AMD Vitis™ Design Implementation (GitHub)

AMD
together we advance_

Agenda

1. Introduction to GitHub
2. AMD Vitis™ DSP Library Overview
3. Introducing AMD Vitis Tools Tutorials on GitHub
4. What is a Polyphase Channelizer?
5. Demo

Introduction to GitHub

What is GitHub

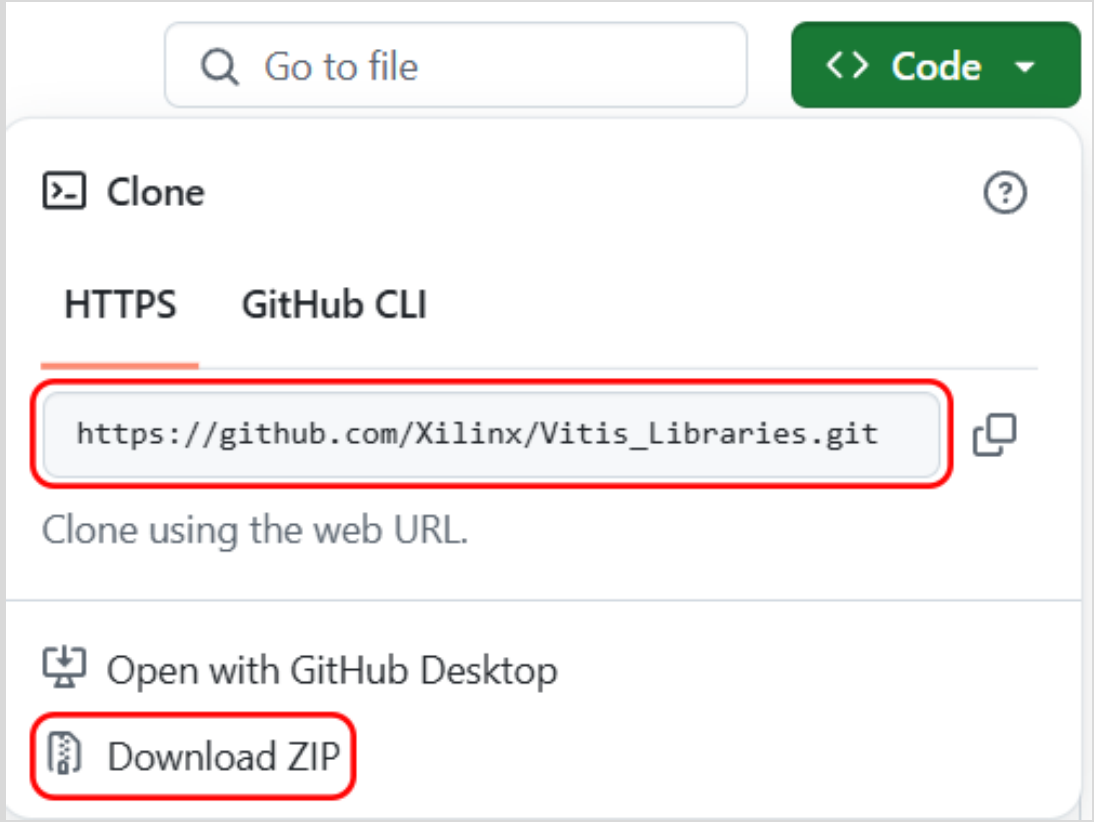
- GitHub is a web-based platform for version control and collaborative software development
- Uses Git for tracking changes in code
- Centralized repositories for storing projects

Key Concepts

Repository	Fork	Clone	Branch	Commit	Pull Request
Folder containing all project files	Create your own version of a project	Process of creating a local copy of a repository	Branches are parallel versions of a repository	Snapshot of the changes made to the files in a repository	Pull requests are a way to propose changes made on a branch to be merged into another branch

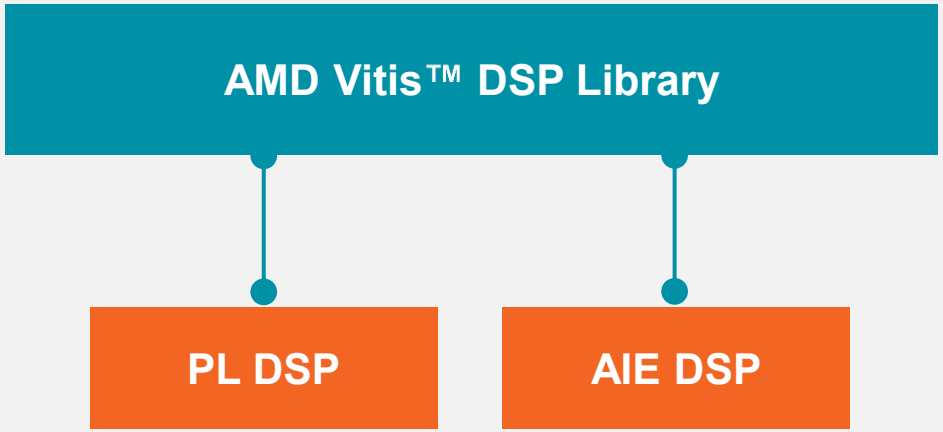
How to Download / Import Libraries from GitHub

- **Option 1:** Clone Repository
  - git clone
    - https://github.com/username/repo.git
- **Option 2:** Download ZIP
  - Code > Download ZIP > Extract files
- **Option 3:** GitHub Desktop (GUI tool)



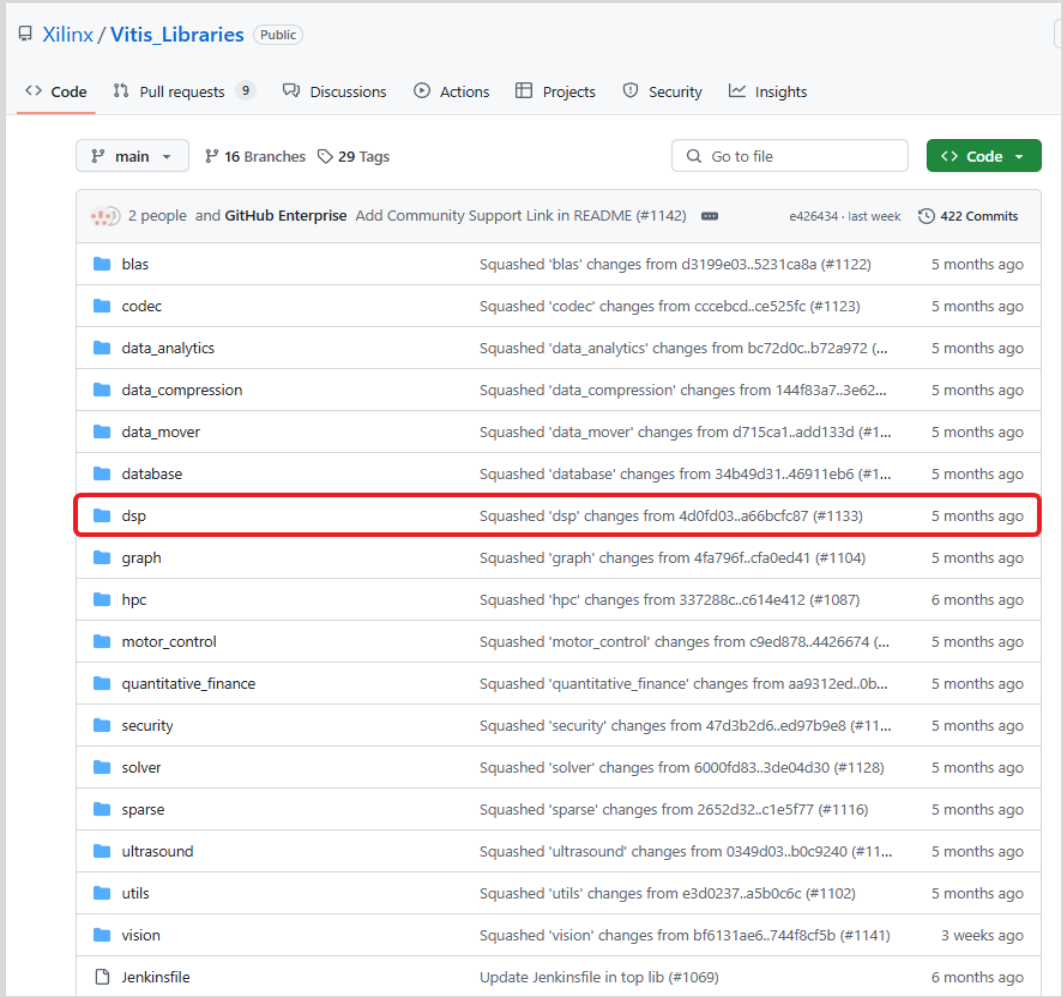
AMD Vitis™ DSP Library Overview

https://github.com/Xilinx/Vitis_Libraries



- Three types of implementations:
  - L1 PL primitives
  - L2 PL kernels

- Three types of implementations:
  - L1 AIE kernels
  - L2 AIE graphs and VSS makefiles



What Is the AMD Versal™ AI Engine DSP Library?

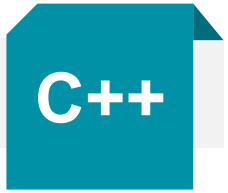
01

Configurable library of kernels to develop applications on AMD Versal™ AI Engines

02

Open-source library for DSP applications

03



Kernels are coded in C++, and intrinsics give access to AI Engine vector processing capabilities

04

Kernels can be combined to construct graphs for developing complex designs

05

Example design is also provided with this library

06

Use the library element's graph as the entry point

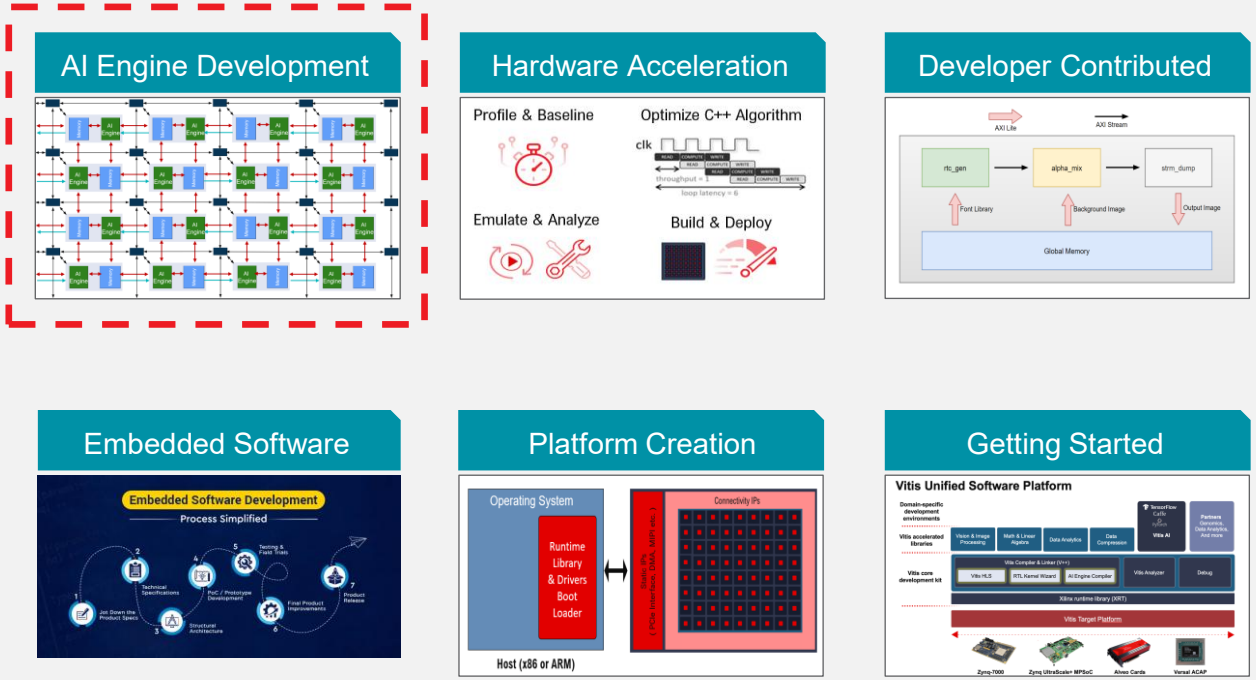
DSP Library Functions (Supports AIE and AIE-ML Devices)

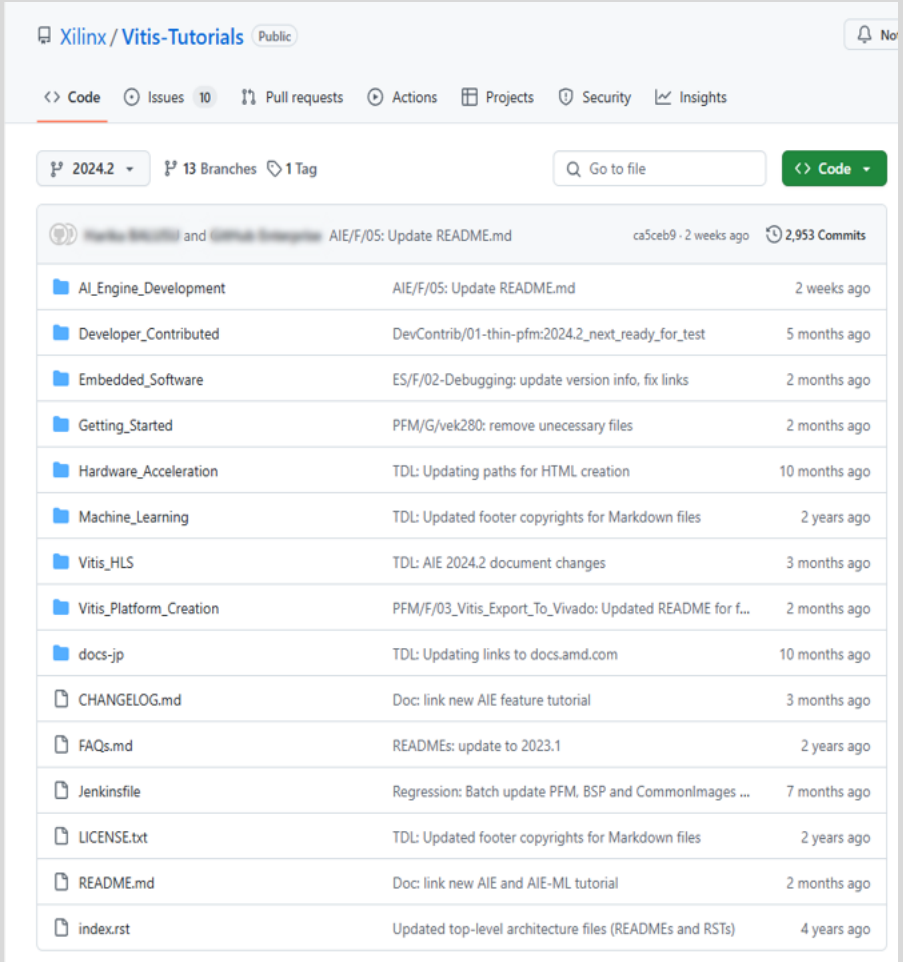
L2-AIE-DSP-Library-User-Guide

Function	Graph Entry Point
Bitonic Sort	xf::dsp::aie::bitonic_sort::bitonic_sort_graph
Convolution / Correlation	xf::dsp::aie::conv_corr::conv_corr_graph
DFT	xf::dsp::aie::fft::dft_graph
FIR TDM	xf::dsp::aie::fir::tdm::fir_tdm_graph
Function Approximation	xf::dsp::aie::func_approx
Hadamard Product	xf::dsp::aie::hadamard::hadamard_graph
Kronecker	xf::dsp::aie::kronecker::kronecker_graph
Matrix Multiply	xf::dsp::aie::blas::matrix_mult::matrix_mult_graph
Matrix-Vector Multiply	xf::dsp::aie::blas::matrix_vector_mul::matrix_vector_mul_graph
Mixed Radix FFT	xf::dsp::aie::fft::mixed_radix_fft_graph

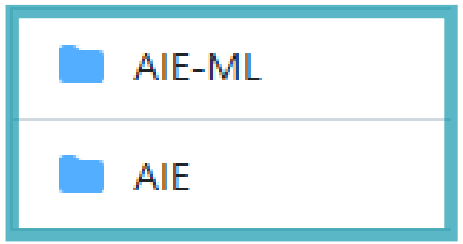
AMD Vitis™ Tools Developer Tutorials on GitHub

https://github.com/Xilinx/Vitis-Tutorials





AMD Versal™ AI Engine Development Categories



Categories	Description
AIE-ML	The tutorials in the AIE-ML folder help you learn how to target, develop, and deploy advanced algorithms using an AMD Versal™ AI Edge Device that features an AIE-ML array in conjunction with PL IP/kernels and software applications running on the embedded processors.
AIE	The tutorials under the AIE folder help you learn how to target, develop, and deploy advanced algorithms using a Versal AI Core Device that features an AIE array in conjunction with PL IP/kernels and software applications running on the embedded processors. To successfully deploy AI Engine applications in hardware, you need to be aware of the AMD Vitis™ and AI Engine tools and flows.

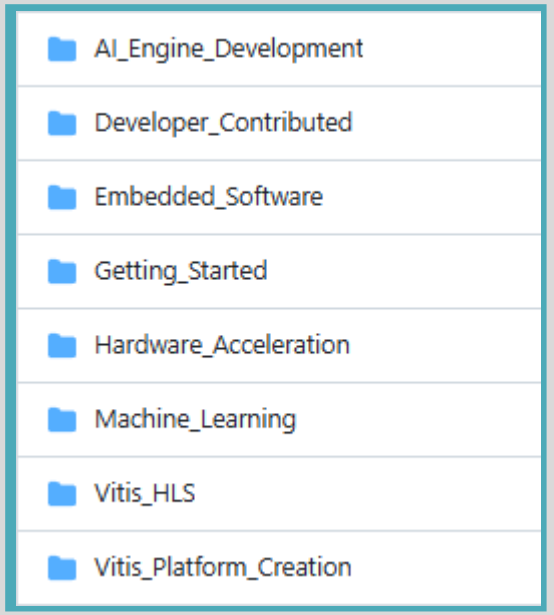
Tutorial Structures

- The tutorials are divided into categories, with each category containing two sections.

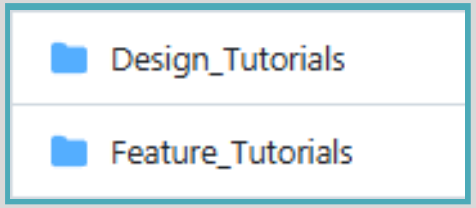
- **Feature Tutorials**
  - Tutorials illustrate a specific feature or some advanced features that not all designs require but are still useful in special use cases.

- **Design Tutorials**
  - Tutorials illustrate higher-level concepts or design flows, walkthrough specific examples or reference designs. More complex and full designs may use multiple features of this subject.

Category



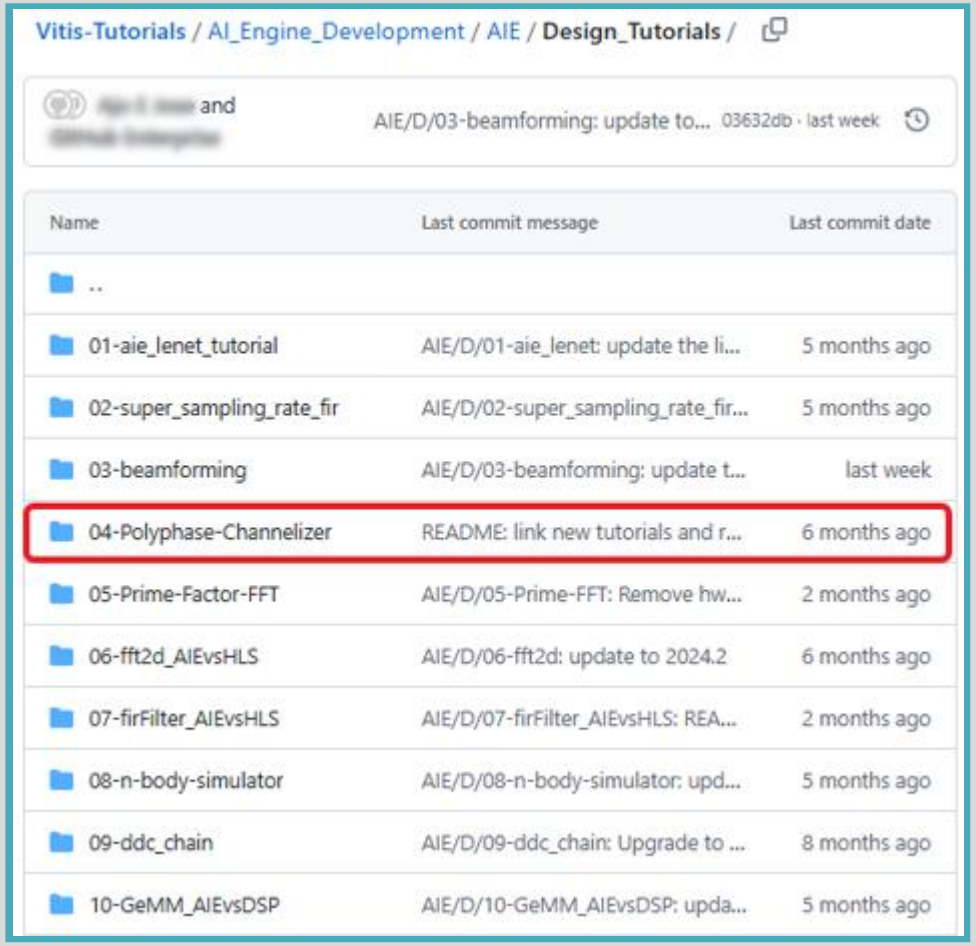
Section



Design Tutorials

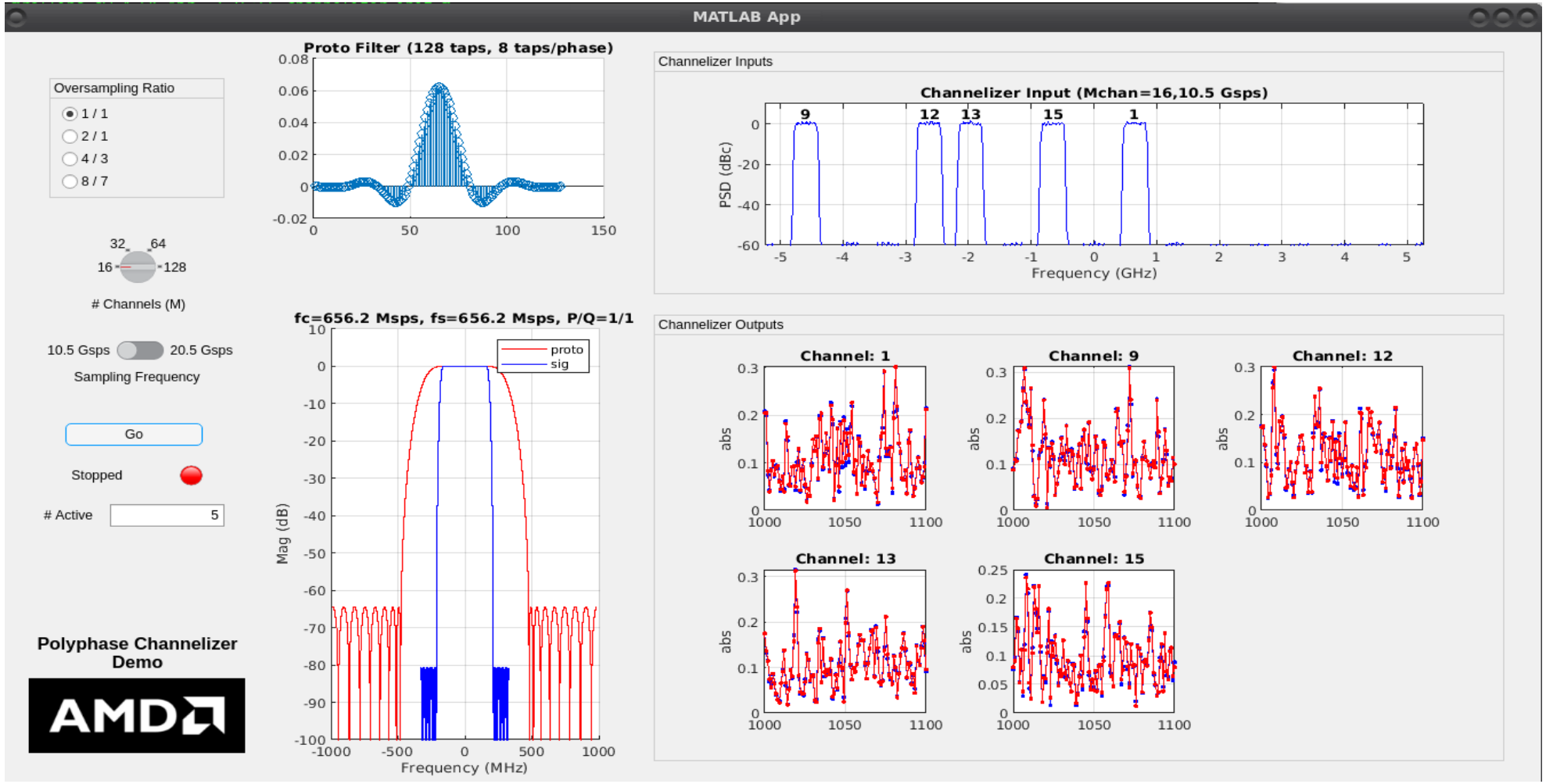
- The AI Engine Development Design Tutorials showcase the two major phases of AI Engine application development: architecting the application and developing the kernels.

- **Polyphase Channelizer**
  - This tutorial demonstrates an implementation of a system-level design (such as Polyphase Channelizer) using a combination of AI Engine and PL/HLS kernels.



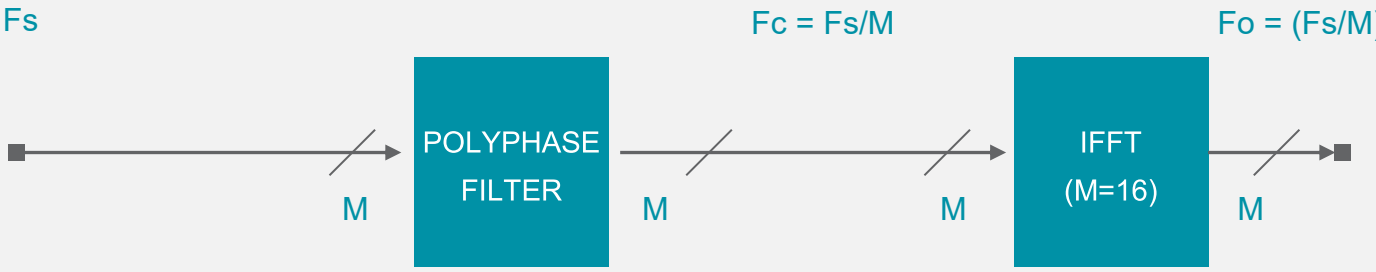
What is a Polyphase Channelizer?

Down-converts several frequency-division multiplexed signals in parallel to baseband

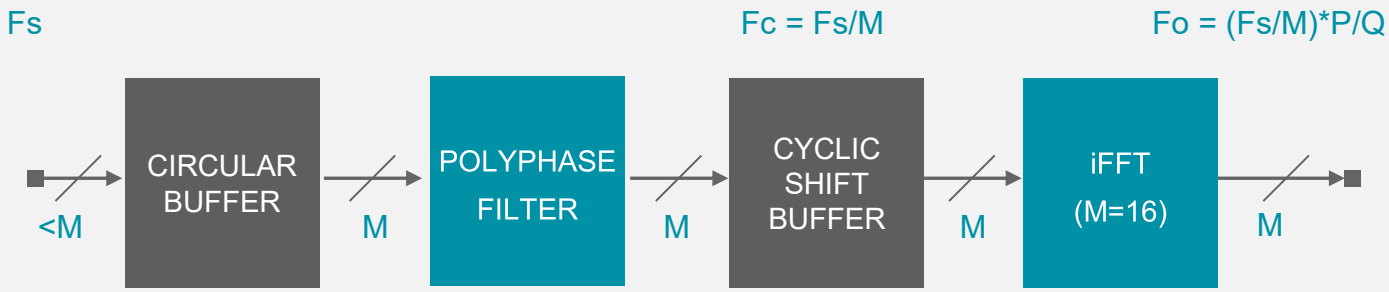


What is a Polyphase Channelizer?

Simple Case



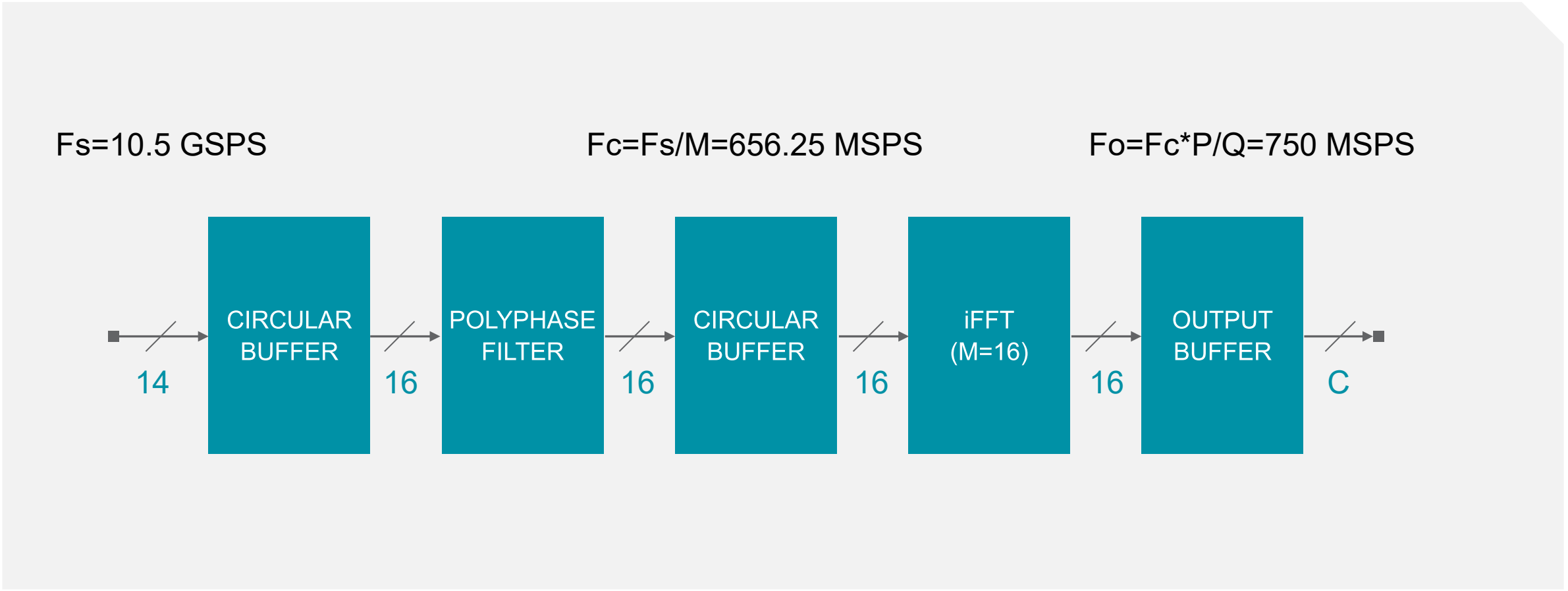
Fractional P/Q Oversampling



Channelizer Design Specifications

Parameter	Value / Units
Input Sampling Rate (Fs)	10.5 GSPS
# of Channels (M)	16 Channels
Interpolation Factor (P)	8 n/a
Decimation Factor (Q)	7 n/a
Channel Bandwidth	656.25 MHz
Output Sampling Rate	750 MSPS
# of Taps per Phase (K)	8 n/a

Block Diagram of the Polyphase Channelizer



DEMO

https://www.youtube.com/watch?v=4ndqWSCz4rw

Summary

01 GitHub is a web-based platform using Git for version control and to track changes and store projects in repositories.

02 The AMD Vitis™ Digital Signal Processing Library offers L1 and L2 elements optimized for AMD FPGAs, adaptive SoCs (PL DSP library), and Versal™ AI Engines (AI Engine DSP library).

03 AMD Vitis tool tutorials are available on GitHub in the Vitis Tutorials repository, covering various categories like AI Engine Development and Hardware Acceleration.

04 A polyphase channelizer uses DSP techniques to down convert multiple frequency-division multiplexed signals to baseband in parallel.

05 The video demonstrates downloading the Vitis Tutorials ZIP file from GitHub, extracting it in Ubuntu®, navigating to the polyphase channelizer design directory.

General Disclaimer and Attribution Statement 2025

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18u.


AMD