XMOS — британская компания, без собственного производства, занимающаяся разработкой многопоточных многоядерных процессоров, предназначенных для решения нескольких задач в режиме реального времени.

В компании XMOS разработали новое поколение 32-разрядных, многоядерных, многопоточных встраиваемых процессоров с возможностью легкого масштабирования через межпроцессорную шину, предназначенных для выполнения нескольких задач в реальном времени, цифровую обработку сигналов, управление различными процессами одновременно. Их отличает событийная модель управления процессами, плотность команд RISC-архитектуры, вычислительные возможности DSP и гибкость периферии FPGA. Многопоточная архитектура XMOS обеспечивает параллельное выполнение 8 задач одним ядром в реальном масштабе времени, при этом каждый поток выполняется не реже, чем раз в 11 нс.

К процессорам прилагаются готовые примеры решений и стартовые комплекты.

Отдельная особенность XMOS — возможность объединения нескольких процессоров благодаря межпроцессорной связи XLink,обеспечивающей связь с быстродействием более 1 гигабита в секунду (gbps) при решении одной сложной задачи на нескольких ядрах.

Разработка программного обеспечения осуществляется с помощью среды разработки XDE (на основе платформы Eclipse), которая имеет: инструменты для симуляции и отладки программы, логический анализатор, приложение xScope, утилиты для работы с платами, возможность работы с помощью командной строки или с помощью графического интерфейса. Правда здесь стоит оговориться, что на данный момент xTIMEcomposer не поддерживает дэвборду XK-EVK-XU316,которую мы взяли в работу. Для этого производитель предлагает свою тулу для поддержки новых чипов - XTC Tools 15.1.4 (https://www.xmos.ai/software-tools/), которую рекомендуют прикрутить к Eclipse, либо VS Code.

У производителя также есть свободный доступ к библиотекам модулей, готовых решений для работы со звуком и периферией (<https://github.com/xmos/xcore_sdk>). Для того чтобы скачать репу нужно будет завести SSH ключ на гит хабе.

Разработка программы осуществляется на языках С, C++ или XC. ХС является версией языка С, разработанной XMOS.

В нём используется тот же самый синтаксис и большая часть типов данных. ХС, в отличие от языка С, обеспечивает удобную работу с параллелизмом, взаимосвязью, синхронизацией и портами, а также поддерживает событийно-управляемое программирование, в том числе через каналы.

Для исследования были использован платы: расширенный вариант XK-EVK-XU316 с дебагером xtag4 на борту и расширенной переферией, а также шилд для малинки … (<https://www.xmos.ai/download/XVF3610-Product-Brief(1.0).pdf)> для использования в качестве звукового устройства, с набором всего функционала о котором заявлено производителем. Прошивка представлена в двух версиях XVF3610–INT (для RPI4) и XVF3610–UA (USB устройство, поддерживается в Windows, Linux & Mac) можно скачать по следующей ссылке https://www.xmos.ai/vocalfusion-voice-interfaces.

Результаты можно посмотреть в приложенном аудиофайле, в котором записан голос(Audacity) с микрофонов в процессе воспроизведения музыкального стрима через динамик, который используется устройством в качестве рефернсного для подавления эха которое попадает в микрофон и и выделения речи говорящего. А также условно тэстилась система автомотической подстройки усиления микрофонов при одинаковой тональности говорящего на разном расстоянии с шагом в один метр.

Описание работы устройства.

Шилдт общается с хостом по шине I2S, при сэмплировании данных с частотой 48кГц и битрэйтом 32 бита. Сэмплирование можно понизить до 16кГц, но для этого нужно сгенирировать data\_partition\_int\_factory\_vxx\_xx\_xx.bin из специального конфигурационного json файла, который заливается специальной утелитой во флэш вместе с прошивкой

Прошивка XVF3610–INT по умолчанию работает при следующих характеристика

# Установка среды разработки

При попытке настроить среду разработки под XMOS мы столкнулись с проблемой нахождения программатора XTAG4. В диспетчере устройств он распознается, но возникает проблема с XTAG3 сервисом

Эта проблема решалась следующим образом:

1. Устанавливаем WSL2 (<https://learn.microsoft.com/ru-ru/windows/wsl/install>), по умолчанию вероятнее что он есть, если установлен Docker
2. Устанавливаем USBIPD-WIN и USBIP (<https://learn.microsoft.com/ru-ru/windows/wsl/connect-usb>)
3. Отключаем файэрвол, если возникает следующая проблема
4. Проверяем что XTAG4 определяется
5. Открываем [\\wsl.localhost\Ubuntu\home\user](file:///\\wsl.localhost\Ubuntu\home\user) на локальном ПК и сюда устанавливаем XTC Tools 15.1.4
6. Здесь же создаем папку в которой будем вести разработку, стягиваем tamplate с проектом vs code и подмодулем xcore\_sdk.
7. Через Ubuntu терминал открываем следующий путь XMOS/XTC/15.1.4, запускаем скрипт для инициализации переменных среды и из него же запускаем VS Code

Для разработки выбрана VS Code (<https://code.visualstudio.com/download>) . Ecnfyf