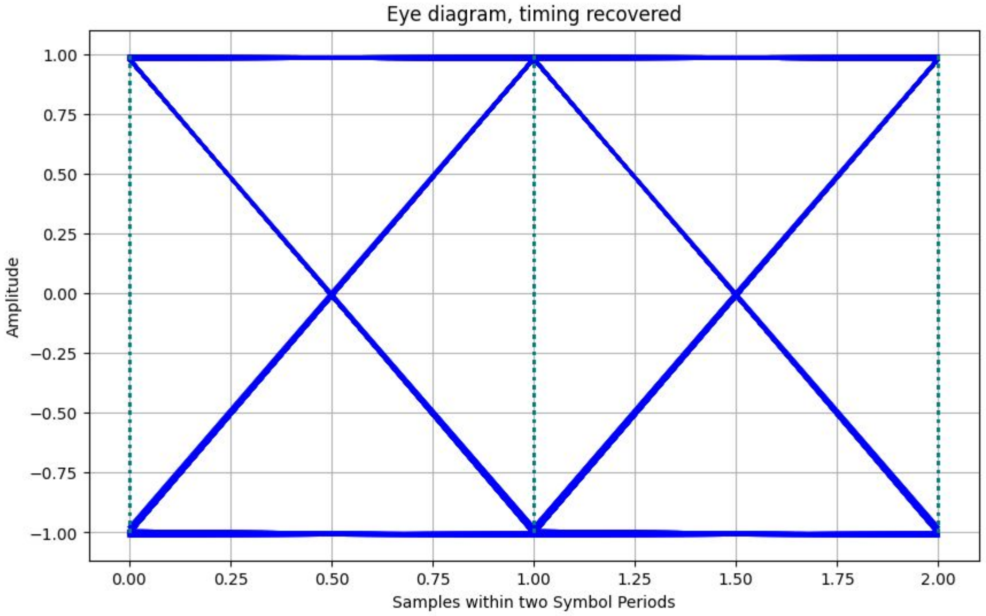
Results after timing recovery:


Fixed-point constellation symbol recovery output

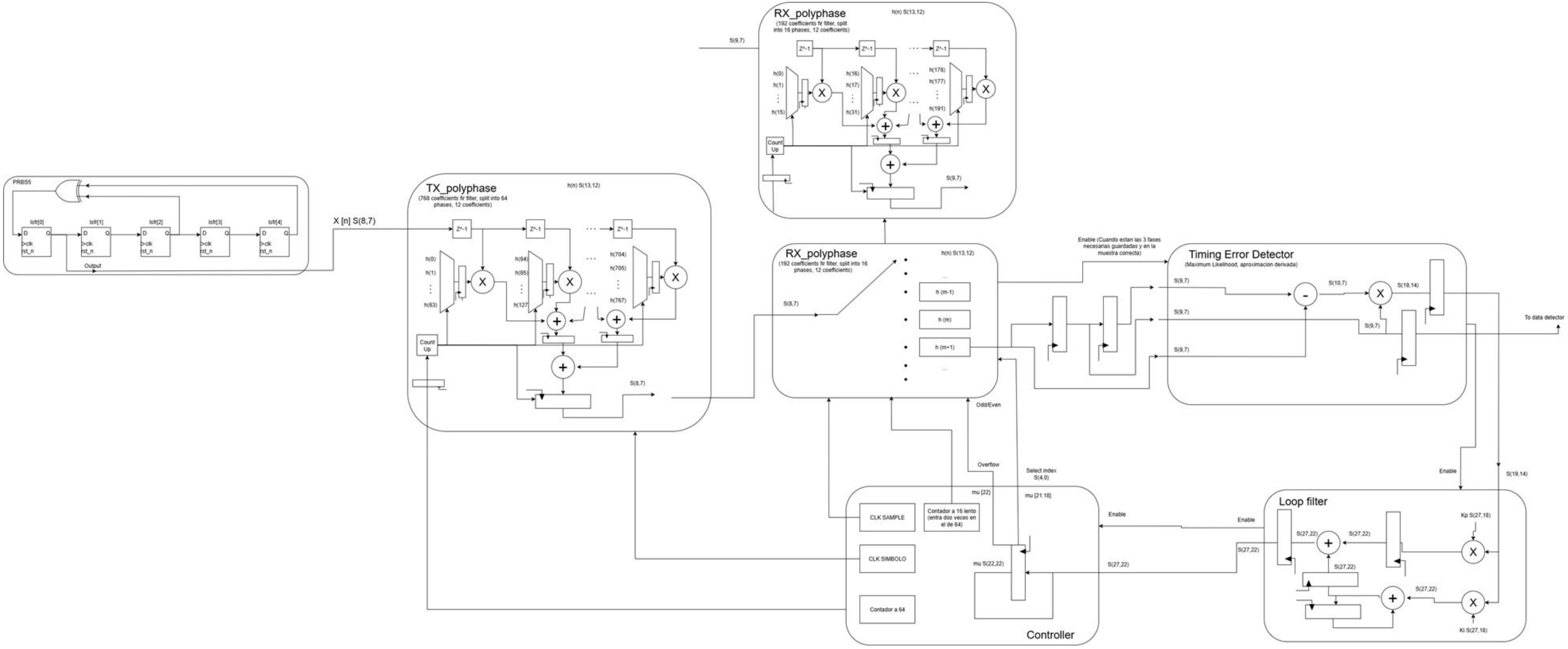
First 1000 samples in blue, remaining in orange


Mu value fixed-point symbol recovery

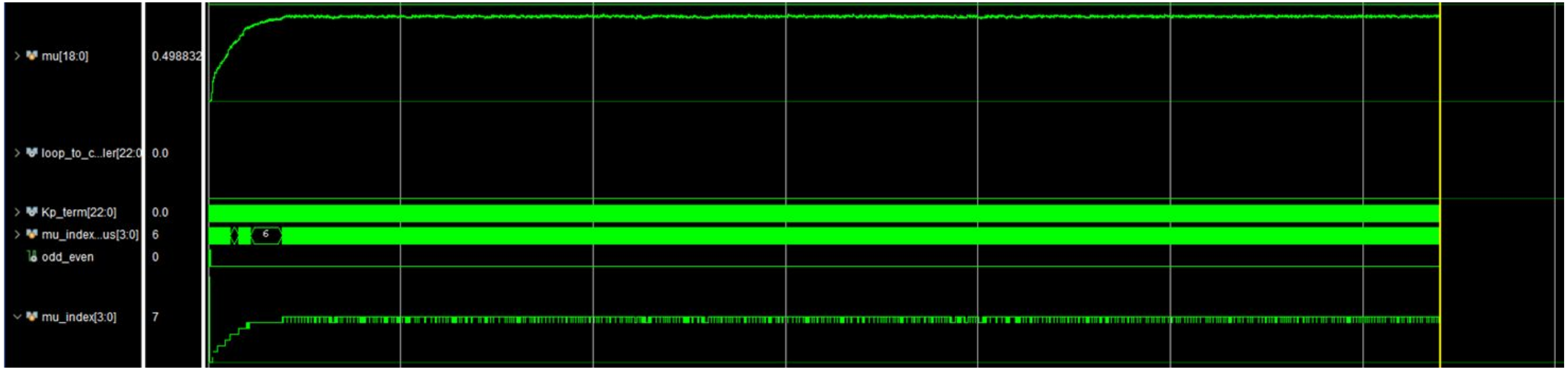

Eye diagram after RX interpolation w/o compensation (FXP)


Eye diagram, timing recovered

Run example - testbench

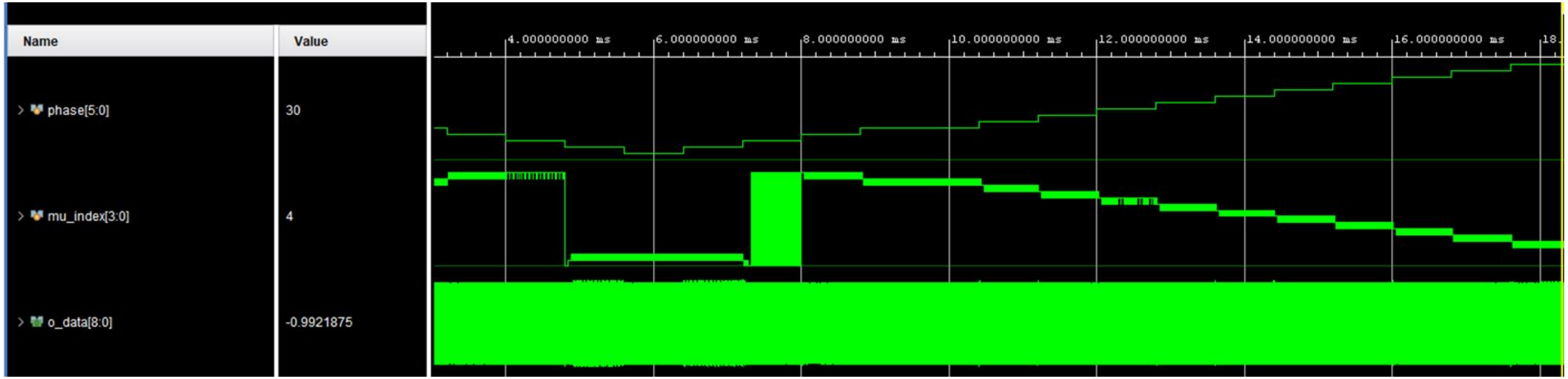


Run example - testbench



mu[18:0]	0.498832
loop_to_c...ler[22:0]	0.0
Kp_term[22:0]	0.0
mu_index...us[3:0]	6
odd_even	0
mu_index[3:0]	7

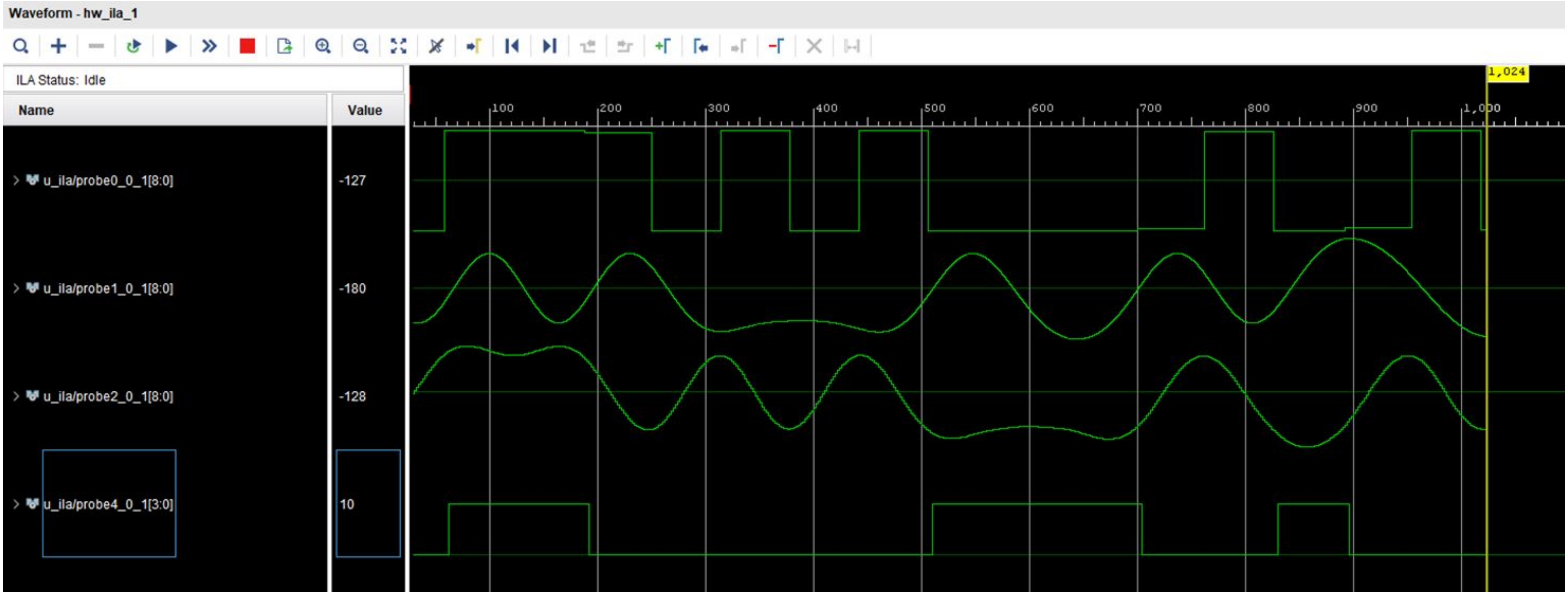
mu (phase error value) starts accumulating and settles as it’s corrected



Name	Value
phase[5:0]	30
mu_index[3:0]	4
o_data[8:0]	-0.9921875

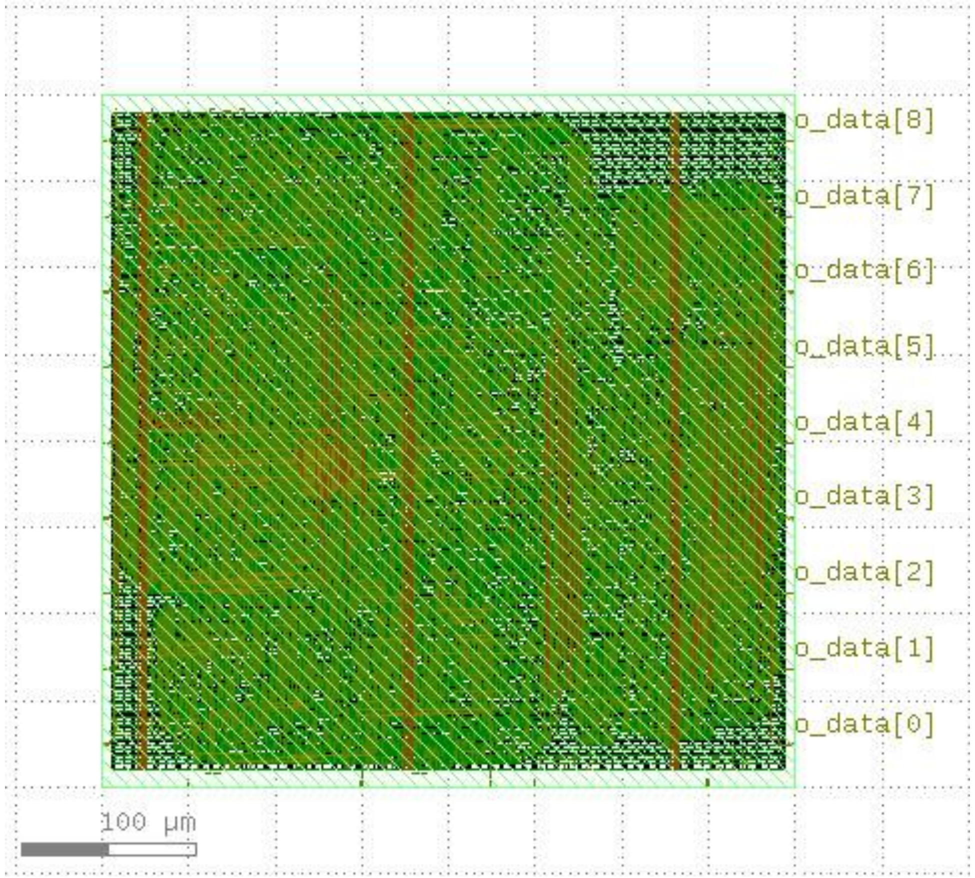
mu keeps being corrected as input phase offset changes

Run example - testbench



Design with autonomous stimulus implemented at 100 MHz in FPGA Artix 7 35T, design recovers successfully the optimal sampling instant from the reconstructed signal (with vector matching with fixed point simulation)

Physical Design Run example



Cell Type report:	Count	Area (µm²)
Fill Cell	10596	71195.78
Tap Cell	2190	2740.13
Antenna Cell	23	57.56
Clock Buffer	82	1217.42
Timing Repair Buffer	316	2712.60
Inverter	150	624.35
Clock Inverter	58	728.20
Sequential Cell	487	10476.30
Multi-input Combinational Cell	8589	64115.24
Total:	22491	153867.57

Design implemented in Skywater 130 nm