

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Схемотехника

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

АДАПТИВНАЯ СИСТЕМА ОСВЕЩЕНИЯ
БГУИР КП 1-40 02 01 310 ПЗ

Студент: гр. 050503 Деруго Д. В.

Руководитель: ассистент каф. ЭВМ
Жук Д.С.

Минск 2022

Оглавление

ВВЕДЕНИЕ	3
1 ОБЗОР ЛИТЕРАТУРЫ	4
1.1 Используемые алгоритмы	4
1.2 Внутреннее устройство ПЛИС.....	5
1.3 Протокол RS-232.....	6

ВВЕДЕНИЕ

Вычислительная техника используется повсеместно уже не первый десяток лет, от элементарных детских игрушек и компьютеров с процессорами общего назначения до станков с ЧПУ и узкоспециализированных схем для самых требовательных к вычислительной мощности задач. Возникает все больше проблем, требующих обработки информации, и в то время как старые задачи нашли приемлемое на данный момент решение, нынешние тенденции развития науки и техники бросают вызов не только программному, но и аппаратному обеспечению. На слуху у инженеров вычислительной техники масса инновационных решений в сфере «железа», например, нейропроцессоры и продвинутые модули цифровой обработки сигналов.

Алгоритмы обработки данных продвинулись далеко вперед, но могут ли они обеспечить достаточный результат на неэффективном аппаратном обеспечении? Безусловно, можно улучшить старые методы, выпустив новую линейку процессоров, но можно пойти другим путем, например, в сторону параллельных вычислений. Однако заказывать отдельные схемы для задач узкого спектра было бы крайне накладно. Для этих пограничных задач применяются схемы программируемой логики.

ПЛИС — программируемая логическая интегральная схема — является в достаточной мере эффективным средством решения задач специализированных вычислений. Ее структуру мы изучаем в текущем курсе схемотехники. Помимо самого чипа с программируемой логикой и массы электронных компонентов, служащих для корректной работы чипа, отладочная плата содержит индикаторы и множество портов различного назначения, некоторые из которых будут использоваться в данном курсовом проекте.

Проект представляет собой вычислитель квадратного корня. Загрузка и получение значений будет производиться через устаревший, но очень подходящий для учебных целей протокол RS-232, а обработка данных в разработанном блоке, который и является самой сутью курсового проекта. В ходе работы будут применены знания как нынешнего курса схемотехники, так и навыки, приобретенные на других дисциплинах.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Используемые алгоритмы

Мной рассматривались только те алгоритмы вычисления квадратного корня, подходящие для реализации на аппаратной основе. Безусловно, все вычисления так или иначе проходят через соответствующую аппаратуру, однако проводить расчеты с помощью методов, основанных на экспоненциальной функции или ряде Тейлора, намного проще на процессоре, чем на специальной схеме.

Многие итерационные алгоритмы требуют начального значения. Очевидно, это значение должно быть больше единицы для целого числа и как можно ближе к значению самого корня числа, чтобы для его вычисления на основе оценки потребовалось меньше итераций. Поэтому выгодно иметь первоначальную оценку, что ускоряет сходимость ряда, выстраивающегося в ходе вычислений.

Так, для аппаратного воплощения был выбран итеративный вавилонский метод, или же метод Герона, с предшествующим ему вычислением двоичной оценки.

Двоичная оценка имеет достаточную точность и дает сбой только с малыми целыми и дробными значениями, например, корень числа 0.5 такой метод оценивает в 0.5, что мало чем полезно для последующих вычислений. Пусть S — число, корень которого требуется. Представим его в двоичном виде так:

$$S = a * 2^{2n}$$

Тогда корень числа будет выглядеть так:

$$\sqrt{S} \approx (0.5 + 0.5 * a) * 2^n$$

Значения, вычисленные на этом этапе, перейдут во второй подблок вычислителя, то есть в блок вавилонского метода.

Формула для вавилонского метода выглядит следующим образом:

$$x_{n+1} = 0.5 * (x_n + \frac{S}{x_n})$$

Где x_n означают значение корня, приближающегося к своему реальному значению с каждой итерацией.

Как видно из формул, реализация алгоритмов потребует проектирования логики с использованием сдвиговых регистров, делителей и умножителей. Также требуется решить, на каком принципе будет построена логика итеративных операций.

1.2 Внутреннее устройство ПЛИС

ПЛИС — это программируемая логическая интегральная схема.

Строение схем с программируемой логикой в достаточной мере сложное, поэтому будет рассмотрена только требуемая основа.

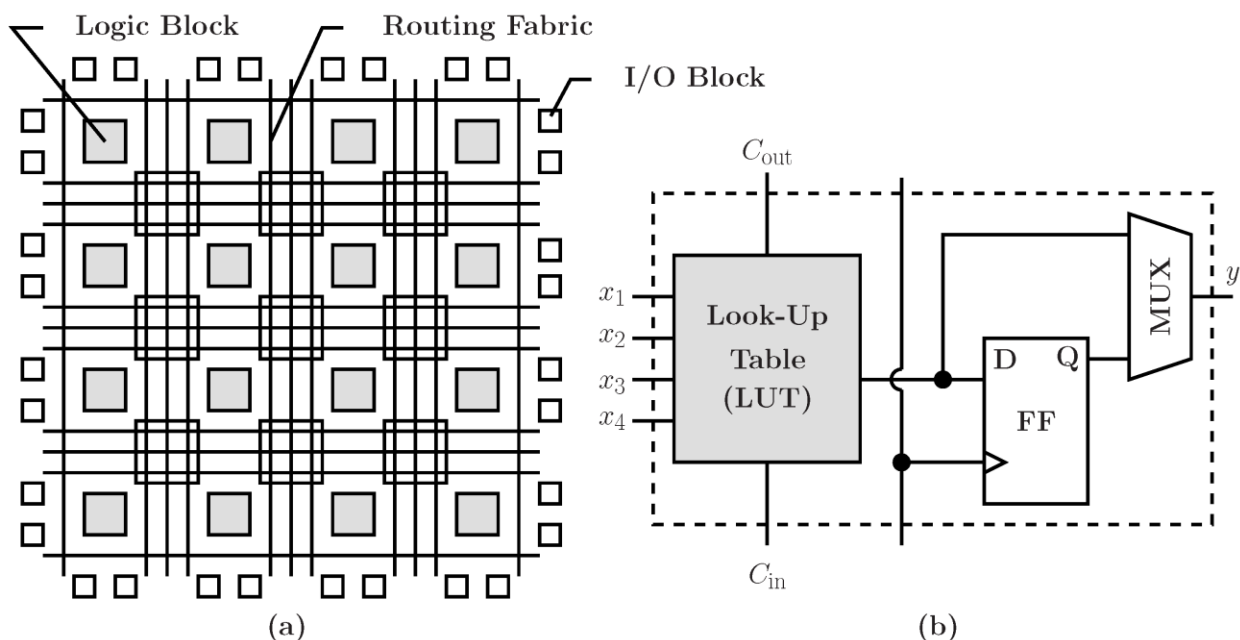


Рисунок 1.2.1 — устройство логической ячейки и массива ЛЯ

На рисунке 1.2.1 в упрощенном виде представлено внутреннее строение ПЛИС. Помимо блоков ввода-вывода присутствуют программируемые логические ячейки и линии связи (коммутационное поле), которые их соединяют в нужном виде. В отличие от программируемых логических матриц, ПЛМ, которые содержат массив элементарных логических элементов И-НЕ или ИЛИ-НЕ, в базе которых можно отобразить любую логическую функцию, ПЛИС содержит небольшой объем памяти, которая содержит таблицу истинности для требуемой функции.

Поскольку мало какой проект в наше время являет собой чисто комбинационную схему, в блок включен D-триггер, на основе которых строятся, например, сдвиговые регистры. Помимо этого, крупные схемы содержат блоки RAM, которые используются по решению разработчика или в ходе оптимизации схемы синтезатором кода, чтобы не расходовать на память ценные программируемые ячейки.

Память LUT строится на SRAM, то есть она энергозависима. Линии связи также основаны на программируемой логике. По этой причине прошивка стирается при выключении, но она может быть загружена во внешнюю энергонезависимую память, откуда будет загружаться обратно при запуске.

В зависимости от сложности своего исполнения, ПЛИС содержат некоторое количество готовых блоков умножения-суммирования, широко применяющиеся в задачах цифровой обработки сигналов.

Такая гибкая архитектура позволяет создать требуемую схему без трудоемкого процесса разработки заказного чипа и еще более затратного его изготовления. ПЛИС стали известны обществу с набором популярности майнинга криптовалют, так как нужный алгоритм требовал серьезного распараллеливания вычислений, которое не могли предоставить даже графические процессоры GPU.

1.3 Протокол RS-232

Существует немало протоколов передачи данных. Многие из них используются в наши дни, многие уже устарели и их порты давно не предусматриваются в электронной технике. Тем не менее, RS-232 оставил достаточный след в индустрии, чтобы все еще быть востребованным, хоть и активно вытесняемым современными интерфейсами. Этот протокол прост в аппаратной реализации, потому для данного курсового был выбран именно он.

Протокол RS-232 является последовательным интерфейсом, то есть биты данных в нем передаются друг за другом. Самая элементарная посылка, без дополнительного кодирования, представлена на рисунке 1.3.1.

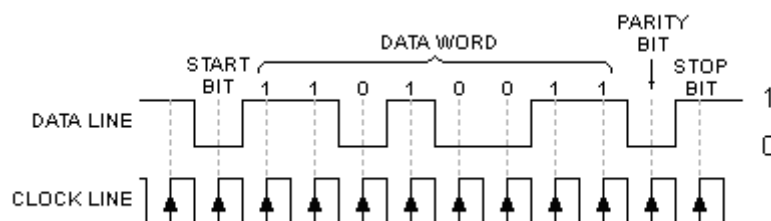


Рисунок 1.3.1 — посылка протокола RS-232

При отсутствии передачи линия содержит значение логической единицы. Начало передачи сигнализируется стартом-битом в виде логического нуля. Стандарт предусматривает длину передаваемого слова в 5-9 бит без учета бита четности.