

Министерство образования и науки Российской Федерации

КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

КАФЕДРА РАДИОЭЛЕКТРОНИКИ

Специальность: 011800.62 — радиофизика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(Бакалаврская работа)

**РАЗРАБОТКА ЛАБОРАТОРНОГО ПРАКТИКУМА ПО СОЗДАНИЮ
ГЕНЕРАТОРОВ СТАНДАРТНЫХ СИГНАЛОВ НА БАЗЕ ПРОГРАММИРУЕМОЙ
АНАЛОГОВОЙ ИНТЕГРАЛЬНОЙ СХЕМЫ ANADIGM AN221K04-DVLP2**

Работа завершена:

"__" _____ 2016 г. _____ (Р.С. Гордеев)

Работа допущена к защите:

Научный руководитель

К. ф.-м. н., доцент

"__" _____ 2016 г. _____ (И.А. Насыров)

Заведующий кафедрой

Д. ф.-м. н., профессор

"__" _____ 2016 г. _____ (М.Н. Овчинников)

Казань — 2016

СОДЕРЖАНИЕ

Введение

1 Программируемая микроэлектроника

1.1 Программируемые логические интегральные схемы

- 1.1.1 История программируемых логических интегральных схем
- 1.1.2 Классификация программируемых логических интегральных схем
- 1.1.3 Сложные программируемые логические устройства
- 1.1.4 Программируемые пользователем вентильные матрицы
- 1.1.5 Преимущество программируемых логических интегральных схем

1.2 Программируемые аналоговые интегральные схемы

- 1.2.1 История программируемых аналоговых интегральных схем
- 1.2.2 Программируемые аналоговые интегральные схемы компании Anadigm
- 1.2.3 Обобщенная архитектура программируемых аналоговых интегральных схем Anadigm AN221k04
- 1.2.4 Конфигурируемые аналоговые блоки программируемых аналоговых интегральных схем Anadigm AN221k04
- 1.2.5 Назначение и структура конфигурационного интерфейса программируемых аналоговых интегральных схем Anadigm AN221k04

2 Основные схемотехнические решения

2.1 Схемы на переключаемых конденсаторах

2.2 МОП структура интегратора

3 Разработка генераторов стандартных сигналов

3.1 Система автоматизированного программного обеспечения AnadigmDesigner2

3.2 Динамическое программирование аналоговых схем Anadigm алгоритмическим методом

3.3 Снятие характеристик генераторов стандартных сигналов при помощи программно-аппаратной среды NI ELVIS II

3.4 Исследование осциллограмм сигналов

3.5 Исследование спектра сигналов

3.6 Исследование амплитудно-частотной характеристики Bode

Заключение

Литература

Приложение А

ВВЕДЕНИЕ

В последние годы интегральные микросхемы с программируемой пользователем внутренней конфигурацией прочно вошли в современную электронику. Прежде всего, это относится к изделиям цифровой техники, таким как программируемые логические интегральные схемы, микроконтроллеры и сигнальные процессоры. Однако на практике в большинстве случаев приходится иметь дело с реальными аналоговыми сигналами, которые нужно усиливать, фильтровать, оцифровывать и т.д. Поэтому аналоговая электроника в современных системах занимает не менее важное место, чем цифровая. В первую очередь это относится к системам промышленной автоматики и управления, медицинскому оборудованию, контрольно-измерительной аппаратуре и аудиотехнике. Традиционно схемы аналоговой обработки сигналов выполняются на дискретных компонентах — операционных усилителях, компараторах, мультиплексорах и т. п. При этом в ряде случаев аналоговая часть занимает большую часть площади печатной платы и имеет высокую стоимость. Решить проблему создания разнообразных аналоговых устройств, снизив стоимость и габариты, позволяет использование программируемой логической интегральной схемы — ПЛИС. В настоящее время в различных областях науки и техники широко применяются системы на кристалле с процессорными блоками на основе программируемых логических интегральных схем. В таких системах вычислительные операции разделяются так, чтобы процессор выполнял функции управления алгоритмически сложные и в относительно медленные вычисления. Операции, требующие максимального быстродействия, выполняются аппаратно на ресурсах ПЛИС и ПАИС. Для выполнения таких операций требуются блоки для вычисления функций.

Программируемые аналоговые микросхемы могут стать незаменимым инструментом в приложениях, для которых необходимо низкое время отклика при обработке аналоговых сигналов с высокой точностью: это PID- регуляторы, аналоговые фильтры высокого порядка, аналоговые нейронные сети и т.п. Сложные аналоговые устройства, требующие точной обработки аналогового сигнала, можно полностью реализовать на одной микросхеме ПАИС Anadigm. Причем параметры схемы мы можем менять (в том числе в реальном времени) программно, без изменения топологии печатной платы. Также САПР AnadigmDesigner2 с понятным и дружелюбным интерфейсом, который имеет в своем составе полный набор библиотек и инструментов для проектирования аналоговых схем. Рассмотренный способ создания блоков генерации стандартных сигналов функций позволяет для ряда практических случаев получить выигрыш по точности при относительно простой аппаратной реализации.

Целью выпускной квалификационной работы является разработка методических указаний для выполнения лабораторного практикума на базе аналоговой интегральной схемы Anadigm AN221k04.

Для достижения поставленной цели следует выполнить следующие задачи:

1. Изучить структуру программируемой аналоговой интегральной схемы Anadigm AN221k04.
2. Изучить структуру системы автоматизированного проектирования AnadigmDesigner2.
3. Спроектировать генераторы синусоидальных, прямоугольных и пилообразных сигналов в системе автоматизированного проектирования.
4. При помощи программно-аппаратной среды NI ELVIS II снять амплитудно-частотные характеристики стандартных сигналов.

1 ПРОГРАММИРУЕМАЯ МИКРОЭЛЕКТРОНИКА

1.1 Программируемые логические интегральные схемы

1.1.1 История программируемых логических интегральных схем

По мере развития цифровых микросхем возникло противоречие между возможной степенью интеграции и номенклатурой выпускаемых микросхем. Экономически оправдано было выпускать микросхемы средней интеграции, таких как регистры, счетчики, сумматоры. Более сложные схемы приходилось создавать из этих узлов. Разместить более сложную схему на полупроводниковом кристалле не имело проблем, но это было оправдано либо очень большой серийностью аппаратуры, либо ценой аппаратуры (военная, авиационная или космическая). Заказные микросхемы не могли удовлетворить возникшую потребность в миниатюризации аппаратуры. Решение могло быть только одним — предоставить разработчикам аппаратуры возможность изменять внутреннюю структуру микросхемы по средством программирования.

История развития программируемых логических интегральных схем (ПЛИС) начинается с появления программируемых постоянных запоминающих устройств. Первое время программируемые ПЗУ использовались исключительно для хранения данных, однако вскоре их стали применять для реализации цифровых комбинационных устройств с произвольной таблицей истинности. В качестве недостатка подобного решения следует отметить экспоненциальный рост сложности устройства в зависимости от количества входов. Добавление одного дополнительного входа цифрового устройства приводит к удвоению требуемого количества ячеек памяти ПЗУ. Это не позволяет реализовать многовходовые комбинационные цифровые схемы. [1]

1.1.2 Классификация программируемых логических интегральных схем

В настоящее время программируемые логические интегральные схемы развиваются по нескольким направлениям, поэтому возникла необходимость как то различать эти микросхемы. Классификация программируемых логических интегральных схем (ПЛИС) приведена на рисунке 1.



Рисунок 1- Классификация программируемых логических интегральных схем (ПЛИС)

Следует отметить, что программируемые логические матрицы (ПЛМ) реализуют хорошо известные принципы создания цифровой комбинационной схемы. Применение постоянных

запоминающих устройств (ПЗУ) в качестве комбинационной схемы позволяет вообще обойтись без составления комбинационной функции и ее минимизации. Области применения этих микросхем сразу определились. ПЗУ применялись для создания комбинационных схем с малым количеством входов. При росте количества входов сложность внутреннего устройства ПЗУ и его цена резко возрастали (по квадратичному закону). ПЛМ позволяли реализовывать таблицы истинности с относительно малым количеством единичных сигналов на выходе и большим количеством входных сигналов, либо хорошо минимизирующиеся логические функции.

Первоначально цифровые устройства с памятью реализовывали либо на нескольких ПЛМ, либо добавляли внешние триггеры, затем стали включать их в состав программируемых логических устройств. По мере увеличения степени интеграции микросхем возникла необходимость объединять полученные сложные микросхемы на одном кристалле. В результате возникли сложные программируемые логические устройства (CPLD – Complex Programmable Logic Device). В них появилась возможность программировать не только таблицу истинности комбинационного устройства, но и линии соединения входов и выходов между собой. Таким образом можно считать CPLD дальнейшим развитием ПЛМ.

Точно так же развивались цифровые устройства на ПЗУ. Сначала к их выходам добавили триггер. Естественно предоставили возможность отключать его при необходимости. Затем несколько этих схем разместили на одном кристалле и предоставили возможность программировать соединения их входов и выходов между собой. Подобные устройства называются FPGA.

Для реализации цифровых комбинационных устройств с большим числом входов были разработаны программируемые логические матрицы (ПЛМ). В иностранной литературе они получили название — Programmable Logic Arrays (PLA). Именно программируемые логические матрицы можно считать первыми программируемыми логическими интегральными схемами (Programmable Logic Devices — PLDs). ПЛМ получили широкое распространение в качестве первых универсальных микросхем большой интеграции.

Первым представителем большого класса программируемых логических устройств (ПЛИС) стали программируемые логические матрицы (ПЛМ). В зарубежной литературе они называются PLA — ProgrammablelogicArray. Обобщенная структура ПЛМ приведена на рисунке 2. [2]

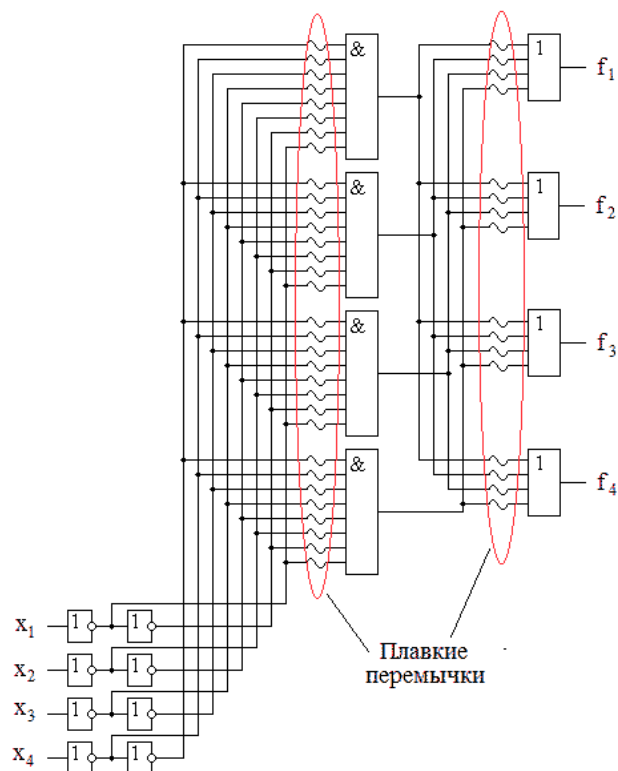


Рисунок 2 - Обобщенная структура программируемых логических матриц [2]

Основная идея работы ПЛМ заключается в реализации логической функции, представленной в СДНФ — дизъюнктивной нормальной форме. На рисунке 2 четко прослеживаются логические элементы "И", способные реализовать любой минтерм СДНФ и логические элементы "ИЛИ", осуществляющие суммирование термов, требующихся по логическому выражению СДНФ.

Следует отметить, что полная принципиальная схема ПЛМ получается достаточно громоздкой (рисунок 2). Поэтому в зарубежной литературе обычно применяется шинное представление проводников. Логический элемент "И", реализующий минтерм СДНФ, изображается как одиночная горизонтальная строка с условно-графическим обозначением схемы "И". К входам этого элемента подводится многоразрядная шина, а на выходе подключен одиночный проводник. Если входной проводник подключается ко входу логического элемента "И" (перемычка сохранена), то это место обозначается крестиком 'x', а если соединение отсутствует (перемычка сожжена), то крестик не проставляется. Аналогично обозначаются и многовходовые элементы "ИЛИ". Пример подобного изображения схемы ПЛМ (PLA) приведен на рисунке 3. [3]

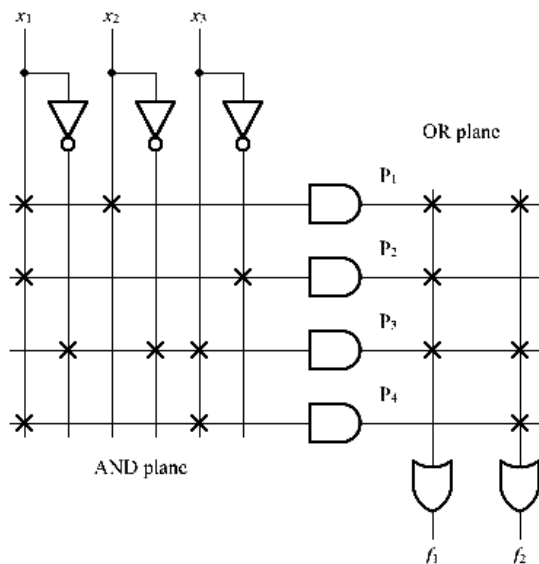


Рисунок 3 - Представление внутренней структуры схем ПЛМ [3]

По схеме четко можно восстановить реализуемую им логическую функцию. На рисунке 3 реализованы две логические функции f_1 и f_2 :

$$f_1 = x_1x_2 + x_1\bar{x}_3 + \bar{x}_1\bar{x}_2x_3$$

$$f_2 = x_1x_2 + \bar{x}_1\bar{x}_2x_3 + x_1x_3$$

Недостаток рассмотренной архитектуры ПЛМ — слабое использование ресурсов программируемой матрицы логических элементов "ИЛИ". Данное обстоятельство привело к появлению ещё одного вида программируемых матриц логики.

ПЛИС, в которых матрица логических элементов "И" является программируемой, а матрица логических элементов "ИЛИ" фиксированной, получили название программируемых матриц логики ПМЛ или PAL — Programmable Array Logic. Так как изготовить такие устройства проще, чем ПЛМ, они имеют меньшую стоимость и обладают улучшенными характеристиками, что привело к их высокой популярности. Пример ПМЛ с тремя входами, четырьмя логическими элементами "ИЛИ", реализующими четыре конъюнктивного одночлена шестого ранга и двумя выходами представлен на рисунке 4.

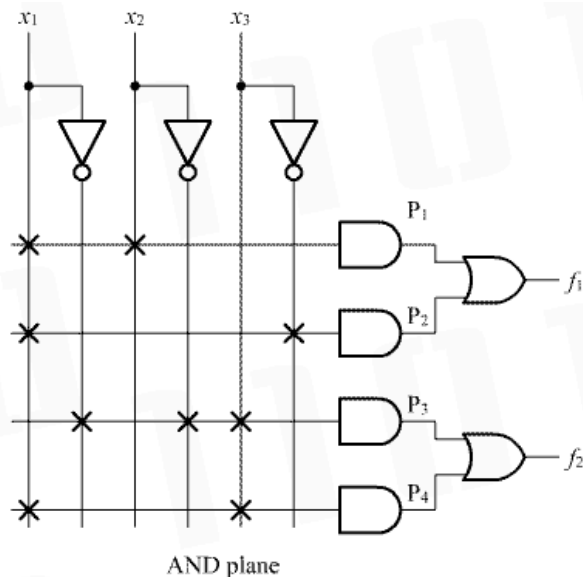


Рисунок 4 - Обобщенная структура программируемых матриц логики [4]

В схеме, приведенной на рисунке 4, действует следующее соглашение: Количество входов у логических элементов "И" равно количеству проводников на входе схемы (в данном случае шесть). Соединения, помеченные крестиком подключены к входам логического элемента, остальные входы логического элемента "И" подключены через резистор к источнику питания, а значит не влияют на результат работы. [4] Представленные на рисунке 4 запрограммированные соединения реализуют следующие логические функции:

$$f_1 = x_1 x_2 + x_1 \bar{x}_3$$

$$f_2 = \bar{x}_1 \bar{x}_2 x_3 + x_1 x_3$$

1.1.3 Сложные программируемые логические устройства

Программируемая логическая интегральная схема CPLD состоит из нескольких макроячеек, расположенных на одном кристалле. Каждая макроячейка соединена с блоками ввода-вывода, осуществляющими формирование необходимого вида входов или выходов для работы с внешними схемами. Кроме того, все макроячейки и блоки ввода-вывода связаны между собой внутренними параллельными шинами. Пример внутренней схемы CPLD приведен на рисунке 5. [4]

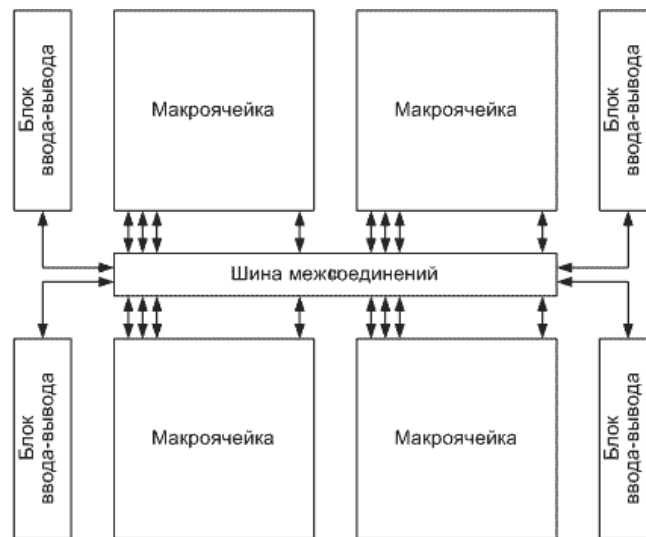


Рисунок 5 - Пример внутренней схемы CPLD [4]

Приведенная на рисунке 5 микросхема CPLD состоит из четырех макроячеек, которые связаны между собой внутренними шинами и соединяются с блоками ввода-вывода. Макроячейка построена подобно ПЛИМ микросхеме, к которой на выходе подключен

D-триггер. На рисунке 6 приведен пример внутренней схемы макроячейки и ее подключение к шине межсоединений.

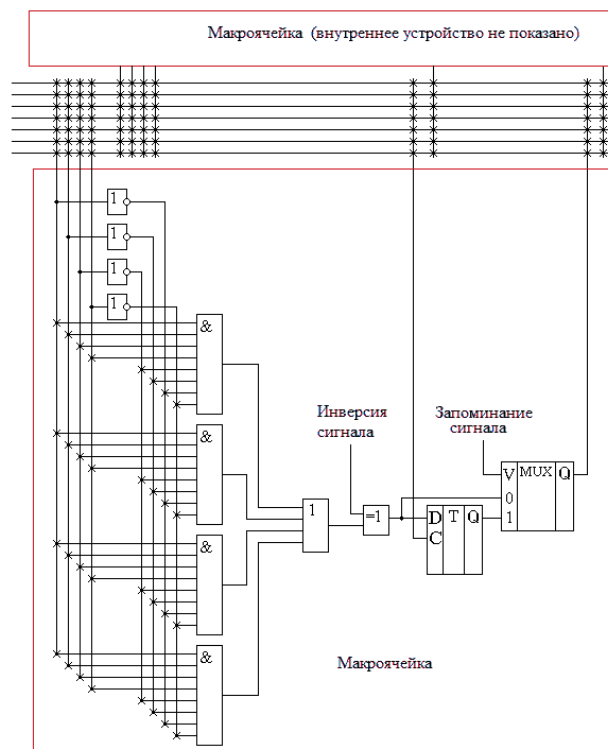


Рисунок 6 - Внутренняя схема макроячейки микросхемы CPLD [4]

В состав приведенной на рисунке 6 макроячейки входят четыре шести входовых логических элемента "6-И" и 4-входовый логический элемент "4-ИЛИ". Его выход соединен со входом логического элемента "ИСКЛЮЧАЮЩЕЕ ИЛИ". Этот логический элемент

предназначен для реализации инверсии логической функции, реализованной ПЛМ - подобной схемой макроячейки. Для этого на его второй вход может подаваться логический ноль или логическая единица. Если этот вход запрограммирован на подачу логической единицы, то логический элемент "ИСКЛЮЧАЮЩЕЕ ИЛИ" будет инвертировать значение, поступающее с выхода логического элемента "4-ИЛИ". Если же на него поступает логический ноль, то схема "ИСКЛЮЧАЮЩЕЕ ИЛИ" будет передавать на выход сигнал без изменения. На выходе макроячейки поставлен двухвходовый мультиплексор, который позволяет передавать на выход текущее значение сигнала с выхода ПЛМ-схемы, или сохраненное в D-триггере. Блок ввода-вывода состоит из элементов с тремя состояниями (буферов). Буфер позволяет настраивать выводы микросхемы на ввод или и на вывод сигналов. Чтобы вывод микросхемы настроить на ввод информации, достаточно перевести буфер, подключенный к данному выводу, в высокоимпедансное состояние. Дальнейший путь прохождения сигнала по микросхеме может быть запрограммирован МОП-ключами, размещенными на пересечениях вертикальных проводников и горизонтальных проводников шины межсоединений. На схеме это соединение помечается символом 'х'. Необходимо отметить, что если вывод запрограммирован на прием информации, то он не может быть использован в качестве выхода микросхемы. Обычно фирмы-изготовители указывают максимально возможное число входных и выходных линий.

Разработка цифровых устройств на программируемых логических интегральных схемах CPLD практически не отличается от разработки обычных цифровых устройств. Для этого не требуется разбираться в особенностях внутренней структуры микросхемы или проектировать матрицу межсоединений. Разработчики ПЛИС предоставляют пакет САПР в составе которого можно вести разработку цифрового устройства в виде обычных схем в схемном редакторе, а затем транслировать эту схему в файл коммутаций внутренней матрицы межсоединений CPLD. Этот файл загружается в ПЗУ микросхемы CPLD и микросхема превращается в разработанное нами цифровое устройство.

В настоящее время разработка цифровых устройств чаще ведется с применением языков программирования схем. Применение языков программирования вместо примитивов, являющихся аналогами микросхем средней интеграции, позволяет значительно оптимизировать внутреннюю структуру прошивки микросхемы. [4]

1.1.4 Программируемые пользователем вентиляные матрицы

Программируемые интегральные микросхемы CPLD предназначены для применения в цифровых устройствах с большим количеством логических элементов. Кроме того, в CPLD трудно обеспечить эффективное применение всех макроячеек. Всегда часть макроячеек остается неиспользуемыми. Остальная часть схемы зря занимает площадь кристалла и потребляет ток от источника питания.

Для реализации сложных цифровых схем удобно ПЛИС, обладающие большей логической вместимостью и большей степенью использования кристалла. В качестве подобных микросхем и можно назвать программируемые пользователем вентиляные матрицы (FPGA — Field- ProgrammableGateArray). Принцип работы FPGA существенно отличаются от принципа работы CPLD. FPGA является наследником комбинационных схем, реализованных на ПЗУ. Обобщенная структура микросхем FPGA приведена на рисунке 7. [5]

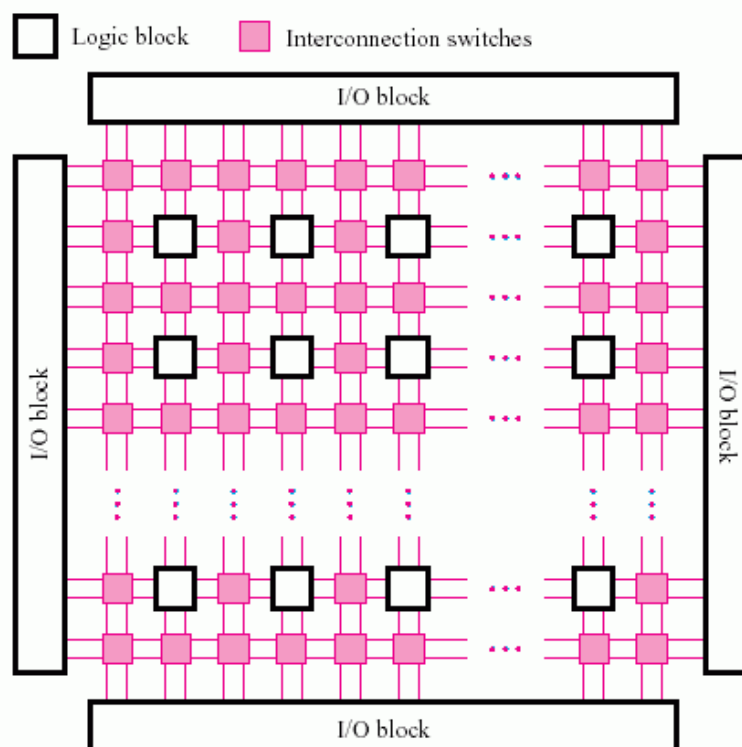


Рисунок 7 - Обобщенная структура микросхем FPGA [5]

Типовая схема программируемой интегральной микросхемы FPGA состоит из трех видов блоков: логические блоки, блоки ввода-вывода, предназначенные для обмена сигналами через внешние выводы микросхемы и программируемые электронные ключи, предназначенные для создания соединений между внутренними блоками микросхемы FPGA. Логические блоки размещаются в узлах решетки вертикальных и горизонтальных шин проводников. Благодаря возможности соединения проводников при помощи электронных

ключей друг с другом, можно создавать нужные нам соединения между логическими блоками. Соединения осуществляемые электронными ключами можно программировать, замыкая и размыкая эти ключи. Матрица соединений хранится в ПЗУ конфигурации и может быть изменена при программировании ПЛИС. Блоки ключей, замыкающие проводники соединительных шин FPGA между собой, обозначены на рисунке 7 квадратами серого цвета.

Соединение внутренней цифровой схемы с блоками ввода-вывода также программируется электронными ключами. У различных микросхем FPGA реальное число программируемых внутренних соединений значительно отличается. Это определяет насколько полно может быть использованы внутренние ресурсы микросхемы и может оказать влияние на предельное быстродействие разработанного цифрового устройства. Эти особенности необходимо учитывать в процессе проектирования цифрового устройства при выборе фирмы-изготовителя и конкретных микросхем FPGA, предназначенных для реализации этого устройства.

Особенностью структуры FPGA является то, что каждый логический блок обычно имеет небольшое число входов и один выход. Это позволяет более полно использовать внутренние ресурсы микросхемы. Типичный логический блок строится на основе ПЗУ, в ячейках которого записана таблица истинности комбинационной схемы. Подобный блок ПЗУ обычно называется LUT (LookUpTable). Каждая ячейка способна хранить значение одной строки таблицы истинности, логический '0' или '1'. Размер LUT определяется числом входов, которое изменяется в зависимости от типа выбранной микросхемы и фирмы-производителя. В качестве примера на рисунке 8 показано внутреннее устройство небольшого LUT ПЗУ с тремя входами x_1, x_2, x_3 , и одним выходом f . Крестиками обозначены электронные ключи, включенные между источником питания и входом мультиплексора. [6]

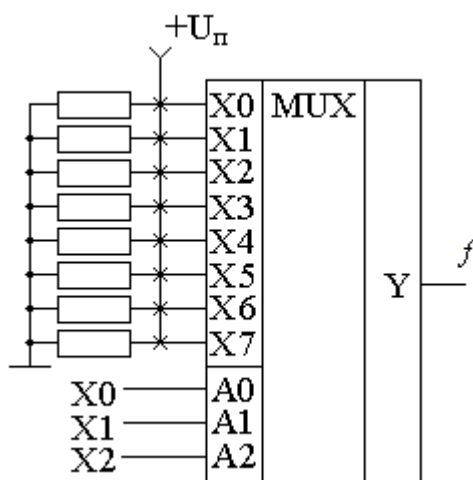


Рисунок 8 - Пример внутреннего устройства LUT ПЗУ [6]

LUT ПЗУ, изображенное на рисунке 8, способно реализовать любую логическую функцию трех переменных. Поскольку таблица истинности функций трех переменных имеет восемь строк, то LUT ПЗУ состоит из восьми запоминающих ячеек. Одна ячейка содержит значение выходного сигнала в каждой строке таблицы истинности. Входные сигналы x_1 , x_2 и x_3 подаются на адресные входы мультиплексора 8×1 . В зависимости от комбинации логических сигналов на адресных входах A_0 , A_1 , A_2 , на выход мультиплексора поступает сигнал с одного из его информационных входов $X_0 \dots X_7$.

Для программирования LUT необходимо при помощи электронных ключей записать таблицу истинности реализуемой функции в ячейках памяти. Электронный ключ представляет собой полевой транзистор с плавающим затвором, что позволяет сохранять состояние ключа при отключении питания. При замыкании электронного ключа на вход мультиплексора поступает напряжение питания, что соответствует логической единице. При размыкании этого ключа на вход мультиплексора через резистор R поступает потенциал общего провода микросхемы, что соответствует логическому нулю. После программирования при $x_1=x_2=x_3=0$ на выход LUT ПЗУ будет передан сигнал с самого верхнего входа мультиплексора, $x_2=x_3=0$, $x_1=1$ на выход LUT ПЗУ будет передан сигнал с второго сверху входа мультиплексора, и т.д..

LUT ПЗУ современных микросхем FPGA обычно имеют четыре или пять входов и состоят из 16 или 32 запоминающих ячеек соответственно. Как и в CPLD микросхемах, кроме комбинационной цифровой схемы логический блок содержит запоминающее устройство, такое как D-триггер. На рисунке 9 приведен пример схемы логического блока FPGA микросхемы.

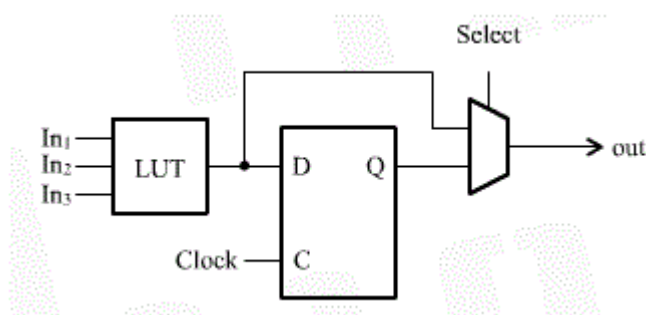


Рисунок 9 - Пример схемы логического блока FPGA микросхемы [6]

Логический сигнал, поступающий с выхода LUT ПЗУ на вход D триггера, запоминается по сигналу синхронизации, который может формироваться как внутренними элементами схемы, так и поступать с внешних выводов FPGA микросхемы. Мультиплексор, стоящий на выходе логического блока, позволяет либо применять D-триггер при построении

регистровых схем, либо отключать его от схемы при проектировании комбинационных логических схем.

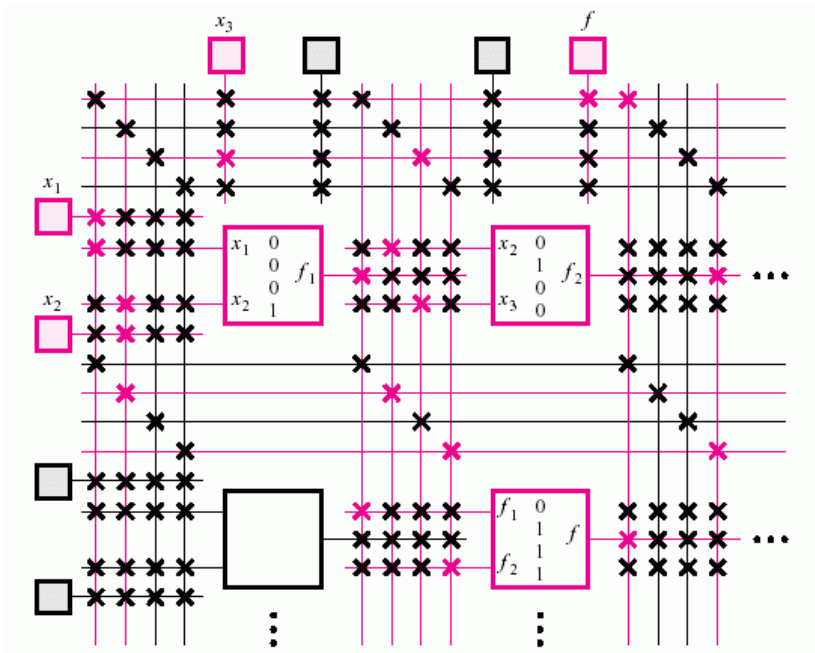


Рисунок 10 - Пример запрограммированного участка FPGA [7]

На рисунке 10 приведен участок внутренней структуры микросхемы FPGA, запрограммированный для реализации комбинационной схемы. Данная микросхема FPGA состоит из LUT ПЗУ с двумя входами и четырьмя программируемыми линиями соединений. Программируемые электронные ключи обозначены символом X. Каждый замкнутый ключ, соединяющий вертикальную и горизонтальную линии показан красным цветом. Не используемые линии соединений и ключи на рисунке показаны черным. Первые два блока LUT запрограммированы на реализацию функций $f_1 = x_1x_2$ и $f_2 = x_2x_3$. Третий LUT реализует функцию $f = f_1 + f_2$. [7]

1.1.5 Преимущество плис

Преимущество ПЛИС это возможность создания на их основе самых различных микросхем при наличии развитых средств проектирования. Именно это, наряду с низкой стоимостью самих кристаллов, и обусловило их широкое распространение во второй половине прошлого века. В ПЛИС заложены возможности, которые позволяют превратить ее в ИС с любой функцией цифровой логики. Проектирование сводится к выявлению программируемых элементов (перемычек или запоминающих ячеек), после удаления которых в структуре схемы остаются только те связи, которые необходимы для выполнения требуемых функций. На практике, для проектирования обязательно применяют системы автоматизированного проектирования (САПР ПЛИС). Благодаря

наличию различных систем автоматизированного проектирования, а также структурным и технологическим особенностям, ПЛИС представляют технологию рекордно-короткого цикла разработки радиоэлектронной аппаратуры. Причем весь цикл проектирования и изготовления готового устройства осуществляется самим разработчиком, что значительно снижает стоимость радиоэлектронной аппаратуры.[8]

Наиболее эффективные области применения Плис:

- Изделия, требующих нестандартных схемотехнических решений и нет необходимых стандартных микросхем.
- Эмуляции схем, подлежащих последующей реализации на другой элементной базе.
- Проектирование на их основе устройств для защиты программного обеспечения и аппаратуры от несанкционированного доступа и копирования.
- Разработка контроллеров, адресных дешифраторов, логики обрaмления микропроцессоров, формирователи управляющих сигналов и др.
- Изготовление микропрограммных автоматов и других специализированных устройств. например, цифровые фильтры, схемы обработки сигналов и изображения, процессоры быстрого преобразования функций Фурье и т.д.
- Применение в аппаратуре уплотнения телефонных сигналов.

1.2 Программируемые аналоговые интегральные схемы

1.2.1 История программируемых аналоговых интегральных схем

Современные измерительные системы и схемы управления широко используют принцип цифрового преобразования сигналов. Цифровые устройства применяются везде, где требуется высокая точность и хорошая повторяемость преобразований, а результаты должны храниться в течение длительного времени, а также в тех случаях, когда алгоритм преобразования сложен или требуется высокая помехоустойчивость. Поскольку подавляющее большинство сигналов окружающего мира имеет аналоговую природу, сигналы, предназначенные для цифровой обработки, должны быть подвергнуты дискретизации, квантованию и кодированию. И обратно: если на выходе системы требуется получить аналоговый сигнал, то должна быть выполнена операция декодирования и, возможно, интерполяция. Подобные операции выполняются АЦП и ЦАП соответственно, которые могут быть построены на переключаемых конденсаторах.

Первые попытки создания безрезистивных преобразователей были предприняты в середине 1970-х годов. Эти схемы в каком-то смысле были предвестниками широкого

распространения схем с переключаемыми конденсаторами, датируемого 1977 г., когда был изготовлен в интегральном исполнении фильтр с переключаемыми конденсаторами. Базовыми элементами для построения АЦП и ЦАП с использованием схемотехники переключаемых конденсаторов являются конденсаторы, матрицы конденсаторов, ключи, ОУ, источники опорного напряжения.

Для управления работой преобразователей на переключаемых конденсаторах применяют многофазные генераторы, сигналы которых могут иметь весьма сложный вид. Применение многофазных генераторов в АЦП обусловлено алгоритмом работы таких преобразователей. Некоторые фазы могут быть использованы для исключения различных паразитных эффектов. Например, МОП ОУ имеет достаточно большое напряжение смещения нуля и температурный дрейф. Поэтому рекомендовано применять схемы с переключаемыми конденсаторами с компенсацией дрейфа.

Обработка сигнала внутри ПАИС осуществляется схемами на переключаемых конденсаторах. В отличие от цифровых систем, где сигнал дискретен по времени и квантован по уровню, в дискретно-аналоговых системах сигнал дискретен только по времени (рисунок 11), в силу этого выходной аналоговый сигнал можно восстановить без искажений по его выборкам [9]. Так, при входном сигнале 1 мВ точность преобразования будет не хуже $\pm 0,1\%$.

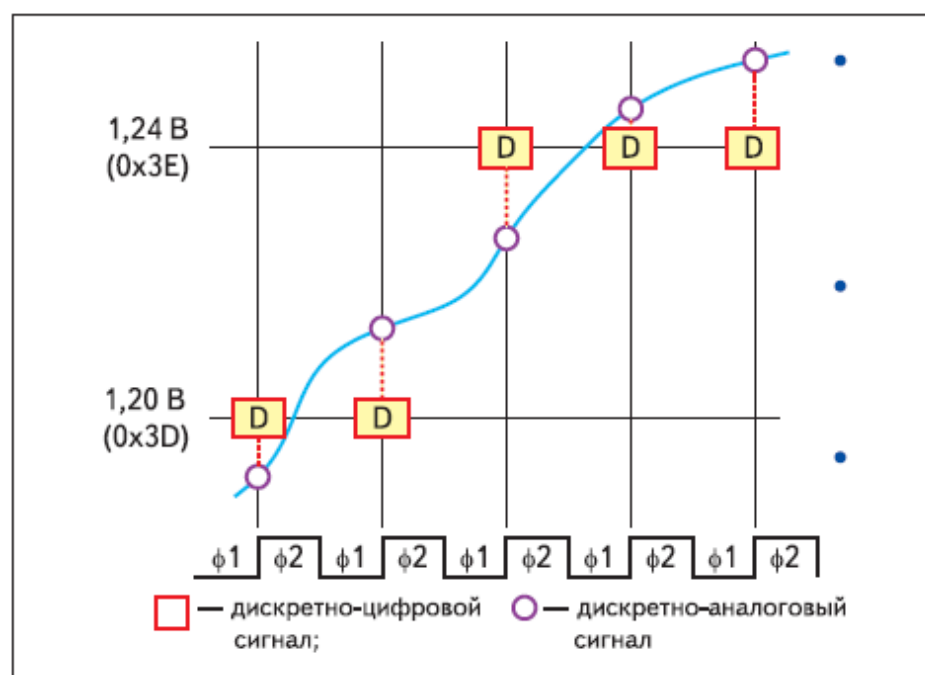


Рисунок 11 -Представление процесса аналоговой и цифровой дискретизации [9]

1.2.2 Программируемые аналоговые интегральные схемы компании Anadigm

Традиционно схемы аналоговой обработки сигналов выполняются на дискретных компонентах – операционных усилителях, компараторах, мультиплексорах и т.п. При этом в ряде случаев аналоговая часть занимает большую часть площади печатной платы, имеет высокую стоимость и требует сложной настройки. Решить проблему создания разнообразных аналоговых устройств, кардинально снизив стоимость и габариты, позволяет использование программируемых аналоговых интегральных схем – ПАИС (FPAА). ПАИС — набор базовых ячеек, которые могут быть сконфигурированы и соединены между собой для реализации наборов аналоговых функций: фильтров, усилителей, интеграторов, сумматоров, ограничителей, делителей, выпрямителей. Особенностью схемы является изменение аналоговой схемы во время функционирования или изменение характеристик некоторых элементов схем в режиме реального времени. Компания Anadigm является лидером в производстве программируемых аналоговых интегральных схем (ПАИС) - аналоговых эквивалентов ПЛИС. Аналоговые схемы ПАИС, которые производит компания, делятся на два типа ПАИС - динамически и статически конфигурируемые. Отличает аналоговые схемы разных типов то, что динамически конфигурируемая схема позволяет изменять полностью или частично функциональную структуру в реальном времени в работающем устройстве, что дает возможность создавать уникальные схемы аналоговой обработки сигналов. Средство разработки и необходимые библиотеки, предоставляемые компанией Anadigm бесплатно, позволяют проектировать аналоговые схемы без дополнительных знаний в области цифровых схем или языков программирования.[4] Компания Anadigm выпускает ПАИС 2-го и 3-го поколения. AN120E04 - статически перепрограммируемая ПАИС с четырьмя входными и двумя выходными ячейками, AN121E04 – статически перепрограммируемая ПАИС с четырьмя универсальными конфигурируемыми ячейками ввода/вывода двумя выходными ячейками, AN220E04 – динамически перепрограммируемая ПАИС с четырьмя входными и двумя выходными ячейками, AN221E04 – динамически перепрограммируемая ПАИС с четырьмя универсальными конфигурируемыми ячейками ввода/вывода двумя выходными ячейками. [10]



Рисунок 12 -ПАИС Anadigm AN221E04

AN221E04 – ПАИС второго поколения(рисунок 12), построенная по схемотехнике на переключаемых конденсаторах, обеспечивает широкую полосу частот обработки сигналов, низкий коэффициент гармоник и интермодуляционных искажений, возможность создавать практически любые схемы аналоговой обработки, снижая затраты на разработку и последующий редизайн изделий. Она позволяет реализовать динамическое переконфигурирование с высокой скоростью – за один такт синхронизации (50 нс). Помимо основного – конфигурационного ОЗУ, она содержит специальное теневое ОЗУ, в которое записываются данные новой конфигурации. С приходом очередного такта синхронизации эти данные записываются в конфигурационное ОЗУ, что приводит к мгновенному обновлению внутренней структуры или параметров схемы. Они включают в себя четыре конфигурируемых аналоговых блока (КАБ), матрицу статически программируемых ключей внутренних соединений и матрицу динамически программируемых ключей, осуществляющих коммутацию переключаемых конденсаторов при реализации тех или иных аналоговых функций. Управление матрицами ключей осуществляется специальной логической схемой, преобразующей информацию, хранящуюся в конфигурационном ОЗУ в адреса состояний. Каждый КАБ имеет доступ к общей области памяти – таблице коэффициентов, информация из которой используется для реализации нелинейных устройств, таких как каскады с передаточными функциями, определяемыми пользователем и генераторы периодических сигналов произвольной формы. Аналоговые сигналы могут подаваться в ПАИС через универсальные входные ячейки. Одна из ячеек имеет на входе 4-х канальный дифференциальный мультиплексор, позволяющий подключать одновременно до 4-х дифференциальных или синфазных сигналов.

1.2.3 Обобщенная архитектура программируемых аналоговых интегральных схем Anadigm AN221k04

Структурная схема динамически конфигурируемой ПАИС AN221E04 показана на рисунке 13. Ее основу составляют конфигурируемые аналоговые блоки (КАБ), которые содержат наборы элементов для реализации стандартных устройств – операционных усилителей, компараторов, источников образцового напряжения, АЦП, а также конфигурационную память LUT и специальный интерфейс. Входные аналоговые сигналы подаются в КАБ через конфигурируемые двунаправленные I/O ячейки. Ячейки I/O_Cell_1 – I/O_Cell_3 осуществляют передачу сигнала напрямую в один из блоков. Ячейка I/O_Cell_4 содержит специальный мультиплексор, который позволяет подключать до четырех дифференциальных или несимметричных входных сигналов или нагрузок (в режиме выходов).

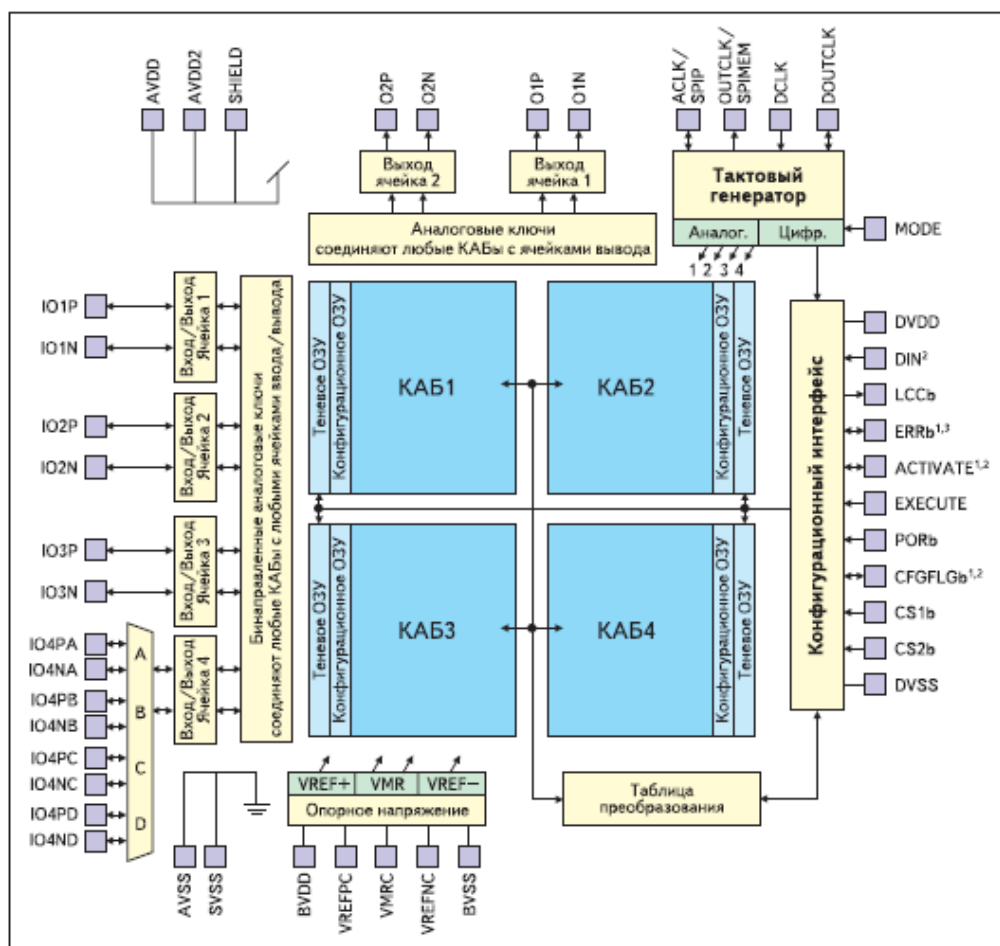


Рисунок 13 -Внутренняя структура программируемой аналоговой микросхемы AN221E04

[11]

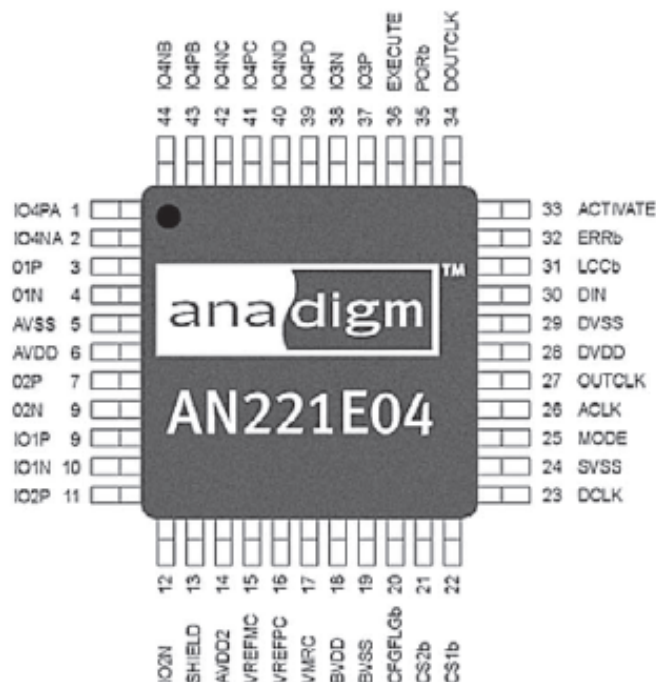


Рисунок 14 - Назначение выводов AN221E04 [11]

Каждый вход-выход ячейки может непосредственно подключаться к одному из блоков или предварительно обрабатываться с помощью набора стандартных устройств в любой комбинации: буферный усилитель, усилитель с программируемым коэффициентом усиления, программируемый фильтр, прецизионный усилитель со стабилизацией прерыванием. Последний специально разработан для усиления сигналов, требующих ультранизкого входного напряжения смещения и высокостабильного коэффициента усиления.

Синхронизация в ПАИС может осуществляться с помощью внешнего источника или от собственного тактового генератора с внешним кварцевым резонатором. Частота внутреннего генератора может быть поделена четырьмя синхронными программируемыми делителями, а каждая из частот – выведена на внешний выход.

Режимы работы КАБ, значения тактовых частот, направления передачи сигналов, назначения и конфигурация I/O_Cell, Output_Cell хранятся в так называемой конфигурационной памяти (ConfigurationSRAM). Копия содержимого конфигурационной памяти хранится в теновом ОЗУ (ShadowSRAM), которое может перезаписываться без нарушения процесса обработки сигнала. Это позволяет динамически изменять конфигурацию ПАИС в работающем устройстве во время работы предыдущей версии. После загрузки в теновое ОЗУ новых данных конфигурация устройства изменяется за один цикл тактовой синхронизации.

Архитектура ПАИС имеет простой и гибкий конфигурационный интерфейс. Он предназначен для работы, как в автономном режиме, так и для связи с внешними интерфейсами. В режиме FPGA EPROM после включения питания конфигурация из EPROM будет автоматически загружена в FPAА, и устройство сразу же начнет работать. Конфигурационный интерфейс также выполняет функцию связи FPAА с внешним микроконтроллером через SPI порт в режиме ведомого устройства. С его помощью возможно наращивание количества ПАИС для создания больших систем аналоговой обработки. [11]

1.2.4 Конфигурируемые аналоговые блоки программируемых аналоговых интегральных схем Anadigm AN221k04

Основу ПАИС составляют четыре конфигурируемых аналоговых блока (КАБ), каждый из которых содержит наборы элементов — 8 программируемых конденсаторов, 2 операционных усилителя, 1 компаратор и 1 регистр последовательного приближения (рисунок 15).

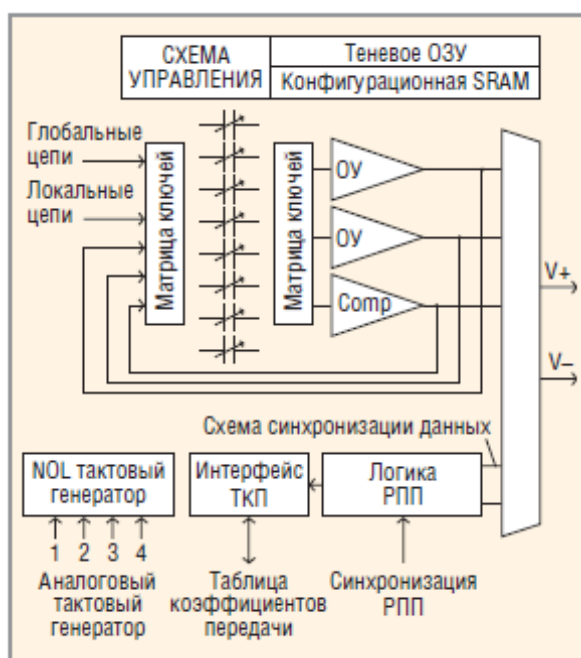


Рисунок 15 - Обобщенная структура одной из ячеек матрицы КАБ [12]

Используя этот набор элементов, можно создавать конфигурируемые аналоговые модули (КАМ): усилители, выпрямители, интеграторы, дифференциаторы, сумматоры, перемножители и т. п. Необходимо отметить, что на базе одного КАБ можно создать несколько КАМ. Программируемые конденсаторы, применяемые в КАБ, могут принимать относительное значение емкости от 0 до 255 единиц. В отличие от традиционных аналоговых схем, в схемах на переключаемых конденсаторах не важны абсолютные значения емкости, а

только соотношение между ними, которое выдерживается с точностью не хуже 0,1%. Параметры таких схем практически не будут зависеть от времени работы или изменения температуры окружающей среды, так как отношение емкостей останется неизменным. Каждый КАБ содержит статические и динамические ключи. Динамические ключи управляются входными и тактовыми сигналами, а также логикой регистра последовательного приближения. Статические ключи определяют общие схемы коммутации блоков, значения емкости конденсаторов, подключение входов. [12]

При включении питания микросхемы SRAM очищается, после этого с помощью конфигурационной логики данные загружаются из внешнего EPROM в теневое ОЗУ, а из него копируются в конфигурационное ОЗУ. Во время работы ПАИС теневое ОЗУ может быть перезагружено новыми данными, которые впоследствии могут использоваться для перепрограммирования структуры ПАИС. В этом случае содержимое теневого ОЗУ копируется в конфигурационную память, и с приходом очередного такта синхронизации микросхема начинает работать в новой конфигурации, без прерывания процесса обработки сигнала. Аналоговые сигналы направляются в ближайший блок по кратчайшему пути с помощью матрицы первой группы входных аналоговых ключей. На эту же матрицу выводятся обратные связи двух внутренних операционных усилителей и компаратора.

Вторая матрица ключей предназначена для организации внутренней топологии и выполняет соответствующие внутренние коммутации цепей. Основу ячейки КАБ составляют два операционных усилителя и компаратор. Их выходы заводятся обратно в матрицу входных ключей для организации обратных связей, а также в соседнюю ячейку КАБ. Обработка сигнала внутри КАБ осуществляется схемами на переключаемых конденсаторах. Для корректной работы такие схемы нуждаются в так называемой неперекрывающейся синхронизации (nonoverlappingclocks, NOL). Такой тактовый генератор является частью КАБ и вырабатывает все необходимые NOL сигналы для его функционирования. ПАИС содержит специальную область памяти объемом 256 байт, называемую таблицей коэффициентов передачи (ТКП), с помощью которой можно реализовывать нелинейные аналоговые функции, такие как перемножение, сжатие, линеаризация, автоматическая регулировка усиления. ПАИС содержит специальную область памяти объемом 256 байт, называемую таблицей коэффициентов передачи, с помощью которой можно реализовывать нелинейные аналоговые функции, такие как перемножение, сжатие, линеаризация, автоматическая регулировка усиления. [13]

1.2.5 Назначение и структура конфигурационного интерфейса программируемых аналоговых интегральных схем Anadigm AN221k04

Отличительной особенностью ПАИС Anadigm является развитый конфигурационный интерфейс, позволяющий максимально упростить и ускорить процесс загрузки данных в микросхему, а также открыть возможность для использования широкой номенклатуры источников данных – от стандартных ПЗУ до микропроцессоров. Конфигурационный интерфейс предназначен для загрузки данных аналоговой структуры в конфигурационное ОЗУ. Он поддерживает автоматическую загрузку конфигурации из внешней Serial EEPROM, а также из микроконтроллера через стандартный интерфейс SPI или SSI или через типовую микропроцессорную шину данных. Выбор варианта загрузки данных осуществляется выдачей соответствующего сигнала на вывод MODE. Оба варианта загрузки данных доступны как для статических, так и для динамических ПАИС. Динамически перепрограммируемые микросхемы дополнительно имеют возможность смены конфигурации при управлении от внешнего МК в режиме реального времени. После загрузки данные хранятся в конфигурационном ОЗУ. В динамически перепрограммируемых микросхемах имеется дополнительное теневое ОЗУ. Данные из внешнего источника загружаются в теневое ОЗУ и затем с приходом очередного такта синхронизации копируются в конфигурационное ОЗУ. Этот метод позволяет загружать данные в работающем устройстве и при необходимости изменять аналоговую структуру без перезагрузки системы. [15]

2 ОСНОВНЫЕ СХЕМОТЕХНИЧЕСКИЕ РЕШЕНИЯ

2.1 Схемы на переключаемых конденсаторах

Схемы на переключаемых конденсаторах это обширный класс схемотехнических решений, основанный на периодической коммутации конденсаторов.

Наибольшее распространение получил с освоением в промышленности интегральных микросхем по технологии с оксидной изоляцией (например, КМОП). Низкий уровень диэлектрической абсорбции и малые утечки диэлектрика позволили создавать высококачественные конденсаторы с хорошей повторяемостью. При этом с резисторами в рамках данной полупроводниковой технологии все было гораздо хуже с точки зрения занимаемой площади, повторяемости и стабильности номиналов, паразитных ёмкостей. Такая ситуация быстро привела к выработке ряда специфических схемотехнических решений. Следует заметить, что решения на переключаемых конденсаторах и ранее применялись в дискретном исполнении в специальных случаях.

Замена резисторов в интегральном исполнении происходит следующим образом. Известно, что сила тока в проводнике прямо пропорциональна приложенному напряжению и обратно пропорциональна сопротивлению проводника (Закон Ома для однородного участка цепи рисунок 16). В то же время сила тока равна отношению заряда, переносимого через проводник за интервал времени.

$$I = \frac{U}{R} \text{ и } I = \frac{\Delta Q}{\Delta t}$$

Рисунок 16 – Закон Ома для однородного участка цепи

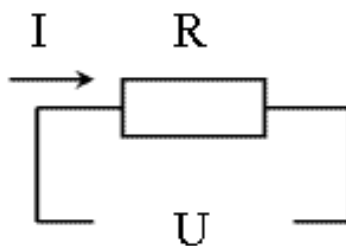


Рисунок 16- Иллюстрация закона Ома

Сопротивление цепи рассчитывается по формуле:

$$R = \frac{U \cdot \Delta t}{\Delta Q}$$

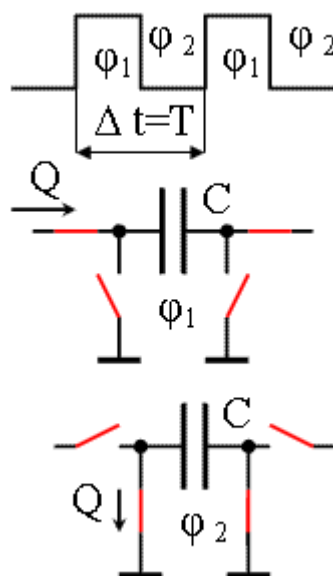


Рисунок 17 - Иллюстрация переноса заряда переключаемым конденсатором

Перенос заряда через конденсатор по схеме на рисунке 17 можно рассчитать по формуле:

$$Q = C \cdot U$$

Используя равенства получаем:

$$R = \frac{U \cdot T}{C \cdot U} = \frac{T}{C} = \frac{1}{f \cdot C}$$

Следовательно, сопротивление цепи с переключающим конденсатором обратно пропорционально произведению частоты переключения конденсатора на значение его ёмкости.

Замена резисторов конденсаторами позволяет повысить точность и уменьшить количество подключаемых к микросхеме внешних элементов. В качестве примера можно привести замещение схемы инвертирующего усилителя схемой на переключаемых конденсаторах (рисунок 18). [16]

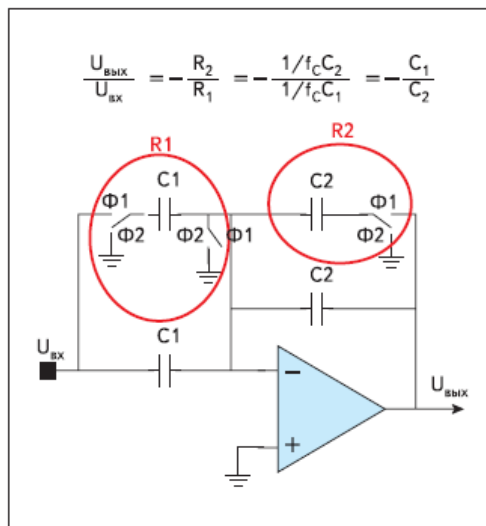


Рисунок 18 - Инвертирующий усилитель на переключаемых конденсаторах [16]

В ПАИС Anadigm AN221E04 обработка сигнала происходит при помощи схем на переключаемых конденсаторах. Принципиальные схемы исследуемых генераторов приведены на рисунках. Операционный усилитель работает в двух режимах: в инвертирующем и в не инвертирующем. На рисунках 19, 20, 21 представлены принципиальные схемы генераторов синусоидального, прямоугольного и пилообразного сигналов на переключаемых конденсаторах соответственно.

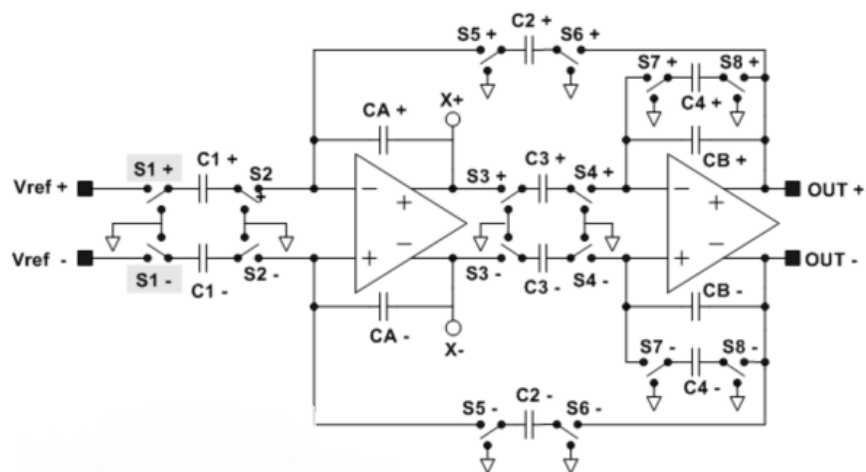


Рисунок 19 – Схема генератора синусоидальных сигналов

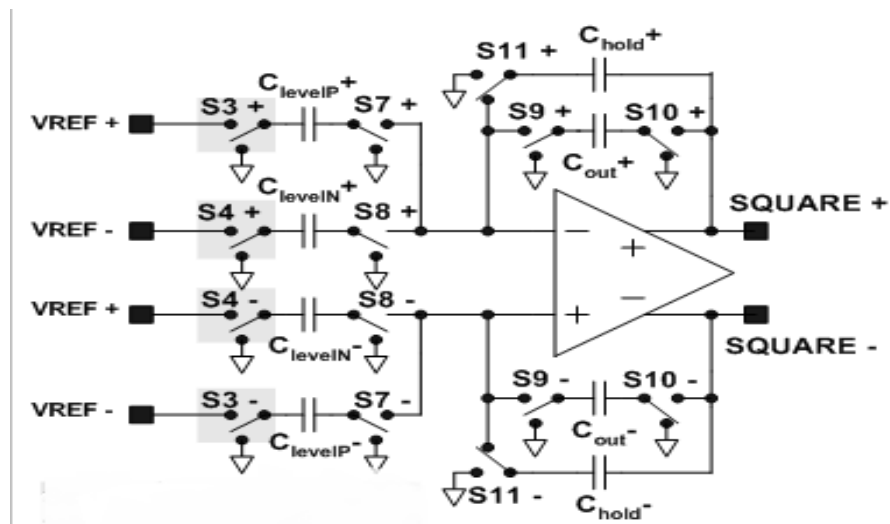


Рисунок 20 – Схема генератора прямоугольного сигнала

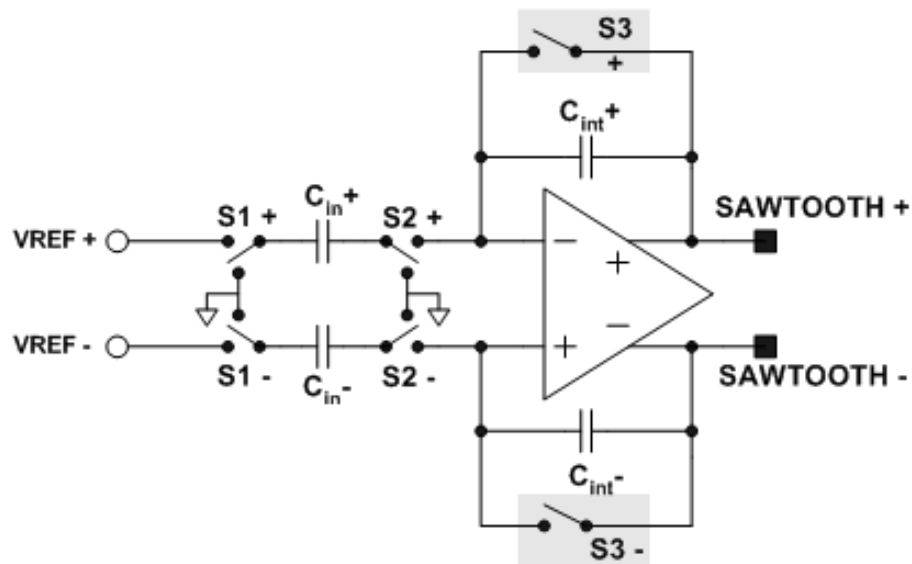


Рисунок 21 – Схема генератора пилообразного сигнала

2.2 МОП структура интегратора

В последнее время наблюдается исключительно быстрый рост производства и применения МОП-структур, имеющих много преимуществ перед биполярными схемами. У МОП-структур большой входной импеданс, и они управляются напряжением (в отличие от биполярных схем, управляемых по существу током). Комплементарные МОП-структуры практически не потребляют мощности в статическом режиме. Технология МОП-структур обеспечивает большую плотность упаковки, чем биполярных. Наконец, эта технология позволяет простым способом реализовать в ИМС конденсаторы относительно большой емкости. Такие МОП-конденсаторы в сочетании с МОП-ключами позволяют заменить резисторы в некоторых типах ИМС и построить аналоговые вычислительные схемы со

значительно лучшими точностными и эксплуатационными характеристиками. Замена резисторов конденсаторами, в частности, позволяет повысить точность аналоговых и аналого-цифровых устройств и уменьшить количество внешних элементов, подключаемых к микросхеме. В таблице представлены сравнительные характеристики интегральных резисторов и МОП-конденсаторов.

Таблица Сравнительных характеристик интегральных резисторов и МОП-конденсаторов

Элемент	Технология изготовления	Точность изготовления, %	Температурный коэффициент $10^{-6}K^{-1}$	Коэффициент влияния напряжения $10^{-6}B^{-1}$
Резистор	Ионная имплантация с шириной 40 мкм	+/-0,12	400	800
Конденсатор	МОП с толщиной диэлектрика 0,1 мкм	+/-0,06	26	10

Высокая точность изготовления интегральных МОП-конденсаторов и их стабильность способствовали тому, что в последние годы получили развитие способы обработки сигналов, использующие явление дискретного переноса зарядов. Один из путей реализации этих способов состоит в применении схем с переключаемыми конденсаторами.

Рассмотрим реализацию аналогового интегратора с применением переключаемого конденсатора. На рисунке 22а приведена схема обычного аналогового интегратора.

Передаточная функция этой схемы имеет вид,

$$K(s) = -\frac{1}{sR_1C_2}$$

а частотная характеристика

$$K(j\omega) = -\frac{1}{j\omega R_1C_2}$$

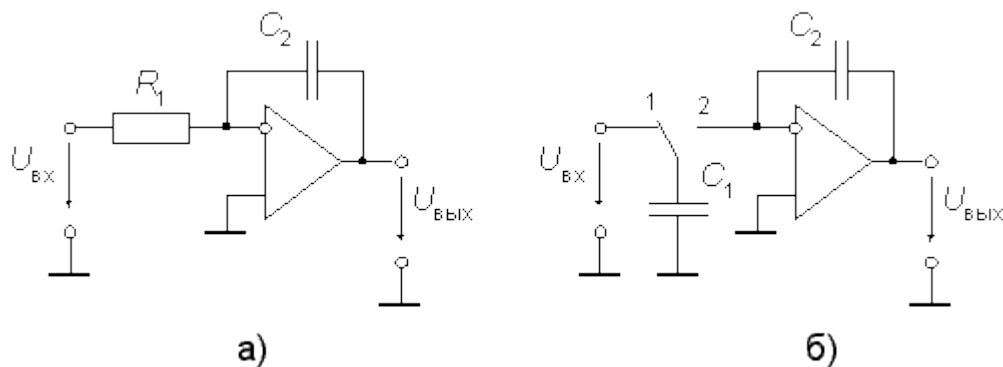


Рисунок 22 - Схемы интеграторов: а) - на RC-цепи, б) - с коммутируемым конденсатором

На рисунке 22б показан интегратор, в котором резистор R1 имитируется с помощью схемы с переключаемым конденсатором. Этот интегратор работает следующим образом. Коммутатор периодически переключается из положения 1 в положение 2 и обратно с периодом T. В момент nT конденсатор C1 заряжается до напряжения $u_{вх}(nT)$, поэтому накопленный на нем заряд составляет $C_1 u_{вх}(nT)$. После переключения коммутатора из положения 1 в положение 2 в момент nT+T/2 конденсатор C1 разряжается на вход ОУ с конденсатором C2 в обратной связи. Поскольку входное дифференциальное напряжение и входные токи идеального ОУ равны нулю, конденсатор C1 разрядится полностью и его заряд суммируется с зарядом, накопленным на конденсаторе C2. В результате в момент (n+1)T справедливо следующее уравнение зарядов:

$$C_2 u_{вых}[(n+1)T] = C_2 u_{вых}(nT) - C_1 u_{вх}(nT).$$

Здесь знак "-" обусловлен отрицательной обратной связью. Применив к обеим частям уравнения z-преобразование, получим уравнение

$$z C_2 U_{вых}(z) = C_2 U_{вых}(z) - C_1 U_{вх}(z).$$

Определенная из этого уравнения передаточная функция имеет вид

$$W(z) = \frac{U_{вых}(z)}{U_{вх}(z)} = - \frac{C_1}{C_2(z-1)}$$

Перейдем к частотным характеристикам, подставив в передаточную функцию $z = \exp(j\omega T)$.

Получим

$$W(j\omega) = - \frac{C_1}{C_2(e^{j\omega T} - 1)}$$

При ωT стремящемся к 0 выражение в скобках в знаменателе правой части уравнения неограниченно приближается к $j\omega T$. Таким образом, для частот входного сигнала, низких относительно частоты переключения коммутатора $f=1/T$, можно приближенно записать

$$W(j\omega) \approx -\frac{C_1}{C_2 j\omega T}$$

Сравнивая выражения, находим, что в схеме на рисунке 22б коммутируемый конденсатор имитирует входной резистор схемы на рисунке 22а, с сопротивлением, равным T/C_1 . Поэтому, увеличивая частоту переключения коммутатора, мы уменьшаем эквивалентную постоянную времени интегрирования интегратора.

Применение интеграторов с переключаемыми конденсаторами в ИМС фильтров вместо обычных интеграторов дает два существенных преимущества. Во-первых, коэффициент передачи интегратора зависит только от отношения двух конденсаторов, а не от их абсолютных величин. Вообще говоря, можно достаточно просто создать на кремниевой подложке ИМС пару любых одностипных согласованных элементов, в то время как получение разнотипных элементов (резистора и конденсатора) с точными значениями и высокой стабильностью весьма затруднительно (различия температурных коэффициентов сопротивления (ТКС) и емкости (ТКЕ) могут быть значительными). Поэтому ИМС фильтров на переключаемых конденсаторах значительно дешевле.

Второе преимущество фильтров на переключаемых конденсаторах состоит в возможности настройки их характеристической частоты (т.е. центральной частоты полосового фильтра или точки -3 дБ фильтра нижних частот) изменением только тактовой частоты. Это объясняется тем, что характеристическая частота фильтра, построенного на основе метода переменных состояния, пропорциональна коэффициенту передачи интегратора (или, что то же, обратно пропорциональна постоянной времени интегрирования). Это позволяет выпускать фильтры 8-го порядка в корпусе с восемью выводами без внешних времязадающих элементов, в то время как ИМС фильтров с RC-интеграторами имеют значительно больше выводов и требуют подключения значительного количества точных резисторов.

3 РАЗРАБОТКА ГЕНЕРАТОРОВ СТАНДАРТНЫХ СИГНАЛОВ

3.1 Система автоматизированного программного обеспечения AnadigmDesigner2

Программное обеспечение AnadigmDesigner2 представляет собой систему автоматизированного проектирования (САПР) второго поколения средств разработки Anadigm, которая позволяет создавать новые или вносить изменения в уже имеющиеся программируемые аналоговые схемы буквально в считанные минуты. Разработчик может создавать схемы, используя имеющиеся конфигурируемые аналоговые модули (КАМ), каждый из которых выполняет целый ряд аналоговых функций в зависимости от передаваемых параметров. AnadigmDesigner2 включает программу функционального моделирования по времени – удобный инструмент для оценки схем без необходимости проведения физических измерений и макетирования. Интерфейс функциональной модели – интуитивно понятный и легкий для использования. Большинство этапов моделирования аналогичны этапам обычного макетирования. Независимо от того, является пользователь специалистом или нет, он сможет быстро разрабатывать сложные аналоговые схемы, сразу же моделировать их работу; затем, выбрав нужную, загрузить ее в ПАИС для дальнейшего проведения испытаний и проверки достоверности.

САПР AnadigmDesigner2 – первый в мире продукт EDA, позволяющий разрабатывать схемы с использованием программируемых аналоговых матриц, которые могут быть переконфигурированы с помощью микропроцессора или микроконтроллера в режиме реального времени. Это позволяет изменять их функции в системе или создавать адаптивные схемы. AnadigmDesigner2 автоматически переводит проект в С-код, позволяя, в свою очередь, микропроцессору внутри системы производить настройку и контролировать работу устройства.

Программное обеспечение AnadigmDesigner2 позволяет быстро и просто конструировать сложные аналоговые схемы путем выбора, размещения и соединения стандартных элементов – конфигурируемых аналоговых модулей. Разработанные аналоговые схемы могут быть загружены в поле программируемой аналоговой матрицы ПАИС. В дальнейшем ПАИС будет функционировать аналогично разработанной и отлаженной схеме. Результат разработки можно посмотреть сразу же, используя для этого генератор сигналов и осциллограф.

Для хранения конфигурационных данных ПАИС используется внешняя SRAM, что позволяет перепрограммировать микросхему столько раз, сколько потребует разработка.

Важно отметить, что AnadigmDesigner2 позволяет в одном проекте создавать несколько независимых аналоговых устройств, с использованием нескольких ПАИС, которые будут

иметь общую память и загружаться общим потоком конфигурационных данных. Это дает возможность разрабатывать сложные системы аналоговой обработки.

Также AnadigmDesigner2 генерирует «разумные» конфигурационные данные в виде готового С-кода для загрузки в микропроцессор, что позволяет выполнять переконфигурирование устройств.

Программа функционального моделирования позволяет обойтись без какого-либо дополнительного оборудования. Она обладает понятным интерфейсом и графически отображает результаты, полученные через определенные промежутки времени.

В комплект AnadigmDesigner2 входят две подпрограммы – системы автоматизированного проектирования фильтров и пропорционально интегрально дифференциальных регуляторов.

Последняя доступная версия САПР AnadigmDesigner2 имеет следующие особенности:

- Позволяет разрабатывать сложные схемы с применением простого графического интерфейса drag-and-drop.
- Включает в себя надежные и легкие в использовании инструменты для разработки аналоговых схем.
- Не нуждается в аналоговом экспертном анализе для разработки законченных аналоговых схем.
- Доступна растущая и постоянно обновляемая библиотека КАМ.
- Экспертные системы синтеза AnadigmFilter и AnadigmPID автоматизируют сложный процесс разработки схем.
- Встроенный симулятор по времени позволяет работать с несколькими ПАИС одновременно.
- Интерфейс четырех канального осциллографа, имитация сигналов случайной формы.
- Автоматическая генерация С-кода для управления и настройки аналоговых функций непосредственно микропроцессором внутри системы.

Для проектирования аналоговой схемы ПАИС понадобится бесплатное программное обеспечение AnadigmDesigner2, которое является полнофункциональным симулятором, позволяющим отладить электрическую схему без программирования ПАИС [5, 6]. На этом этапе проектирования необходимо выбрать тип используемой микросхемы (AN221E04), установить ее параметры, а затем приступить к построению аналоговой структуры.

Для разработки генераторов стандартных сигналов в САПР AnadigmDesigner2 необходимо выполнение следующих этапов:

1. Настроить параметры платы.
2. Выбрать, настроить и разместить внутри кристалла конфигурируемые аналоговые модули.
3. Подключить и настроить генераторы сигналов.
4. Соединить КАМ и ячеек I/O между собой.
5. Настроить параметры разработанной схемы и запустить процесс моделирования.
6. Загрузить конфигурационные данные в ПАИС.

Чтобы открыть окно нового проекта, необходимо выбрать меню *File*→ *New*, как показано на рисунке 23.

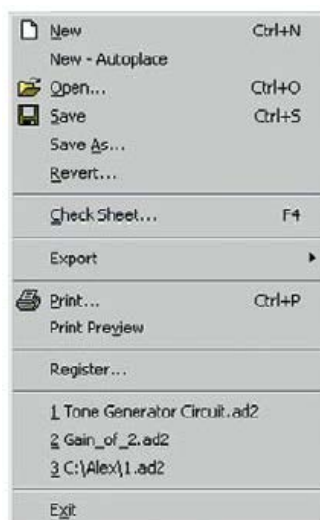


Рисунок 23 – Окно нового проекта

В раскрывающемся меню *Settings* выберем пункт *ActiveChipsettings...* В открывшемся окне, во вкладке *Clocks* в разделе *MasterClock* устанавливаем значения частот как показано на рисунке 24. так как все вычисления программируемых параметров являются функциями от частоты синхронизирующего сигнала схем на переключаемых конденсаторах, определяемого как раз этой частотой.

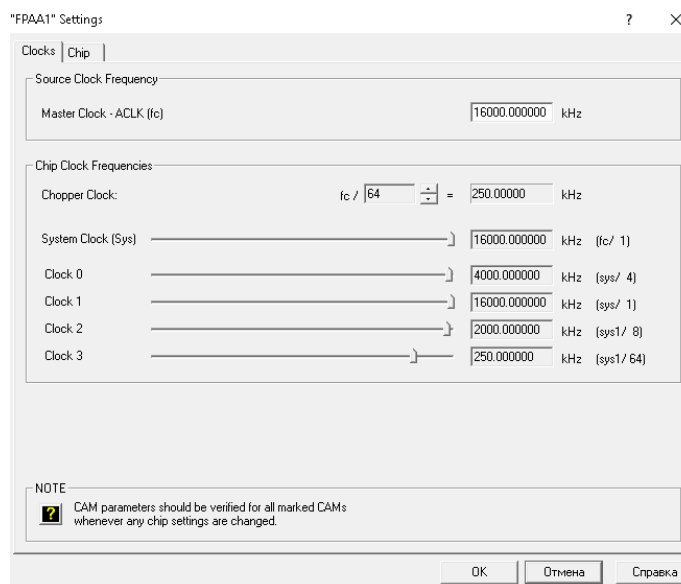


Рисунок 24 – Установка значений частот

Во вкладке **Chip** настраиваем дополнительные параметры ПАИС, включающие в себя элементы управления энергопотреблением, подтягивающими резисторами и поддержку конфигурации ПЗУ, как показано на рисунке 25.

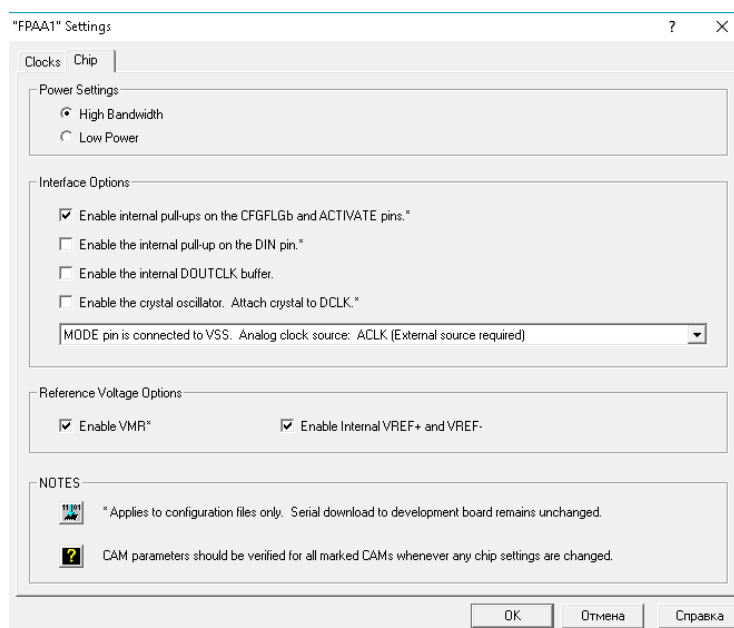


Рисунок 25 – Настройка дополнительных параметров

Затем в диалоговом окне «preferences...» последовательно настраиваем пять подчиненных диалоговых окон доступ к которым осуществляется через соответствующие закладки. Закладка **display** содержит элемент типа «флажок» «showwirelabels», который управляет отображением всех этикеток проводов. На этой же закладке есть управляющий элемент «enablesound», позволяющий включить или отключить звуковое сопровождение всех действий в среде. На закладке **wire** настраиваем элементы управления, устанавливающие значения «по умолчанию» для шрифта этикеток, цвета проводов и

графического типа соединения. Закладка serialport используется для установки типа последовательного порта. Производим настройку как показано на рисунке 26.

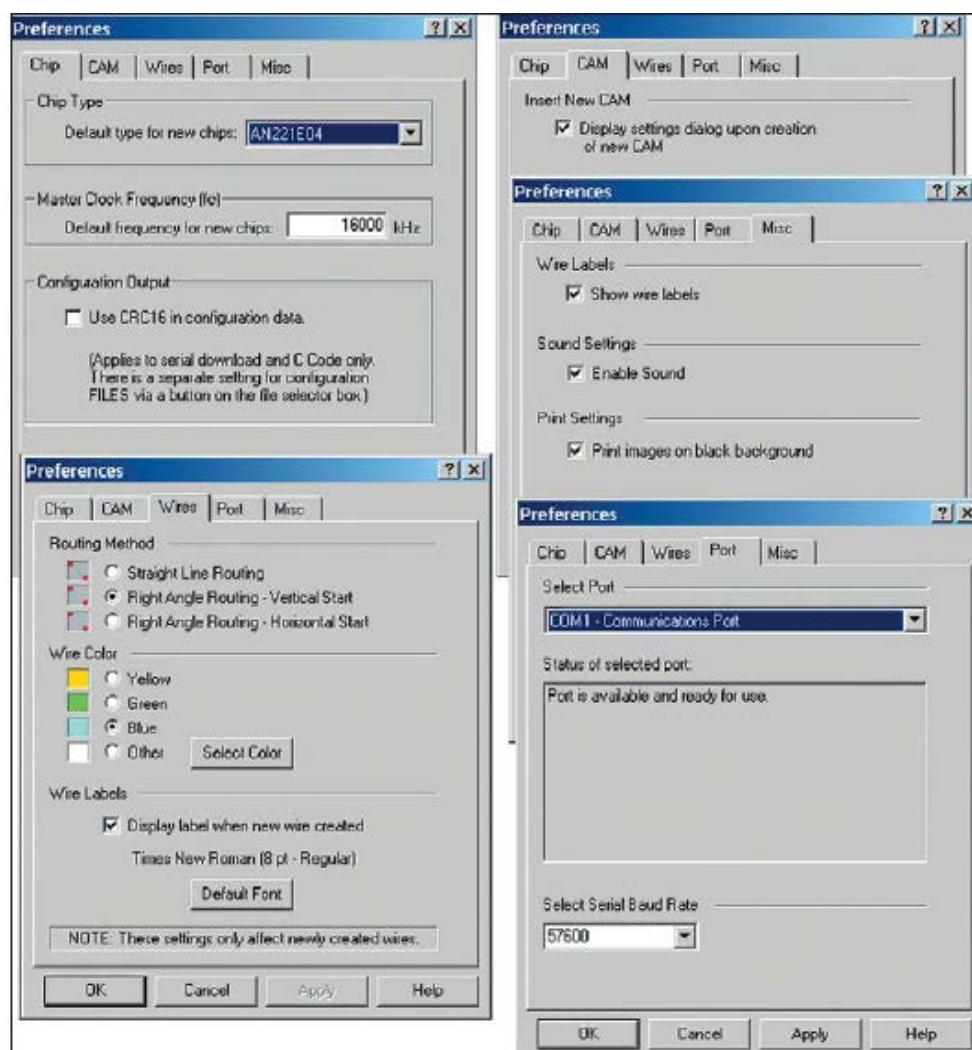


Рисунок 26 – Настройка параметров в диалоговом окне «preferences...»

Для выбора генераторов стандартных сигналов нажать клавишу «m» на клавиатуре для того, чтобы открылось окно выбора КАМ (рисунок 27).

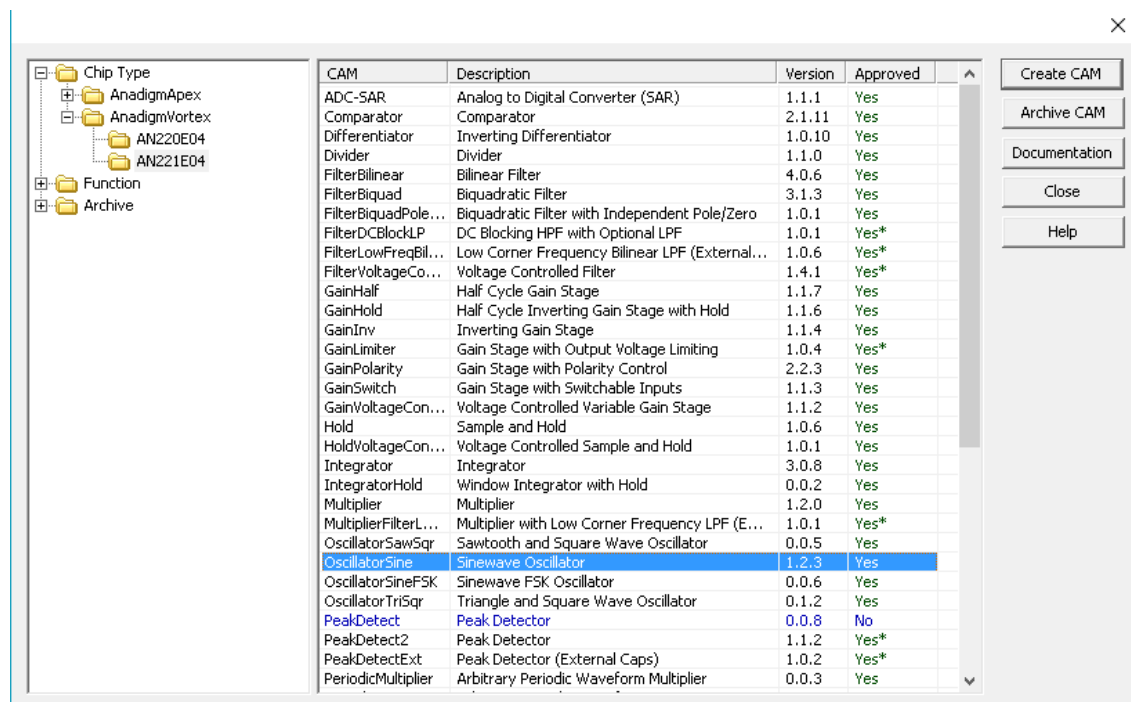


Рисунок 27 – Выбор генераторов стандартных сигналов

Для поиска нужного генератора КАМ использовал полосу прокрутки справа. Подвел к нему курсор мыши и щелкнул по нему левой кнопкой. После щелчка левой кнопкой мыши по кнопке *Create CAM* (или нажатия клавиши *Enter*) возле курсора появилось бледное изображение КАМ. Это означает, что выбранный КАМ теперь можно разместить в поле ПАИС, представленном в окне разработки. После того как выбранный КАМ установлен в нужное место (еще один щелчок левой кнопкой) (рисунок 28), появилось окно *Set CAM Parameters* (рисунок 29), предназначенное для установки параметров КАМ. Перечень параметров зависит от настроек в *Settings*→*Preferences...* на закладке *CAM*. Содержимое окна *Set CAM Parameters* зависит от типа выбранного КАМ, но, как правило, в нем есть все доступные пользователю настройки параметров для выбранного модуля. В генераторах выставляем параметр частоты равный 50 кГц и амплитуду сигнала равную 2 В. В управляющем элементе для выбора одного из четырех значений частоты внутреннего тактового генератора выбираем 2000 кГц. После нажатия кнопки ОК заданные параметры будут установлены, окно автоматически закроется.

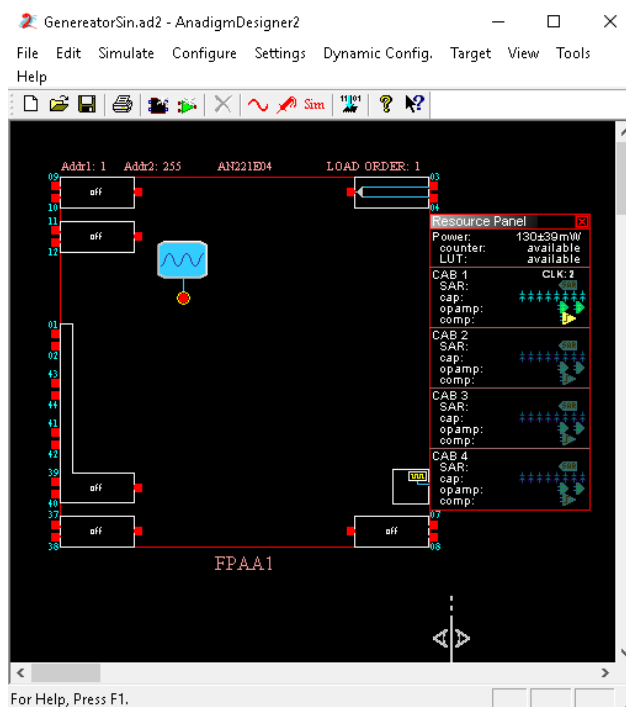


Рисунок 28 – Установка выбранного КАМ

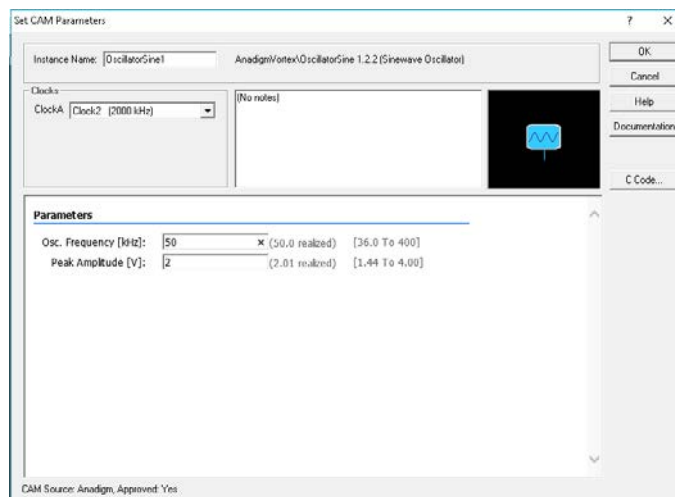


Рисунок 29 – Установка параметров КАМ

Разводка схемы осуществляется при помощи нажатия клавиши «w» на клавиатуре. Курсор мыши теперь выглядит как карандаш. Разводка соединений осуществляется с помощью левой кнопки мыши. Соединил выход генератора со входом ячейки OutputCell (рисунок30). Выход ячейки OutputCell идет к ножкам платы O1P и O1N.

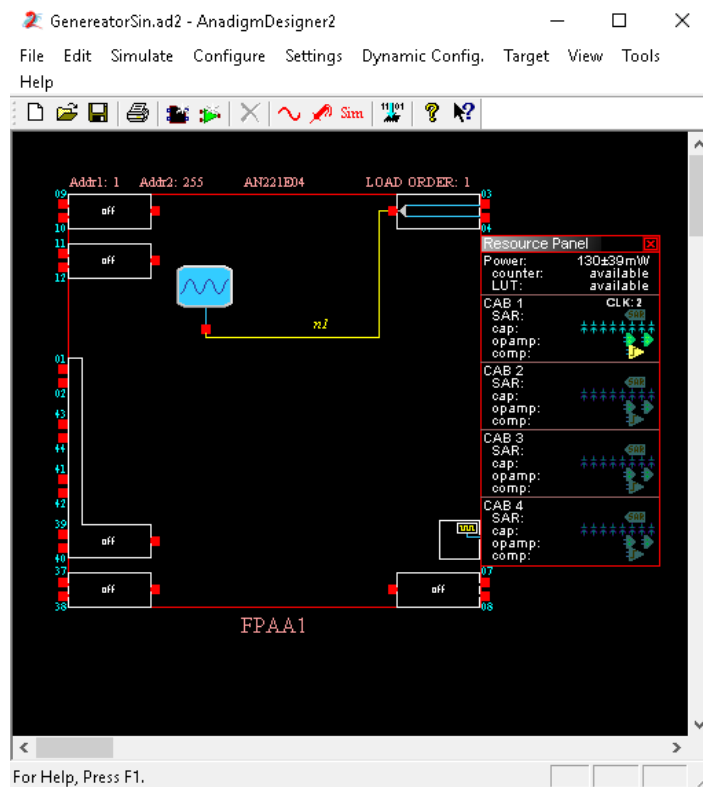


Рисунок 30 –Соединения КАМ с выходом

В верхней вкладке программы выбрана кнопка симуляции, после чего появилось окно симуляции, в котором представлены осциллограммы стандартных сигналов. (рисунок 31, 32, 33)

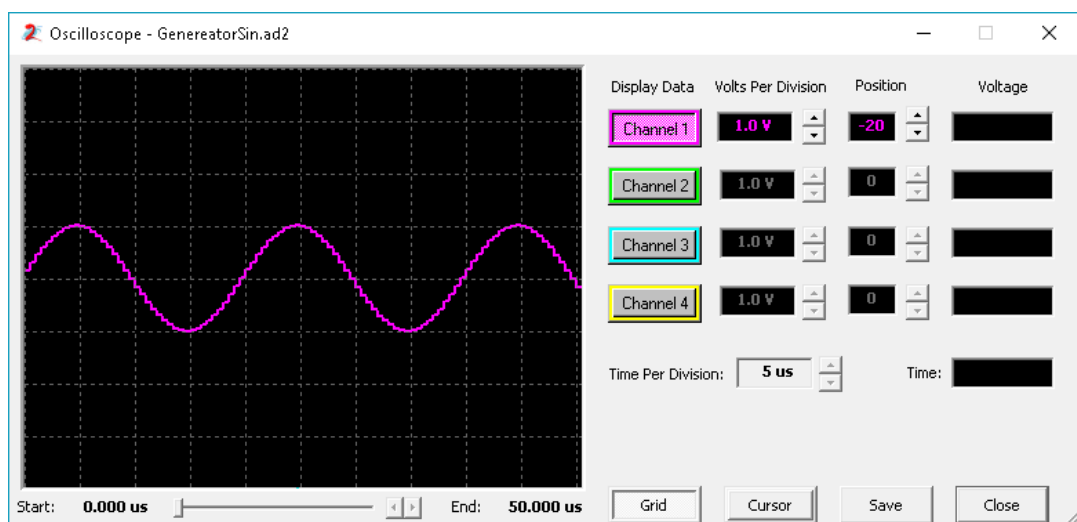


Рисунок 31 – Осциллограмма синусоидального сигнала

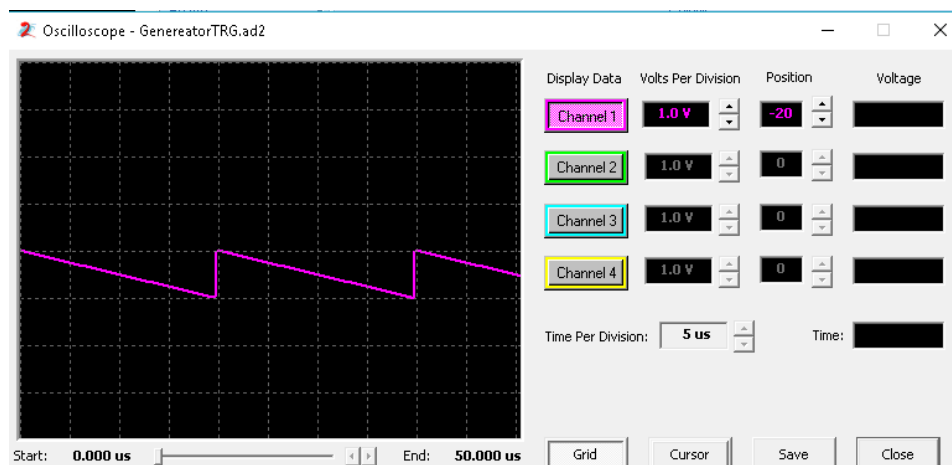


Рисунок 32 – Осциллограмма пилообразного сигнала

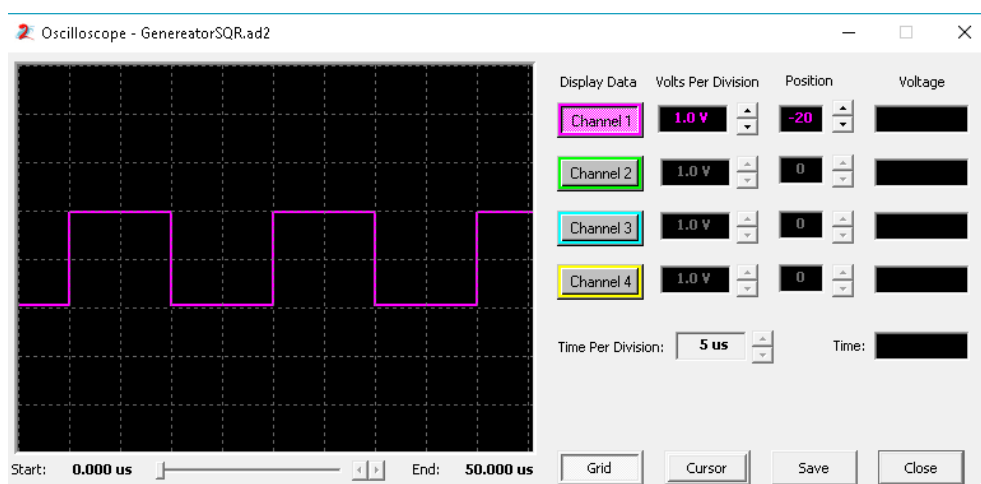


Рисунок 33 – Осциллограмма прямоугольного сигнала

3.2 Динамическое программирование аналоговых схем Anadigm алгоритмическим методом

Для динамического конфигурирования программируемых аналоговых микросхем Anadigm рекомендуется управляющий метод (State-Driven), который задействует предварительно сконфигурированные данные. Алгоритмический метод более сложен в реализации и для выполнения сложного алгоритма требует быстродействующего микроконтроллера, который с помощью специально сгенерированного алгоритма будет рассчитывать конфигурационный код в зависимости от входных параметров, а затем передавать его в ПАИС. Такое решение позволит гибко изменять параметры схемы.

Для макетирования схемы и ее отладки будем использовать комплект разработчика AN221K04-DVLP2 (рисунок 34), для создания аналоговой схемы и получения специального

алгоритма — бесплатно распространяемый САПР AnadigmDesigner2, а так же программу Visual C++, с помощью которой мы сможем создать приложение с графическим интерфейсом для управления параметрами нашей схемы.

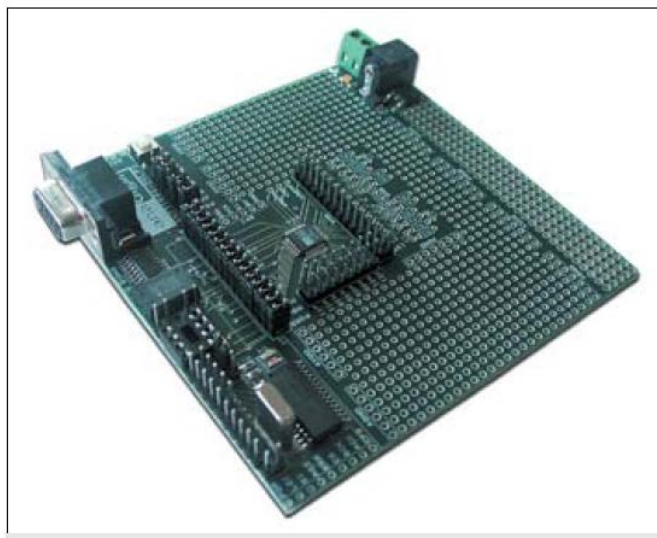


Рисунок 34 -Отладочный комплект AN221K04-DVLP2

Конфигурационные данные, рассчитанные программой на персональном компьютере, передаются на отладочный комплект AN221K04- DVLP2 по кабелю через интерфейс RS232 или USB.

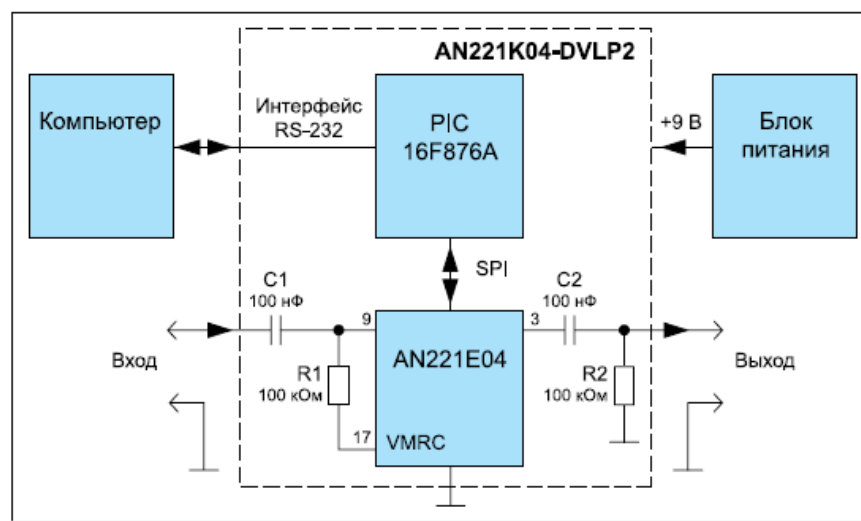


Рисунок 35 - Электрическая блок-схема подключения отладочного комплекта AN221K04-DVLP2 для динамического программирования алгоритмическим методом

Обработка сигнала в ПАИС AN221E04 осуществляется при помощи однополярных дифференциальных сигналов, с постоянной составляющей +2 В. С учетом особенностей работы ПАИС Anadigm разработаем электрическую схему и блок-схему подключения отладочного комплекта. Для работы с недифференциальным униполярным аналоговым сигналом необходимо настроить входную ячейку для работы с таким сигналом, а так же добавить во входной сигнал постоянную составляющую от внутреннего источника опорного напряжения. Питание отладочного комплекта осуществляется от внешнего блока с рабочим током не менее 0,5 А и выходным напряжением 10,5–20 В с использованием 3,5- мм разъема типа Jack (положительный центральный контакт) или стабилизированным напряжением 5,5–10 В через терминальный разъем, установленный на печатной плате. На рисунке 35 представлена схема подключения платы к персональному компьютеру и блоку питания.

После создания схемы в программе AnadigmDesigner2, необходимо определить метод динамического конфигурирования. Для выбора динамического программирования алгоритмическим методом необходимо в меню программы AnadigmDesigner2 выбрать DynamicConfig («Динамическое конфигурирование») а затем в открывшемся подменю выбрать Visual C++ Prototype... («Создание приложения с помощью VisualC++»). В открывшемся окне (рисунок 36) необходимо задать название проекта (ProjectName) и путь для сохранения проекта (Dirrectory), затем нажать кнопку Generate, чтобы создать необходимые файлы для проекта, после чего проект будет автоматически открыт в программе Visual C++.

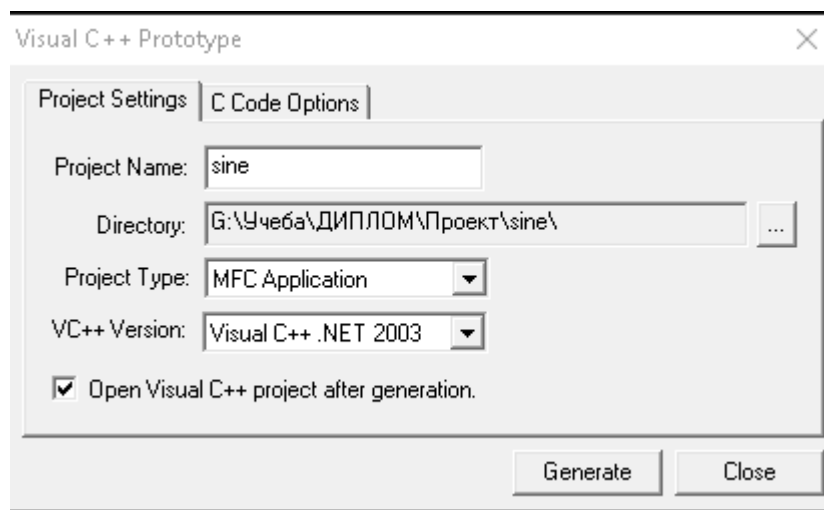


Рисунок 36 – Окно для создания приложения C++

Для удобства пронумеруем этапы создания приложения.

1. В открывшемся окне Visual C++ в SolutionExplorer откроем файл AnadigmSineDlg.cpp, найдем функцию OnInitDialog и добавим в нее код, устанавливающий для генератора предельные значения изменения частоты.

2. В функцию Update Sample Controls добавил строчки, вызывающие функцию расчета новой конфигурации в зависимости от изменившегося значения частоты и загрузки полученного конфигурационного кода в ПАИС Anadigm.
3. Чтобы добавить слайдер для изменения параметра коэффициента передачи фильтра, необходимо вызвать пользовательский графический интерфейс приложения путем двойного клика в окне Resource View на его названии IDD_ANADIGMSine_DIALOG (рисунок 37)

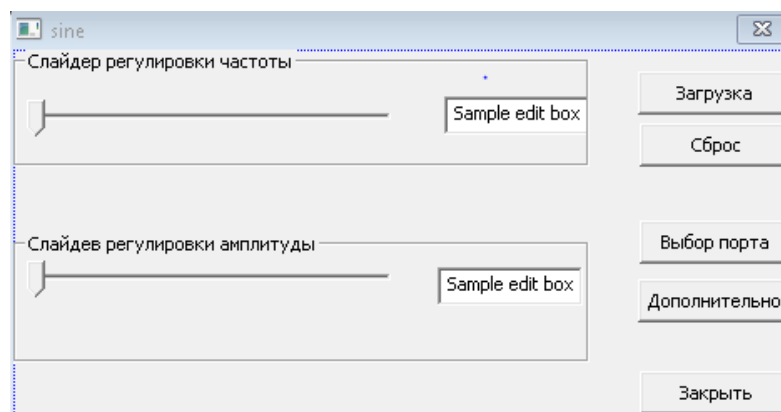


Рисунок 37 – Графический интерфейс

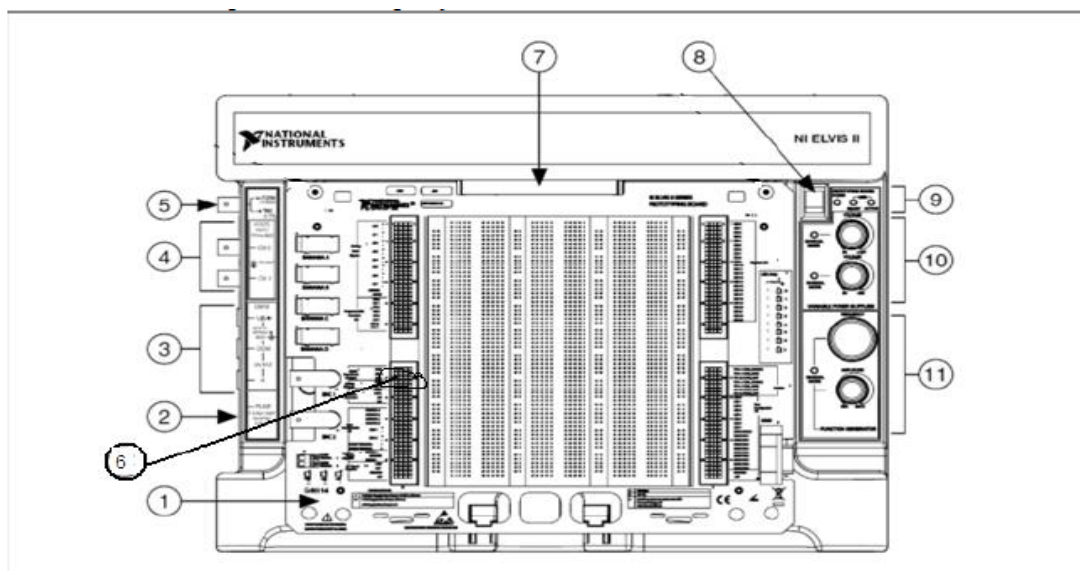
4. Нужно скопировать текущий слайдер, регулирующий частоту генератора и вставить его копию чуть ниже; при необходимости поменять его название. Затем, вызвав правой кнопкой дополнительное меню AddVariable... , добавить параметр «m_gainSlider» в поле Variablename и поменять параметр в поле Access на «private».
5. Добавим необходимые переменные в файл AnadigmSineDlg.h. Для этого найдем строку CSliderCtrlm_gainSlider и добавим за ней строчки, декларирующие переменные максимального и минимального значения передачи.
6. Затем в файле AnadigmSineDlg.cpp находим функцию OnHScroll и добавляем в нее строки, позволяющие изменять слайдером значения переменной m_gainValue.
7. В файле AnadigmSineDlg.cpp создадим новую функцию, подобную UpdateSampleControls, и назовем ее UpdateGainControls. С ее помощью мы будем менять амплитуду генератора слайдером, используя логарифмическую шкалу.
8. Задекларируем функцию UpdateGainControls, добавив в файл AnadigmSineDlg.h строку в раздел private.
9. Выполним проект и убедимся в отсутствии предупреждений и ошибок.
10. Теперь необходимо переключить работу VisualStudio из режима Debug в режим Release, чтобы получить исполняемый файл приложения с меньшим размером. Для

этого в программе VisualStudio заходим в меню Build и выбираем Configuration Manager. В открывшемся окне меняем «Active solution configuration» на «Realise» и запускаем выполнение проекта заново. Теперь в поддиректории Realise нашего проекта будет содержаться исполняемый файл, с помощью которого мы можем управлять через компьютер настройками фильтра нижних частот, выполненного на базе отладочного комплекта AN221K04- DVLP2.

Теперь, с помощью программы мы можем менять параметры генератора ползунками или задавать точные значения в текстовых полях. Конфигурационные данные, рассчитанные компьютером в соответствии с нашими установками, будут автоматически переданы по кабелю в ПАИС, установленную на отладочном комплекте AN221K04- DVLP2, при этом генератор будет изменять параметры, не прерывая своей работы. Полный код программы представлен в Приложении А.

3.3 Снятие характеристик генераторов стандартных сигналов при помощи программно-аппаратной среды NI ELVIS II

Программно-аппаратная среда NI ELVIS II включает в себя аппаратную составляющую для построения макетов радиоэлектронных устройств и выполнения измерений их характеристик, а также программное обеспечение. Основной целью NI ELVIS II является отработка характеристик радиоэлектронных устройств, схемы которых смоделированы в виртуальной среде программы Multisim, после их реализации на реальных электрорадиокомпонентах. Однако этап моделирования может быть пропущен, если отрабатывается схема, разработанная традиционным способом. Аппаратная часть NI ELVIS II состоит из трех компонентов – рабочей станции, макетной платы и компьютера. ПО обеспечивает сопряжение работы всех трех частей для функционирования всей среды NI ELVIS II. Схема расположения компонентов на панели управления рабочей станции NI ELVIS II приведена на рисунке 38.



- | | |
|---|---|
| 1. Макетная плата NI ELVIS II | 6. Разъемы для измерения емкости и индуктивности |
| 2. Плавкий предохранитель мультиметра | 7. Разъем для подключения макетной платы |
| 3. Клеммы для подключения к мультиметру | 8. Выключатель питания макетной платы |
| 4. Разъемы для подключения к осциллографу | 9. Световые индикаторы статуса |
| 5. Разъем выхода функционального генератора или входа цифрового запуска | 10. Элементы управления регулируемыми блоками питания |
| | 11. Элементы управления функциональным генератором |

Рисунок 38 –Рабочая станции NI ELVIS II с макетной платой

Основные входы макетной платы описаны на рисунке 39 . Входы сгруппированы по функциональному назначению так же, как они расположены.

Название входа	Тип	Описание
AI<0..7>+	Аналоговые входы	Положительный и отрицательный входы дифференциальных аналоговых каналов 0 [^] 7 (+).
AI SENSE	Аналоговые входы	Аналоговый входной уровень - опорный уровень аналоговых каналов, сконфигурированных для работы в режиме несимметричного входа без заземления.
AI GND	Аналоговые входы	Общая цепь аналоговых входов модуля ввода-вывода -"аналоговая земля".
DUT+	Цифровой мультиметр.	Возбуждающий вход измерения емкости и индуктивности (цифровой мультиметр).
DUT-	Цифровой мультиметр.	Виртуальная "земля" и потенциал токовой цепи для измерения емкости и индуктивности (цифровой мультиметр).
FGEN	Функциональный генератор	Выход функционального генератора
+15 V	Источники постоянного тока	Выход источника питания +15 В, нерегулируемый.
-15 V	Источники постоянного тока	Выход источника питания -15 В, нерегулируемый.
GROUND	Источники постоянного тока	Общий контакт нерегулируемых блоков питания ("земля").
+5 V	Источники постоянного тока	Выход источника питания +5 В, нерегулируемый.
DIO <0..23>	Цифровой ввод/вывод	Линии цифрового ввода/вывода общего назначения, используемые для чтения и записи данных.

LED <0..7>	Пользовательские гнезда ввода/вывода	Светодиоды - питание 5 В, ток управления 10 мА.
+5 V	Источник постоянного тока	Выход источника питания +5 В, нерегулируемый.
GROUND	Источник постоянного тока	Общий контакт ("земля").

Рисунок 39 - Основные входы макетной платы

Поскольку в системе используются дифференциальные аналоговые входные каналы, то где-то в схеме подключения выводов необходимо предусмотреть точку заземления. Измерение осуществляется правильно, если один из выводов источника измеряемого сигнала соединен контактом AI GND. На практике это может означать, что землю испытуемого макета схемы необходимо соединить с аналоговой землей осциллографа (AIGND) на макетной плате.

Все необходимые приборы находятся на рабочем столе компьютера в папке «Приборы». Чтобы запустить какой-нибудь прибор, просто нажмите на соответствующую иконку (рисунок 40).



Рисунок 40 - Ярлыки линейки виртуальных приборов.

Для измерения характеристик построенных генераторов будем использовать следующие приборы:

1. Осциллограф Scope.
2. Анализатор спектра DSA.
3. Анализатор амплитудно-частотной характеристики Bode.

Перед измерением основных характеристик генераторов построенных в AnadigmDesigner2 следует загрузить конфигурацию в саму ПАИС Anadigm. Во-первых, следует подключить питание платы при помощи 3.5 разъема. Во-вторых, подключить кабель USB к самой плате и персональному компьютеру, с которого будет происходить загрузка всех параметров в ПАИС. Для этого используем команду Write configuration data to serial port(или по соответствующей ей «горячей» клавише Ctrl+w) происходит передача конфигурационных данных с FPAА-устройства на последовательный порт компьютера. После проделанных операций все данные будут загружены в ПАИС и она будет готова к работе.

Отладочные комплекты Anadigm содержат встроенную схему с USB портом. По команде Writeconfigurationdatatoserialport(selectedchipsonly) происходит передача

конфигурационных данных выбранных ПАИС на последовательный порт компьютера. По этой команде происходит запись конфигурационных данных ПАИС в файл данных. По умолчанию, этот файл будет сохранен в той же директории, что и активный файл разработки (файл с расширением .ad2) и унаследует имя корневого файла. Этот файл конфигурационных данных может быть использован для программирования последовательных загрузочных ПЗУ или в качестве файла данных для микроконтроллера, управляющего ПАИС.

3.4 Исследование осциллограмм

На рабочем столе в папке «Приборы» кликните иконку с надписью Score. На экране появится лицевая панель виртуального прибора NI ELVISmxOscilloscope. Осциллограф обладает всей функциональностью традиционных осциллографов, использующихся в учебных лабораториях. Частота оцифровки сигналов осциллографом определяется возможностями многофункциональной платы ввода/вывода. Имеется возможность регулирования чувствительности и временной развертки, а также положения кривых на экране. Пользователь может осуществлять среднеквадратичные измерения напряжения, размах входных сигналов и их частоту. Для осуществления прецизионных измерений могут быть использованы курсоры. Имеется возможность задавать источник (аналоговый, цифровой) и тип запуска осциллографа. Кроме этого система NI ELVIS II способна осуществлять автоматическую коммутацию выходов функционального генератора и анализатора импеданса на входы осциллографа.

Для исследования осциллограмм генераторов следует правильно подключить ПАИС с рабочей станции. Для этого при помощи проводов соединим выходные ножки ПАИС 01P и 01N с аналоговыми входами рабочей станции AI 4 + и AI 4 –

На лицевой панели виртуального прибора NI ELVISmxOscilloscope выберем соответствующий канал AI 4, нажмем кнопку RUN на виртуальной панели. После этого в окне осциллографа появится функция. Так как мы использовали генераторы трех типов, то осциллограммы будут выглядеть так, как показано на рисунках 41, 42, 43.

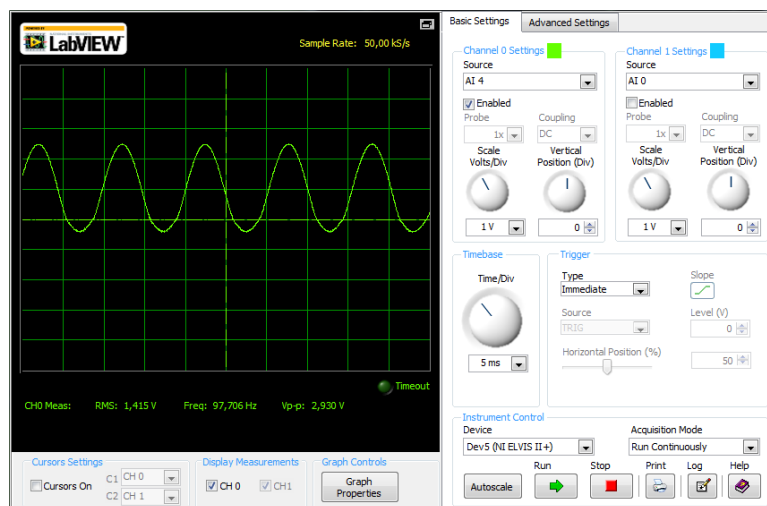


Рисунок 41 – Осциллограмма синуса

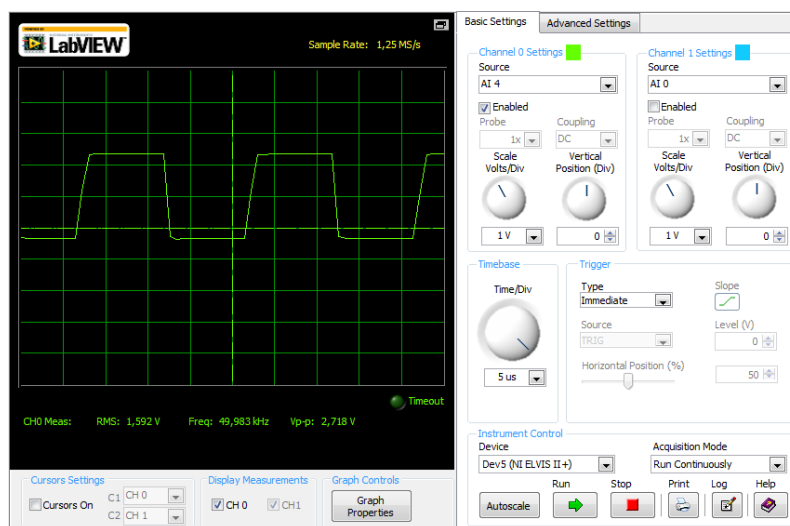


Рисунок 42 – Осциллограмма прямоугольного сигнала

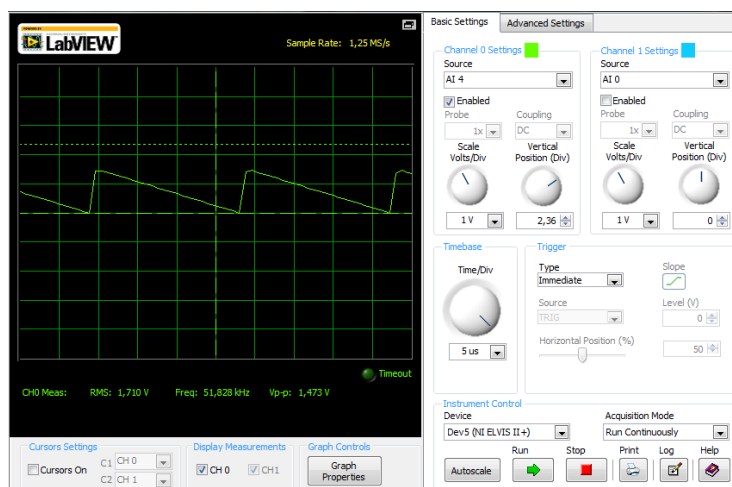


Рисунок 43 – Осциллограмма пилообразного сигнала

3.5 Исследование спектра

Исследование спектров генераторов будет происходить при помощи анализатора спектра. Анализатор спектра (DSA)— прибор для наблюдения и измерения относительного распределения энергии электромагнитных колебаний в полосе частот. Известно, что только идеальная синусоида имеет один отклик на частотной шкале, и именно на собственной частоте. Все реальные сигналы имеют спектр в виде откликов, распределенных по частотной шкале, поскольку содержат сумму нескольких идеальных синусоид (гармоник) с кратными частотами и разными фазами относительно основной гармоники. Анализатор спектра применяется при анализе искажений сигнала в высококачественных мощных усилителя в аудиотехнике, при анализе гармоник в преобразователях частоты приемников.

Основными характеристиками анализаторов спектра являются: диапазон анализируемых частот, разрешение по частоте и амплитуде, тип измерений (параллельный или последовательный) и тип проводимого анализа (скалярный или векторный). Скалярные анализаторы дают информацию только об амплитудах гармонических составляющих спектра, а векторные предоставляют также информацию и о фазовых соотношениях. Анализатор спектра позволяет определить амплитуду и частоту спектральных компонент, входящих в состав анализируемого процесса. Важнейшей его характеристикой является разрешающая способность: наименьший интервал Δf по частоте между двумя спектральными линиями, которые ещё разрешаются анализатором спектра. Анализатор спектра может дать истинный спектр только тогда, когда анализируемое колебание $x(t)$ периодически, либо существует только в пределах временного интервала проведения измерений T .

Рассмотрим анализатор спектра, созданный на базе NI ELVIS II. Он относится к цифровым анализаторам параллельного типа в виде БПФ-анализатора, который вычисляет спектр с помощью алгоритмов быстрого преобразования Фурье (БПФ).

Для исследования спектра сигналов генераторов следует правильно подключить ПАИС с рабочей станции. Для этого при помощи проводов соединим выходные ножки ПАИС 01P и 01N с аналоговыми входами рабочей станции AI 4 + и AI 4 –

На лицевой панели виртуального прибора NI ELVISmxDynamicSignalAnalyzer выберем соответствующий канал AI 4, выставим интервал по частоте, нажмем кнопку RUN на виртуальной панели. После этого в окне анализатора появится функция зависимости уровня сигнала от частоты. Так как мы использовали генераторы трех типов, то функции будут выглядеть так, как показано на рисунках 44, 45, 46.

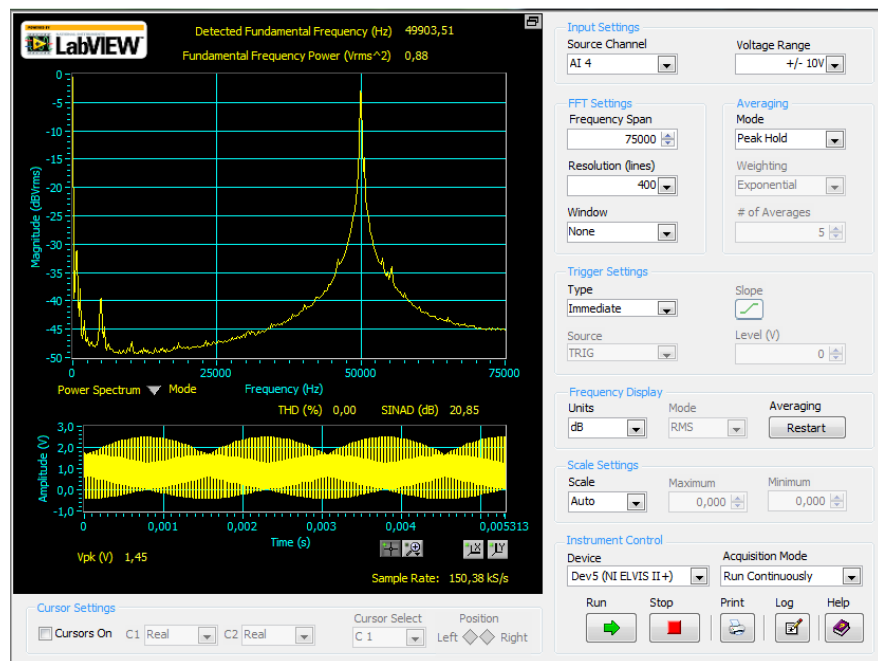


Рисунок 43 – Спектр синусоидального сигнала



Рисунок 44 – Спектр прямоугольного сигнала

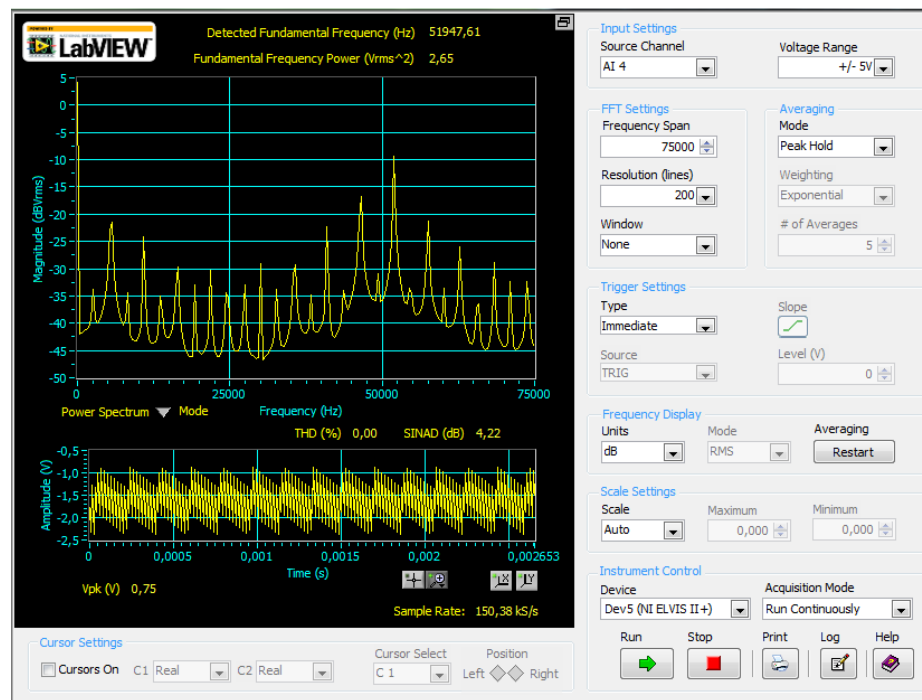


Рисунок 45 – Спектр пилообразного сигнала

3.6 Исследование амплитудно-частотной характеристики Бode

Амплитудно-частотная характеристика (АЧХ) или график Бode представляет собой очень удобный графический формат частотных характеристик цепей переменного тока. Амплитудно-частотная характеристика (АЧХ) или амплитудный отклик – это зависимость коэффициента передачи или усиления схемы, выраженного в децибелах, от частоты. Фазово-частотная характеристика (ФЧХ) или фазовый отклик – это зависимость разности фаз между входным и выходным сигналами от частоты.

Анализатор Бode позволяет выполнить перестройку в целом диапазоне частот от начальной до конечной с шагом ΔF . Также можно выбрать амплитуду тестирующего синусоидального сигнала. В анализаторе Бode используется ИПУ (интерактивная панель управления) functiongenerator для создания тестирующей осциллограммы. Поэтому обязательно выходные разъемы FGEN необходимо присоединить к входам исследуемой схемы, а также к контактам +AI 1 и к земле. Выход схемы подается на контакты +AI 0 и к земле. Пример соединения ПАИС и рабочей станции приведен на рисунке 46.

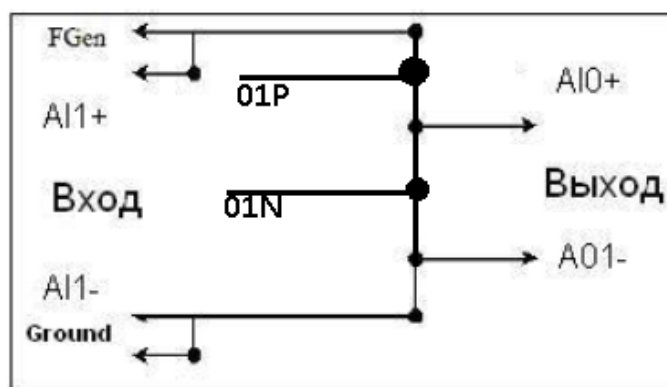


Рисунок 46 - Пример соединения ПАИС и рабочей станции

После того как подключение было произведено следует запустить анализатора АЧХ, выбрать функцию BodeAnalyzer из меню запуска инструментов NI ELVIS II. Установить начальную и конечную частоту равную 100 Гц и 200 кГц соответственно. И установить число точек на декаду равную 10. Затем нажать кнопку RUN на виртуальной панели. Результаты снятия АЧХ для генераторов представлены на рисунках 47, 48, 49.

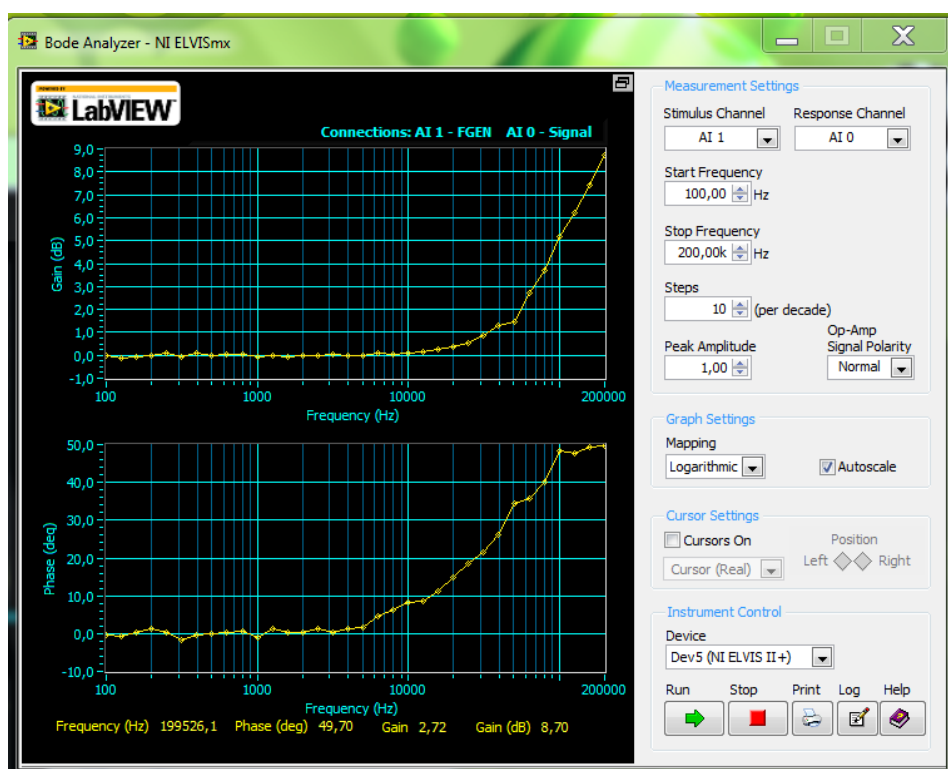


Рисунок 47 - Результаты снятия АЧХ с генератора синуса

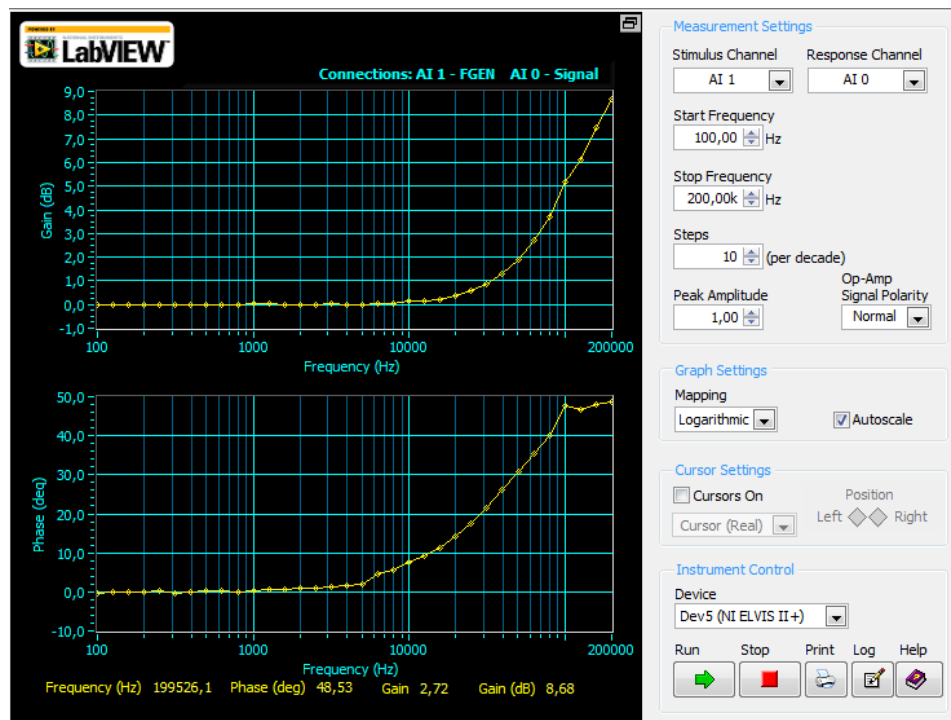


Рисунок 48 - Результаты снятия АЧХ с генератора прямоугольных сигналов

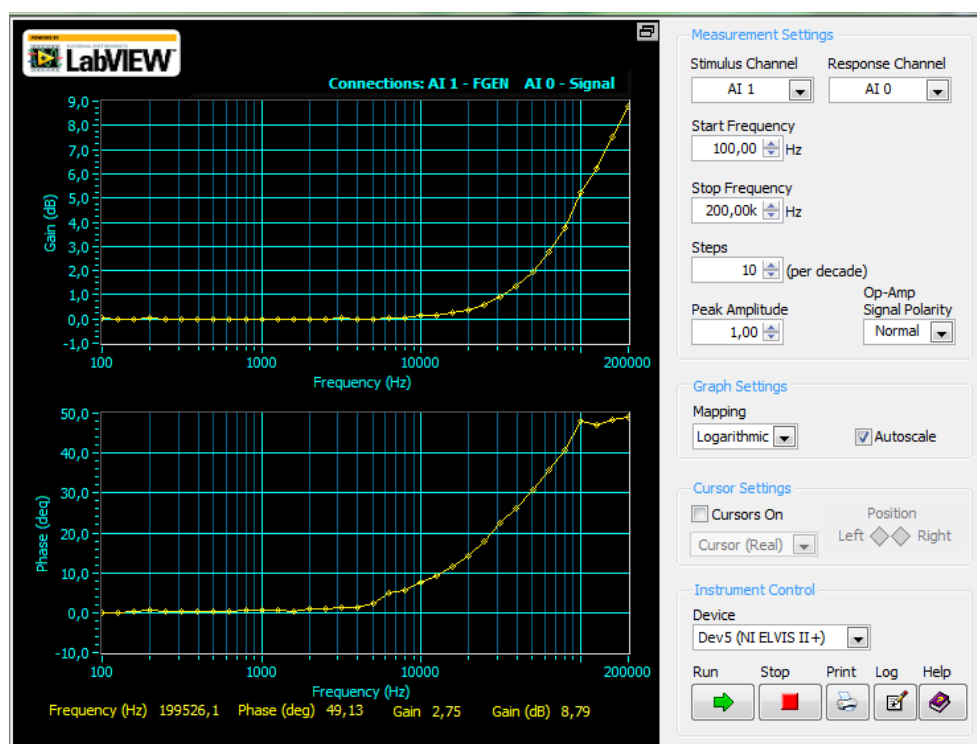


Рисунок 49 - Результаты снятия АЧХ с генератора пилообразных сигналов

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы было сделано следующее:

1. Изучена структура программируемой аналоговой интегральной схемы Anadigm AN221k04.
2. Освоена система автоматизированного проектирования AnadigmDesigner2.
3. Спроектированы и реализованы генераторы синусоидальных, прямоугольных и пилообразных сигналов.
4. Исследованы осциллограммы, спектр и амплитудно-частотные характеристики генераторов при помощи программно-аппаратной среды NI ELVIS II.

Таким образом, были разработаны методические указания для выполнения лабораторного практикума на основе программируемой аналоговой интегральной схемы Anadigm AN221k04.

Рассмотренный способ создания блоков генерации стандартных сигналов позволяет для ряда практических случаев получить выигрыш по точности при относительно простой аппаратной реализации.

ЛИТЕРАТУРА

1. Немудров В. Системы на кристалле / В. Немудров, Г. Мартин. – М. : Техносфера, 2004. – 216 с.
2. Максфилд, К. Проектирование на ПЛИС. Курс молодого бойца / К. Максфилд. – М. : ДОДЭКА, 2007. – 408 с.
3. Шагурин, И. И. Системы на кристалле. Особенности реализации и перспективы применения / И. И. Шагурин // Электронные компоненты. – 2009. – № 1. – С. 37–39.
4. Гауси М., Лакер К. Активные фильтры с переключаемыми конденсаторами / Пер. с англ. М.: Радио и связь, 1986.
5. Микушин А.В., Сажнев А.М., Сединин В.И. Цифровые устройства и микропроцессоры. СПб, БХВ-Петербург, 2010.
6. Угрюмов Е. П. Цифровая схемотехника. СПб, БХВ-Петербург, 2004.
7. Курбатов. А. Программируемые аналоговые интегральные схемы. Жизнь продолжается. - Компоненты и технологии, 2000, №2.
8. Полищук А. Система автоматизированного проектирования программируемых аналоговых интегральных схем AnadigmDesigner2. Часть 1. Первый шаг: знакомство с интерфейсом // Компоненты и технологии. 2005. № 5. http://kit-e.ru/articles/plis/2005_5_162.php
9. Полищук А. Система автоматизированного проектирования программируемых аналоговых интегральных схем AnadigmDesigner2. Часть 2. Особенности разработки проектов в среде программирования ПАИС Anadigm. Компоненты и технологии, 2005, №9.
10. Щерба А. Построение входных и выходных цепей программируемых аналоговых схем Anadigm // Компоненты и технологии. 2008. № 12.
11. Полищук А. Программируемые аналоговые интегральные схемы Anadigm. Часть 1. Структура и характеристики. – Компоненты и технологии, 2005, №1
12. Кулапин В.И., Князьков А.В., Маньков А.М. Использование программируемых аналоговых интегральных схем (ПАИС) в составе устройства для измерения проводимости кондуктометрического датчика. 2012. <http://cyberleninka.ru/article/n/ispolzovanie-programmiruemyyh-analogovyh-integralnyh-schem-pais-v-sostave-ustroystva-dlya-izmereniya-provodimosti>
13. Полищук А. Программируемые аналоговые ИС Anadigm: применение конфигурируемых аналоговых модулей в составе программы AnadigmDesigner2. – Компоненты и технологии, 2007, №12 http://kit-e.ru/articles/plis/2007_12_12.php

14. Щерба А. Динамическое программирование аналоговых схем Anadigm алгоритмическим методом //Компоненты и технологии. 2012. № 8. http://kit-e.ru/articles/plis/2012_8_68.php
15. Полищук А. Система автоматизированного проектирования программируемых аналоговых интегральных схем AnadigmDesigner2. Часть 1.2. Знакомство с интерфейсом //Компоненты и технологии. 2005. № 6. http://kit-e.ru/articles/plis/2005_6_78.php
16. Александр Полищук, "Программируемые аналоговые ИС Anadigm: весь спектр аналоговой электроники на одном кристалле. Первое знакомство.", СОВРЕМЕННАЯ ЭЛЕКТРОНИКА , Декабрь 2004 © СТА-ПРЕСС, с.8-11.
17. Александр Полищук, "Программируемые аналоговые ИС Anadigm: структура и принцип построения.", СОВРЕМЕННАЯ ЭЛЕКТРОНИКА № 1 2005, © СТА-ПРЕСС., с. 24-26.

Полный код программы для изменения параметров генератора

```
#include "stdafx.h"
#include "sine.h"
#include "sineDlg.h"
#include "AboutDlg.h"
#include "DownloadManager.h"
#include ".\sinedlg.h"

#ifdef _DEBUG
#define DEBUG_NEW
#undef THIS_FILE
staticchar THIS_FILE[] = __FILE__;
#endif

//crs 10/4/2007
#define AUTO_EXEC 1
#define ZERO_CROSS 0
//crs 10/4/2007 end

// Dave Lovell added these defines
#define RESET_FIRST 0
#define NO_RESET 1

BEGIN_MESSAGE_MAP(sineDlg, CDialog)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_PORT, OnPort)
    ON_BN_CLICKED(IDC_ABOUT, OnAbout)
    ON_BN_CLICKED(IDC_DOWNLOAD, OnDownload)
    ON_BN_CLICKED(IDC_RESET, OnReset)
    ON_WM_HSCROLL()
    ON_EN_KILLFOCUS(IDC_SAMPLE_EDIT, OnSampleEditChanged)

    ON_EN_CHANGE(IDC_SAMPLE_EDIT, OnEnChangeSampleEdit)
    ON_NOTIFY(NM_CUSTOMDRAW, IDC_SAMPLE_SLIDER, OnNMCustomdrawSampleSlider)
    ON_NOTIFY(NM_CUSTOMDRAW, IDC_SAMPLE_SLIDER2, OnNMCustomdrawSampleSlider2)
    ON_BN_CLICKED(IDOK, OnBnClickedOk)
END_MESSAGE_MAP()

sineDlg::sineDlg(CWnd* pParent)
    : CDialog(sineDlg::IDD, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void sineDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_SAMPLE_SLIDER, m_sampleSlider);
    DDX_Control(pDX, IDC_SAMPLE_EDIT, m_sampleEdit);
}

BOOL sineDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Setup application icon
    SetIcon(m_hIcon, TRUE);
    SetIcon(m_hIcon, FALSE);

    // Add the "About..." menu item
    ConfigureAboutMenu();

    // Initialize the sample slider\edit controls
    m_sampleMin = 2;
    m_sampleMax = 4;
    m_sampleValue = 10.0;

    // The range [0, 100] will be percent. The slider control cannot use
    // floating point values, so we map [0, 100] on to our desired scale
```

```

// to achieve floating point results.
m_sampleSlider.SetRange(0, 100);

// Initial update of the sample controls
UpdateSampleControls();

// return TRUE unless you set the focus to a control
return TRUE;
}

void sineDlg::ConfigureAboutMenu()
{
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }
}

void sineDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        AboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.
void sineDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this function to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR sineDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void sineDlg::OnPort()

```

```

{
DownloadManager::Instance()->ShowPortDlg();
}

void sineDlg::OnAbout()
{
AboutDlg dlg;
dlg.DoModal();
}

void sineDlg::OnDownload()
{
m_FPAA1.ExecutePrimaryConfig(RESET_FIRST);
}

void sineDlg::OnReset()
{
m_FPAA1.ExecuteReset();
}

void sineDlg::UpdateSampleControls()
{
// Map the actual sample value to the range [0, 100] for the slider
int percent = (int)(100.0 * (m_sampleValue - m_sampleMin) / (m_sampleMax - m_sampleMin));

// Update the slider position
m_sampleSlider.SetPos(percent);
m_sampleSlider.Invalidate();

// Format the text for the edit control
CString value;
value.Format("%.2f Units", m_sampleValue);

// Update the edit control text
m_sampleEdit.SetWindowText(value);

// NOTE:
//
// This would be the spot to perform C Code updates
// based on m_sampleValue.
//
// At this point, m_sampleValue holds the value that is
// displayed on the sample slider and edit box.
//
// To change the range of values of the slider change the
// following lines found in the OnInitialUpdate function above:
//
// m_sampleMin = 2;
// m_sampleMax = 4;
//
// EXAMPLE:
//
// m_chip1.GainHalf1.setGain(m_sampleValue);
// m_chip1.ExecuteReconfig(AUTO_EXEC);
}

// This function is used to handle the Enter key when the focus is on
// an edit control.
BOOL sineDlg::PreTranslateMessage(MSG* pMsg)
{
bool eatMessage = true;

if (pMsg->message == WM_KEYDOWN && pMsg->wParam == VK_RETURN)
{
if (GetFocus() == &m_sampleEdit)
{
OnSampleEditChanged();
}
else
{
eatMessage = false;
}
}
else
{
eatMessage = false;
}
}

```

```

    }

    if (eatMessage)
    {
        return true;
    }
    else
    {
        return CDialog::PreTranslateMessage(pMsg);
    }
}

// This function handles ALL horizontal sliders on the window. If you need
// to handle vertical sliders, that should go in the OnVScroll message handler.
void sineDlg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    // This switch determines which horizontal slider is changing
    switch (pScrollBar->GetDlgCtrlID())
    {
        case IDC_SAMPLE_SLIDER:
        {
            // Get the current position of the slider
            int percent = ((CSliderCtrl*) pScrollBar)->GetPos();

            // Map the percent onto the actual sample value
            m_sampleValue = m_sampleMin + (percent / 100.0) * (m_sampleMax - m_sampleMin);

            // Synchronize the slider and the edit control
            UpdateSampleControls();

            break;
        }

        default:
            // Nothing to do
            break;
    }
}

CDialog::OnHScroll(nSBCode, nPos, pScrollBar);
}

// This function is called by MFC when the edit control loses focus.
void sineDlg::OnSampleEditChanged()
{
    // Get the value in the edit box
    CString text;
    m_sampleEdit.GetWindowText(text);

    // Convert to a double
    m_sampleValue = atof(text);

    // Ensure the value is within the allowable range
    m_sampleValue = max(min(m_sampleValue, m_sampleMax), m_sampleMin);

    // Synchronize the slider and the edit control
    UpdateSampleControls();
}

void sineDlg::OnEnChangeSampleEdit()
{
    // TODO: If this is a RICHEDIT control, the control will not
    // send this notification unless you override the CDialog::OnInitDialog()
    // function and call CRichEditCtrl().SetEventMask()
    // with the ENM_CHANGE flag ORed into the mask.

    // TODO: Add your control notification handler code here
}

```