

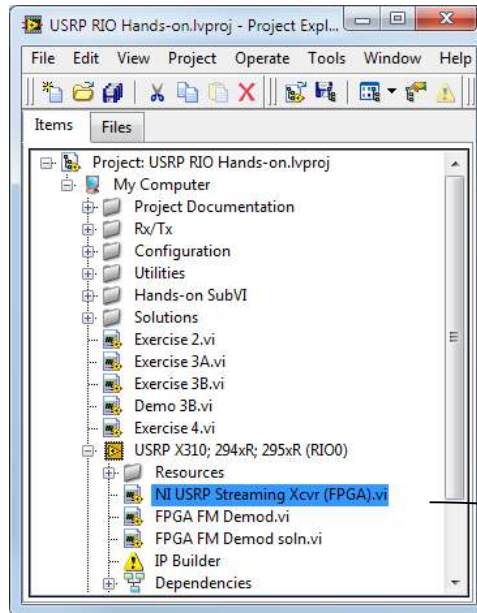
Упражнение 7: Перенос кода на ПЛИС

Примечание: Данное упражнение выполняется на устройстве USRP RIO, например, USRP-2940R. Устройство USRP RIO должно быть включено и подключено к компьютеру перед тем, как вы начнете выполнение упражнения. Не отключайте USRP RIO при включенном компьютере.

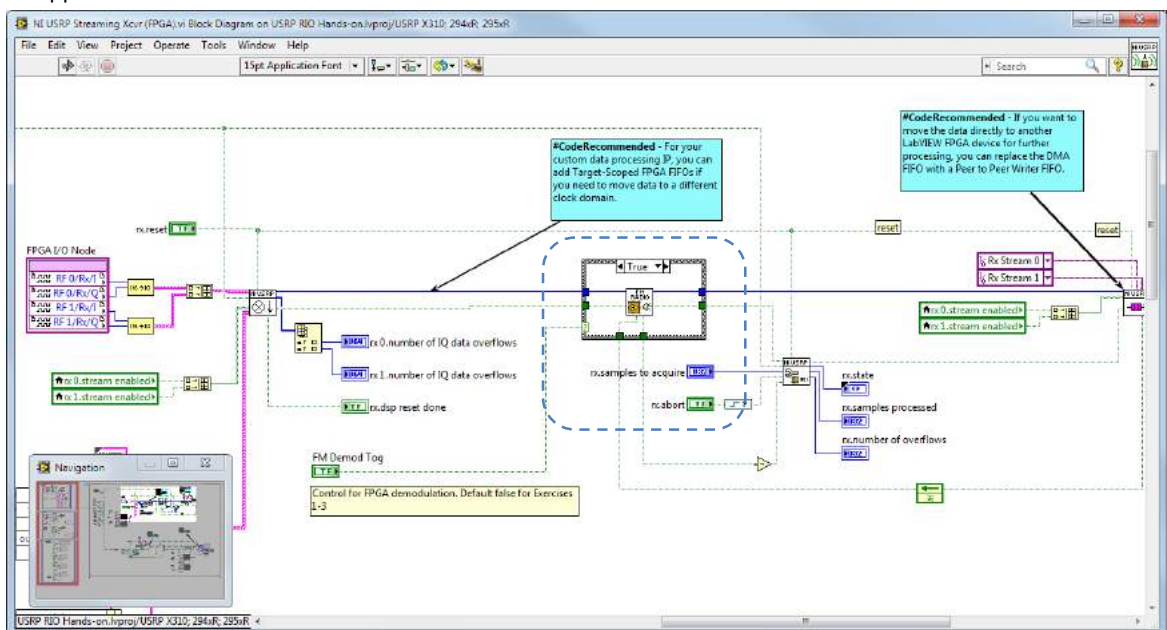
- Цели**
- a) Изучить потоковый проект NI USRP
 - b) Понять принцип потоков данных в ПЛИС и одноктактные циклы
 - c) Перенести демодулятор на ПЛИС

1. Откройте VI для ПЛИС

- a. В браузере проектов LabVIEW разверните целевое устройство **USRP X310; 294xR; 295xR**
- b. Дважды щелкните VI **NI USRP Streaming Xcvr (FPGA)**, чтобы открыть его.

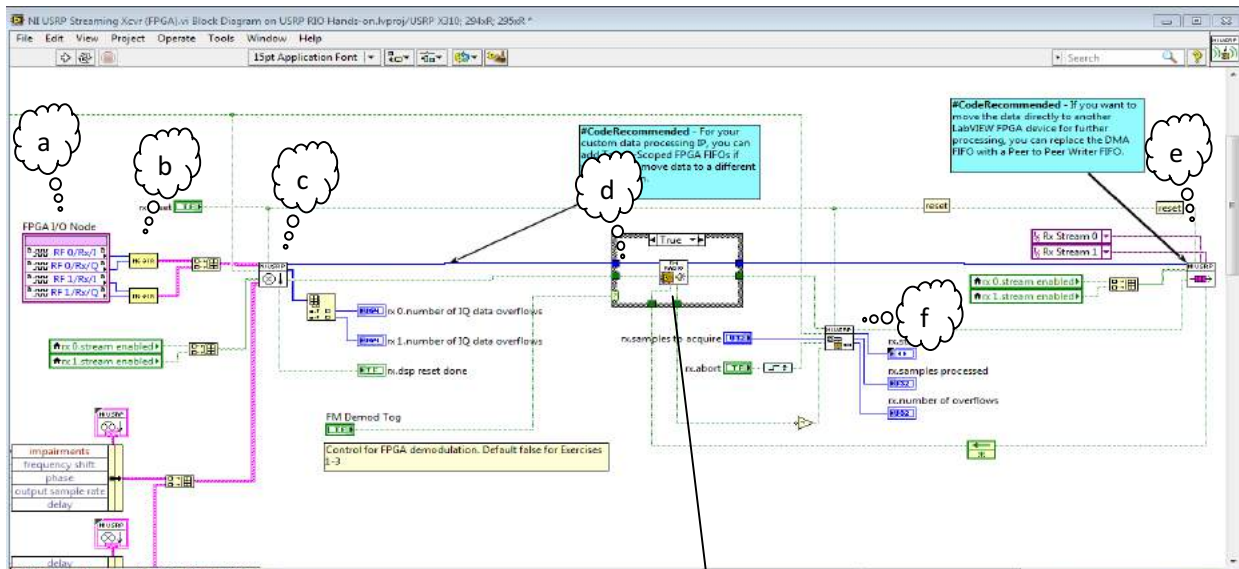


- c. Переключитесь на блок-диаграмму
- d. Прокрутите её до одноктактного цикла потокового приемопередатчика (SCTL)
- e. Найдите subVI **FPGA FM Demod**



2. Анализ потока данных VI верхнего уровня для ПЛИС

- (a) Поток данных от АЦП поступает через узел **FPGA I/O Node**
- (b) Данные конвертируются в узле **I16 to fixed-point IQ** для получения квадратур в фиксированной точке
- (c) VI **DDC (Multi Rate)** цифровым алгоритмом переносит частоту данных.
- (d) Полученные данные демодулируются в VI **FM Demod**
- (e) Демодулированный поток данных передается в хост-программу
- (f) VI **Input Stream Control** мониторит и аккумулирует переполнения, например, если валидные данные не были приняты последующими функциями.

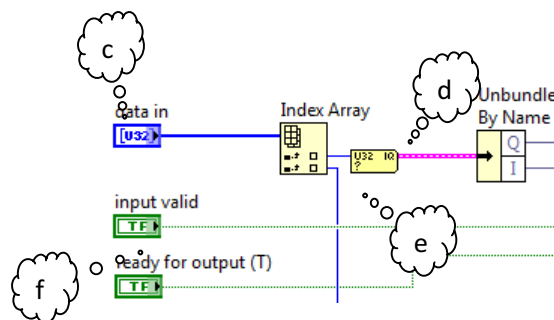


3. Анализ потока данных в subVI FM-демодуляции.

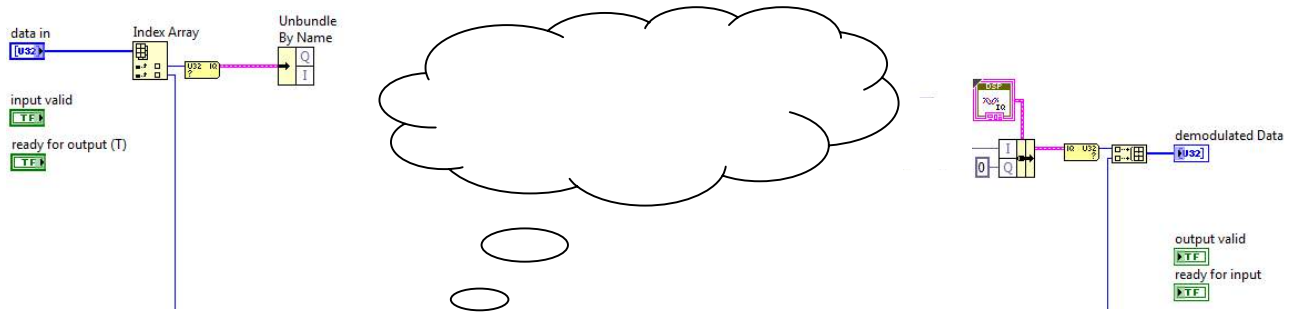
- Дважды щелкните VI **FPGA FM Demod**



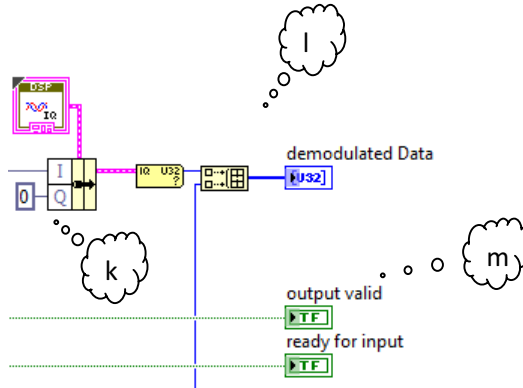
- Переключитесь на блок-диаграмму (Ctrl+E)



- (c) Каналы A и B приходят в subVI как 2 целых 32-битных числа
- Функция **index array** разделяет эти два канала
- (d) Канал A конвертируется в сэмплы с фиксированной точкой и пониженной частотой.
- (e) Канал B проходит без изменений
- Функция **unbundle by name** разделяет сэмплы канала A на компоненты I и Q
- (f) Логическая переменная **input valid** используется для определения валидности текущих данных
- (f) Логическая переменная **ready for output** используется для определения готовности нижеследующих функций принимать данные.



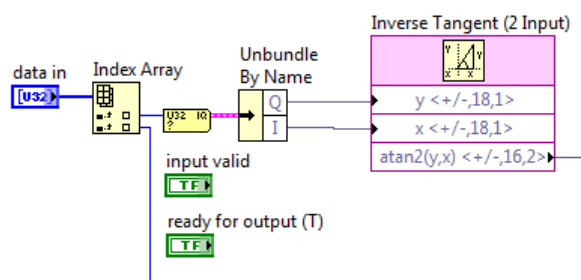
- Вставьте свои функции демодуляции в это пустое место. Для этого выполните следующие шаги.



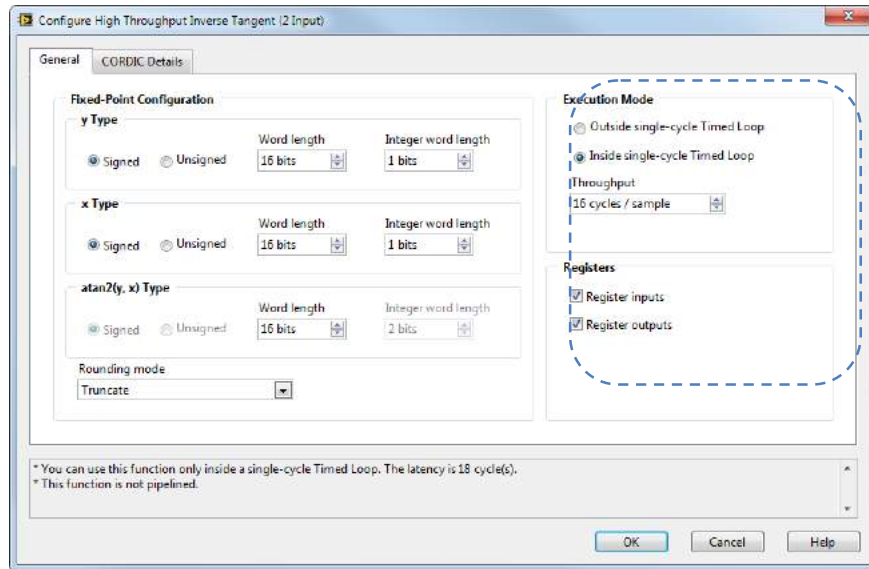
- (k) Поскольку демодулированные данные больше не являются комплексными, вам нужно добавить константу в качестве заполнителя Q дабы не менять дальнейшие функции обработки.
- (l) Каналы A и B комбинируются перед отправкой на хост
- (m) Логическая переменная **output valid** отображает валидность выходных данных
- (m) Логическая переменная **ready for input** отображает, есть ли ещё данные для вывода

4. Вычислите фазу IQ сигнала

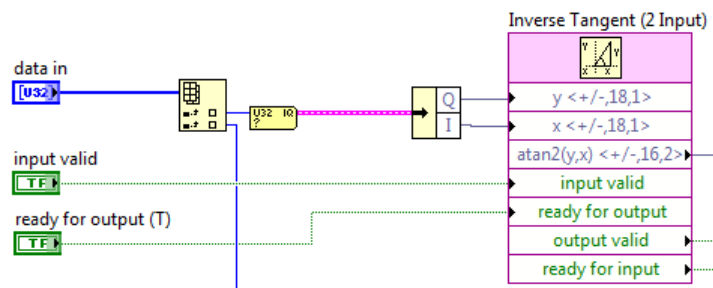
- Через палитру FPGA Math»High Throughput добавьте функцию **Atan2** или **Inverse Tangent (2 Input)**. Она заменит аналогичную функция из программы для Host. Соедините разъемы **Q** и **I** с разъемами **y** и **x** соответственно



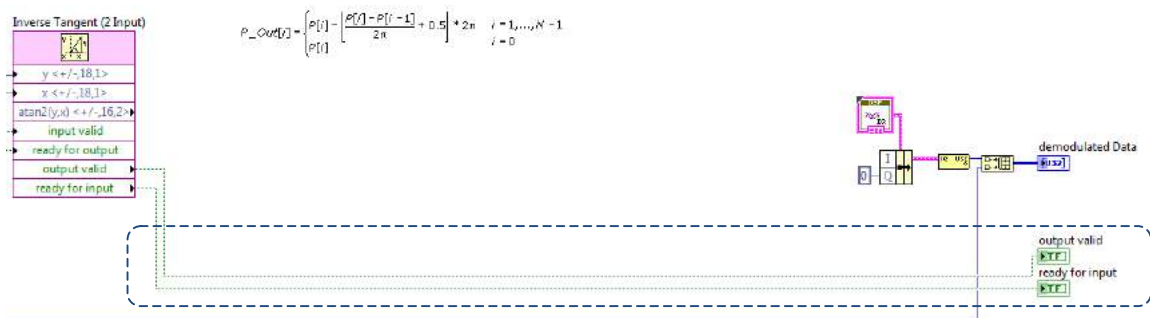
5. Настройте функцию **Inverse Tangent** для однократного цикла.
 - a. Дважды щелкните VI **Inverse Tangent (2 Input)** для настройки:



- b. Нажмите **OK**
- c. Подключите элемент управления **input valid** к терминалу **input valid**
- d. Подключите элемент управления **ready for output (T)** к терминалу **ready for output**

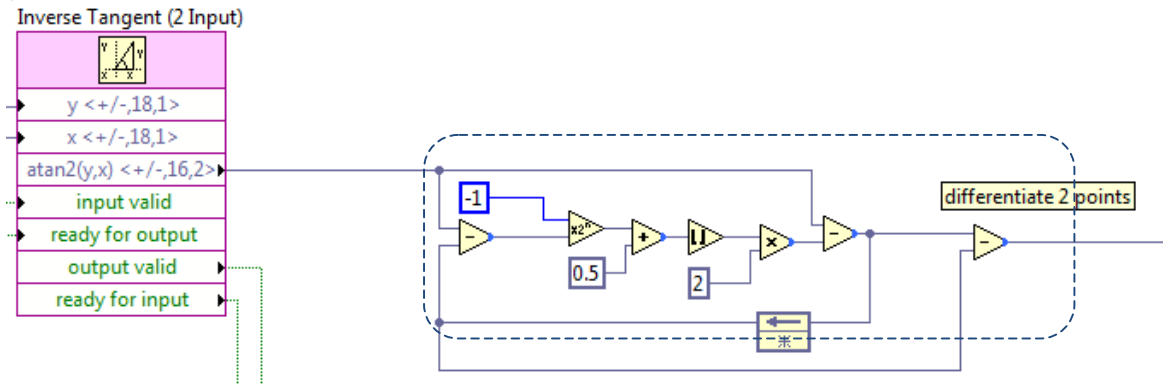


6. Постройте 4-проводной протокол обмена данными.
 - a. Соедините терминал **output valid** VI **inverse tangent** с индикатором **output valid**
 - b. Соедините терминал **ready for output** с индикатором **ready for output**



7. Скопируйте оставшуюся часть алгоритма.

a. Справа от функции **Inverse Tangent**, добавьте код развертки и дифференцирования фазы из упражнения 6

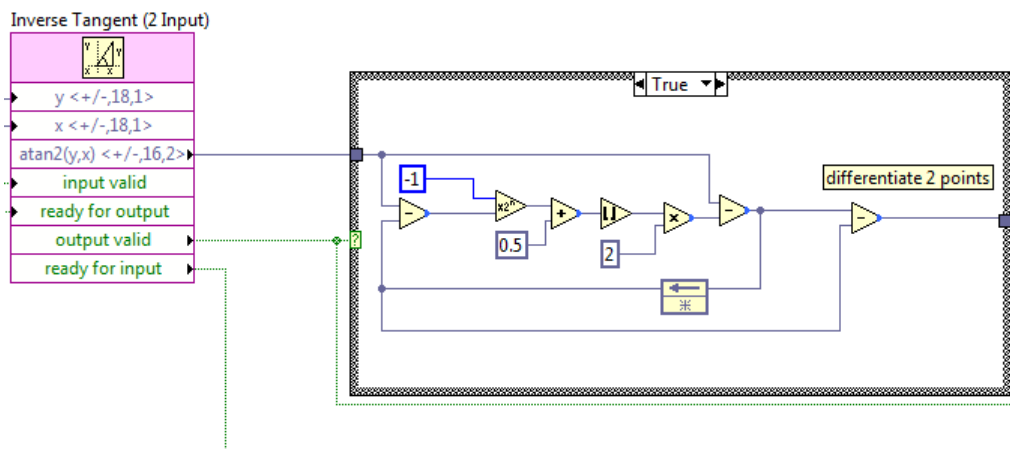


8. Добавьте защиту от обработки некорректных сэмплов.

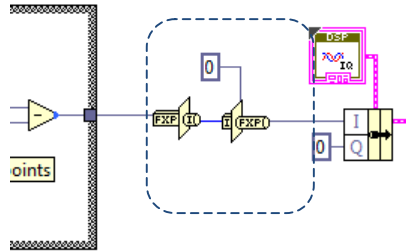
a. Создайте структуру **Case** вокруг кода **unwrap phase** и функции производной.

b. Соедините разъем **output valid** с селектором структуры **Case**

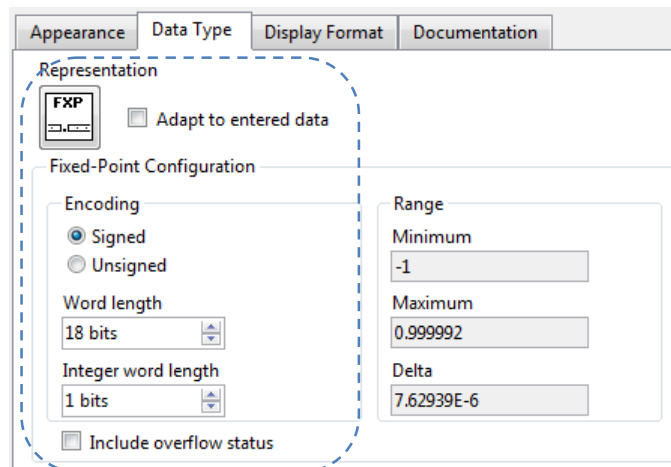
c. Для состояния **False**, создайте выходную константу 0.



9. Тип выходных данных является типом с фиксированной точкой, но он не имеет правильно заданного количества целочисленных и десятичных бит для отправки на хост без преобразования. Добавьте функции **Fixed Point to Integer** и **Integer to Fixed Point** для выполнения преобразования
- Добавьте узел **Fixed-Point to Integer Cast**
 - Добавьте узел **Integer to Fixed-Point Cast**
 - Создайте константу для входа **fixed-point type**

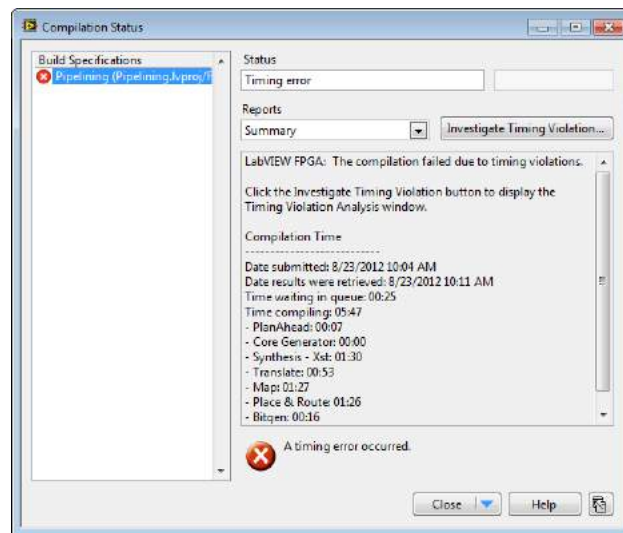


- Щелкните правой кнопкой по нулевой константе и выберите **Properties**
- Откройте закладку **Data Type**
- Сконфигурируйте константы, как указано на рисунке ниже.



Конвейеризация: Необходимая оптимизация ПЛИС

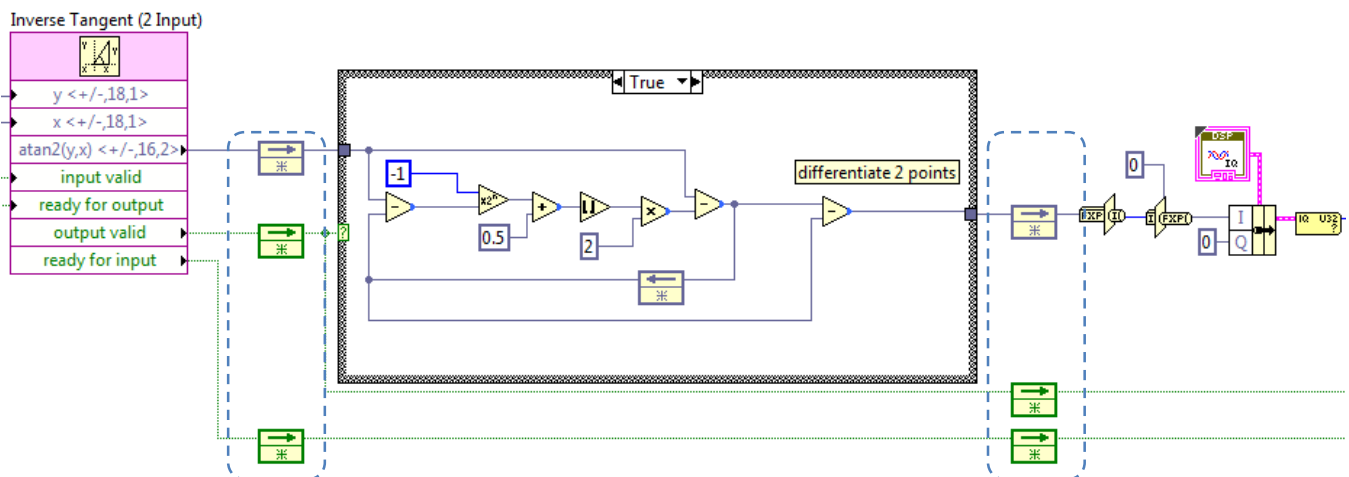
Пока что вы, вероятно, не сможете скомпилировать программу, поскольку получите следующее сообщение об ошибке:



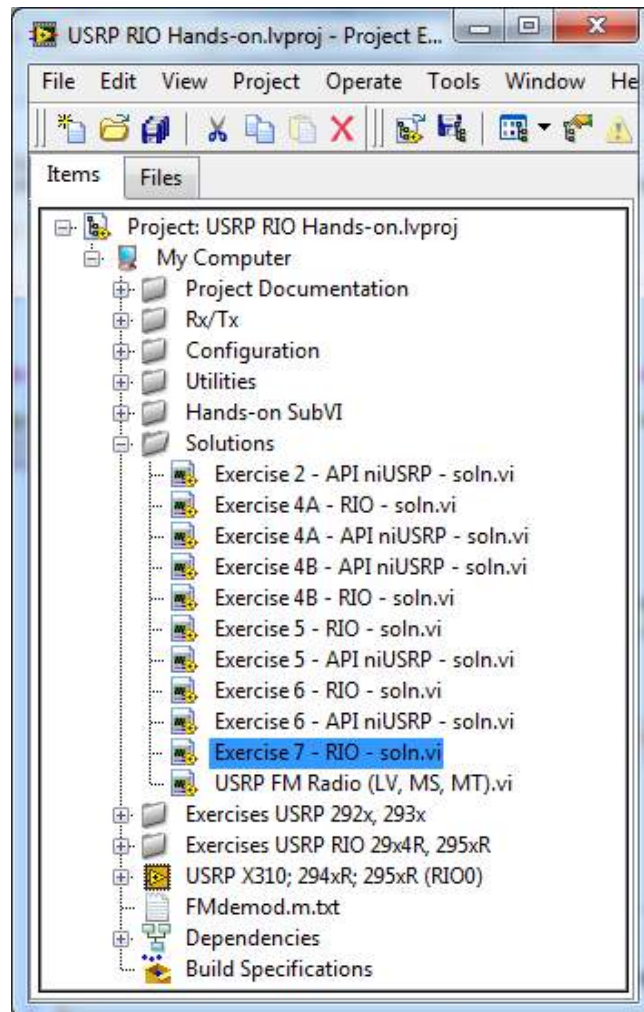
Это упражнение подчеркивает важную особенность программирования под ПЛИС, поскольку ПЛИС имеет строгие требования по времени. Вы должны использовать технику программирования с высокой пропускной способностью, которая называется *Конвейеризация*.

10. Реализуйте Конвейеризацию с помощью узлов обратной связи.

- Добавьте модули обратной связи на места, как на рисунке ниже.
- Щелкните по каждому из них правой кнопкой мыши и выберите **change direction**
- Соедините узлы обратной связи



- В нормальных условиях, вы скомпилируете VI для ПЛИС с помощью Build Specifications, но этот процесс может быть долгим.
- В рамках этого семинара, вы задействуете предварительно скомпилированный бинарный файл, открыв VI с решением для Упражнения 7.



11. Слушайте радио!

- Закройте subVI **FM Demod** и **Streaming Xcvr**
- В браузере проекта откройте VI **Exercise 7**
- Используйте настройки радиостанции из Упражнения 2
- Нажмите Run, чтобы демодулировать FM-радио с помощью ПЛИС!

Пожалуйста, дождитесь инструктора перед продолжением работы

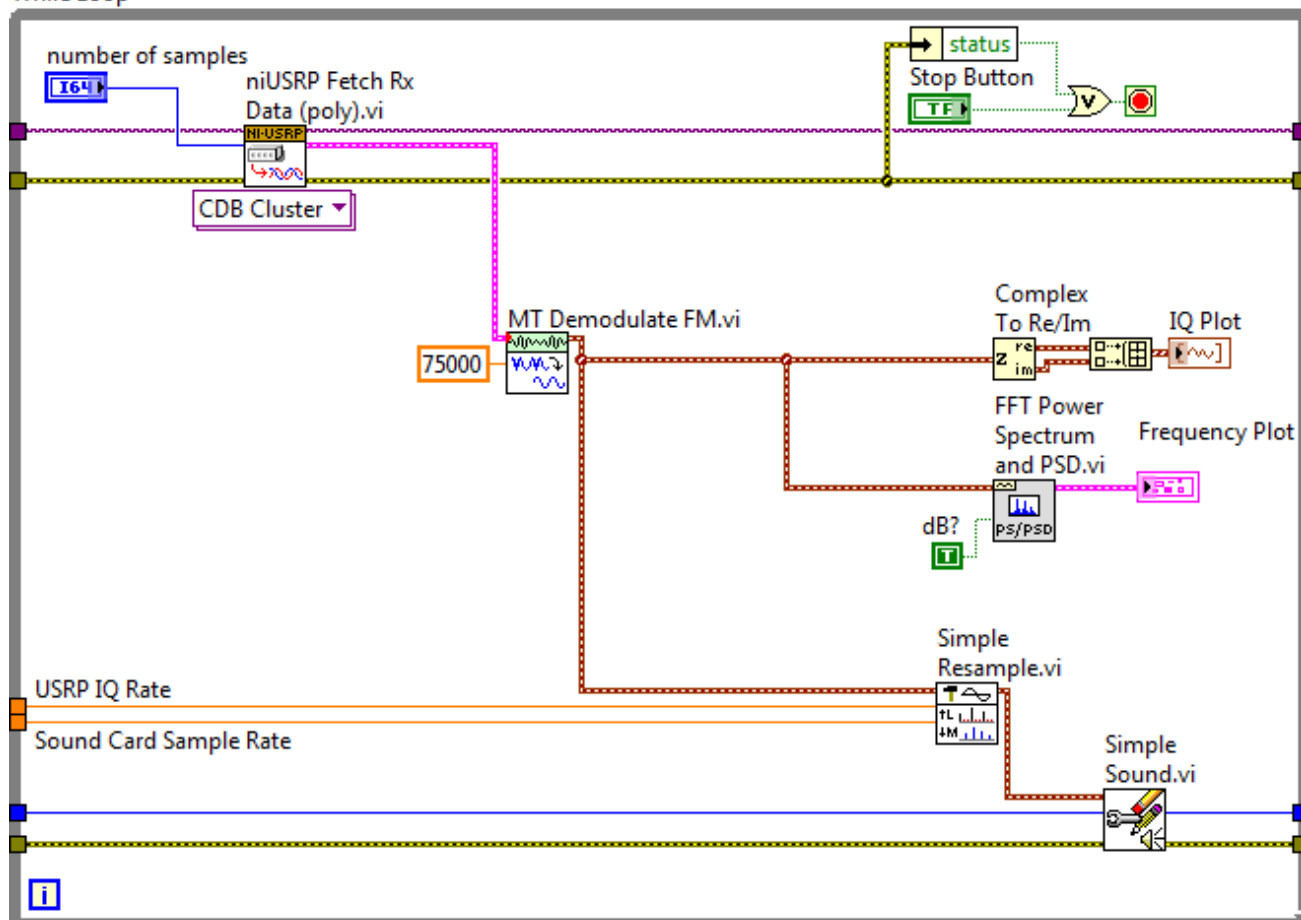
Ответы к упражнениям

Упражнение 1

Упражнение не требует программирования

Упражнение 2

While Loop

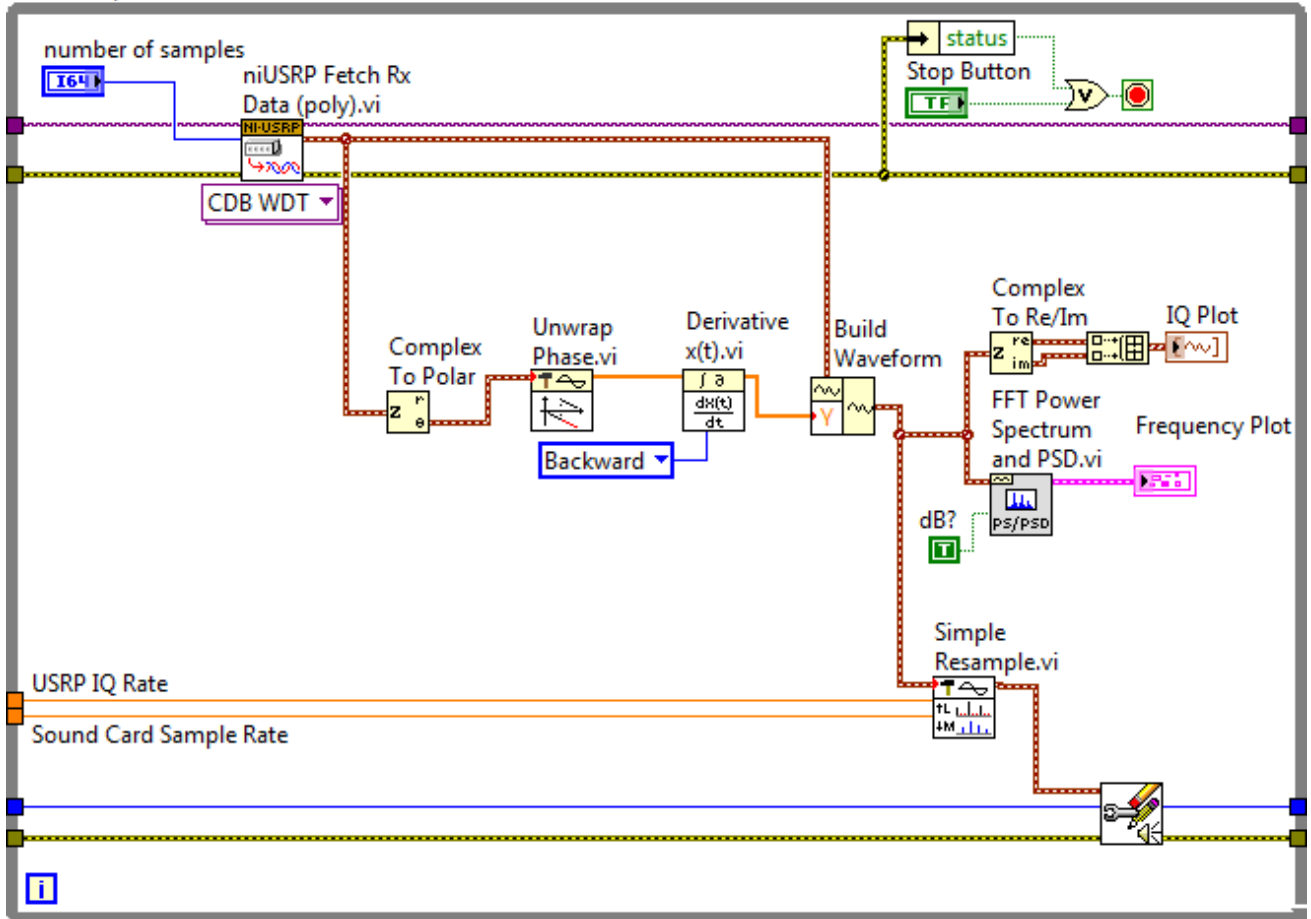


Упражнение 3

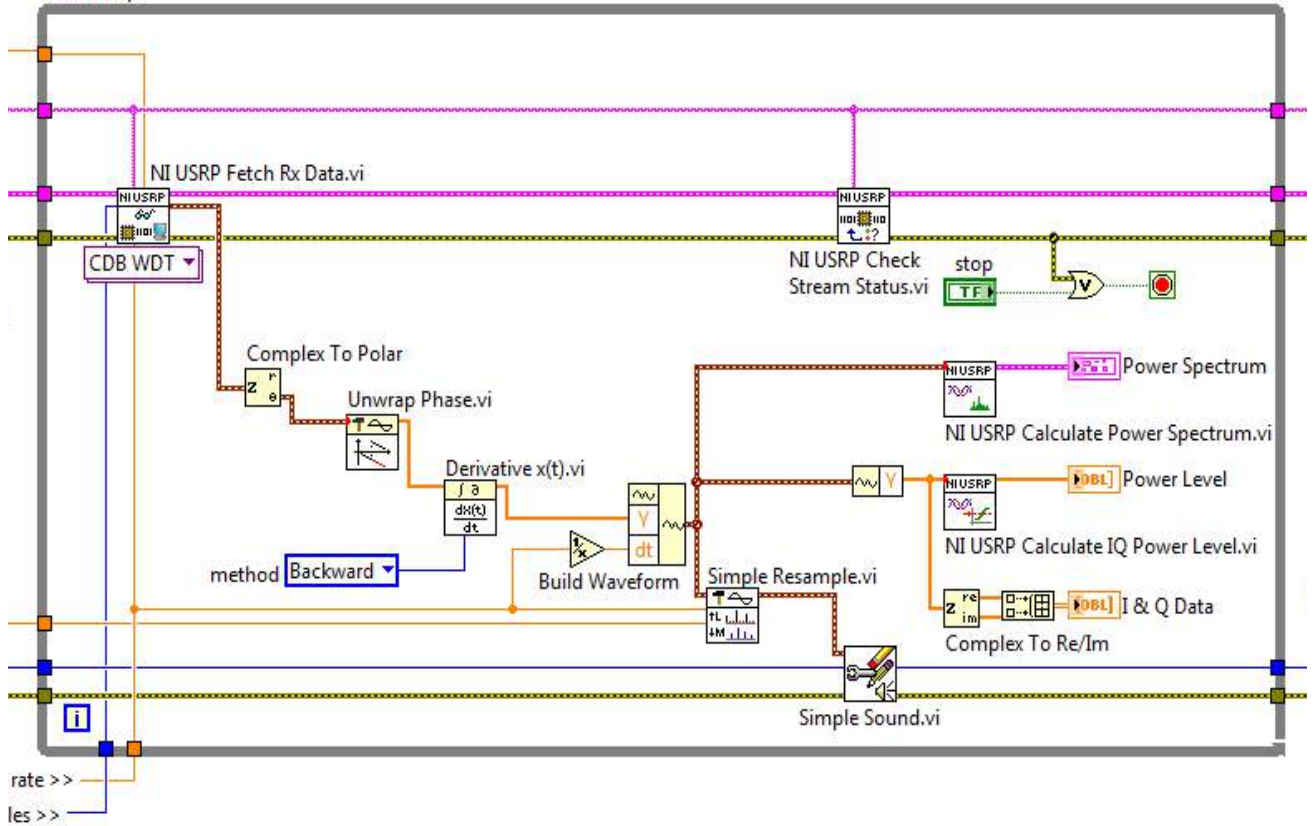
Упражнение не требует программирования

Упражнение 4А

While Loop

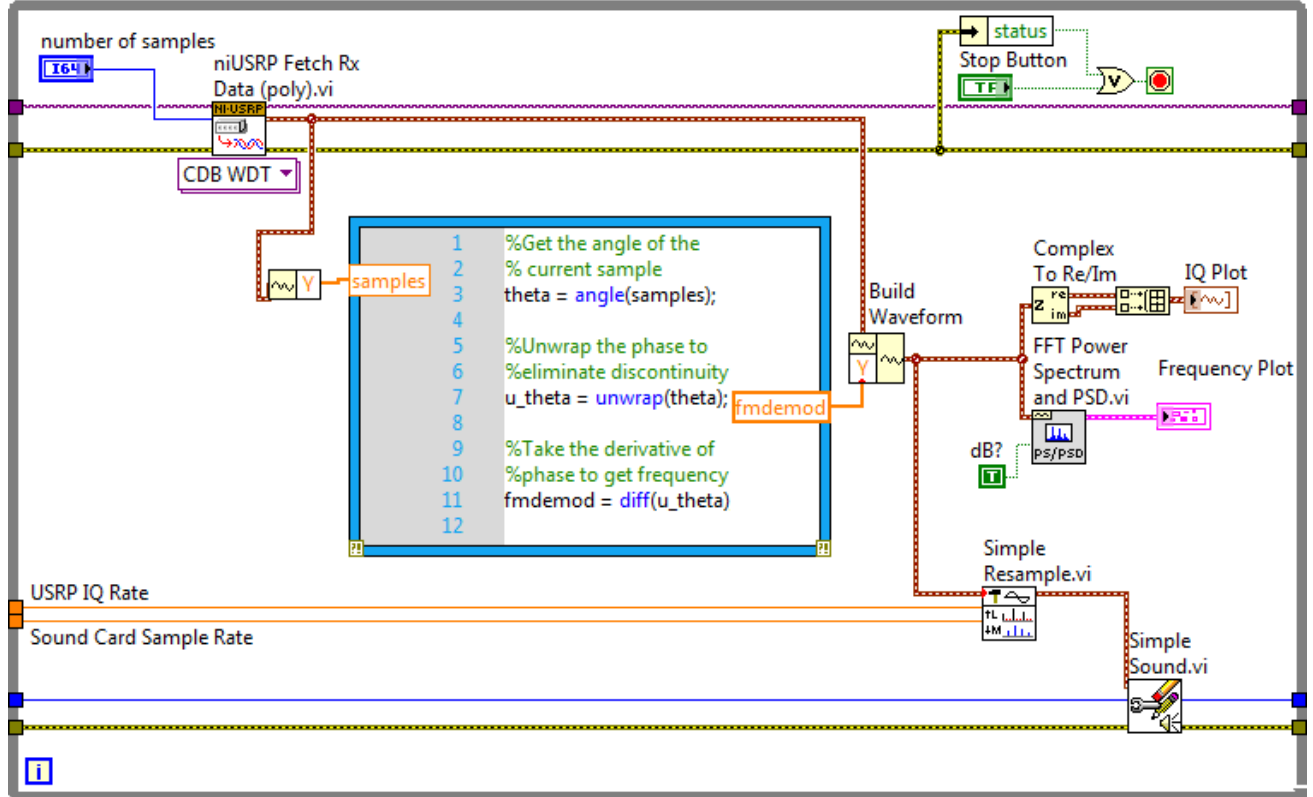


While Loop



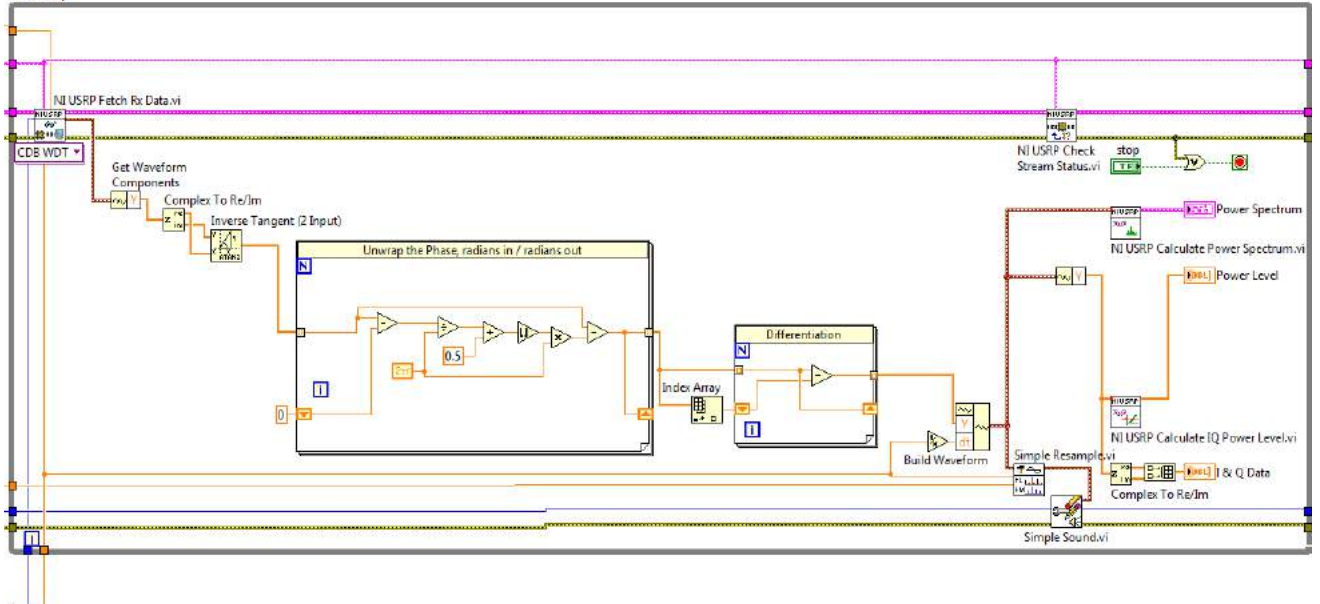
Упражнение 4В

While Loop

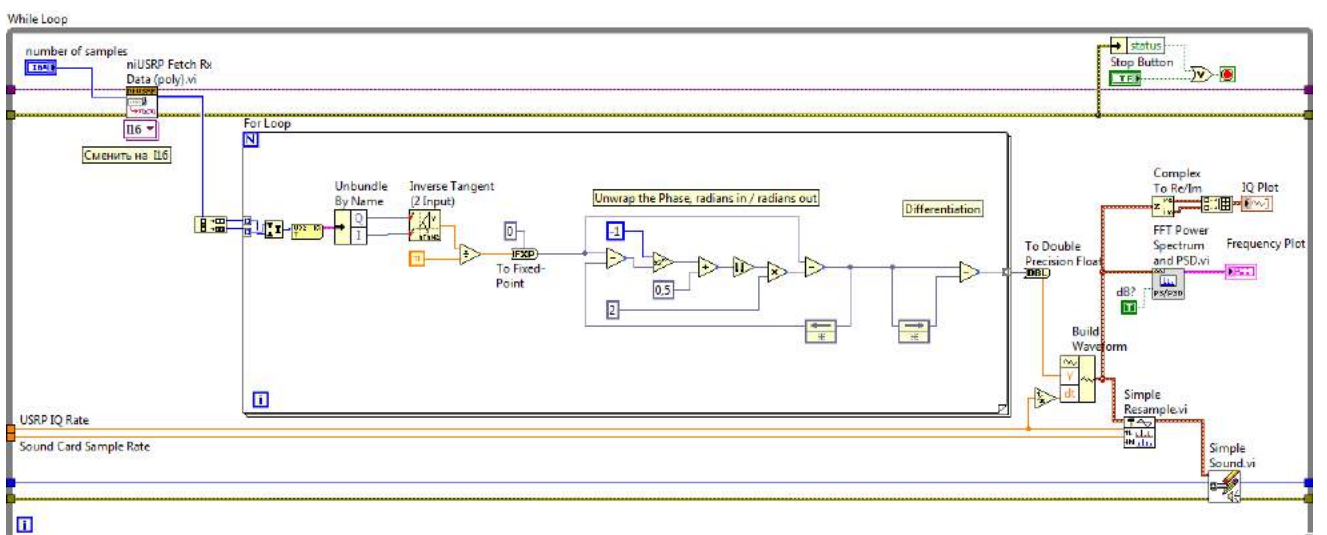
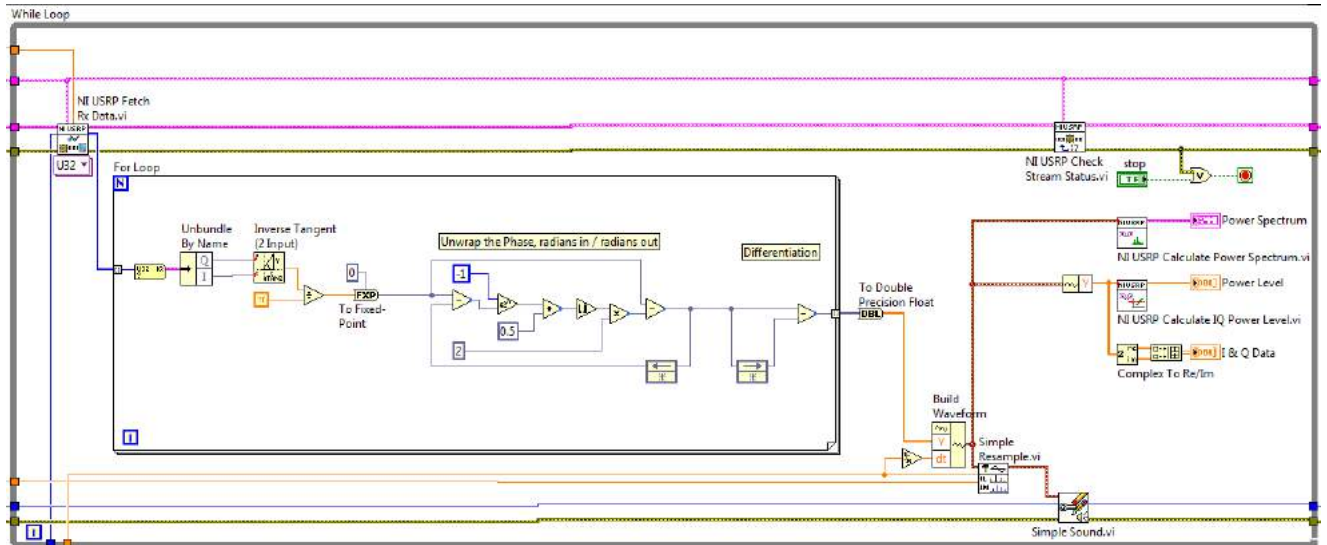


Упражнение 5

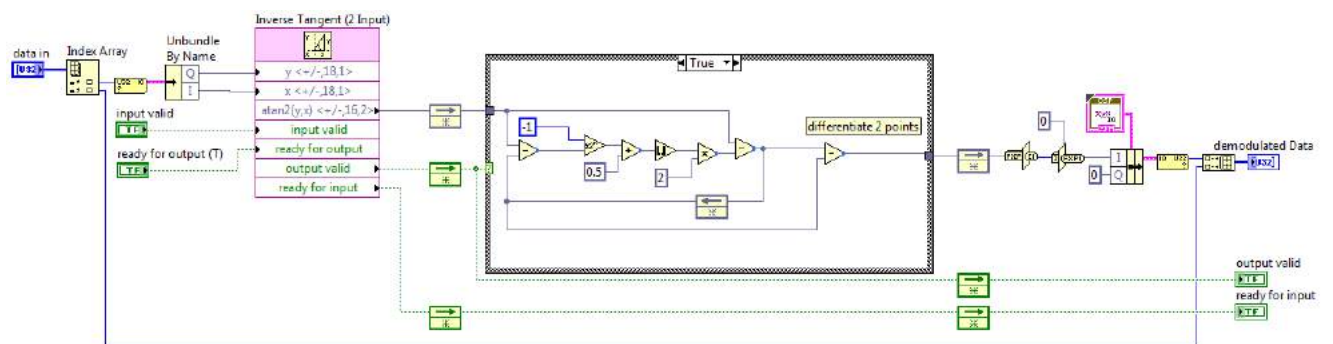
While Loop



Упражнение 6



Упражнение 7



Часто задаваемые вопросы

Как узнать, поддерживает ли ПЛИС нужную мне функцию?

В приведенных упражнениях вы использовали LabVIEW чтобы написать и выполнить код на хост-компьютере или на ПЛИС, установленном в устройстве NI USRP RIO. Тем не менее, ПЛИС может не поддерживать функции высокого уровня, доступные на хост-компьютере. Чтобы проверить, доступна ли функция на ПЛИС, откройте в браузере проектов VI для устройства USRP RIO и попытайтесь найти функцию на палитре.

Также важно помнить, что в программировании ПЛИС часто используются одноплатные циклы в целях оптимизации. Поскольку весь код в цикле должен выполняться в один такт, некоторые функции не могут быть выполнены корректно. Кроме того, при использовании не поддерживаемых функций, на этапе компиляции будет отображено сообщение об ошибке.

Почему представления данных с фиксированной и плавающей точкой работают по-разному?

Понимание различных типов данных и как они используются — важный концепт взаимодействия между хост-компьютером и целевым устройством. В то время, как компьютер может оперировать как данными с фиксированной, так и с плавающей точкой, NI USRP RIO и встроенная ПЛИС не поддерживает тип данных с плавающей точкой и операции с ними. Нужно учитывать эти особенности при проектировании и написании кода.

Почему важно помнить об имеющихся ресурсах ПЛИС?

Преимуществом ПЛИС на устройстве NI USRP RIO является возможность применять перенастраиваемые алгоритмы на детерминированном высокоскоростном устройстве. Поскольку ПЛИС разделена на логические блоки и имеет временные ограничения к циклу, нужно быть внимательным к функциональности, потоку данных и задержкам вашего кода. Поэтому мы придерживались последовательности программирования, которую можно будет использовать в дальнейшем.

Дополнительные ресурсы

Обучающее видео для LabVIEW:

<http://www.ni.com/academic/students/learn>

Основы ПЛИС:

<http://www.ni.com/white-paper/6983/en>

Типы данных с фиксированной точкой, часть 1:

<http://www.ni.com/newsletter/50303/en>

Типы данных с фиксированной точкой, часть 1:

<http://www.ni.com/newsletter/50363/en>

Конвейеризация для оптимизации ПЛИС:

http://zone.ni.com/reference/en-XX/help/371599J-01/lvfpgaconcepts/fpga_pipelining

Обучение и сертификация National Instruments

<http://russia.ni.com/training>

Обучение и сертификация NI

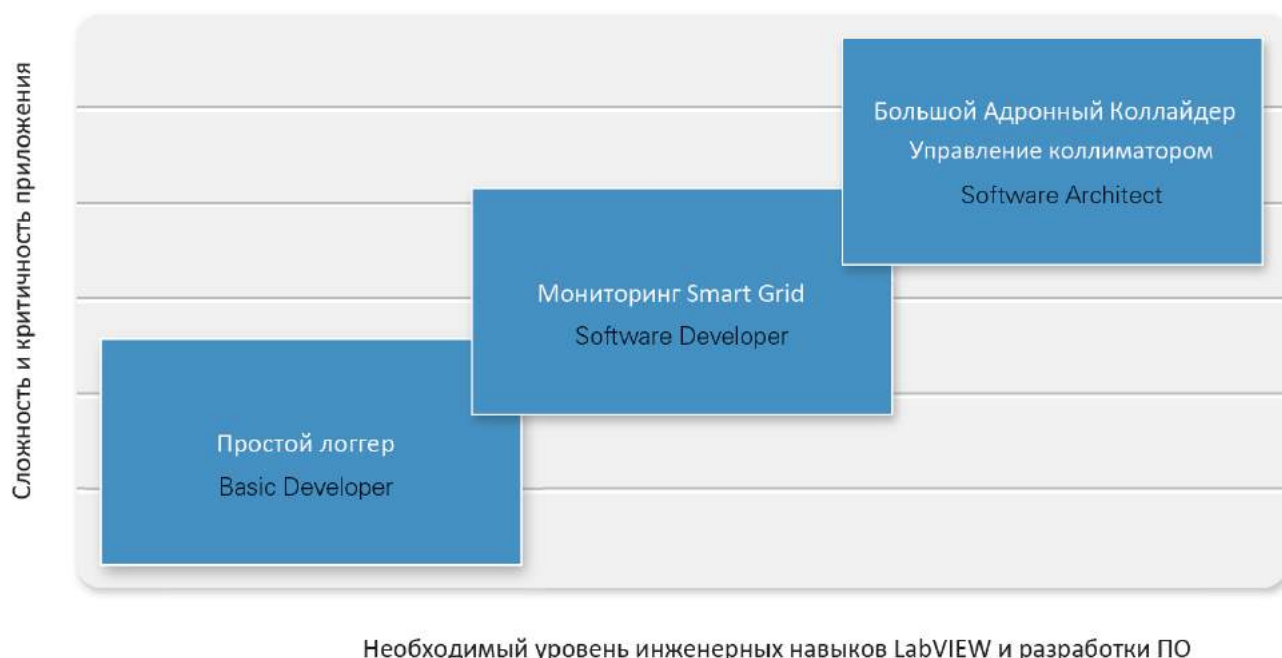
Ваш успех зависит от опыта

Успешно создать приложение с помощью системы графического программирования это не только сгенерировать программу, отвечающую заданным требованиям. Успешный проект также минимизирует время разработки и снижает стоимость обслуживания, в то же время максимально используя преимущества продуктов National Instruments, включая увеличение производительности с помощью NI LabVIEW и доступ к продвинутым технологиям с помощью оборудования NI. Опыт в использовании продуктов National Instruments делает такой успех наиболее возможным.

Форматы обучения

Чтобы лучше соответствовать вашим запросам, NI предлагает обучающие курсы в различных форматах и на разных языках, в том числе: очные курсы и тренинги в наших офисах или на вашем предприятии, онлайн курсы и виртуальные тренинги. Какой бы формат вы ни выбрали, обучение от NI поможет вам достичь повышения продуктивности.

Посетите страничку ni.com/training/options, чтобы получить более подробную информацию.



Курсы

National Instruments проводит много различных курсов, помогающих вам наиболее эффективно использовать ваше оборудование NI.

Development Courses	
NI LabVIEW Development Courses <ul style="list-style-type: none">▪ LabVIEW Core 1▪ LabVIEW Core 2▪ LabVIEW Core 3▪ LabVIEW Connectivity▪ LabVIEW Performance▪ Object-Oriented Design and Programming in LabVIEW▪ Advanced Architectures in LabVIEW▪ Managing Software Engineering in LabVIEW▪ Using NI LabVIEW for Test and Automation in Regulated Markets	LabVIEW Development With NI Hardware Courses <ul style="list-style-type: none">▪ LabVIEW Real-Time 1▪ LabVIEW Real-Time 2▪ LabVIEW FPGA▪ High-Throughput LabVIEW FPGA▪ NI FlexRIO▪ Data Acquisition and Signal Conditioning▪ LabVIEW Instrument Control▪ Sound and Vibration Fundamentals▪ Modular Instruments: Switches▪ Modular Instruments: Digital Multimeters (DMMs)▪ Modular Instruments: High-Speed Digital I/O▪ RF Measurement Fundamentals▪ RF Application Development▪ LabVIEW Machine Vision and Image Processing
NI TestStand Courses <ul style="list-style-type: none">▪ TestStand 1: Test Development▪ TestStand 2: Framework Development	
NI LabWindows™/CVI Courses <ul style="list-style-type: none">▪ LabWindows/CVI Core 1▪ LabWindows/CVI Core 2	NI Multisim and Ultiboard Courses <ul style="list-style-type: none">▪ Multisim Basics▪ Ultiboard Basics
NI VeriStand Course <ul style="list-style-type: none">▪ NI VeriStand Fundamentals	NI DIAdem Courses <ul style="list-style-type: none">▪ DIAdem Basics▪ DIAdem Advanced

Подготовка к вашей сертификации NI

National Instruments предоставляет множество материалов, которые помогут вам подготовиться к вашему сертификационному экзамену. Посетите страничку ni.com/training/certification_prep, где вы можете найти руководства по подготовке, примеры экзаменационных задач и многое другое.

Зарегистрируйтесь на экзамен недалеко от вас

В Америке и Европе National Instruments предлагает сдачу экзаменов через сертификационные центры Pearson VUE. В России, странах СНГ и Балтии сдать экзамен можно в офисе National Instruments, в авторизованных учебных центрах NI, а также на тематических мероприятиях National Instruments. Посетите страничку ni.com/certificationschedule и russia.ni.com/training/certification, чтобы получить информацию о расписании ближайших экзаменов, или свяжитесь с представителем или офисом National Instruments в вашем регионе.