

МИКРОПРОЦЕССОРНАЯ СИСТЕМА НА ОСНОВЕ ПЛИС

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Базанов М. Г.

Селезнев И. Л. – к. т. н., доцент

В последнее время в реализации встраиваемых систем различного назначения начинают все больше и больше фигурировать системы, основанные на микропроцессорах и представленные в основном микроконтроллерами. В качестве микропроцессора можно использовать и ПЛИС.

На данный момент наиболее популярными интегральными схемами для разработки встраиваемых систем являются: Application-Specific Integrated Circuit (ASIC), System-on-a-Chip (SoC), ПЛИС типа FPGA. Основываясь на преимуществах и недостатках каждой из технологий и поставленной задаче, выбирается система, которая будет относительно недорогой и будет обладать определенной гибкостью.

Технология ПЛИС – это программируемая логическая интегральная схема, применяемая для создания собственных цифровых устройств. Логика работы ПЛИС не определена при ее создании, а определяется путем дополнительного программирования с помощью специальных средств: программаторов и программного обеспечения.

Микросхемы ПЛИС не являются микропроцессорами, в которых заданная программа выполняется всегда последовательно, команда за командой. В ней реализуется конкретная электронная схема, состоящая из логических элементов и триггеров. Однако данная технология при достаточном количестве ресурсов в кристалле, позволяет реализовать в себе программный микропроцессор и другие функциональные блоки, которые далее можно запрограммировать и использовать для определенно поставленных целей.

Обычно, микросхема ПЛИС состоит из: логических блоков, которые можно сконфигурировать так, чтобы они реализовывали требуемую логическую функцию, программируемых электронных связей между конфигурируемыми логическими блоками, программируемых блоков ввода/вывода, которые обеспечивают связь внешнего вывода микросхемы с внутренней логикой. Так же могут присутствовать блоки памяти, блоки Digital Signal Processor (DSP), умножители, блоки Phase-Locked Loop (PLL) для подстройки частоты генератора и другие.

Логические элементы, как и запоминающие элементы, могут быть собраны в регулярные матрицы. Если можно запрограммировать соединения между логическими элементами, то такие матрицы можно конфигурировать для реализации различных логических функций, причем не будет необходимости изменять соединения между микросхемами на плате. Программируемые матрицы производятся в больших количествах, что обеспечивает их малую стоимость.

Структура FPGA представляет собой матрицу конфигурируемых Логических элементов (ЛЭ), каждый ЛЭ может сконфигурировать для выполнения функций комбинационной или последовательной схемы. ЛЭ окружены элементами ввода/вывода (ЭВВ), которые предназначены для организации обмена информацией между FPGA и прочими компонентами системы. Теоретически, ЛЭ может быть очень простым - как отдельный транзистор. Или он может быть очень сложным - как целый процессор. Сложность и структура ЛЭ определяется производителем.

В первом случае требуется огромное число программируемых связей, чтобы потом из отдельных транзисторов собрать требуемую схему. Во втором случае связей необходимо значительно меньше, но теряется гибкость проектирования пользовательской схемы. Как правило, ЛЭ является чем-то средним. Он достаточно сложен для конфигурирования определенной логической функции и довольно мал, чтобы разместить множество таких блоков внутри FPGA и связать их в единую схему.

Матрицы FPGA подходят не для любого проекта. К примеру, при вычислениях с плавающей запятой и сложной программной реализацией. Это создает довольно большие затраты на вычисление сложных, не однотипных операций. Учитывая гибкость FPGA и преимущества вычислительной мощности микропроцессоров – чаще всего разработчики прибегают к компромиссу, используя гибридную платформу, содержащую оба устройства. Гибридные платформы лучше всего подходят для систем, где требуется связующее программное обеспечение и полный стек протоколов обмена данными, а также синхронизирующие схемы.

Одним из наилучших решений данной задачи являются встраиваемые, вычислительные ядра и функциональные модули в матрицу FPGA. Xilinx и Altera предоставляют матрицы, содержащие синтезируемые на логических элементах процессорные блоки. Архитектура процессоров и модулей может быть любой: стандартной, заказной или собственной разработки. Гибридные системы могут использоваться в качестве альтернативы микроконтроллерным устройствам, что значительно упрощает разработку. В том числе этот подход оказывается выгодным, так как позволяет сэкономить на лицензии процессорного ядра, которая выдается вместе с покупкой FPGA. При использовании внешнего процессора или внешнего функционального модуля – их пришлось бы лицензировать отдельно. Еще одна из привлекательных особенностей ядер в FPGA – возможность реализовать аппаратные ускорители. Они могут быть синтезированы на блоках RTL либо определены как отдельная функция.

Для работы со встроенными в матрицу FPGA ядрами необходимо использование дополнительных схем и осуществление верификации проекта. К дополнительным схемам относятся: модули памяти, периферийные

устройства, графический интерфейс пользователя, который помогает связать все блоки системы, создать структуру адресов и так далее.

Множество проектов перестают работать или же начинают работать некорректно после добавления процессорного ядра, шины и периферийных устройств. Это возникает из-за того, что система становится довольно сложной, чтобы ее можно было проверить простым прогоном. Необходимы более совершенные инструменты, быстродействующие модели и методы, анализирующие систему на уровне сверхбольшой интегральной схемы (СБИС). Следует проводить не только моделирование работы всех функциональных блоков, но и верификацию модулей сторонних производителей. Так же стоит использовать испытательный стенд, для отслеживания хода работы всего процесса системы. Гораздо проще, когда производитель предоставляет отлаженные и проверенные примеры кода. Программная отладка обычно производится с помощью инструментов JTAG, позволяющих последовательно отслеживать работу и устанавливать точки останова.

Таким образом, в данной работе рассмотрены некоторые основные моменты структуры ПЛИС и всей системы в целом, а также указаны некоторые проблемы, которые могут возникнуть в процессе разработки.

Список использованных источников:

1. Архитектура ПЛИС (FPGA) [Электронный ресурс]. – Режим доступа: <https://marsohod.org/11-blog/265-fpga>.
2. ПЛИС [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/ПЛИС>.

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ УЧЁТА, АНАЛИЗА И ПЛАНИРОВАНИЯ ЛИЧНОГО БЮДЖЕТА

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Беликова Т.О.

Байрак С.А. – ст. преподаватель

Мобильные приложения - то, что актуально, то, что не имеет границ, то, чем пользуются все обладатели смартфонов. Благодаря такому ПО человек может изучать информацию любого уровня, беззаботно проводить время, а также взаимодействовать с другими людьми. Помимо социальных сетей и игр, существуют полезные программы, которые создаются как помощники для своего обладателя. Одним из таких приложений и является объект данной разработки.

Программное средство для учёта, анализа и планирования личного бюджета – разработано для минимизации времени, которое уходит при манипуляциях с денежными потоками. Приложение работает в режиме оффлайн. Название предполагает, что:

1. Приложение ведет учёт всех входных и выходных потоков денежных средств. Делается это при моментальных изменениях финансового положения, чтобы ничего не забыть.
2. Есть возможность расчёта больших денежных вложений: отпуск, покупка квартиры или автомобиля и т.д. То есть, пользователь анализирует только базовые элементы такие как сроки накоплений и необходимая сумма. Всё остальное берет на себя программное средство. А в результате предоставляет варианты развития событий, при которых пользователь сможет достичь своих целей.
3. Анализ данных происходит при каждом изменении. Поэтому в любой момент времени можно узнать состояние финансовых дел.

Особенности программного средства для учёта, анализа и планирования личного бюджета:

1. Простой и понятный интерфейс.
2. Нет проблем с поиском нужной графы для записи информации о затратах. Т.е. пользователь сам создаёт себе варианты расходов, а не выбирает из общепринятых или из тех, которые возможны в теории.
3. Возможность минимизации затрат, путем анализа статистических данных. Графики или диаграммы, пользователь сам выберет, на что ему удобнее смотреть.
4. Ежедневная обработка данных и учёт потраченных средств с имеющимися.
5. Возможность переноса всех данных из приложения в облако.
6. Ключевая особенность приложения – нельзя уйти в минус. Учет ведется абсолютно по всем сбережениям. И если пользователь использует помощь других людей и их средства, то программное обеспечение позаботится о том, чтобы проинформировать своего обладателя о том, что он уже пользуется не собственными, а заимствованными денежными единицами.

Однако в смартфонах память не безгранична. Поэтому основная проблема данного ПО – память. Рано или поздно, смартфон не сможет принимать информацию от пользователя. Решение этой проблемы