# FINAL PROJECT

## DFX Expanded Hyperbolic Cordic Implementations

Iyad Mansour
Omar Bataineh

OAKLAND UNIVERSITY 12/13/2015

# OUTLINE:

- Objectives
- Status Summary
- Project Timeline
- Extended Hyperbolic Cordic in Math
- DFX VHDL Design
- Error Analysis
- IP Package and Peripherals
- Attention Areas

# Objectives

- The original hyperbolic CORDIC (Coordinate Rotation Digital Computer) algorithm imposes a limitation to the input's domain which renders the algorithm useless for certain applications (**limited convergence**).

- A fixed-point implementation of the hyperbolic CORDIC algorithm with the expansion scheme proposed by Hu is presented, Using the *Iterative* Architecture, which is upgraded to a DFX Version.

- We used DFX because it combines the simplicity of a fixed-point system with the wider dynamic range offered by a floating point system. Using a single bit exponent which selects two different fixed-point representations, it allows dynamic scaling of signals throughout the system.

## Status Summary

- The project is done on time as scheduled.

# Project Timeline

| Project # 1 - Phase A | | | 2015 | | | | | | | | | | | 2016 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Activity | Notes | Status | 9/5 | 10/28 | 11/4 | 11/11 | 11/18 | 11/25 | 11/30 | 12/9 | 12/5 | 12/10 | 12/30 | 1/6 | 1/13 | 1/20 | 1/27 | 2/3 | 2/10 | 2/17 |
| **Start** | | | | | | | | | | | | | | | | | | | | |
| **Project Selection** | | Done | | | | | | | | | | | | | | | | | | |
| **MATLAB implementations for FX** | | Done | | | | | | | | | | | | | | | | | | |
| **MATLAB implementations for Expanded FX** | | Done | | | | | | | | | | | | | | | | | | |
| **Implementation** | | | | | | | | | | | | | | | | | | | | |
| **VHDL implementations for FX** | | Done | | | | | | | | | | | | | | | | | | |
| **VHDL implementations for Expanded FX** | | Done | | | | | | | | | | | | | | | | | | |
| **DFX Design** | | | | | | | | | | | | | | | | | | | | |
| **Design DFX Barrel Shifter** | | Done | | | | | | | | | | | | | | | | | | |
| **Design DFX Add/Sub** | | Done | | | | | | | | | | | | | | | | | | |
| **Parametrization** | | Done | | | | | | | | | | | | | | | | | | |
| **Run full range test case** | | Done | | | | | | | | | Final | | | | | | | | | |
| **Write final report** | | Done | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |

# Extended Hyperbolic Cordic in Math

The method proposed by Hu consists in the inclusion of additional iterations for negative indexes *i*:

$$For\ i \le 0$$

$$X_{i+1} = X_i + \delta_i \left(1 - 2^{i-2}\right) Y_i$$

$$Y_{i+1} = Y_i + \delta_i \left(1 - 2^{i-2}\right) X_i$$

$$Z_{i+1} = Z_i - \delta_i Tanh^{-1}\left(1 - 2^{i-2}\right)$$

$$For\ i > 0$$

$$X_{i+1} = X_i + \delta_i Y_i 2^{-i}$$

$$Y_{i+1} = Y_i + \delta_i X_i 2^{-i}$$

$$Z_{i+1} = Z_i - \delta_i Tanh^{-1}\left(2^{-i}\right)$$

# Extended Hyperbolic Cordic in Math –Cont.

Depending on the mode of operation, the quantities X, Y and Z tend to the following values, for sufficiently large $N$:

| Rotation Mode | Vectoring Mode |
|---|---|
| $x_n = A_n(x_1 cosh z_1 + y_1 sinh z_1)$<br>$y_n = A_n(y_1 cosh z_1 + x_1 sinh z_1)$<br>$z_n = 0$ | $x_n = A_n\sqrt{x_1^2 - y_1^2}$<br>$y_n = 0$<br>$z_n = z_1 + tanh^{-1}(y_1/x_1)$ |

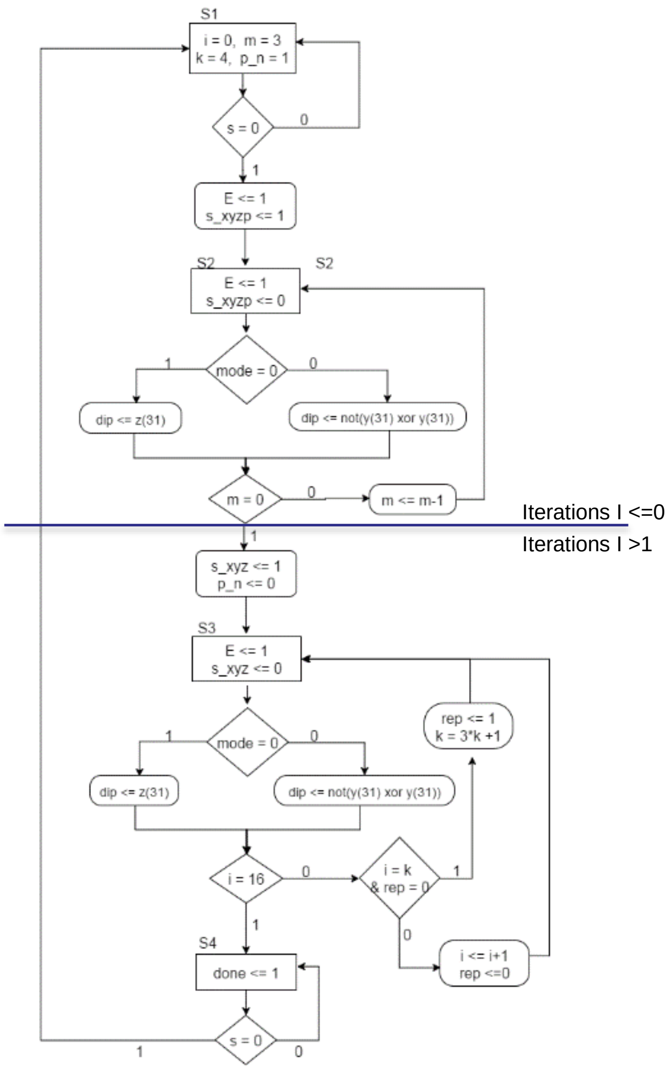
$$A_n \leftarrow \left[\prod_{i=-M}^{0}\sqrt{1-\left(1-2^{(i-2)}\right)^2}\right]\left[\prod_{i=1}^{N}\sqrt{1-2^{-2i}}\right]$$

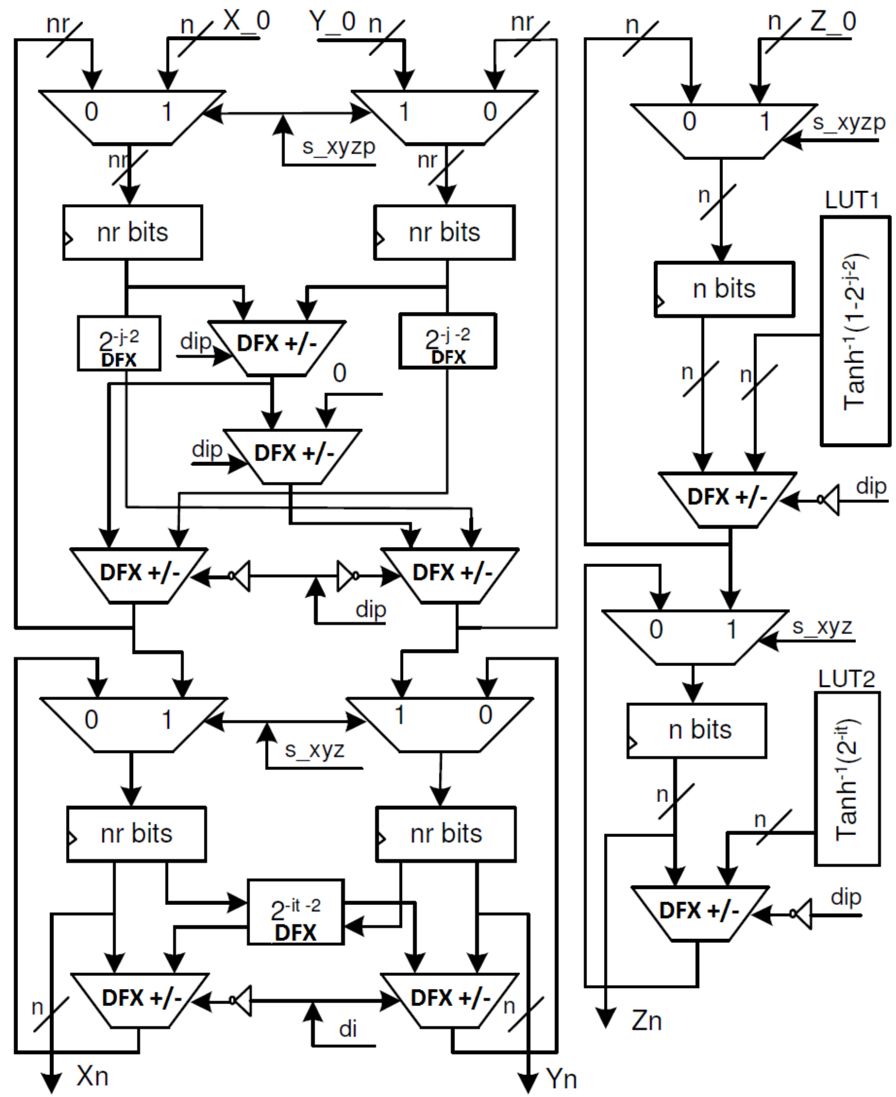
Range of convergence with respect to the negative iterations (M):

| M | $\theta_{max}$ from (19) | M | $\theta_{max}$ from (19) |
|---|---|---|---|
| 0 | 2.09113 | 5 | 12.42644 |
| 1 | 3.44515 | 6 | 15.54462 |
| 2 | 5.16215 | 7 | 19.00987 |
| 3 | 7.23371 | 8 | 22.82194 |
| 4 | 9.65581 | 9 | 26.98070 |
| | | 10 | 31.48609 |

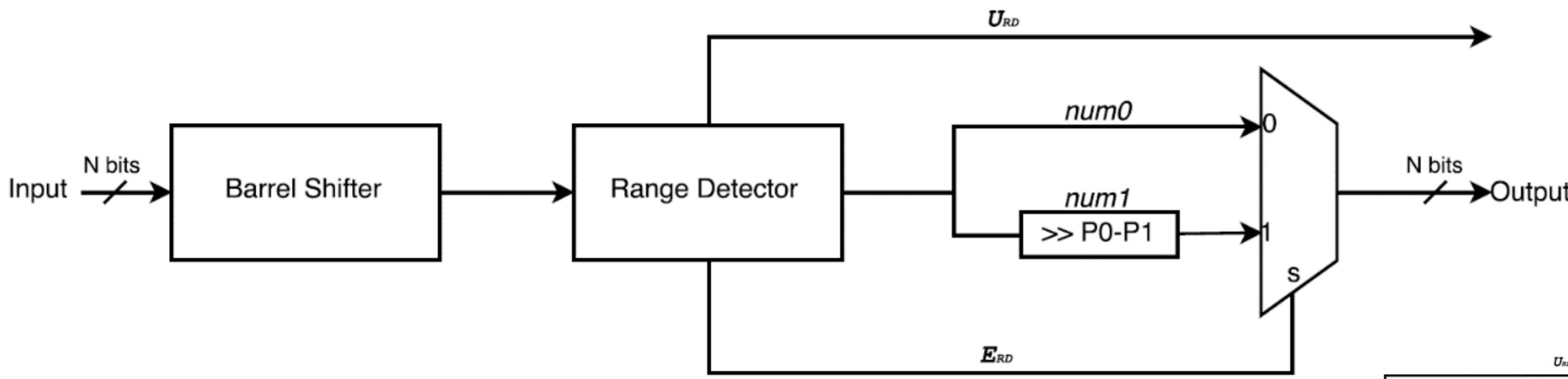
$$\theta_{max} = \sum_{i=-M}^{0} Tanh^{-1}\left(1-2^{i-2}\right) + \left[Tanh^{-1}\left(2^{-N}\right) + \sum_{i=1}^{N} Tanh^{-1}\left(2^{-i}\right)\right]$$

# DFX VHDL Design



# DFX Barrel Shifter

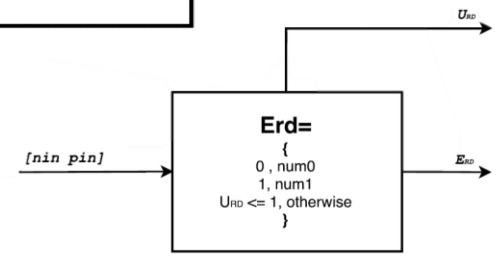



## Range Detector:

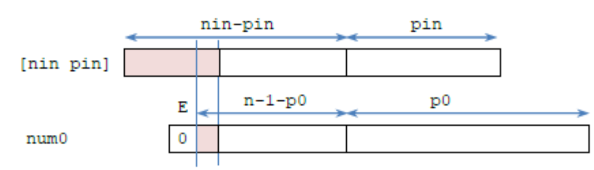
- It determines whether a fixed point (FX) number $[nin\ pin]$ can be represented as a DFX $num0$ number with $n$ bits. Note that if $E$=1, it does not necessarily mean that it is a $num1$ number with $n$ bits, because it might actually need more than $n$ bits.

- For the DFX number to be $num0$ with $n$ bits, the corresponding FX number has to be such that the $nin-pin-(n-1-p0)+1$ MSBs have be all 1 or 0 (due to sign extension). This means only one of those bits is needed.

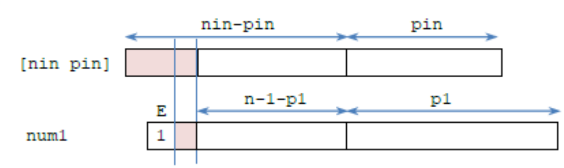
- The figure assumes that: $nin-pin \geq n-1-p0, p0 \geq pin$. If $nin-pin < -1-p0$ then the FX number is a $num0$ DFX number with $n$ bits. If $p0<pin$, we need to get rid of $p0-pin$ LSBs (we lose precision here).




$$E_{RD} = \overline{b_{nin-pin-1}\cdots b_{n-1-p0-1} + \overline{b_{nin-pin-1}\cdots b_{n-1-p0-1}}}$$




If not, we try with $num1$:




If not, it means we need more than $n$ bits.

# Error Analysis

- To measure the accuracy of the implemented DFX hyperbolic CORDIC, error analysis is performed on a different cases that will cover the two formats in the DFX representation (num0, num1).
- The results are compared with the ideal values obtained in MATLAB.
- The error measures will be:

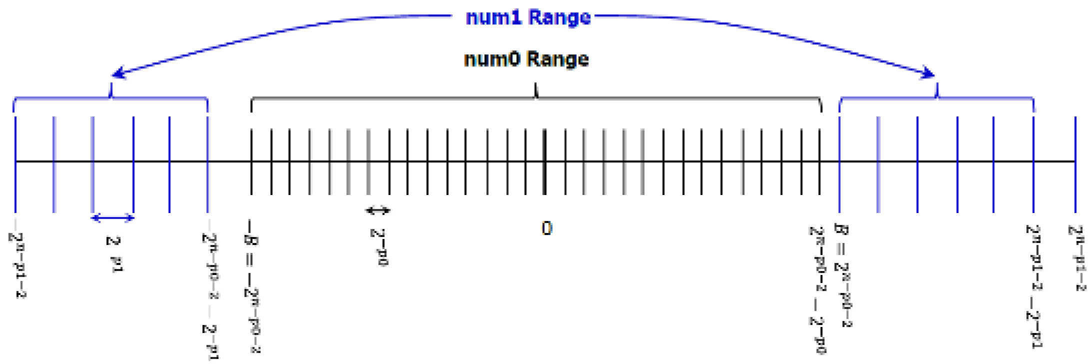
$$Relative\ Error = \frac{|\ ideal\ value - DFX\ CORDIC\ value|}{|ideal\ value|}$$

$$SNR = 20 * \log_{10} \frac{|\ ideal\ value\ |}{|\ ideal\ value - DFX\ CORDIC\ value\ |}$$

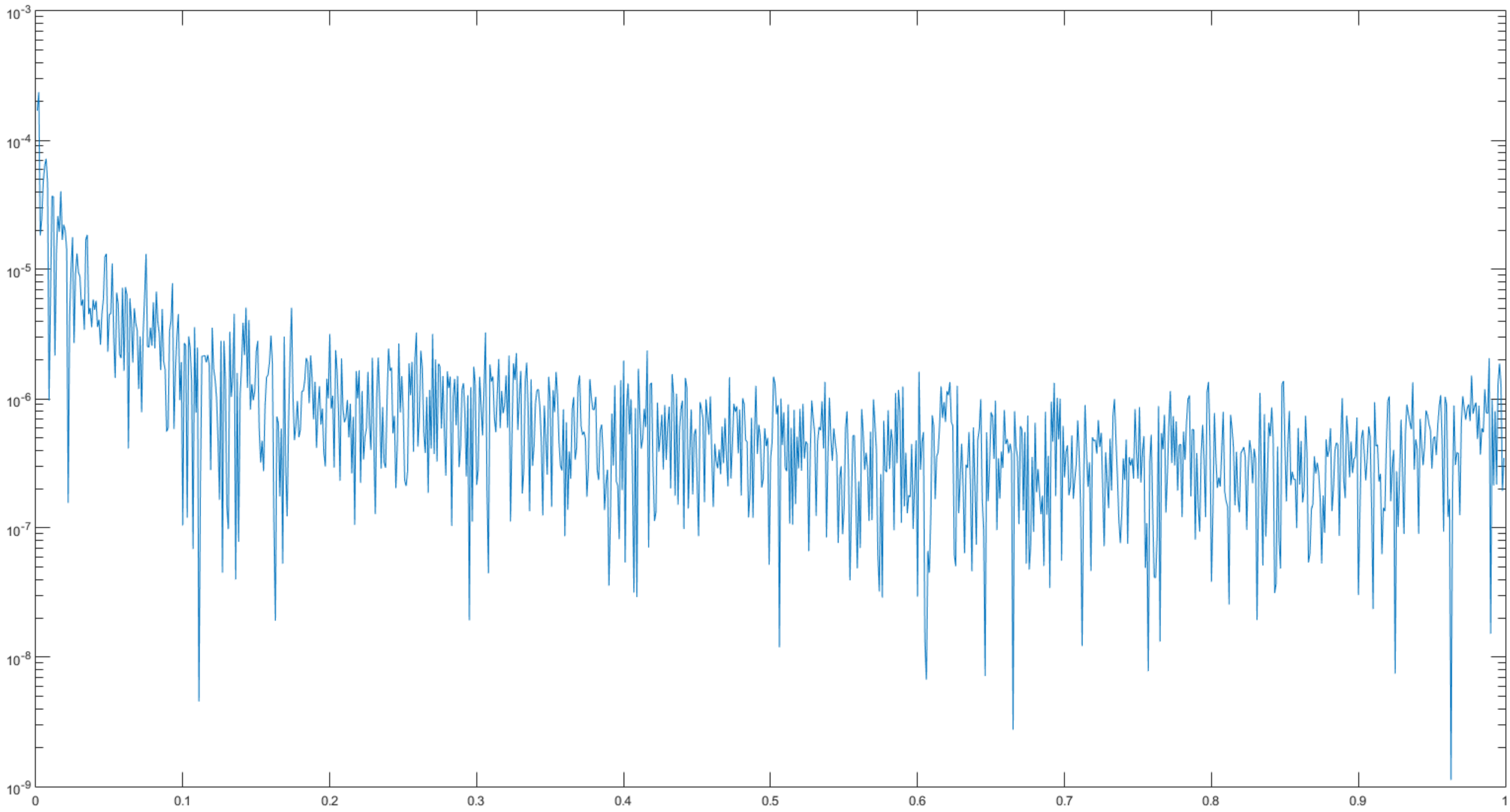
# Format & Range

- Format: [*n p0 p1*]
  - X,Y format [32 29 25]
  - Z format [32 27 25]

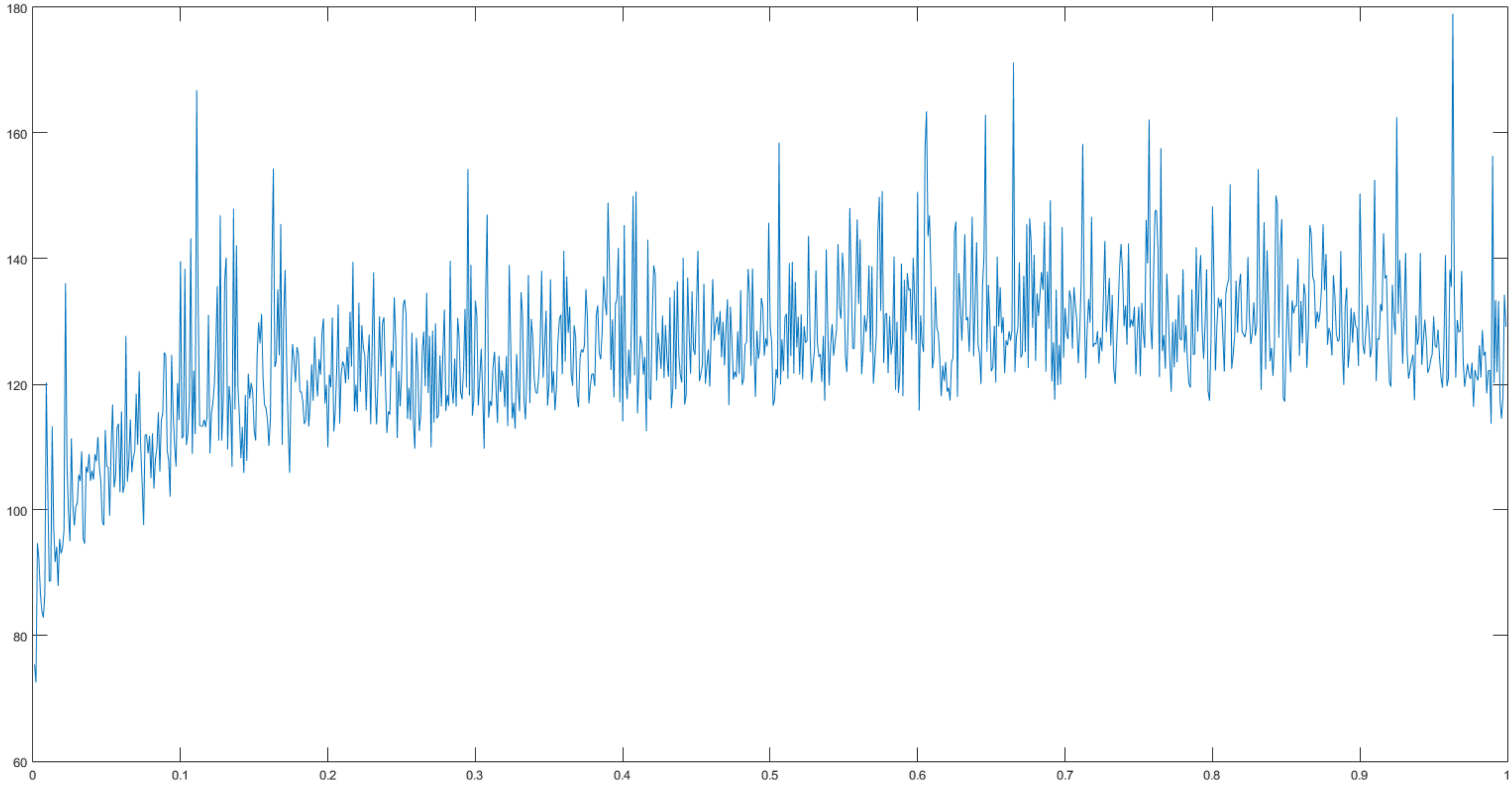
- Range
- X,Y:
  - Num0 : [-2,2)
  - Num1 : [-32,-2)U[2,32)
- Z:
  - Num0 : [-8,8)
  - Num1 : [-32,-8)U[8,32)



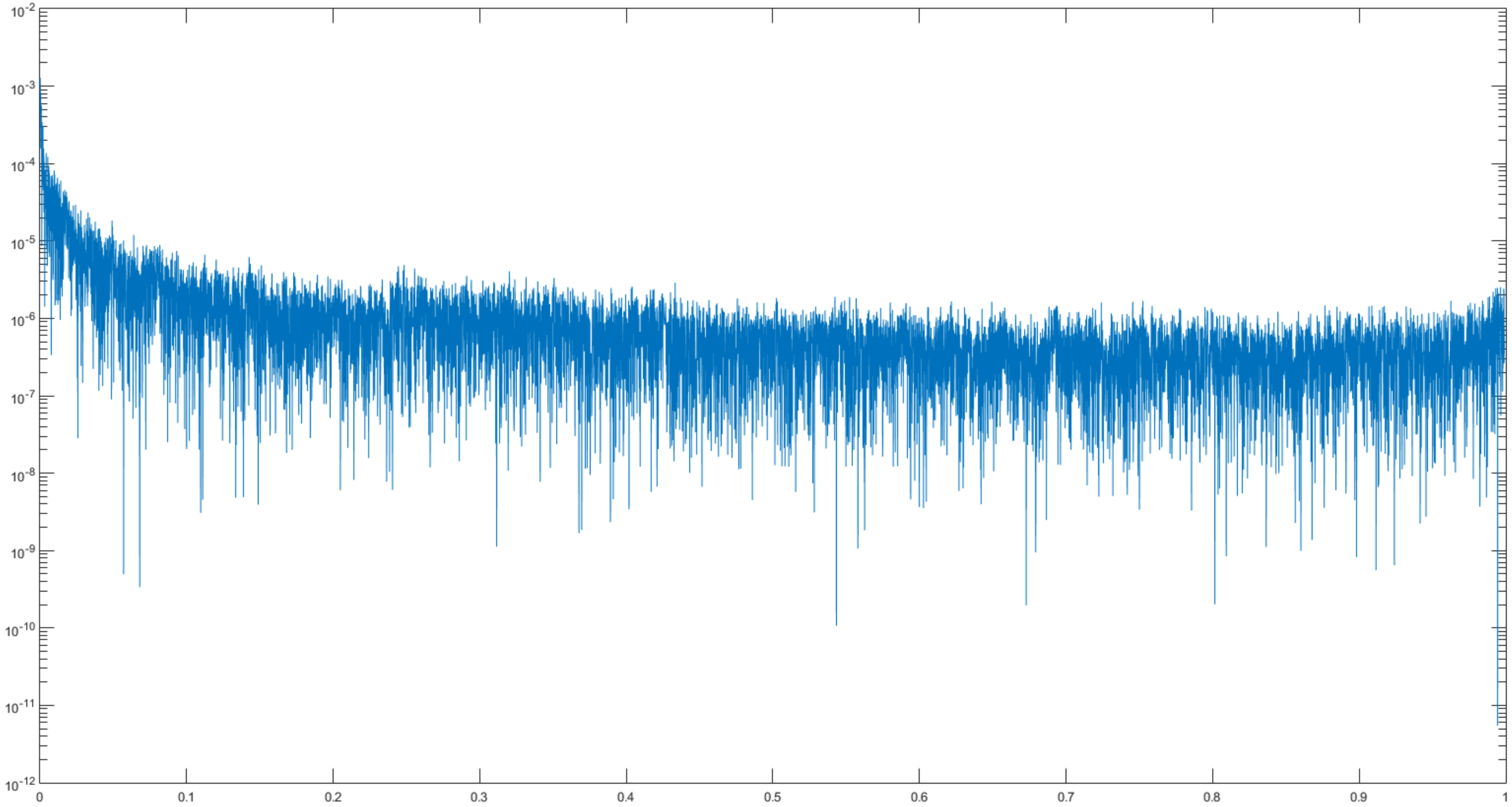
# Results-**Atanh**-Relative error [0 1) 1000 inputs



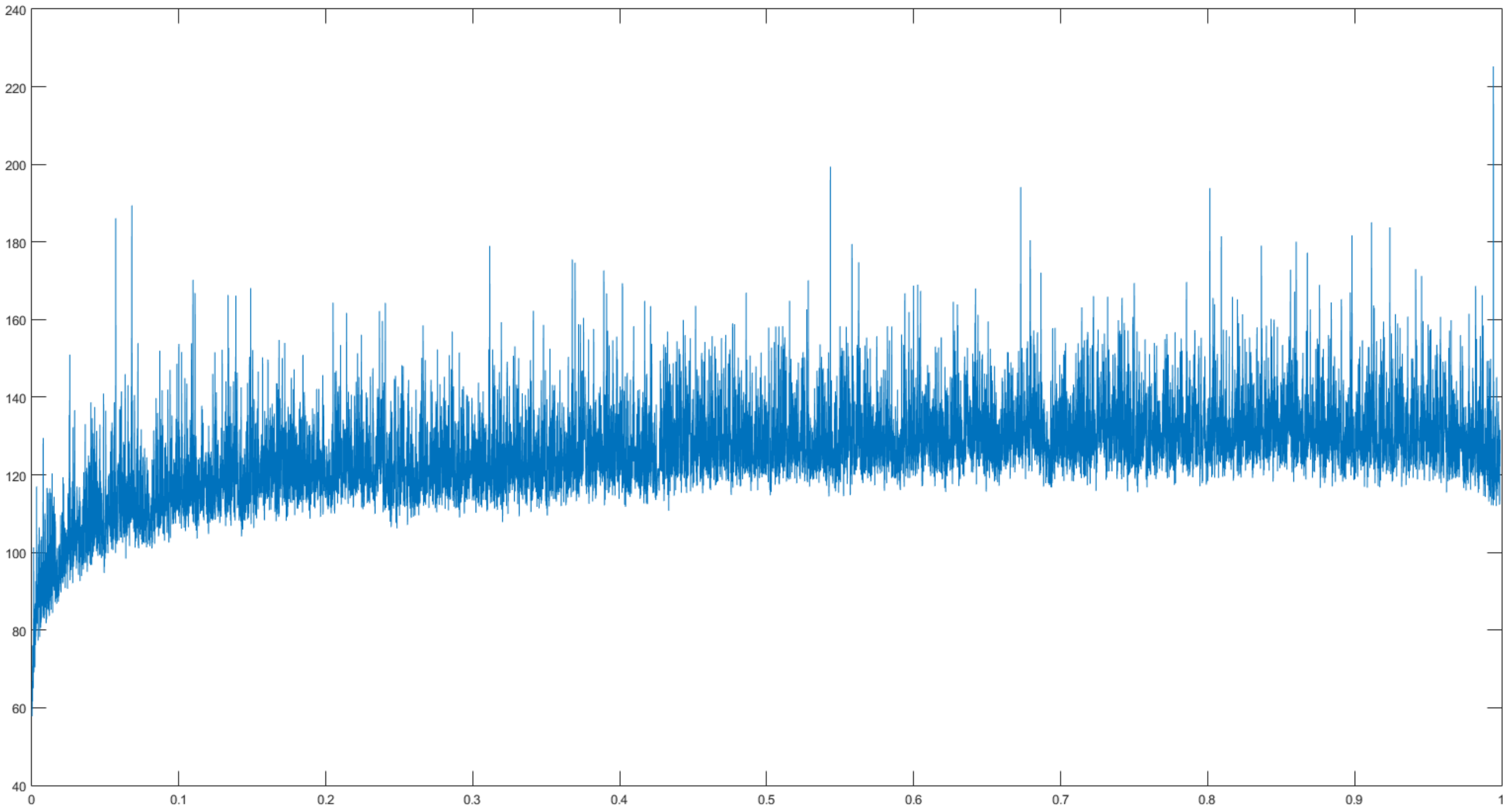
# Results-**Atanh**-SNR [0 1) 1000 inputs



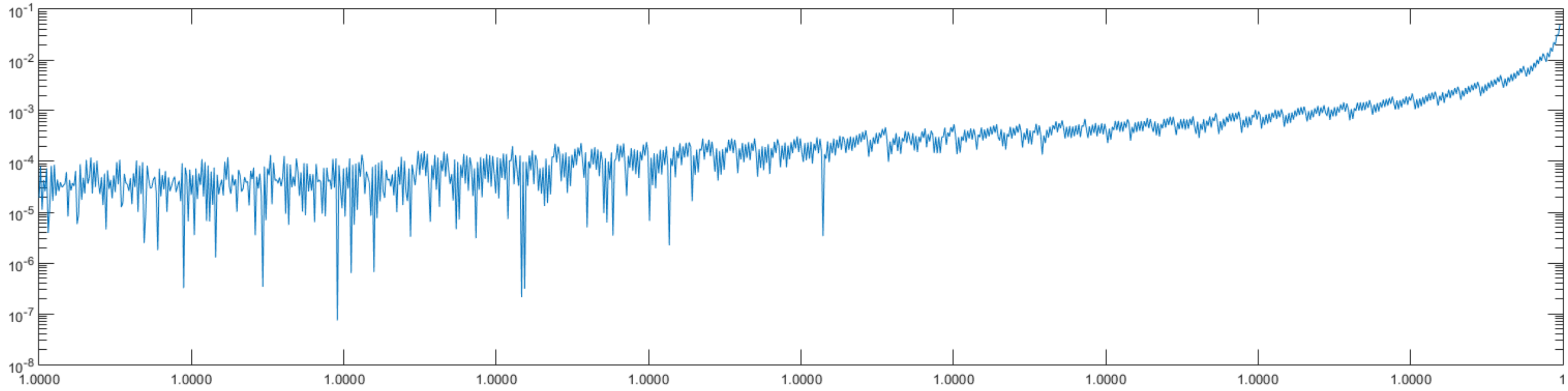
# Results-**Atanh**-Relative error [0 1) 10000 inputs

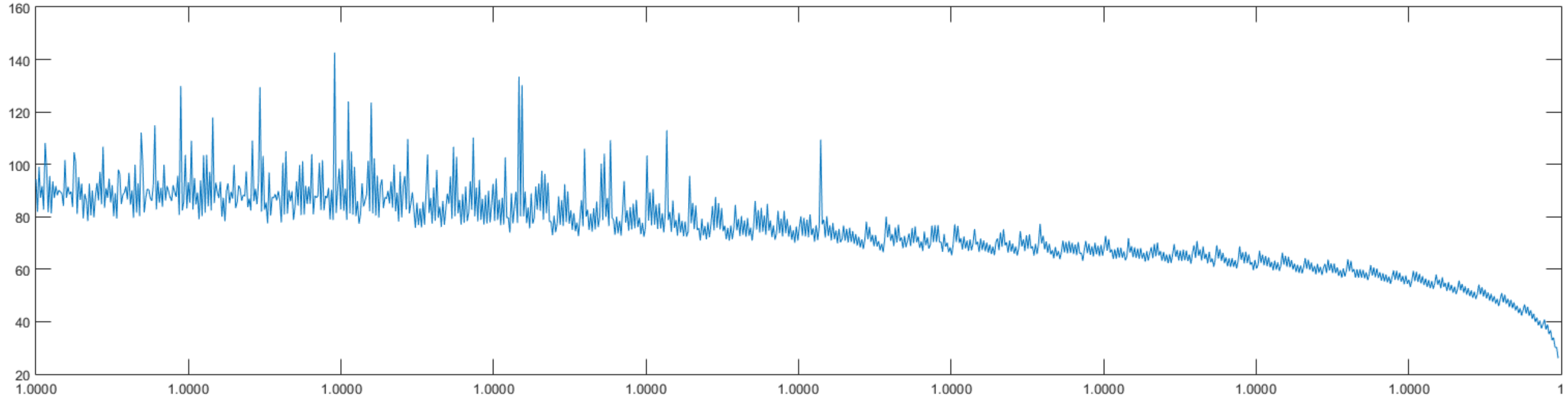


# Results-**Atanh**-SNR [0 1) 10000 inputs

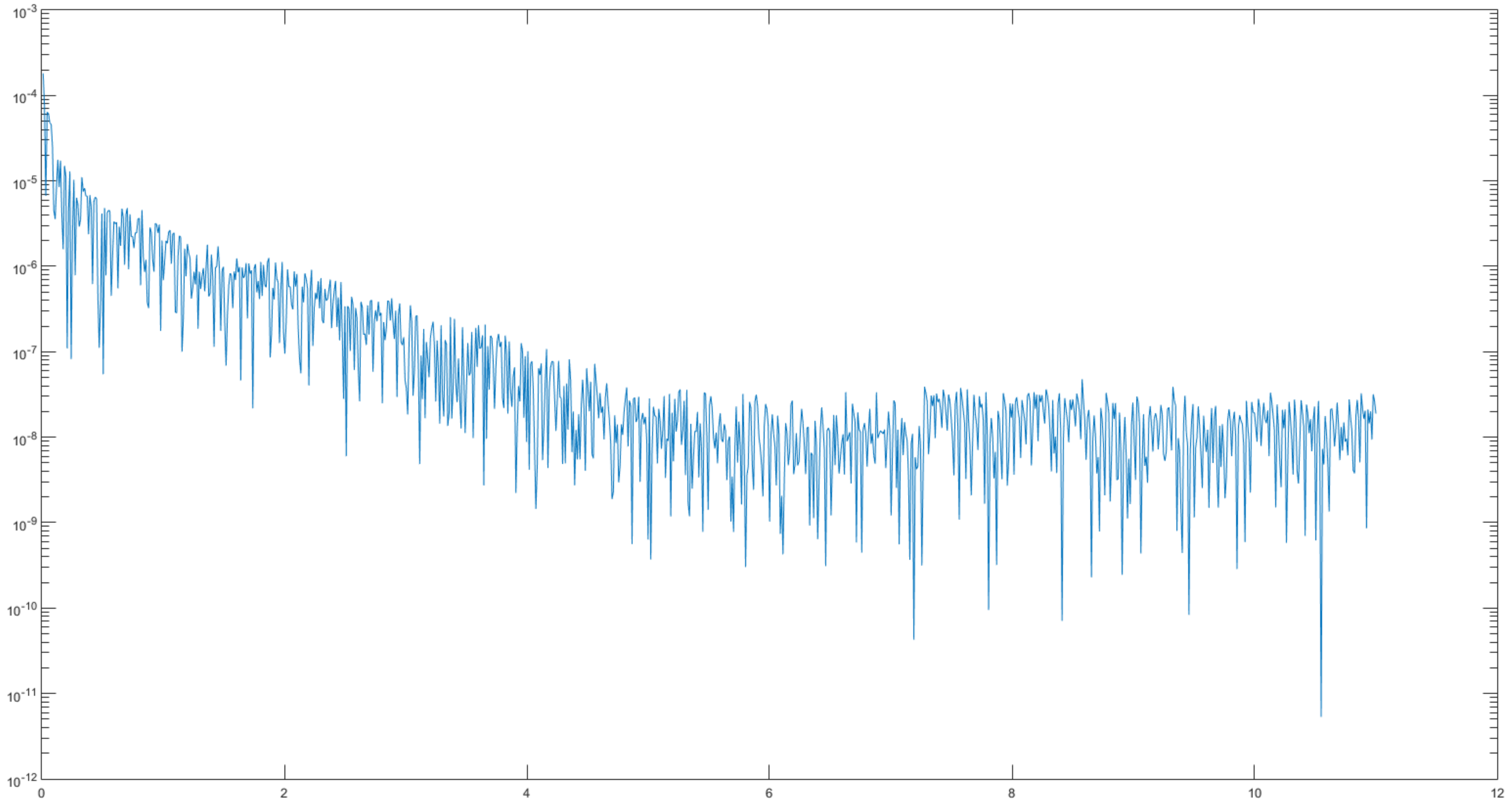


# Results-**Atanh**-Relative error and SNR (0.999999 to 0.999999998)

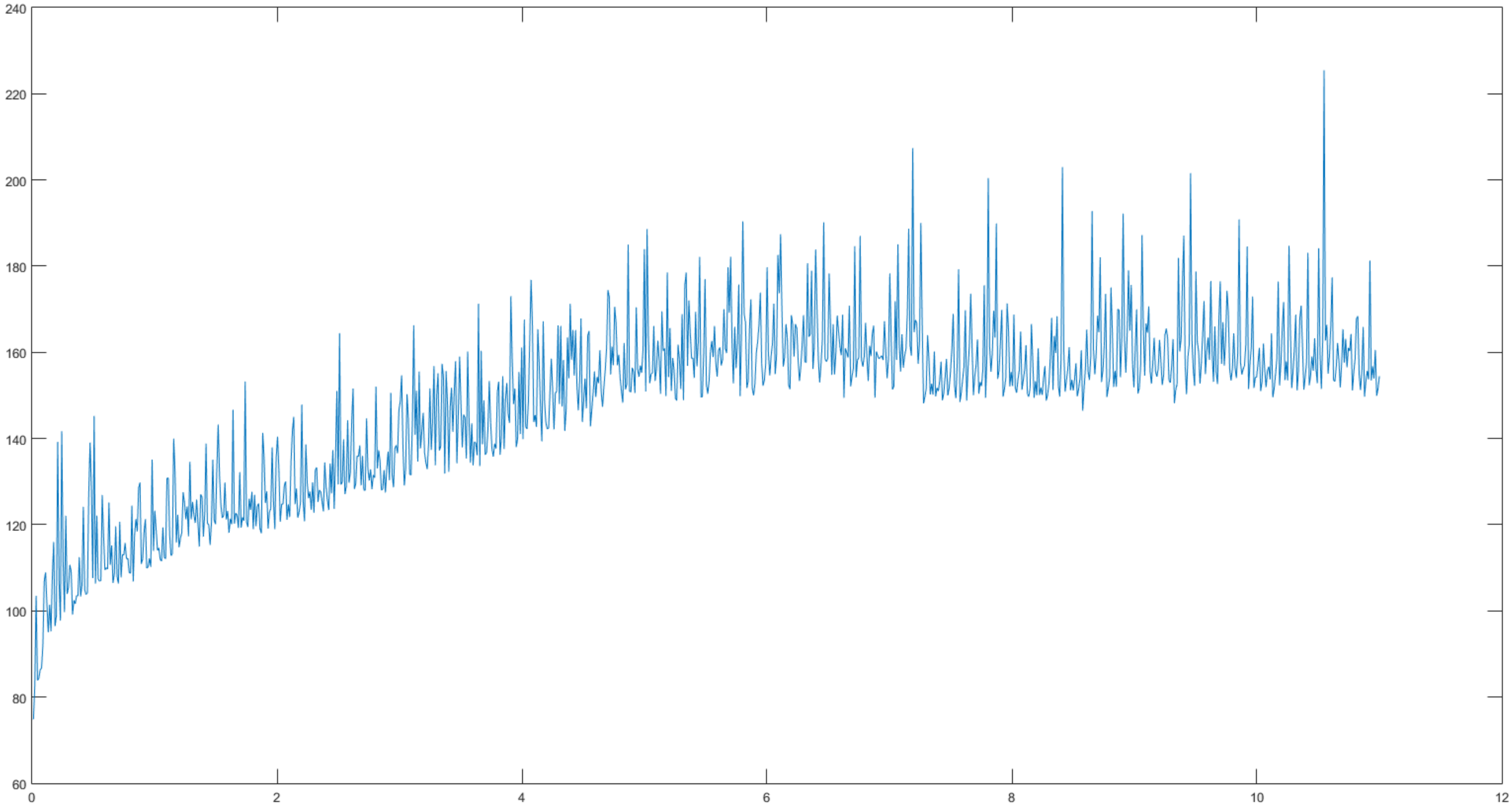


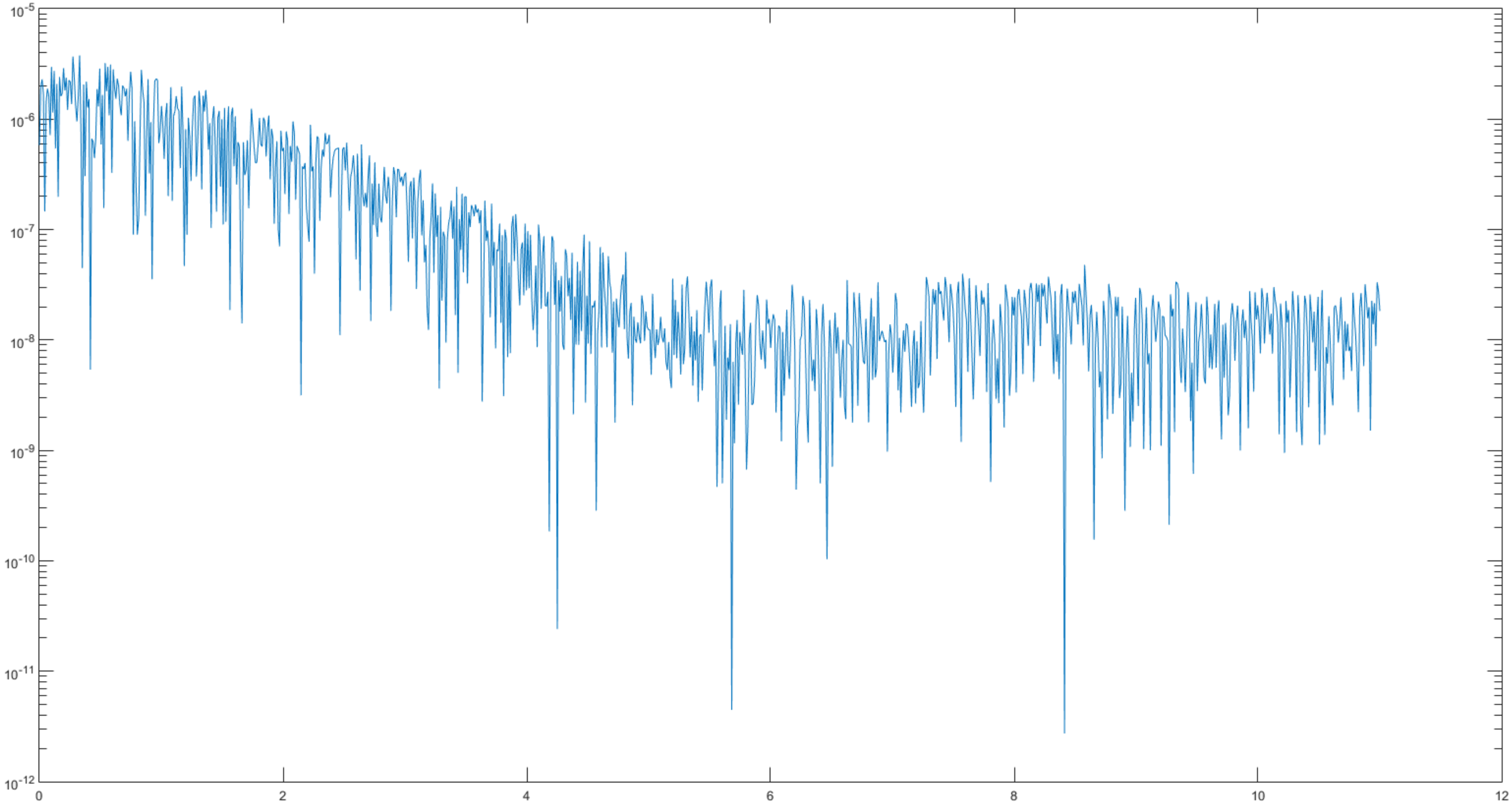
# Results-**Sinh**-Relative error [0 11] 1000 inputs



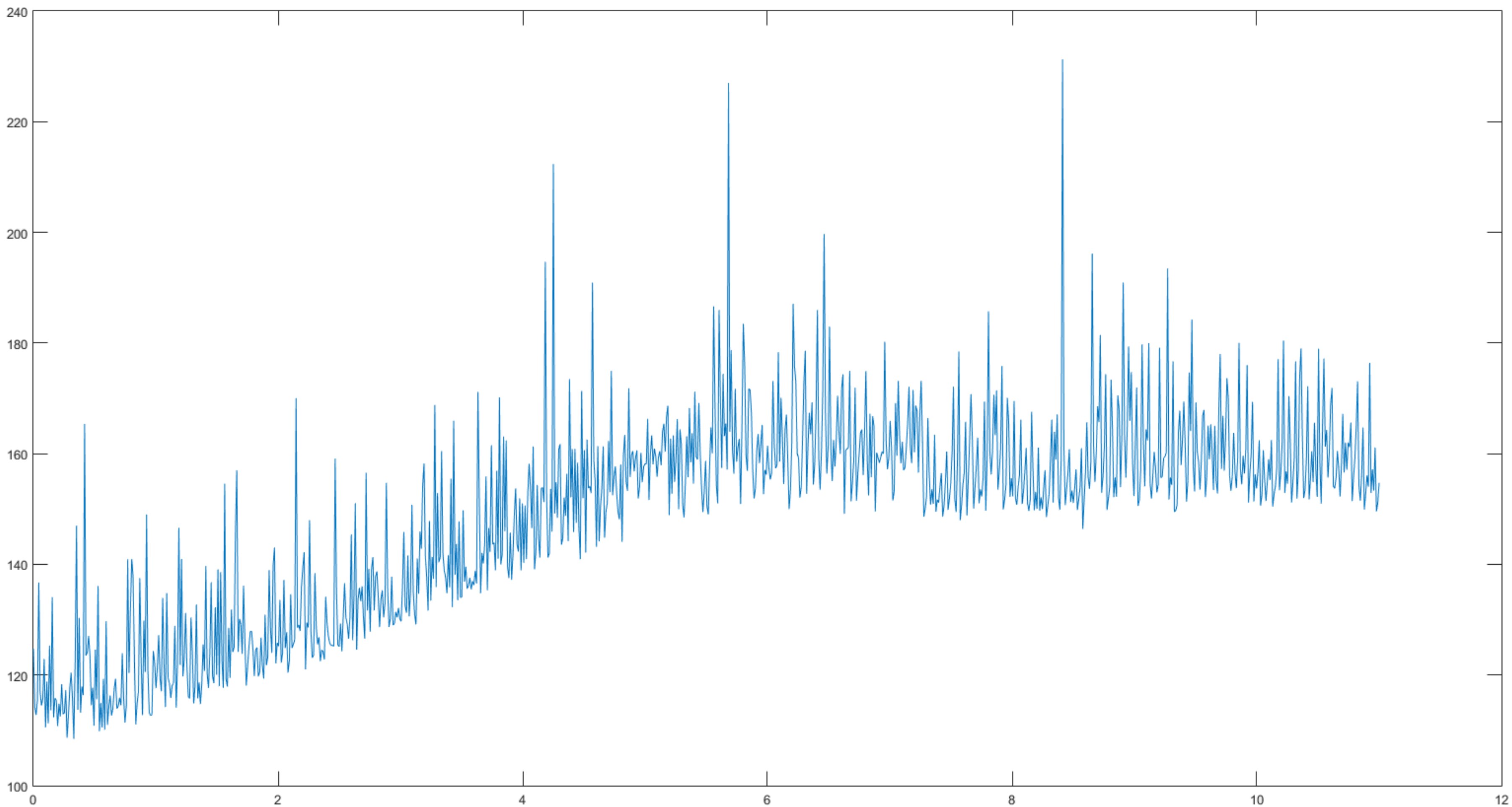
# Results-**Sinh**-SNR [0 11] 1000 inputs



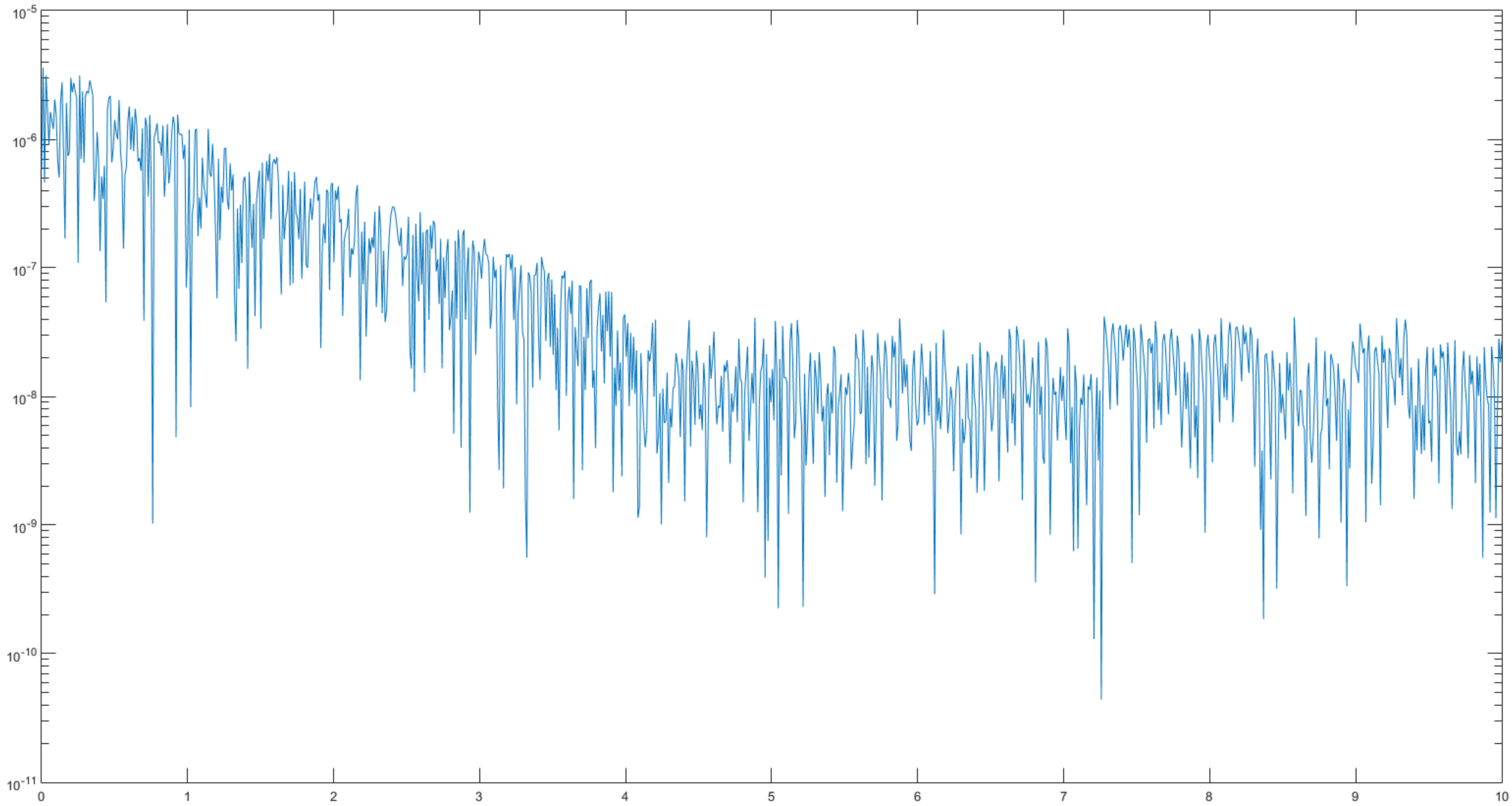
# Results-**Cosh**-Relative error [0 11] 1000 inputs



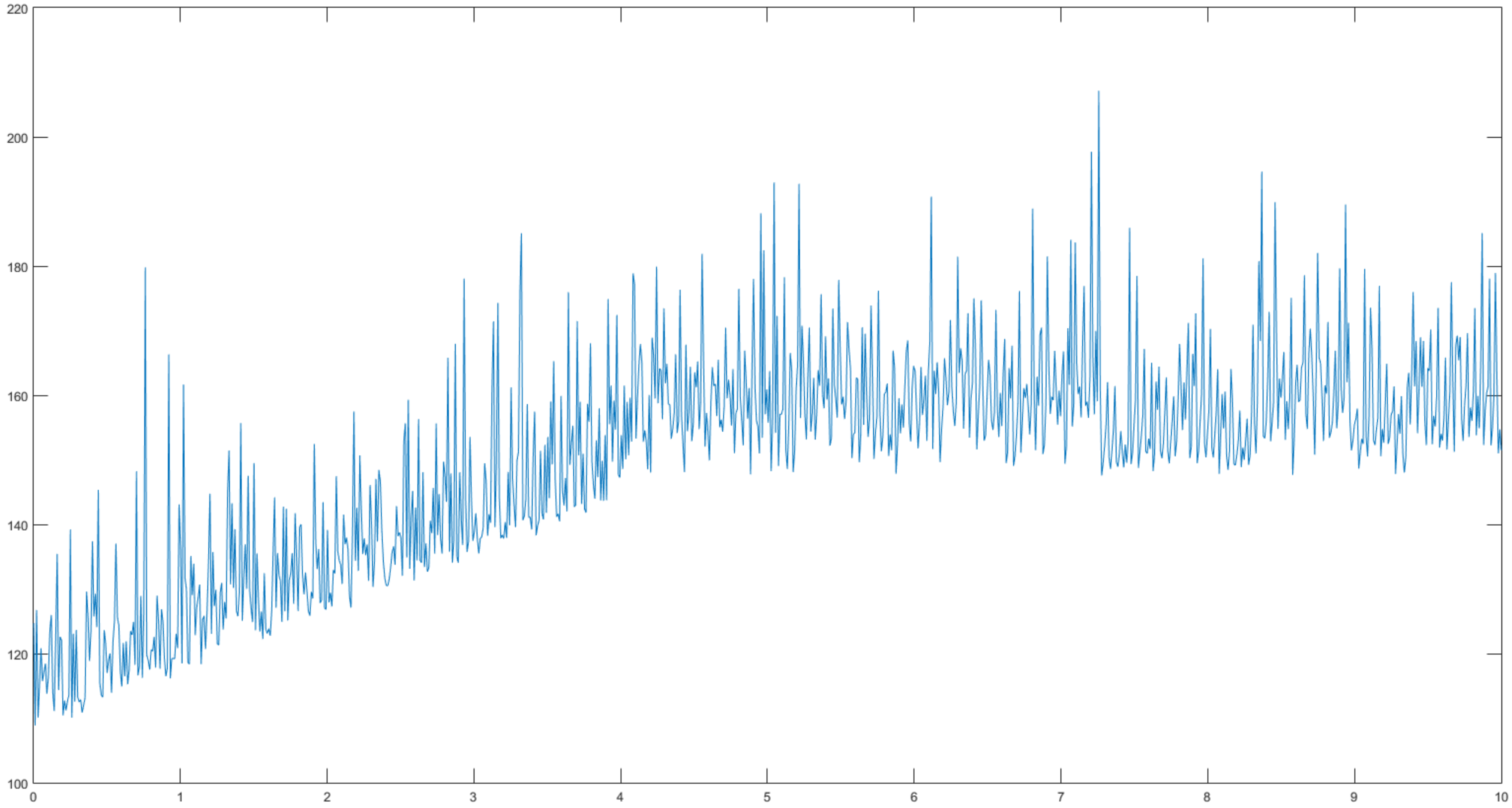
# Results-**Cosh**-SNR [0 11] 1000 inputs



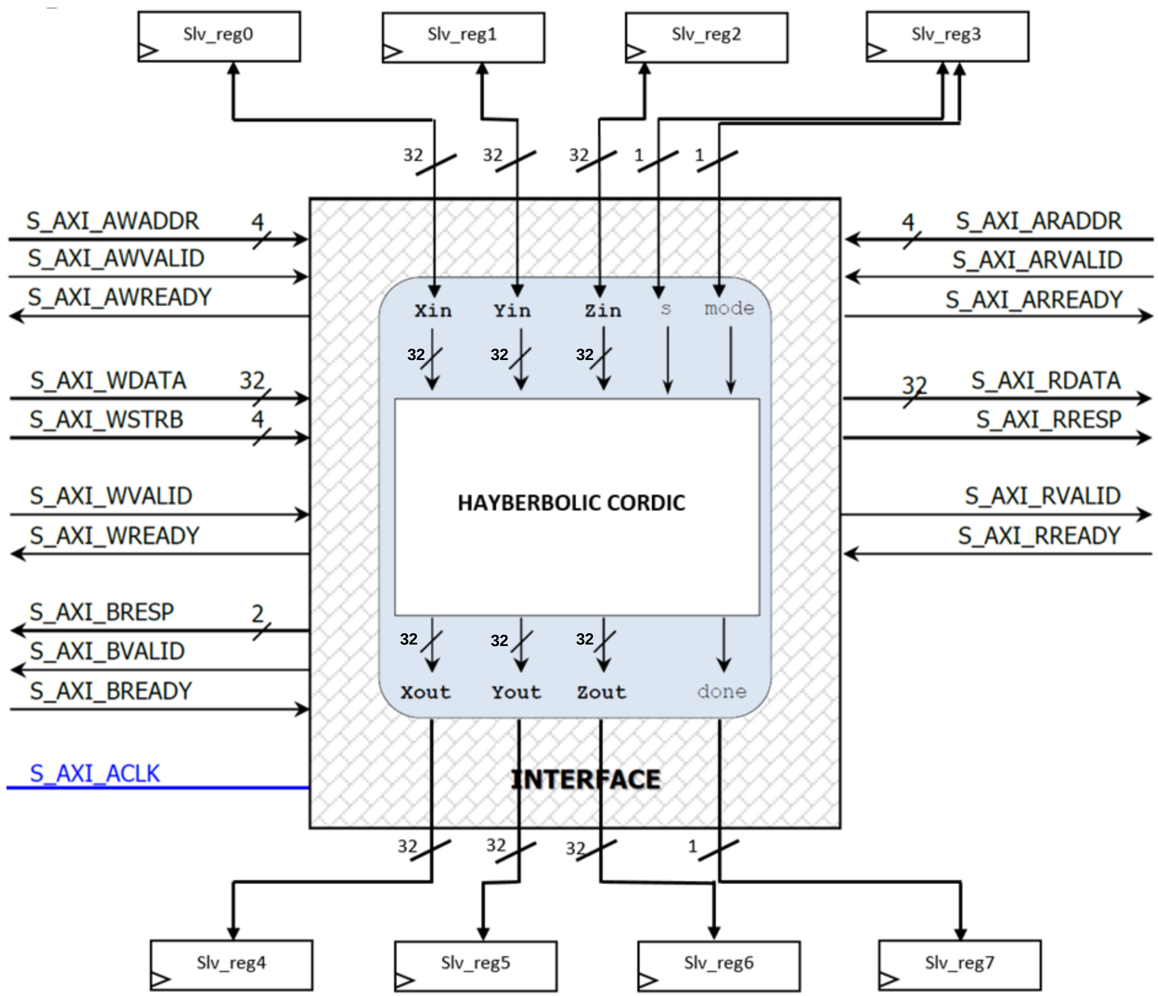
# Results-**Exp**-Relative error [0 10] 1000 inputs



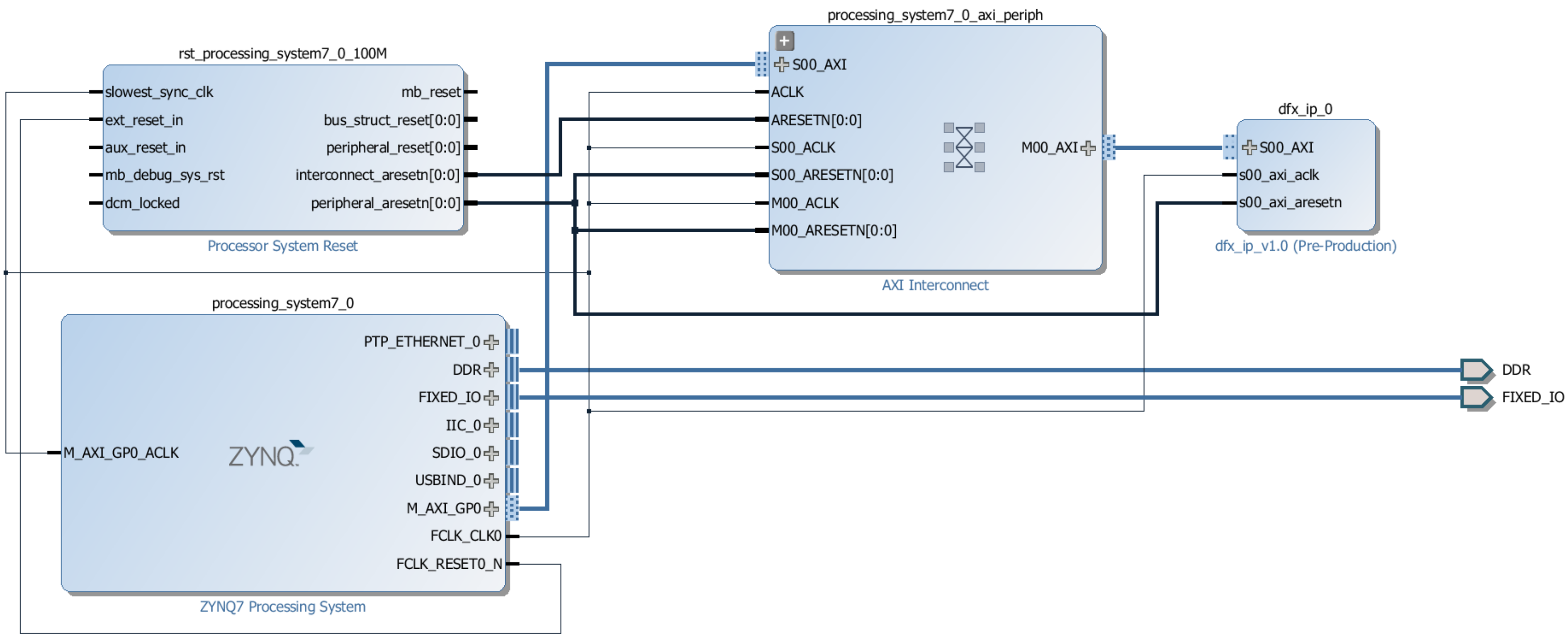
# Results-**Exp**-SNR [0 10] 1000 inputs



# IP Package and Peripherals



# IP Package and Peripherals



# Attention Areas

- Number of Iterations (positives and negatives).
- DFX format used.
- Irregularity due to:
  - Truncation of P0-P1 LSB bits when doing DFX Addition/Subtraction. Can be recovered.
  - Converting *Decimal* numbers to *Binary* and vice versa.

# References


- X. Hu, R.G. Harber, S.C. Bass, “Expanding the range of convergence of the CORDIC algorithm,” *IEEE Transactions on Computers*, vol. 40, no. 1, pp. 13-21, Jan. 1991.
- J. Becker, M. Platzner, S. Vernalde (Eds.): FPL 2004, LNCS 3203, pp. 200–208, 2004. c Springer-Verlag Berlin Heidelberg 2004.
- *D. Llamocca, C. Agurto,* “A FIXED-POINT IMPLEMENTATION OF THE EXPANDED HYPERBOLIC CORDIC ALGORITHM” *Lat. Am. appl. res. v.37 n.1 Bahía Blanca ene. 2007.*

# Thank You

- Any Questions??