

ОСОБЕННОСТИ РАЗРАБОТКИ ЦИФРОВОГО УСТРОЙСТВА НА ПЛИС С ТОЧКИ ЗРЕНИЯ ПРОГРАММИСТА

И.Г. Мещерова

Научный руководитель: А.Н. Мальчуков

Томский политехнический университет

E-mail: igm4@tpu.ru

Введение

Разработка на ПЛИС — не типичное для программиста занятие, и без знания низкоуровневой структуры сложно достичь успехов в этой сфере.

Сама же ПЛИС представляет собой чип, содержащий в себе множество элементов. Какие именно это будут логические элементы, их взаимодействие с внешним миром определяется на этапе программирования ПЛИС. Проектирование ПЛИС отличается от программирования, в котором используются языки высокого уровня, и имеет свои особенности.

Целью данной работы является выявление особенностей разработки цифрового устройства на ПЛИС с точки зрения программиста.

Область применения ПЛИС

В век цифровых технологий расширение сферы применения ПЛИС обусловлено минимальными затратами на проектирование и большим числом выполняемых функций. ПЛИС имеет более гибкую структуру, чем микроконтроллер, и служит для реализации нескольких задач. Такие схемы широко используются для построения различных по сложности и по возможностям цифровых устройств, например, устройств с большим количеством портов ввода-вывода, устройств, выполняющих цифровую обработку сигнала, а также широкое применение ПЛИС находит в системах защиты информации и т.д.[2]

Опыт программирования на языках описания аппаратуры

Для проектирования цифровых устройств используются языки программирования, называемые языками описания аппаратуры.

В данной работе все вопросы будут освещены с опорой на язык SystemVerilog, так как именно на нём разрабатывался первый проект — автомобильно-пешеходный светофор с кнопкой на учебно-лабораторном комплексе SDK 6.1.

Отличие от программирования на языках высокого уровня

Высокоуровневые языки программирования были разработаны для скорости и удобства использования программистом, в то время как низкоуровневые языки более приближены к машинному коду и позволяют работать напрямую с устройством.

Для разработки цифровых устройств используются именно языки низкого уровня, имеющие более сложный и далекий от понимания синтаксис. Если некоторые команды языков высокого уровня могут быть понятны людям, далеким от программирования, то понять HDL (hardware description

language, в переводе – язык описания аппаратуры) будет гораздо сложнее. Для примера в таблице 1 представлен синтаксис двух языков – C++ и SystemVerilog на примере конструкции конечного автомата. Синтаксис C++ является более интуитивно понятным.

Таблица 1. Синтаксис C++ и SystemVerilog

C++	SystemVerilog
switch (state)	always_ff @(posedge
{	clock) begin
case State_Start:	case (present_state)
.....	st_start:
state = State_Next;
break;	next_state <= st_next;
case State_Next:	st_next:
.....
}	end

В языках описания аппаратуры используются некоторые конструкции, которые не имеют аналогов в других языках, такие как провод (wire) и регистры (reg). Чтобы в дальнейшем не допускать ошибок, необходимо понять не только назначение этих элементов, но и базовые принципы работы цифровых устройств.

Особенности работы с ПЛИС

В процессе изучения System Verilog и разработки цифровых устройств можно выделить несколько особенностей, которые отличают программирование на HDL от программирования на языках высокого уровня.

- Необходимость моделирования результатов. В отличие от программ, которые разрабатываются для электронно-вычислительных машин, программы на FPGA (Field-Programmable Gate Array, то же, что и ПЛИС) не дают возможности мгновенно увидеть результат работы. В некоторых случаях возможно провести моделирование в системе, в которой производится разработка программы (рис. 1), но большая часть проектов для просмотра итогов работы требует прошивки ПЛИС. Таким образом, для проектирования программы на ПЛИС недостаточно лишь среды разработки.



Рис.1. Моделирование результатов работы светофора в системе Quartus II

- Другой уровень абстракции. Если программисту не требуется вникать в сложную организацию устройства, для которого разрабатывается данная программа, то взаимодействие с ПЛИС возможно без знания основ схемотехники. Чтобы

результат работы при отладке проекта не стал неожиданностью, важно не только понимать, как запрограммировать логику устройства, но и представлять то, как это будет выглядеть со стороны платы.

- Программы для FPGA сложны в понимании. Из-за особенностей синтаксиса языков описания аппаратуры, для чтения кода требуется больше времени. Как было сказано выше, некоторые синтаксические конструкции нужно рассматривать со стороны взаимодействия элементов платы между собой, что занимает большой объем работы при изучении HDL.

- Разное видение реализации. При постановке какой-либо задачи программист старается представить ее в виде алгоритма, разработать его схему (рис.2), в то время как разработчик цифровых устройств представляет это в виде диаграммы переходов состояний (рис. 3).

Ниже представлен пример того, как одна и та же задача, в данном случае – автомобильный светофор, может быть представлена в двух формах.

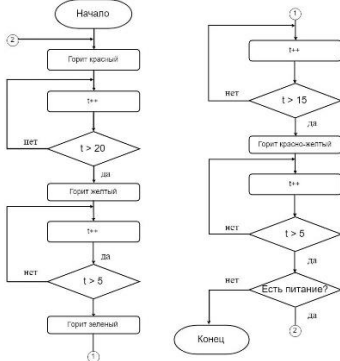


Рис. 2. Схема алгоритма работы светофора

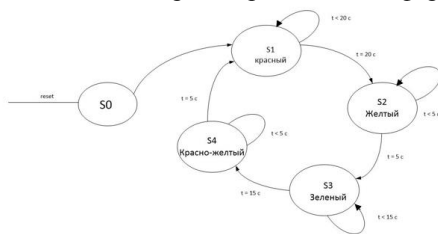


Рис. 3. Диаграмма переходов состояний автомобильного светофора

- Схематизация. Неотъемлемой частью разработки цифрового устройства на ПЛИС является проектирование его структурной схемы. Она определяет основные функциональные части устройства, их назначение и взаимосвязи между ними, а также наглядно отображает принцип действия устройства в общем виде. На рис.4 представлена структурная схема автомобильного-пешеходного светофора с кнопкой, на которой сплошной линией обозначается передача данных, а пунктирной — управляющих сигналов.

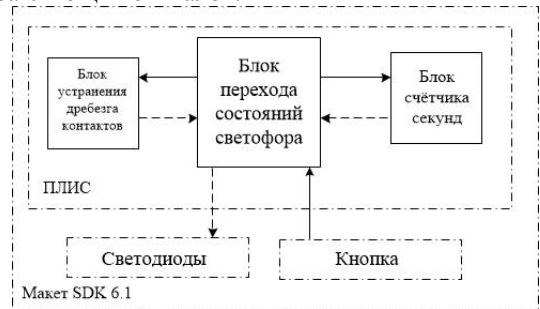


Рис. 4. Структурная схема светофора с кнопкой

Заключение

Таким образом, в этой статье были выявлены основные особенности разработки цифрового устройства на ПЛИС со стороны программиста, а также были рассмотрены отличия программирования на языках описания аппаратуры от программирования на высокоуровневых языках.

Список использованных источников

1. Дэвид М. Харрис, Сара Л. Харрис Цифровая схемотехника и архитектура компьютера. - 2 изд. Morgan Kaufman, 2013.
2. Хоровиц П., Хилл У. Искусство схемотехники. - 6 изд. - Москва: Мир, 2003 – 153-155 с.