

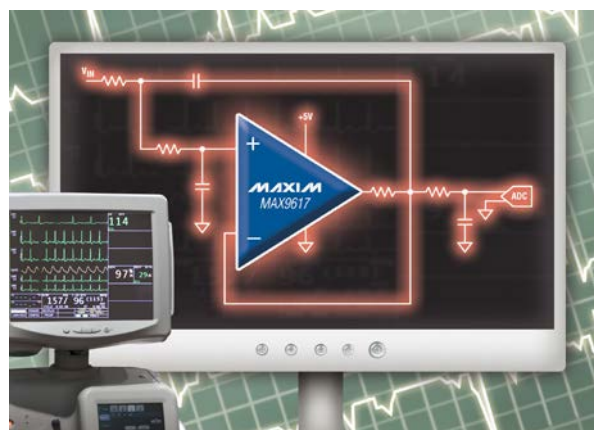
**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ  
БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТРАНСПОРТА»**

**Кафедра  
«Микропроцессорная техника и информационно-управляющие  
системы»**

**М. В. НАФТОЛЬСКИЙ, В. В. ШЕВЦОВ**

**СПЕЦИЗМЕРЕНИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА  
Лабораторный практикум**



**Гомель 2016**

## РАЗРАБОТКА АЦП ПОСЛЕДОВАТЕЛЬНОГО ПРИБЛИЖЕНИЯ В ГРАФИЧЕСКОЙ СРЕДЕ ПРОГРАММИРОВАНИЯ LABVIEW

**Цель работы:** изучение функций цифровой обработки сигналов в *LabVIEW*, преобразование входной физической величины в ее числовое представление с помощью аналого-цифровых преобразователей (АЦП). Приобретение практических навыков создания виртуальных приборов (ВП) для преобразования напряжения в код.

### Основные положения

Аналого-цифровое преобразование – это процесс преобразования входной физической величины в ее числовое представление. Аналого-цифровой преобразователь – устройство, выполняющее такое преобразование. Формально, входной величиной АЦП может быть любая физическая величина – напряжение, ток, сопротивление, емкость, частота следования импульсов, угол поворота вала и т.п. Однако, для определенности, в дальнейшем под АЦП мы будем понимать исключительно преобразователи напряжения в код, над которым микропроцессор и программное обеспечение выполняют определенные действия.

Понятие аналого-цифрового преобразования тесно связано с понятием измерения. Под измерением понимается процесс сравнения измеряемой величины с некоторым эталоном, при аналого-цифровом преобразовании происходит сравнение входной величины с некоторой опорной величиной (как правило, с опорным напряжением).

АЦП - один из самых важных электронных компонентов в измерительном и тестовом оборудовании.

В настоящее время известно большое число методов преобразования напряжение-код. Эти методы существенно отличаются друг от друга скоростью преобразования, разрядностью, сложностью аппаратной реализации. На рисунке 7.1 представлена классификация АЦП по методам преобразования. В основу классификации АЦП положен принцип, указывающий на то, как во времени происходит процесс преобразования аналоговой величины в цифровую. В основе преобразований выборочных значений сигнала в цифровые эквиваленты лежат последовательные операции квантования и кодирования. Они могут осуществляться с помощью последовательной, параллельной, либо комбинированной последовательно- параллельной процедур приближения цифрового эквивалента к преобразуемой величине.

Различные типы измерительного оборудования используют различные типы АЦП. Например, в цифровом осциллографе используется высокая частота дискретизации, но не требуется высокое разрешение. В цифровых мультиметрах нужно большое разрешение, но можно пожертвовать скоростью измерения.

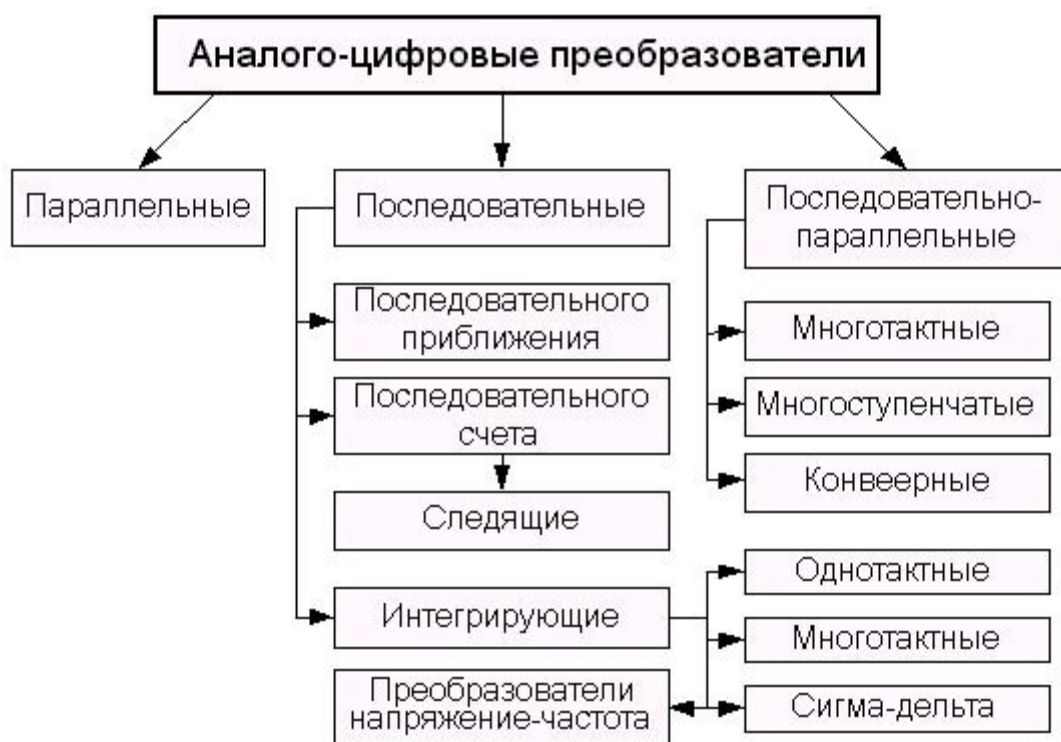


Рисунок 7.1 – Классификация АЦП по методам преобразования

Системы сбора данных общего назначения по скорости дискретизации и разрешающей способности обычно занимают место между осциллографами и цифровыми мультиметрами. В оборудовании такого типа используются АЦП последовательного приближения либо сигма-дельта АЦП. Существуют также параллельные АЦП для приложений, требующих скоростной обработки аналоговых сигналов, и интегрирующие АЦП с высоким разрешением и помехоподавлением.

На рисунке 7.2 показаны возможности основных архитектур АЦП в зависимости от разрешения и частоты дискретизации.

Наибольшим быстродействием и самой низкой разрядностью обладают АЦП параллельного преобразования. Например, АЦП параллельного преобразования *TLC5540* фирмы *Texas Instruments* обладает быстродействием 40MSPS при разрядности всего 8 бит. АЦП данного типа могут иметь скорость преобразования до 1 *GSPS*. Здесь можно отметить, что еще большим быстродействием обладают конвейерные АЦП (*pipelined ADC*), однако они являются комбинацией нескольких АЦП с меньшим быстродействием.

*Примечание* – Частота преобразования обычно выражается в отсчетах в секунду (*samples per second, SPS*), разрядность – в битах.

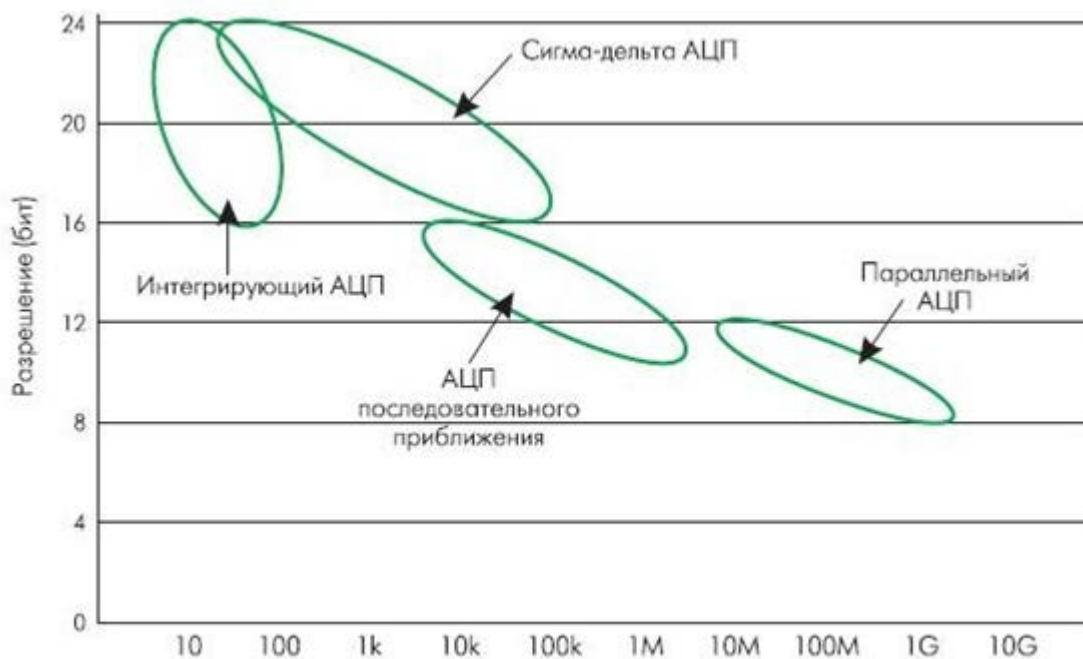


Рисунок 7.2 – Основные архитектуры АЦП в зависимости от разрешения и частоты дискретизации

АЦП последовательного приближения занимают среднюю нишу в ряду разрядность-скорость. Типичными значениями является разрядность 12-18 бит при частоте преобразования  $100KSPS-1MSPS$ .

Наибольшей точности достигают сигма-дельта АЦП, имеющие разрядность до 24 бит включительно и скорость от единиц  $SPS$  до единиц  $KSPS$ .

Еще одним типом АЦП, который находил применение в недавнем прошлом, является интегрирующий АЦП. Интегрирующие АЦП в настоящее время практически полностью вытеснены другими типами АЦП, но могут встретиться в старых измерительных приборах.

### АЦП параллельного (прямого) типа

Большинство высокоскоростных осциллографов и некоторые высокочастотные измерительные приборы используют параллельные АЦП из-за их высокой скорости преобразования, которая может достигать  $5 GSPS$  ( $5 \times 10^9$  отсчетов/сек) для стандартных устройств и  $20 GSPS$  для оригинальных разработок. Обычно параллельные АЦП имеют разрешение до 8 разрядов, но встречаются также 10-ти разрядные версии.

На рисунке 7.3 показана упрощенная блок-схема 3-х разрядного параллельного АЦП (для преобразователей с большим разрешением принцип работы сохраняется). С помощью трех разрядов можно представить восемь различных чисел, включая нуль. Необходимо, следовательно, семь компараторов ( $k_1 - k_7$ ) и семь эквидистантных опорных напряжений, которые

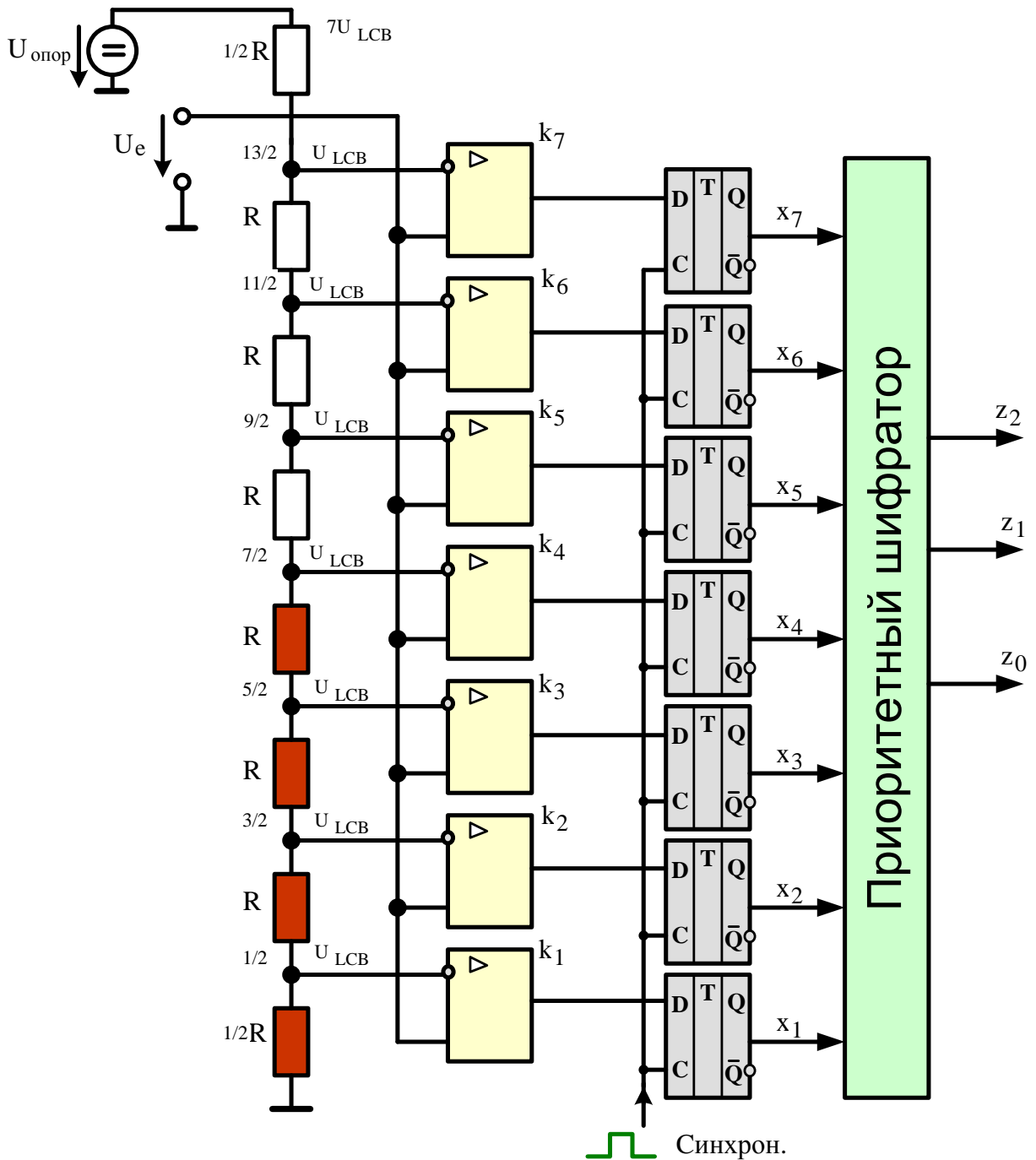


Рисунок 7.3 – АЦП- преобразователь работающий по параллельному методу

образуются с помощью резистивного делителя. Значения опорных напряжений начинаются со значения, равного половине младшего значащего разряда (*LSB*), и увеличиваются при переходе к каждому следующему компаратору с шагом, равным  $1/2 U_{LSB}$ . С увеличением входного напряжения компараторы последовательно устанавливают свои выходы в логическую единицу вместо логического нуля, начиная с компаратора, отвечающего за младший значащий разряд. Можно представить преобразователь как ртутный термометр: с ростом температуры столбик ртути поднимается.

На рисунке приложенное входное напряжение не выходит за пределы диапазона от  $7/2U_{LSB}$  до  $9/2U_{LSB}$  и, таким образом 4 нижних компаратора имеют на выходе "1", а верхние три компаратора - "0". Необходимы логические схемы, преобразующие эти состояния в двоичное число или десятичный эквивалент. Необходимое преобразование можно произвести, используя приоритетный шифратор. Однако приоритетный шифратор нельзя подсоединять непосредственно к выходам компараторов, так как может быть получен ошибочный результат, если произойдет изменение входного напряжения в течение времени измерения.

Этот недостаток можно устранить, если, как показано на рисунке 7.3, после каждого компаратора в качестве промежуточной памяти ввести схему запоминания аналоговой величины – срабатывающий по фронту триггер, который при воздействии тактового сигнала запускает следующие последовательности схемы.

Схема на рисунке 7.3 содержит 8 компараторов и имеет 3 разряда, для получения 8 разрядов нужно уже 256 компараторов, для 10 разрядов – 1024 компаратора, для 24-битного АЦП их понадобилось бы свыше 16 млн.

Параллельные АЦП - достаточно быстрые устройства, но они имеют свои недостатки. Из-за необходимости использовать большое количество компараторов параллельные АЦП потребляют значительную мощность, и их нецелесообразно использовать в приложениях с батарейным питанием.

### **АЦП последовательного (весового) приближения**

Когда необходимо разрешение 12, 14 или 16 разрядов и не требуется высокая скорость преобразования, а определяющими факторами являются невысокая цена и низкое энергопотребление, то обычно применяют АЦП последовательного приближения. Этот тип АЦП чаще всего используется в разнообразных измерительных приборах и в системах сбора данных. В настоящий момент АЦП последовательного приближения позволяют измерять напряжение с точностью до 16 разрядов с частотой дискретизации от 100К ( $1 \times 10^3$ ) до 1М ( $1 \times 10^6$ ) отсчетов/сек.

АЦП последовательного приближения реализует алгоритм «взвешивания», восходящий еще к Фибоначчи. В своей книге «Liber Abaci» (1202 г.) Фибоначчи рассмотрел «задачу о выборе наилучшей системы гирь», то есть о нахождении такого ряда весов гирь, который бы требовал для нахождения веса предмета минимального количества взвешиваний на рычажных весах. Решением этой задачи является «двоичный» набор гирь.

На рисунке 7.4 представлена упрощенная схема АЦП последовательного приближения. В основе АЦП данного типа лежит специальный регистр последовательного приближения. В начале цикла преобразования все выходы этого регистра устанавливаются в логический 0, за исключением первого (старшего) разряда. Это формирует на выходе внутреннего цифроаналогового преобразователя (ЦАП) сигнал, значение которого равно половине входного диапазона АЦП. А выход компаратора переключается в состо-

ание, определяющее разницу между сигналом на выходе ЦАП и измеряемым входным напряжением.

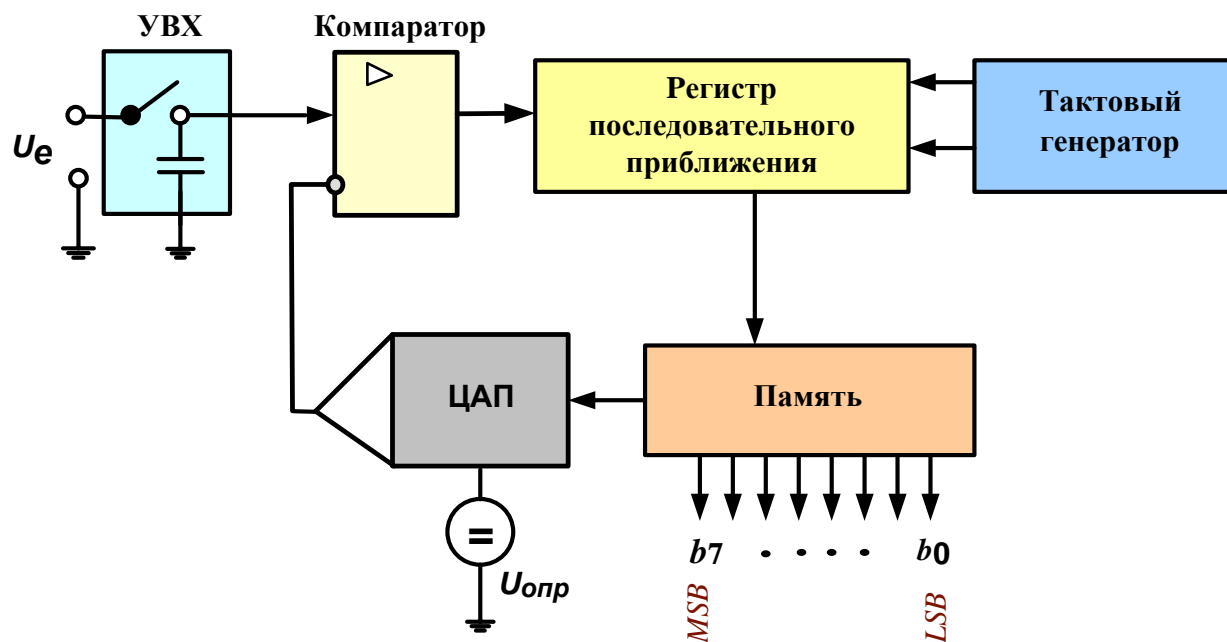


Рисунок 7.4 – Упрощенная схема АЦП последовательного приближения

Для работы данного АЦП принципиально важно, чтобы входное напряжение сохраняло неизменную величину в течение всего цикла преобразования. Однако «реальные» сигналы имеют свойство изменяться во времени. Устройство выборки-хранения (УВХ) «запоминает» текущее значение аналогового сигнала, и сохраняет его неизменным на протяжении всего цикла работы устройства.

На рисунке 7.5 представлен процесс преобразования в АЦП последовательного приближения. Если входное напряжение меньше половины входного диапазона АЦП, тогда выход компаратора примет значение логического 0. Это дает регистру последовательного приближения команду переключить свои выходы в состояние "01000000", что соответственно приведет к изменению выходного напряжения с ЦАП, подаваемого на компаратор. Если при этом выход компаратора по-прежнему оставался бы в "0", то выходы регистра переключились бы в состояние "00100000". Но на этом такте преобразования выходное напряжение ЦАП меньше, чем входное напряжение (см. рисунок 7.5), и компаратор переключается в состояние логической 1. Это предписывает регистру последовательного приближения сохранить "1" во втором разряде и подать "1" на третий разряд. Описанный алгоритм работы затем вновь повторяется до последнего разряда. Таким образом, АЦП последовательного приближения требуется один внутренний такт преобразования для каждого разряда, или N тактов для N-разрядного преобразования.

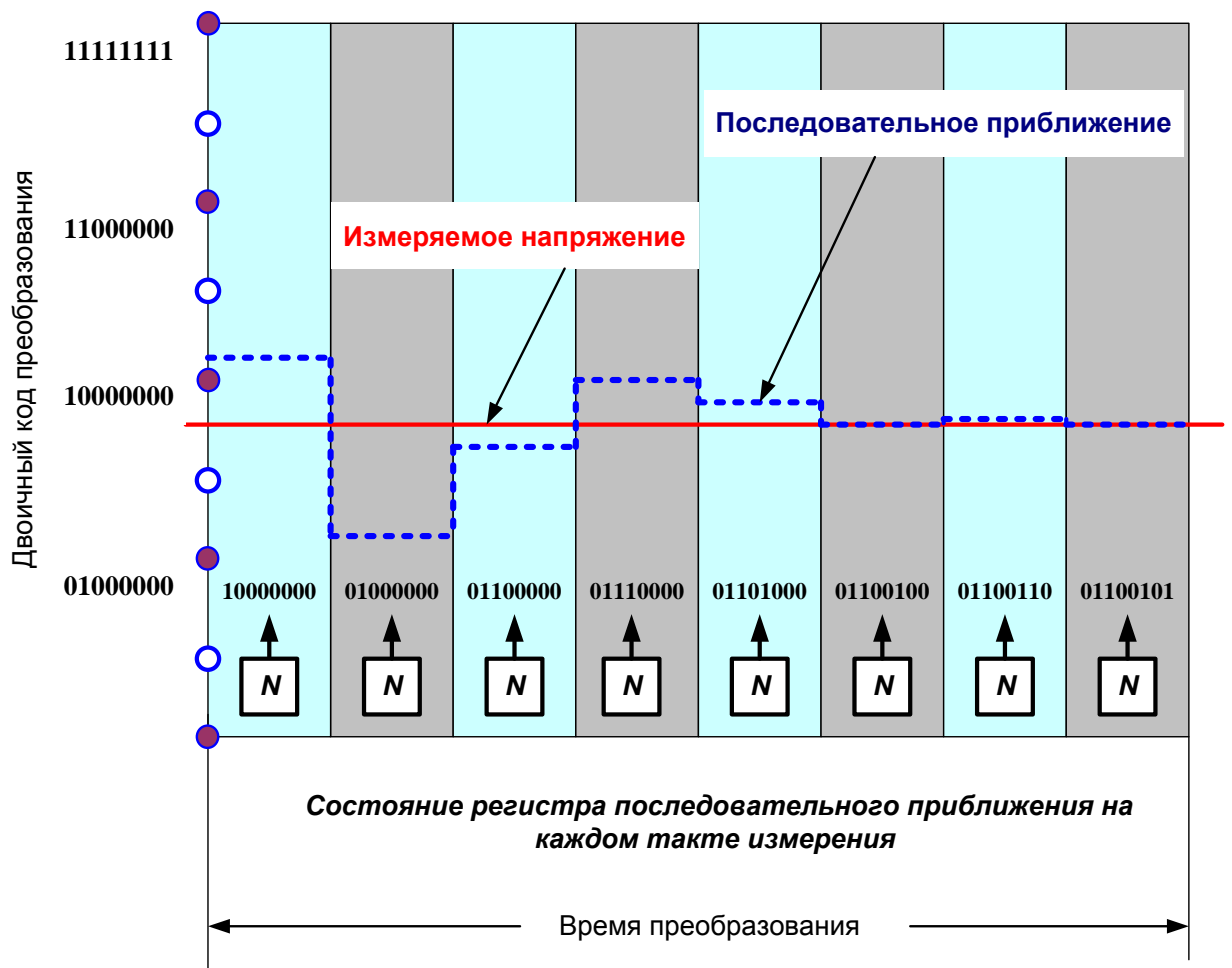


Рисунок – 7.5 Процесс последовательного приближения в АЦП

Работа АЦП последовательного приближения имеет особенность, связанную с переходными процессами во внутреннем ЦАП. Теоретически, напряжение на выходе ЦАП для каждого из  $N$  внутренних тактов преобразования должно устанавливаться за одинаковый промежуток времени. Но на самом деле этот промежуток в первых тактах значительно больше, чем в последних. Поэтому время преобразования 16-разрядного АЦП последовательного приближения более чем в два раза превышает время преобразования 8-разрядного АЦП данного типа.

Промоделируем работу аналого-цифрового преобразователя с использованием графического пакета *LabVIEW*. Алгоритм работы программы моделирующей работу АЦП последовательного приближения, представлен на рисунке 7.6. Программа работает в двух режимах преобразования (ручном и автоматическом), позволяет менять разрядность АЦП, устанавливать опорное напряжение, тактовую частоту, вычислять погрешность преобразования, а так же значения цены младшего разряда. Процесс последовательного приближения и переключение компаратора на каждом такте преобразования



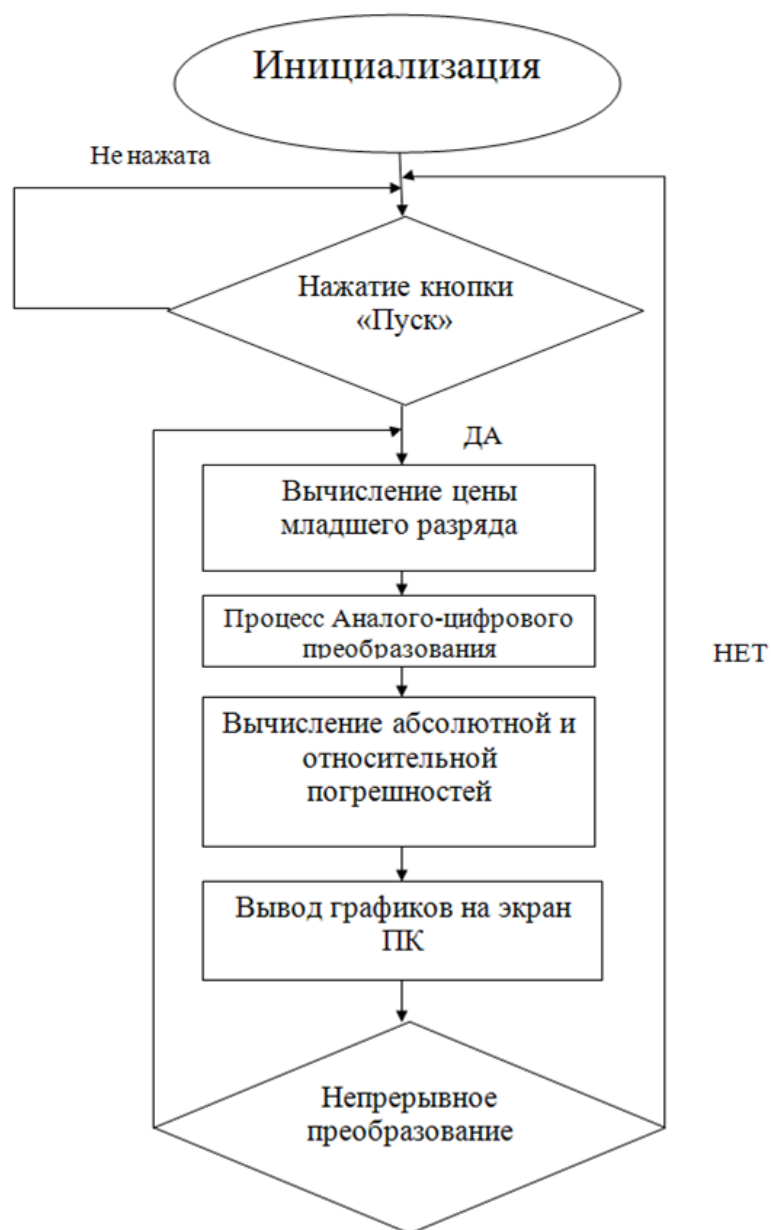


Рисунок – 7.6 Алгоритм работы АЦП последовательного приближения

отображается на графике работы АЦП.

Лицевая панель АЦП последовательного приближения представлена на рисунке 7.7.

Для определения порядка выполнения программы моделирования, осуществления управления потоком при обработке данных и придания наглядности в LabVIEW используется такая процедура, как структура последовательности (*Sequence Structure*). Структура последовательности весьма похожа на кадры киноплёнки и выполняется вначале кадр 0, за которым следует кадр 1, затем кадр 2 и т.д. пока не выполнится последний кадр и лишь после этого

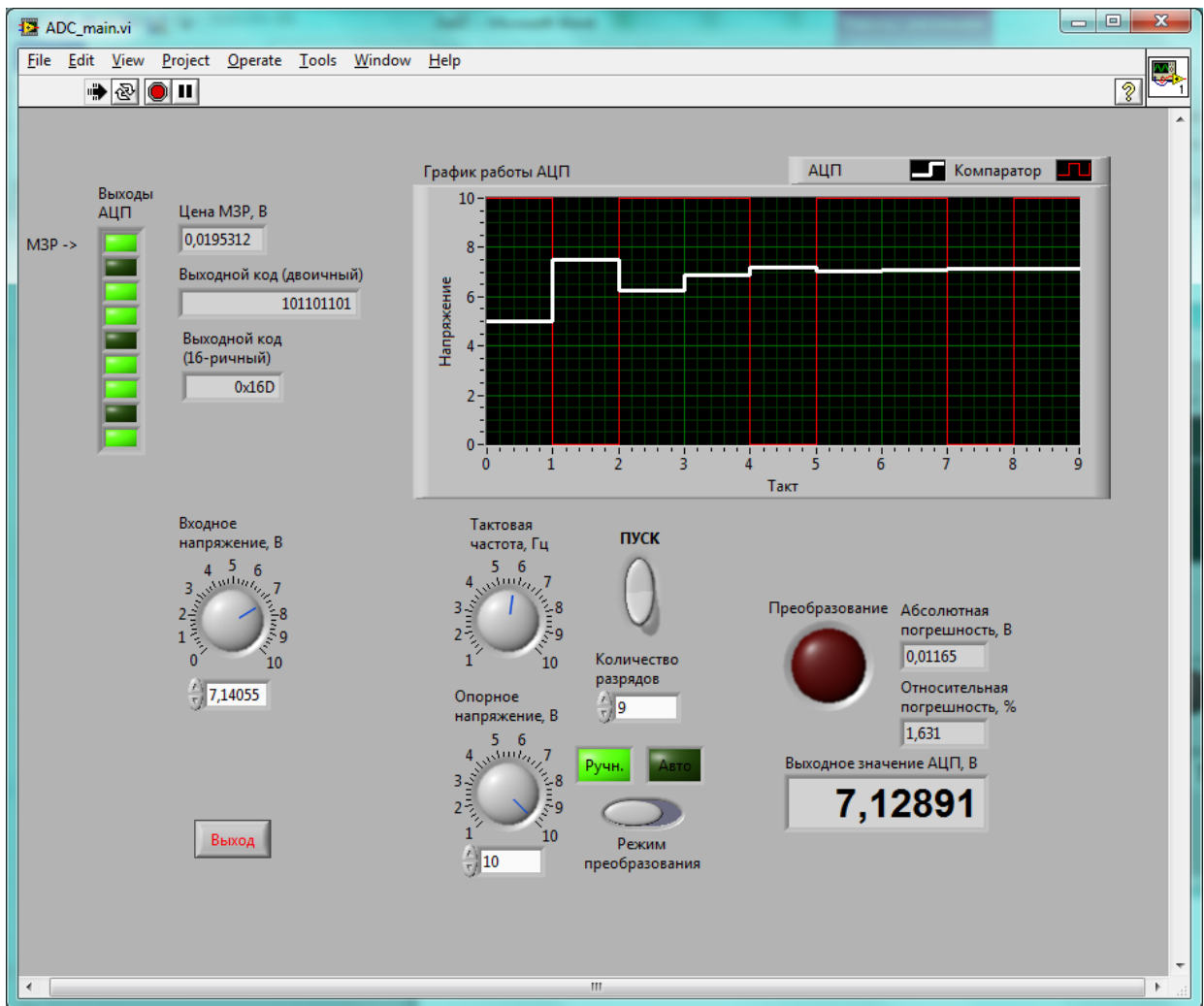


Рисунок 7.7 – Лицевая панель виртуального прибора АЦП

данные покидают структуру. На представленной на рисунке 7.8 блок-диаграмме АЦП последовательного приближения используется структура открытой последовательности, состоящей из трех кадров, которые исполняются подряд: начало преобразования, рабочий цикл и вывод информации. Использование открытой структуры последовательности более предпочтительно по сравнению с многослойной, т.к. в первом случае все части блок-диаграммы видны одновременно. Использование многослойной последовательности может быть оправдано, когда необходимо сэкономить место на блок-диаграмме.

Структура *While Loop* эквивалентна следующему псевдокоду: *do {программа} while {условие} ....*

При помещении структуры на функциональную панель ее контур в виде прямоугольника должен быть растянут так, чтобы охватить существующий фрагмент программы. Внутри структуры размещаются терминал итераций *i* и терминал условия (квадрат с круговой стрелкой). Фрагмент программы, размещенный в структуре, выполняется до подачи на терминал условия заданной булевской переменной True (1) или False (0). Если терминал условия не

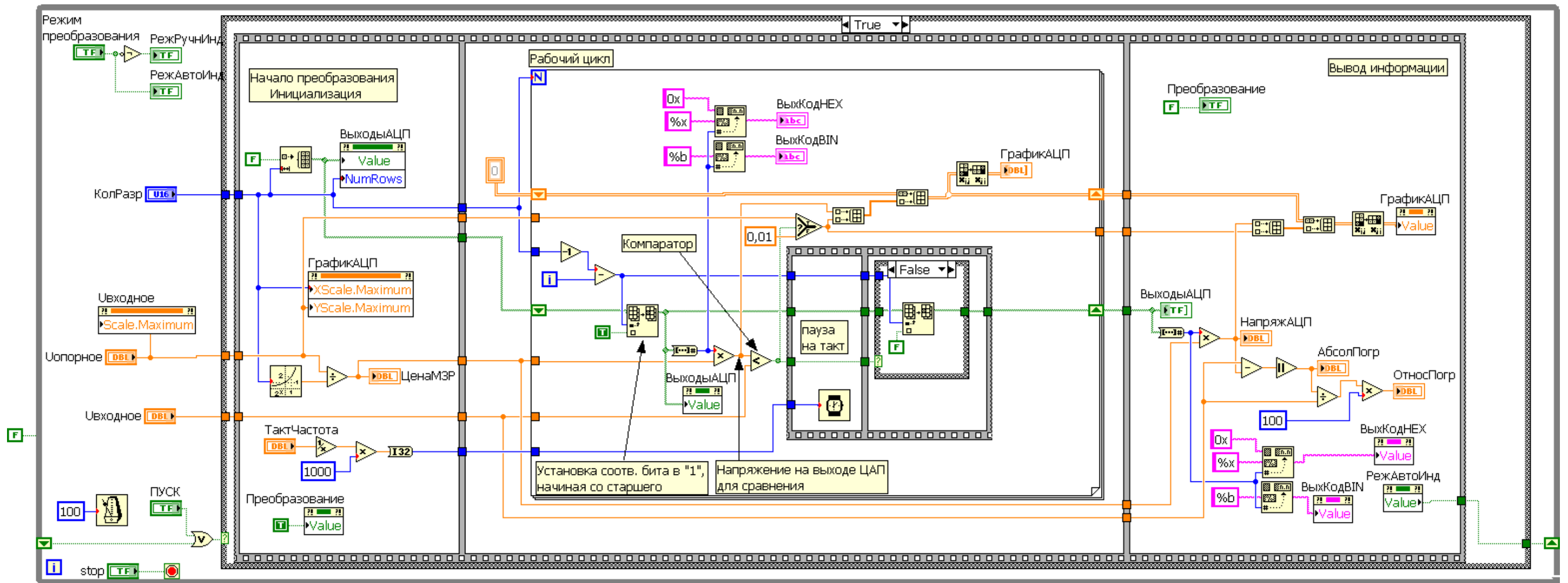


Рисунок 7.8 – Блок- диаграмма АЦП последовательного приближения

подключен к какому либо выходу, программа в структуре выполняется один раз. В структуре *While Loop* могут быть установлены сдвигающие регистры (*Shift Registers*) (СР). Установка СР производится щелчком ПКМ на любой вертикальной стороне прямоугольника структуры. СР используются для передачи данных из текущей итерации в последующую. Последовательность передачи данных через СР показана на рисунке 7.9.

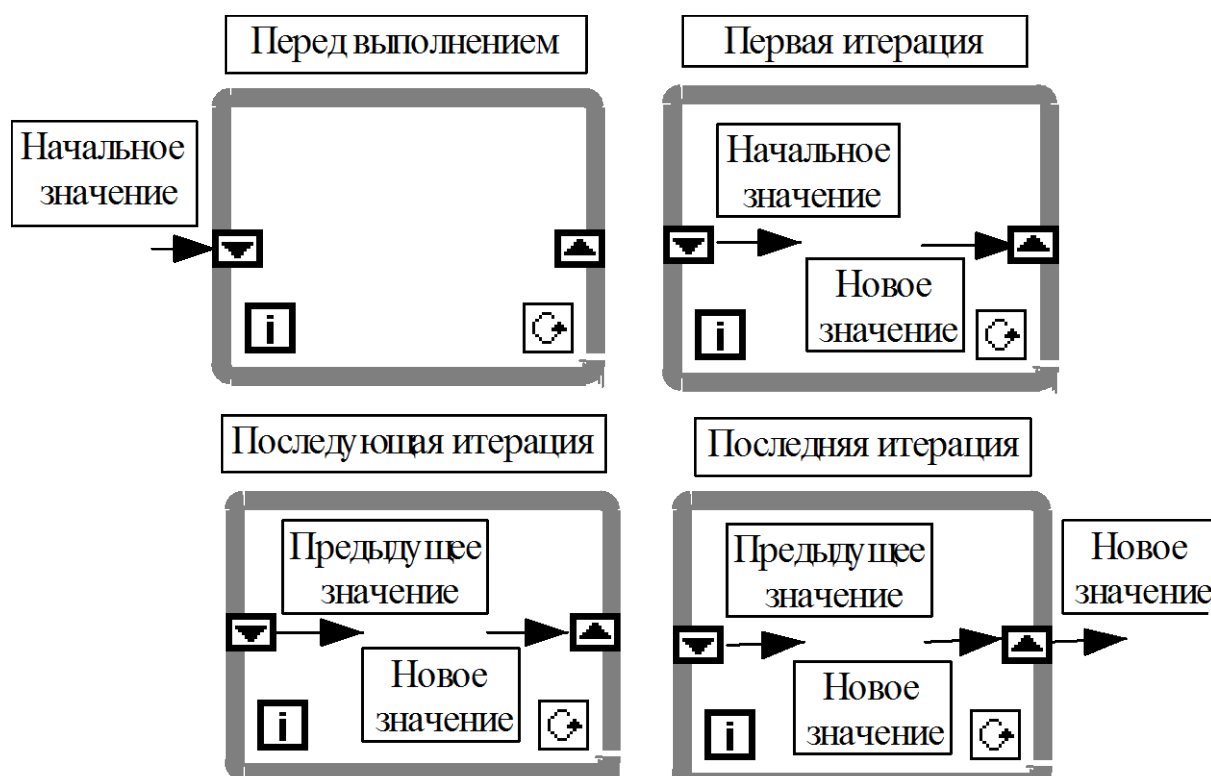


Рисунок 7.9 – Последовательность передачи данных через сдвиговый регистр

Структура *Case* аналогична конструкции *if-then-else* текстовых языков. По умолчанию *Case*-структура является булевой и имеет два варианта – *True* и *False*, выбираемых с помощью терминала выбора (элемент «?»). Структура автоматически преобразуется в числовую при подключении числовой переменной к терминалу выбора. С помощью меню, ПКМ при щелчке на рамке структуры, можно добавить или удалить вариант (*Add Case*), (*Delete This Case*). Одновременно можно наблюдать только один вариант (кадр) структуры. Переход между кадрами производится с помощью стрелок в заголовке, меню ПКМ на рамке структуры или меню ЛКМ на ее заголовке.

Ввод и вывод данных в структуре *Case* производится с помощью тоннелей. Установка тоннеля вывода данных в одном кадре структуры приводит к образованию тоннелей в других кадрах. До подключения данных к выходному тоннелю во всех кадрах он сохраняет белый цвет и воспринимается как ошибка программы. Для устранения необходимости такого подключения вместо тоннелей можно воспользоваться передачей данных с помощью локальной или глобальной переменной.

Структура *For Loop* аналогична текстовому оператору *for i = 0 to N-1 do ..*

При помещении структуры на функциональную панель ее контур в виде прямоугольника должен быть растянут так, чтобы охватить существующий фрагмент программы, который должен выполняться циклически заданное число раз, или чтобы позволить разместить новый фрагмент программы. Количество циклов задается в виде константы или контроля, подключаемых к переменной *N*. Значение *N*, помимо этого, может вычисляться в программе или определяться автоматически в соответствии с размером подаваемого на вход структуры массива (вариант работы с автоинкрементированием). В последнем случае значения массива вводятся в цикл поэлементно.

В структуре *For Loop* могут быть установлены сдвигающие регистры (*Shift Registers*) (СР). Установка СР производится щелчком ПКМ на любой вертикальной стороне прямоугольника структуры. СР используются для передачи данных из текущей итерации в последующую. Тип и размерность СР определяется типом и размерностью данных, подключаемых к правому (входному) элементу СР. Количество разрядов (тактов задержки) левого СР можно изменять мышью в режиме Стрелка или с помощью разделов (*Add Element, Remove Element*) меню ПКМ. По умолчанию начальные значения левых СР равны 0.

Принцип работы СР поясняется рисунком 7.10

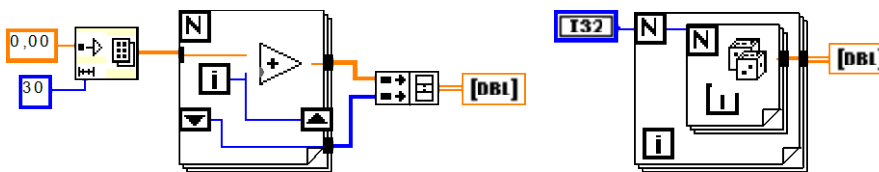


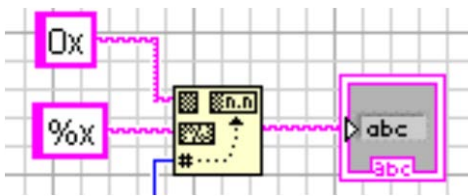
Рисунок 7.10 – Принцип работы сдвигового регистра

Для управления свойствами ВП, его запуска и, при необходимости, остановки, в данном проекте использованы функции *Property Node* (Узел свойств) из палитры *Programming* → *Application Control* (Управление приложением).

*Property Node* – Узел свойств возвращает либо записывает свойства ВП-приложения, на который указывает ссылка. Свойства - это разнообразные данные, например, открыта ли лицевая панель, каково ее название, активна ли она, и масса других. При выборе этой функции из подпалитры *Application Control* (Управление приложением), еще не указано, какое из свойств необходимо вернуть либо записать. Выбор конкретного свойства осуществляется путем нажатия правой кнопки мыши на функции. При этом происходит появление выпадающего меню, где нужно выбрать *Properties*. Число свойств в соответствующем элементе может быть установлено путем растяжения списка свойств, а их состав определен с помощью выбора из списка *Properties* меню ПКМ при ее щелчке на разделе свойств.



Функция программной задержки *Wait Until Next ms Multiple* необходима в большинстве циклических процедур, особенно тех, что ожидают действий пользователя через объекты лицевой панели - без задержки эти циклы будут выполняться с максимально возможной скоростью и тем самым бесполезно загружат большую часть ресурсов компьютера. Функция *Wait Until Next ms Multiple* переводит выполнение цикла в асинхронный режим, даже если вы подключаете величину 0 миллисекунд к терминалу интервала ожидания. Если в цикле производится ожидание действий пользователя, имеет смысл увеличить интервал ожидания до величины, характерной для времени реакции человека – большинство пользователей не сможет заметить задержки 100...200 миллисекунд между щелчком мыши на лицевой панели и началом выполнения соответствующего действия.



Для отображения на лицевой панели виртуального прибора АЦП (см. рисунок 7.7) выходного кода в двоичном или 16-ричном формате, используется функция *Format Value* (Преобразовать значение) из палитры

*Programming* → *String* → *String/Number Conversion* → *Format Value*.

Функция *Format Value* преобразует число, подключенное к входу значение (*value*), в упорядоченную строку в соответствии с форматом, определенным в строке формата (*format string*), и добавляет ее к входной строке (*string*), образуя выходную строку (*output string*).

В данной функции при определении формата используется следующая запись с применением упрощенных синтаксических элементов:

$\%[Width]Conversion\ Code$  ,

где % – символ, с которого начинается определение формата;

[Width] – число, определяющее ширину используемого поля (необязательный параметр);

*Примечание.* LabVIEW сканирует только заданное число символов при обработке параметра. Если ширина не задана или равна 0, то для выходного параметра выделяется такая ширина, какая необходима при его представлении.

*Conversion Code* (Код преобразования) – единичные символы, которые определяют способ сканирования или форматирования параметра.

*Примечание.* Коды преобразования могут прописными или строчными, за исключением кодов формата времени, которые чувствительные к регистру.

Ниже приведены коды преобразования для целых чисел:

- x – шестнадцатеричное целое (например, 16D);
- o – восьмеричное целое (например, 702);
- b – двоичное целое (например, 1111);
- d – десятичное целое со знаком;
- u – десятичное целое без знака.

## Выполнение работы

**Задание 1.** Создайте виртуальный прибор АЦП, представленный на рисунках 7.7 и 7.8. Для отчета сделайте снимки экрана (screenshot) лицевой панели и блок-диаграммы данного прибора. Изучите принцип работы новых для вас узлов диаграммы данного прибора и опишите их. Разберитесь с новыми типами данных, изучите новые элементы лицевой панели прибора. Сохраните проект виртуального прибора для его демонстрации во время защиты работы.

**Задание 2.** По заданию преподавателя внесите одно из перечисленных ниже изменений в созданный виртуальный прибор АЦП:

- структуру открытой последовательности, состоящей из трех кадров, заменить на многослойную;
- режим преобразования АЦП – только ручной;
- режим преобразования АЦП – только автоматический;
- фиксированное число разрядов;
- сократить отдельные узлы свойств ВП-приложений (*Property Node*).

Для отчета сделайте снимки экрана (screenshot) лицевой панели и блок-диаграммы прибора в соответствии с заданием. Сохраните проект виртуального прибора для его демонстрации во время защиты работы.

## Требования к отчету

Наименование и цель работы; краткие сведения из теории аналого-цифрового преобразования сигналов; снимки экрана (screenshot) лицевой панели и блок-диаграммы созданного и преобразованного виртуальных приборов АЦП в соответствии с заданием; описание узлов блок-диаграммы данного прибора; по заданию преподавателя отобразить один из этапов преобразования с использованием анимации потоков данных, пошагового выполнения программы, установкой отладочных индикаторов; заключение.

Отчет оформляется каждым студентом самостоятельно. Защита проходит в начале каждого следующего занятия с демонстрацией работы проекта на ПЭВМ. Студент, не подготовивший или не защитивший отчет по работе, к следующей лабораторной работе не допускается.

## Контрольные вопросы

1. Какой тип АЦП используются в системах сбора данных общего назначения?
2. Назовите основные достоинства и недостатки АЦП параллельного типа.
3. Какой алгоритм реализует АЦП последовательного приближения?






4. Какие управляющие структуры LabVIEW использованы на рисунке 7.9? Какая из структур вложена внутрь другой структуры?
5. Зачем используются сдвиговые регистры “*Shift register*”?
6. С какой целью узел “*Replase Array Subset*” помещен в структуру “*Case Structure*”?
7. Чем меньше цена младшего значащего разряда (МЗР), тем с более высокой точностью устройство может измерить сигнал. Приведите формулу подтверждающее это и продемонстрируйте на примерах в созданном вами виртуальном приборе АЦП.
8. Какой тип данных отображается розовым цветом на блок - диаграмме виртуального прибора?

## СПИСОК ЛИТЕРАТУРЫ



1. Бутырин, П.А. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7 Express / П.А. Бутырин, Т.А. Васьковская, В.В. Каратаев, С.В. Материкин – М.: ДМК Пресс, 2005. - 264 с.
2. Суранов, С.Я. LabVIEW 8.5: справочник по функциям / С.Я. Суранов. - М.: ДМК Пресс, 2007. - 572 с.
3. Батоврин, В.К. LabVIEW: практикум по электронике и измерительной технике / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин. - М.: ДМК Пресс, 2007.
4. Батоврин, В.К. LabVIEW: практикум по основам измерительных технологий / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин. - М.: ДМК Пресс, 2007.
5. Евдокимов, Ю.К. LabVIEW для радиоинженера: от виртуальной модели до реального прибора / Ю.К. Евдокимов, В.Р. Линдваль, Г.И. Щербаков. - М.: ДМК Пресс, 2007.
6. Федосов, В.П. Цифровая обработка сигналов в LabVIEW / В.П. Федосов, А.К. Нестеренко. - М.: ДМК Пресс, 2007.
7. Лупов С.Ю., Муякшин С.И., Шарков В.В. LabVIEW в примерах и задачах. Учебно-методические материалы по программе повышения квалификации “Обучение технологиям National Instruments”. – Нижний Новгород, 2007, 101 с.
8. Теоретические основы цифровой обработки сигналов с помощью LabVIEW. Режим доступа: <http://1labview.info/>.



Описание функций работы с массивами в меню Array

Входы	Иконка	Выходы	Название	Операция
1	2	3	4	5
Массив		Размер массива	<b>Array Size</b>	Определение размера (ов) массива
Массив Индекс		Элемент массива (подмассив)	<b>Index Array</b>	Выделение элемента массива
Массив Новый элемент Индекс		Массив с новым элементом	<b>Replace Array Element</b>	Замена элемента в массиве
Массив Индекс Длина		Подмассив	<b>Array Subset</b>	Выделение подмассива по индексу и длине
Массив размера N Длина		Массив размера M	<b>Reshape Array</b>	Преобразование массива размером N в массив размером M
Элемент		Новый массив	<b>Initialize Array</b>	Создание нового массива
Массив Элемент ....		Массив с добавленными элементами	<b>Build Array</b>	Создание массива из массивов или элементов
1D массив Индекс		Массив с циклич. заменой элементов	<b>Rotate 1D Array</b>	Циклическая замена элементов массива
1D массив		Массив с обратным порядком элементов	<b>Reverse 1D Array</b>	Перестановка элементов в обратном порядке

2D массив		Транспонированный массив	<b>Transpose 2D Array</b>	Транспонирование массива
1D массив Элемент Нач. индекс		Индекс элемента	<b>Search 1D Array</b>	Поиск элемента в массиве
1D массив		Отсортированный массив	<b>Sort 1D Array</b>	Сортировка массива
Числовой массив		Величины и индексы максимального и минимального значений	<b>Array Max&amp;Min</b>	Определение величины и индексов максимального и минимального значений
1D массив Индекс		Первый и второй подмассивы	<b>Split 1D Array</b>	Выделение подмассивов до и после индекса
1D массив Дробный индекс		Интерполированное значение	<b>Interpolate 1D Array</b>	Интерполяция значений массива
1D массив Порог Индекс		Дробный индекс	<b>Threshold 1D Array</b>	Определение положения точки пересечения порога
1D массив 1D массив .....		1D массив с чередующимися элементами исходных массивов	<b>Interleave 1D Arrays</b>	Объединение 1D массивов в 1D массив с чередующимися элементами
1D массив		N подмассивов со сдвинутыми на N элементами	<b>Decimate 1D Array</b>	Выделение N подмассивов со сдвинутыми на N элементами

1D массив		Кластер	<b>Array To Cluster</b>	Преобразование массива в кластер
Кластер		1D массив	<b>Cluster To Array</b>	Преобразование кластера в массив