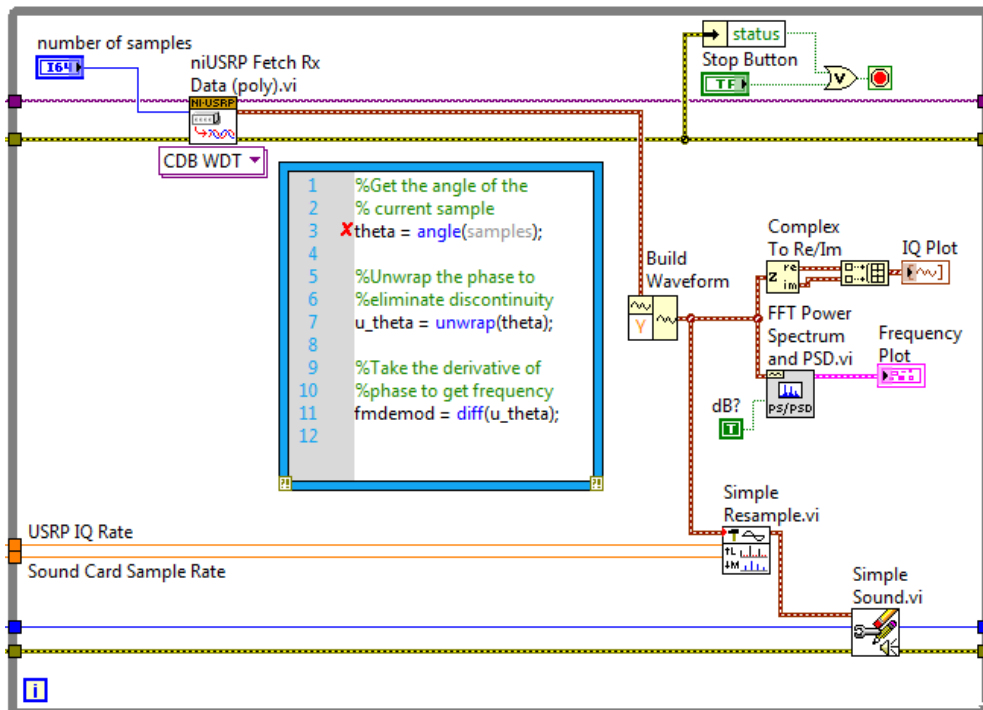


Упражнение 4В (дополнительное): Использование алгоритмов на текстовых языках

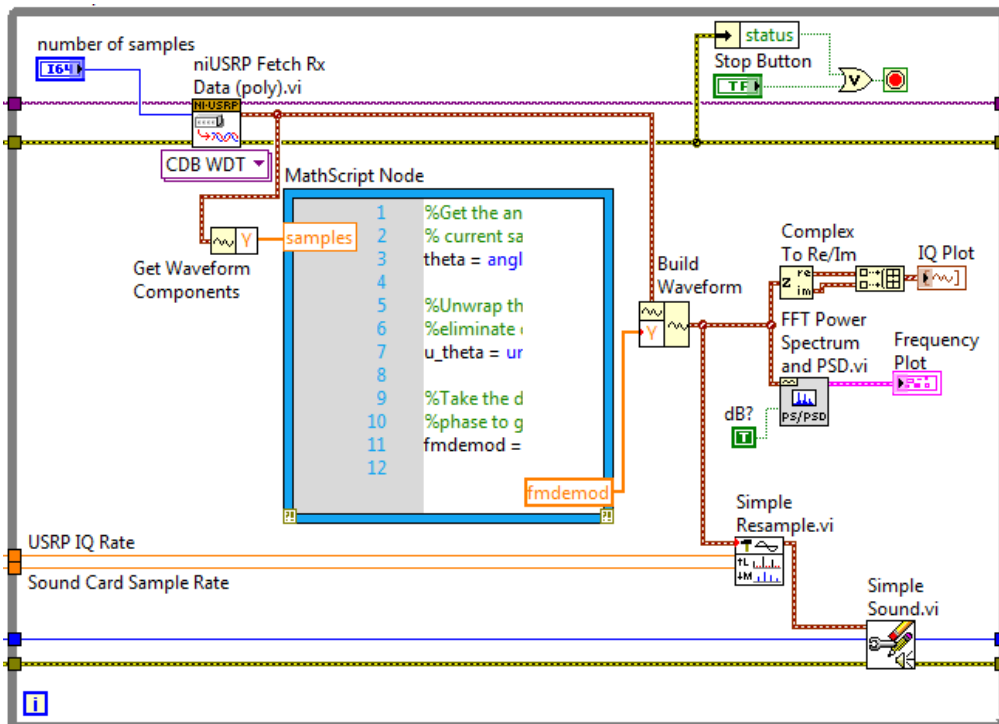
Примечание: Устройство USRP RIO должно быть включено и подключено к компьютеру перед тем, как вы начнете выполнение упражнения. Не отключайте USRP RIO при включенном компьютере.

- Цели**
- Познакомиться с пакетом LabVIEW MathScript RT Module
 - Демодулировать радио с помощью текстовой математики

- Откройте упражнение **Exercise 4B** из папки **Exercises USRP 292x, 293x** или из папки **Exercises USRP RIO 294xR, 295xR**
- Перейдите на блок-диаграмму (Ctrl + E)
- Поместите узел обработки текстовой математики MathScript Node из палитры функций **Programming > Structures**.
- Вставьте m-код одним из способов ниже
 - Кликните правой кнопкой по границе узла и выберите Import. Выберите файл FMdemod.m из папки упражнений
 - Скопируйте код из файла FMdemod.m, подшитого в проект, в тело узла.



- Добавьте входы и выходы
 - Кликните правой кнопкой на левой границе узла и выберите Add Input. Наберите имя ввода **samples**.
 - Кликните правой кнопкой на правой границе узла и выберите Add Output. Выберите **fmdemod**.
- Соедините сигналы
 - Из палитры функций **Waveform** выберите функцию **Get Waveform Components** чтобы выделить массив данных из сигнала.
 - Подайте на нее сигнал от **niUSRP Fetch Rx Data**, а сигнал с ее выхода **Y** подключите к входу узла MathScript Node.
 - Подключите выход узла MathScript Node к входу **Y** функции **Build Waveform**.



7. Сохраните программу.
8. Перейдите на лицевую панель.
9. Настройте параметры приема и запустите программу.

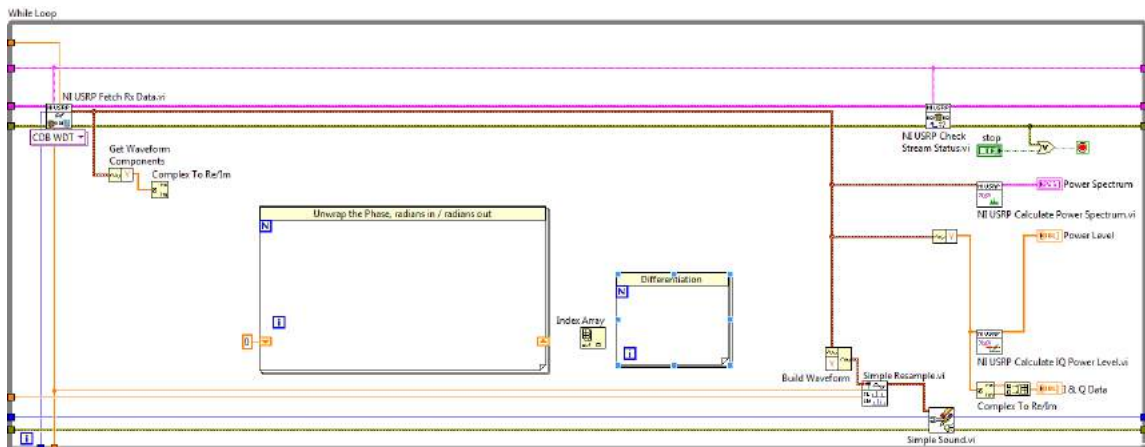
Пожалуйста, дождитесь инструктора перед продолжением работы

Упражнение 5: Прототипирование с плавающей точкой

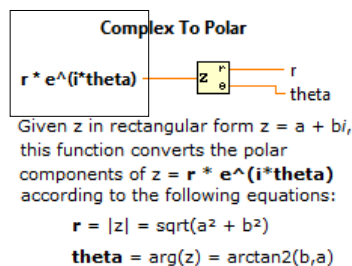
Примечание: Устройство USRP RIO должно быть включено и подключено к компьютеру перед тем, как вы начнете выполнение упражнения. Не отключайте USRP RIO при включенном компьютере.

- Цели**
- Создать имитацию FM-демодулятора с плавающей точкой
 - Убедиться, что симулятор работает корректно

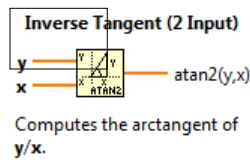
- Закройте все открытые окна передних панелей и блок-диаграмм.
- Откройте упражнение **Exercise 4B** из папки **Exercises USRP 292x, 293x** или из папки **Exercises USRP RIO 294xR, 295xR**
- Перейдите на блок-диаграмму (Ctrl + E)
- Изучите блок-диаграмму
 - Для демодуляции используется три основные функции:
 - Вычисление фазы
 - Развёртка фазы
 - Дифференцирование фазы



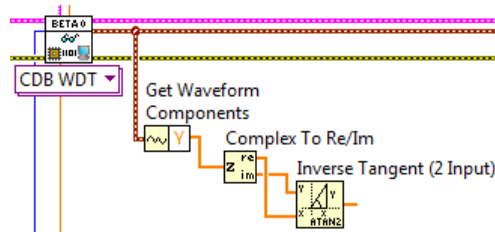
- ПЛИС не может обеспечить все функции обработки высокого уровня, доступные на хосте, поэтому некоторые функции должны быть переведены с хоста на ПЛИС. Рассмотрим функции, требующие разбора.
- Переведите и оптимизируйте функцию Complex to Polar
 - Функция используется для получения фазы входящего сигнала.
 - Откройте справку (**detailed help**) для функции, чтобы увидеть её формулу.



- Необходимое вам выходное значение — это theta, использующее функцию арктангенса.
- Замените функцию на VI Inverse Tangent (с 2 входами), доступную для ПЛИС. (Mathematics > Elementary ... > Trigonometric Functions)

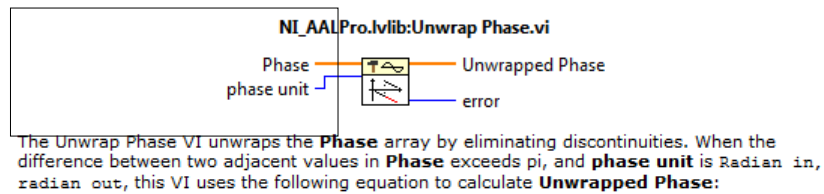


- b. Поместите функцию **Inverse Tangent (2 Input)** внизу справа от **Complex To Re/Im**
- c. Подключите разъемы **real** и **imaginary** к разъемам **x** и **y**.



6. Преобразуйте VI **Unwrap Phase**

- Поскольку функция арктангенса с двумя входами не является непрерывной и её область значений лежит в пределах $[-\pi, +\pi]$, вам нужно *развернуть* выход, чтобы исключить разрывы. Откройте справку для функции Unwrap Phase, чтобы увидеть формулу



$$P_Out[i] = \begin{cases} P[i] - \left\lfloor \frac{P[i] - P_Out[i-1]}{2\pi} + 0.5 \right\rfloor * 2\pi & i = 1, \dots, N-1 \\ P[i] & i = 0 \end{cases}$$

where P_out is the **Unwrapped Phase**, P is the **Phase**, N is the length of **Phase**, and $\lfloor \cdot \rfloor$ is the floor operation.

- Данная функция недоступна на ПЛИС, поэтому вы должны сами создать формулу, используемую в этом VI

 - a. Соедините выход функции **Inverse Tangent (2 Input)** с циклом For
 - b. Создайте код с помощью следующих элементов:

Subtract

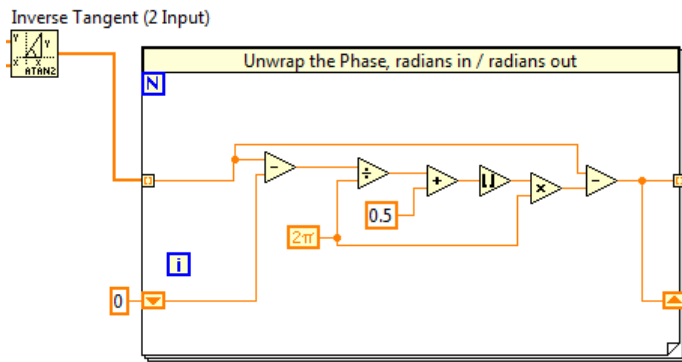
Divide

Pi Multiplied By 2

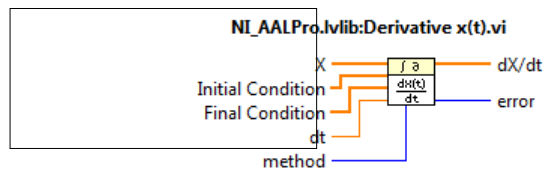
Add

Round Toward -Infinity

Multiply



7. Последняя функция в FM-демодуляторе — это дифференцирование, VI **Derivative x(t)** VI.
- a. Откройте справку, и изучите формулы, которые использует этот VI.



If **method** is Backward, Y is given by the following equation:

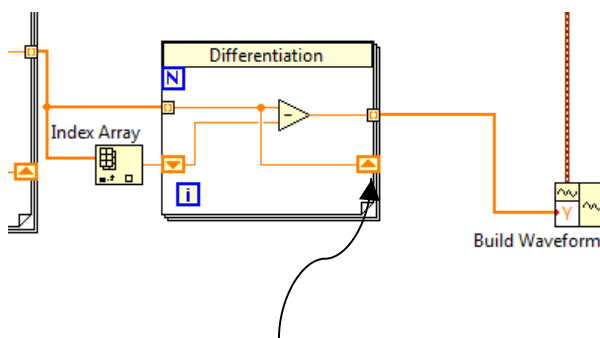
$$Y_i = \frac{1}{dt}(x_i - x_{i-1})$$

for $i = 0, 1, 2, \dots, n - 1$,

where n is the number of samples in $\mathbf{x(t)}$ and x_{-1} is the first element in **Initial Condition**.

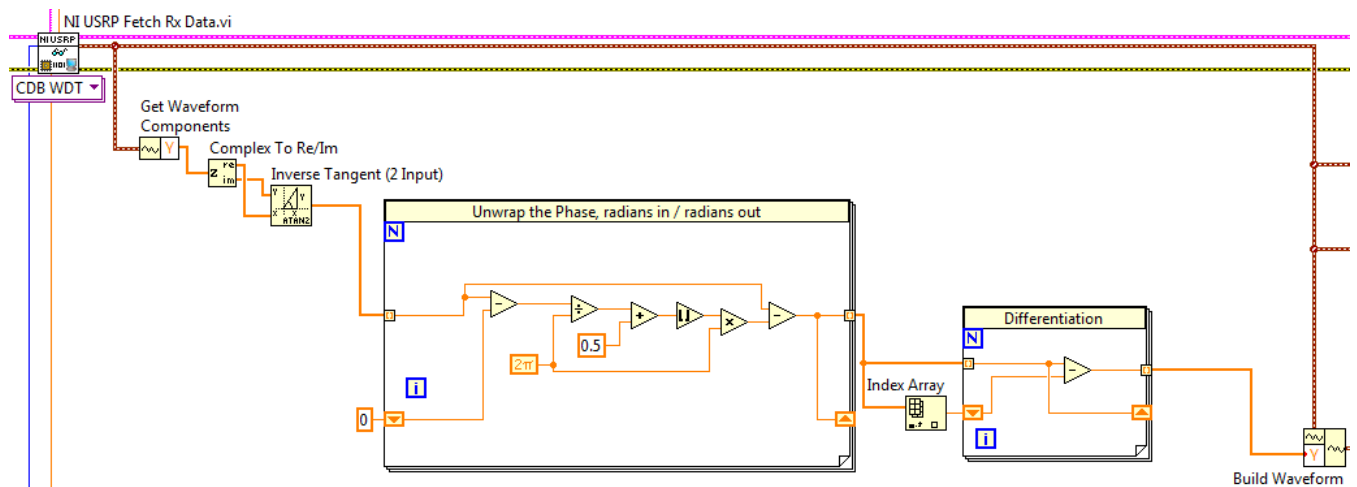
The **Initial Condition** and **Final Condition** minimize the error at the boundaries.

- Если пренебречь константой $\frac{1}{dt}$, данная функция становится функцией одного вычитания.
- b. Поместите функцию вычитания в цикл For под именем Differentiation.
- c. Соедините выход-массив цикла **Unwrap Phase** к функции **Index Array** и функции вычитания в цикле.
- d. Соедините выход функции вычитания с функцией **Build Waveform**.
- e. Переключите нижний выходной терминал в режим **shift register** (если нужно).



Щелчок правой кнопкой мыши -> Change to Shift Register

8. Законченный алгоритм должен выглядеть так:



9. Убедитесь, что модель с плавающей точкой корректно проводит демодуляцию FM-радио.
 - a. Переключитесь на переднюю панель (Ctrl+E)
 - b. Задайте настройки USRP RIO, как в упражнении 2
 - c. Запустите VI

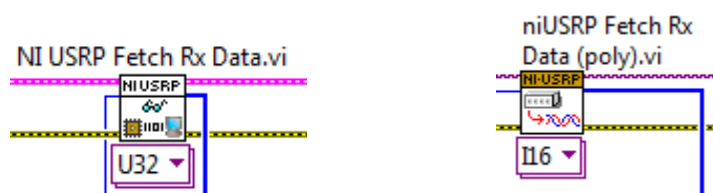
Пожалуйста, дождитесь инструктора перед продолжением работы

Упражнение 6: Преобразование к фиксированной точке

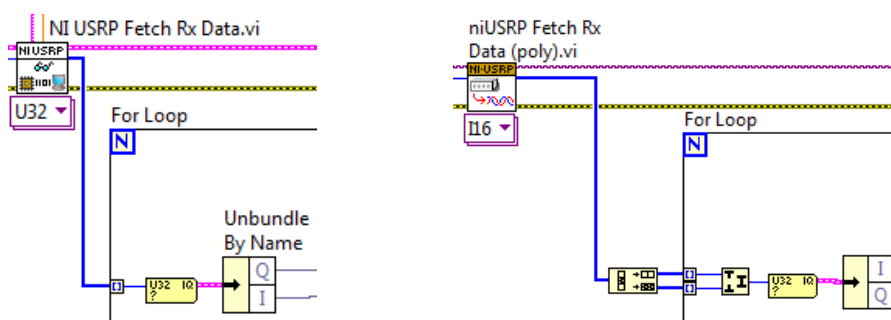
Примечание: Устройство USRP RIO должно быть включено и подключено к компьютеру перед тем, как вы начнете выполнение упражнения. Не отключайте USRP RIO при включенном компьютере.

- Цели**
- a) Смоделировать хост-симуляцию максимально близко к коду на ПЛИС
 - b) Понять представление переменных с фиксированной точкой.

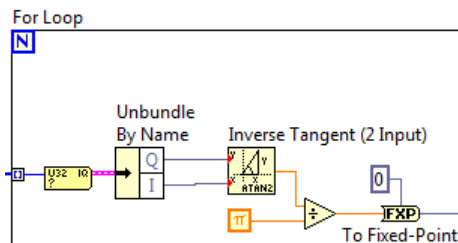
1. Откройте упражнение **Exercise 4B** из папки **Exercises USRP 292x, 293x** или из папки **Exercises USRP RIO 294xR, 295xR**
2. Перейдите на блок-диаграмму (Ctrl + E)
3. Сымитируйте VI поточечного считывания данных на ПЛИС с помощью полиморфного VI NI USRP.
 - a. Для упражнения из папки **Exercises USRP RIO 294xR, 295xR** настройте VI **NI USRP Fetch Rx Data** для выдачи данных в формате **U32** – 32-разрядное целое число без знака.
Для упражнения из папки **Exercises USRP RIO 292x, 293x** настройте VI **NI USRP Fetch Rx Data** для выдачи данных в формате **I16** – 16-разрядные целые числа без знака поочередно.



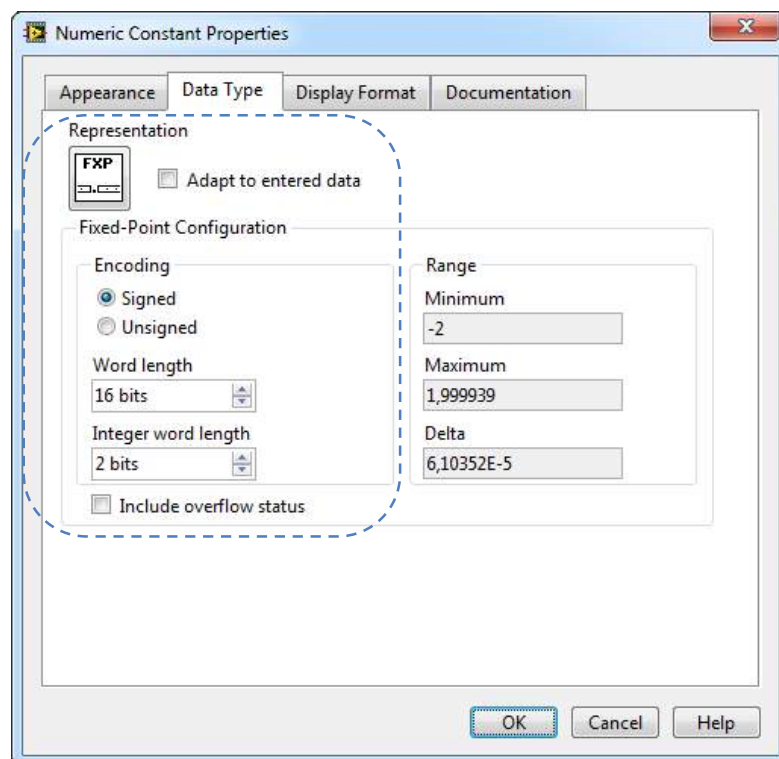
- b. Обратите внимание на цикл **For** для имитации работы с каждой отдельной точкой данных.
- c. Подключите VI **NI USRP Fetch Rx Data** к циклу с включенной автоиндексацией.
- d. Подключите subVI **U32 to IQ** внутри цикла For. Преобразование типа сделано так, чтобы максимально приблизить к коду на ПЛИС, где данные поступают в виде U32.
- e. Поместите функцию **Unbundle By Name** для извлечения компонент I и Q.



4. Сымитируйте VI **Inverse Tangent** для ПЛИС
 - a. Разместите хост-VI **Inverse Tangent (с 2 входами)**
 - b. Подключите **Q** к **y** и **I** к **x**
 - c. Разделите выход функции **Inverse Tangent** на π
 - d. Разместите функцию преобразования **To Fixed Point**
 - e. Создайте константу **fixed-point type**

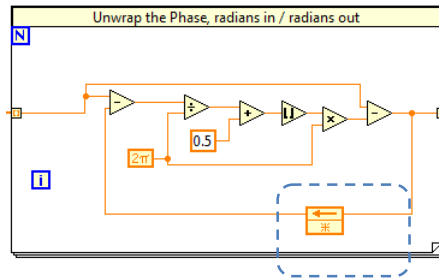


- f. Щелкните по константе правой кнопкой мыши и выберите **Properties**
- g. Откройте закладку **Data Type**
- h. Нажмите **Representation** и выберите **FXP**
- i. Если нужно, снимите флажок **Adapt to entered data** (адаптировать к введенным данным)
- j. Установите значение **16** для **Word length** и **2** для **Integer word length**, что соответствует формату данных этой функции на ПЛИС.

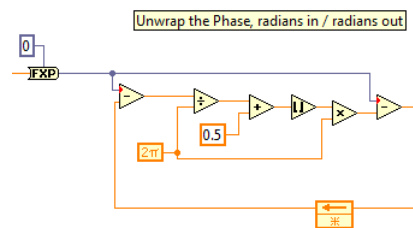


- k. Нажмите **OK**

5. Подготовьте алгоритм развёртки
 - a. Скопируйте алгоритм из Упражнения 5 в цикл **for**
 - b. Замените **shift register** на **feedback node**

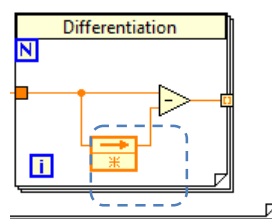


- c. Удалите цикл For
- d. Соедините выход функции **To Fixed-Point** с алгоритмом развёртки

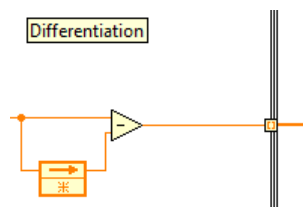


- e. Измените значение константы 2π на 2

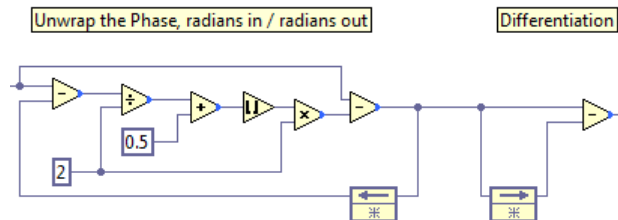
6. Подготовьте алгоритм дифференцирования
 - a. Скопируйте аналогичный алгоритм из упражнения 5 в цикл **For**.
 - b. Соедините алгоритм **unwrap phase** с разъемом цикла For.
 - c. Удалите функции **shift register** и **index array**.
 - d. Добавьте узел обратной связи **feedback node**
 - e. Щелкните по нему правой кнопкой мыши и выберите **change direction**
 - f. Соедините выход функции обратной связи с функцией вычитания



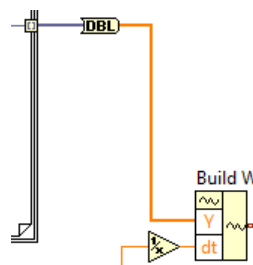
- g. Удалите цикл **For** вокруг алгоритма
- h. Соедините выход с разъемом цикла For



7. Преобразуйте все значения с плавающей точкой в значения с фиксированной точкой.
 - a. Для каждой функции и константы проделайте следующее:
 - i. Щелкните правой кнопкой мыши
 - ii. Выберите **Properties**
 - iii. Откройте закладку **Data Type** или **Output Configuration**
 - iv. Измените значение **representation** на **FXP**
 - v. Установите значение **word length** на 16, а **integer word length** на 12
 - vi. Нажмите **OK**



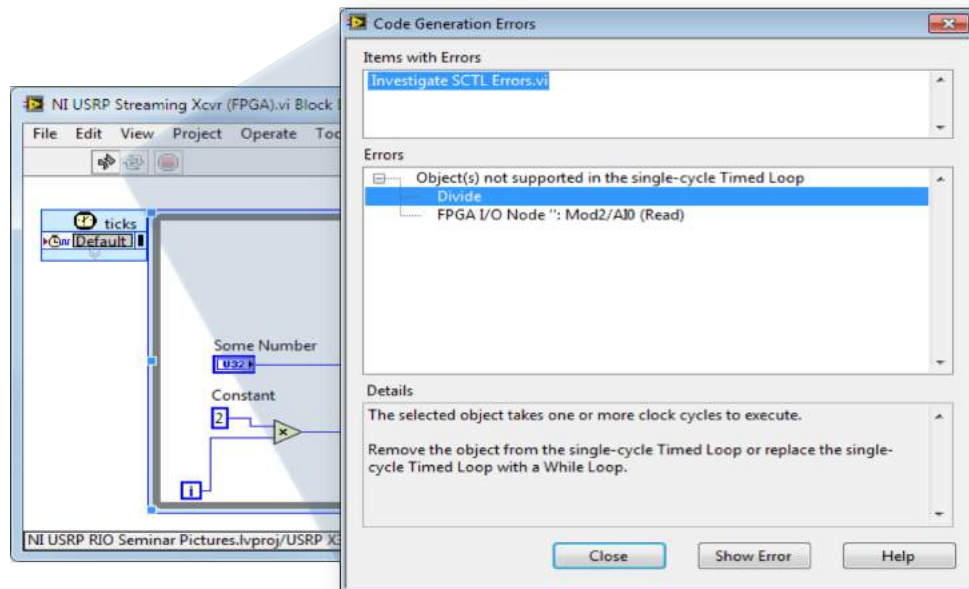
8. Добавьте функцию преобразования **To Double Precision Float**
9. Подключите выход к функции **Build Waveform**



10. Запустите VI и подключитесь к радиостанции из Упражнения 1, чтобы проверить работоспособность алгоритма.

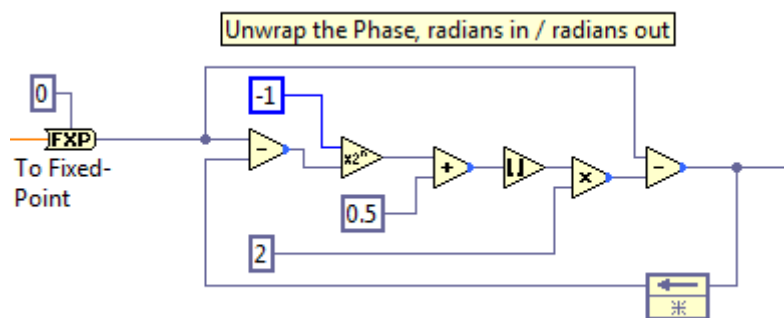
О программировании ПЛИС

При переводе прототипа на операции с фиксированной точкой на ПЛИС, нужно убедиться в том, что ваша ПЛИС поддерживает нужные функции. Например, если бы мы попытались скомпилировать код из упражнения 3В, мы бы увидели следующую ошибку:



Поскольку функция деления на ПЛИС занимает больше одного такта, и она будет помещена внутри одноклокового цикла, LabVIEW не сможет скомпилировать код для ПЛИС. Мы должны использовать функции, поддерживающие работу в один такт.

1. Удалите функцию **divide** из секции развёртки.
2. Поместите на блок-диаграмму функцию **Scale by Power of 2** и подключите выход самой левой функции вычитания с входным разъемом **x**.
3. Щелкните правой кнопкой мыши по входному разъему **n** и выберите константу **-1**



4. *Замените умножение на 2 аналогичным образом

Пожалуйста, дождитесь инструктора перед продолжением работы

