

Разработка проекта ПЛИС FPGA модуля прототипа микропроцессора

А.Н. Петров¹, С.В. Юрлин^{1,2}

¹АО «МЦСТ», ²ПАО «ИНЭУМ им И. С. Брука», Sergey.V.Yurlin@mcst.ru

Аннотация — в данной статье рассмотрен процесс разработки проекта ПЛИС для специализированного прототипа микропроцессора на основе микросхем FPGA Intel Stratix 10 GX и Xilinx Virtex UltraScale. Обозначены основные проблемы эмуляции устройства на нескольких ПЛИС и способы их решения на примере модулей форм-фактора КУБ-ПРО, применяемого в АО «МЦСТ». Рассмотрена задача размещения сигналов на матрице выводов ПЛИС, которая необходима для разработки модуля и напрямую связана с разработкой проекта.

Ключевые слова — эмуляция, прототип, ПЛИС, микропроцессор, КУБ-ПРО.

I. ВВЕДЕНИЕ

Применение специализированных прототипов на основе ПЛИС является одним из общепринятых методов [1] аппаратной верификации микропроцессоров (МП). Описание высокопроизводительного микропроцессора на языке регистровых передач зачастую сильно превышает доступную емкость любой ПЛИС, поэтому прототип должен состоять из нескольких микросхем. Исходное RTL-описание микропроцессора разбивается на блоки логики, соединенные между собой логическими связями. Каждый блок логики размещается на одной ПЛИС. Данный подход к организации прототипа ставит перед разработчиком проекта ПЛИС ряд проблем:

1. Количество логических связей между двумя блоками логики в большинстве случаев превышает количество доступных для соединения с соседней ПЛИС контактов.
2. Необходимо сохранять потактовую целостность RTL-модели в распределённой системе эмуляции путём своевременной передачи состояний логических связей, установленных между полученными после деления блоками логики.
3. Каждой ПЛИС прототипа должен соответствовать проект САПР, определяющий назначение сигналов на матрице выводов, используемые типы интерфейсов и особенности загружаемых блоков логики. В худшем случае их количество равно количеству применяемых ПЛИС, что приводит к резкому увеличению трудоёмкости и соответственно времени разработки.

Сложность данных проблем зависит от особенностей реализации конечной системы. Унификацию проектов ПЛИС и, соответственно, уменьшение сложности проектирования прототипа можно достичь при следовании принципам КУБ-ПРО [2]. В этом случае в качестве базового элемента можно выделить модули эмуляции, на которых расположена одна ПЛИС с фиксированным набором интерфейсов. Используемые интерфейсы можно разделить на две группы по назначению. Первая - интерфейсы прототипа, предназначенных для передачи состояний большого количества логических связей блоков RTL-модели МП между ПЛИС. Вторая - применяется непосредственно для эмуляции интерфейсов самого МП. Для унификации работ по разработке проектов ПЛИС FPGA и топологии таких модулей требуется выполнение следующих задач:

1. Разработать иерархию проекта, позволяющую адаптировать проект для различных ПЛИС в однотипных модулях прототипа.
2. Разработать модульное описание линий связи прототипа, предназначенных для передачи состояний логических связей между блоками RTL-модели МП.
3. Разместить сигналы на матрице выводов ПЛИС с учетом требований конструкции модуля.

Одним из факторов, который усложняет решение данных задач, является редкость публикаций на данные темы. Большая часть работ в этой области посвящена топологии системы и общему маршруту проектирования, изредка с упоминанием способов решения задачи размещения сигналов на выводах ПЛИС [3]. В статье приведены особенности решения указанных задач для проектов ПЛИС FPGA модулей эмуляции, выполненных в форм-факторе КУБ-ПРО, с учётом использования микросхем Intel Stratix 10 GX и Xilinx Virtex Ultrascale.

II. СТРУКТУРА ПРОЕКТА И ПОРЯДОК РАЗРАБОТКИ

Проекты ПЛИС FPGA имеет смысл разделить на два разных типа: базовые и результирующие. Результирующие проекты содержат блок логики микропроцессора, иначе говоря, предназначены для эмуляции поведения части конечного устройства. Их количество может быть большим, вплоть до значения равного числу ПЛИС в прототипе. Суть базовых проектов заключается в тестировании высокочастотных схем аппаратуры прототипа и

подготовке шаблона для создания результирующих проектов. Это особенно важно при введении в эксплуатацию впервые разработанных модулей, работоспособность которых не подтверждена, и при проведении работ по распределению блоков RTL-модели МП по ПЛИС с целью создания распределённой системы эмуляции. Базовый проект универсален для конкретной модели ПЛИС, используемых в модулях эмуляции. Фактически основным отличием базового проекта от результирующего является наличие генератора тестовых данных вместо блока логики. Тем не менее, путем выделения общих частей двух типов проектов достигается унификация, обеспечивающая распараллеливание процессов проектирования.

Учитывая вышесказанное можно выделить следующий порядок разработки проекта модуля прототипа: сначала разрабатывается базовый проект, который выполняет две цели: назначение сигналов на матрице выводов ПЛИС и создание модулей интерфейсов межмодульной связи. Первое необходимо для разработки модуля эмуляции как физического устройства, отладки RTL-описания контроллеров линий связи прототипа и после монтажа модуля, для проверки работоспособности реализованного устройства. После этого создаются необходимые результирующие проекты. Это происходит путём замены блоков генерации данных базового проекта блоками логики, полученными в процессе разделения RTL-описания верифицируемого микропроцессора, с учётом ассоциации их портов с портами в проектах ПЛИС.

Абстрагируясь от различий блока логики и блока генерации тестовых данных можно определить общую структуру проекта (рис. 1). Она позволяет унифицировать общие для всех результирующих проектов части и вносить изменения проекта, связанные с модулем прототипа, только в базовом проекте, а не во всех результирующих. Поэтому в дальнейшем иерархия рассматривается для базового проекта модулей эмуляции в форм-факторе КУБ-ПРО.

Таким образом, разработка проекта ПЛИС начинается с выбора целевой ПЛИС, которая будет применяться в модуле. Данный выбор осуществляется исходя из нескольких факторов:

1. Максимальная логическая емкость – чем больше логическая емкость ПЛИС, тем меньше требуется микросхем для системы эмуляции, тем надежнее и дешевле будет прототип.
2. Поддержка необходимых интерфейсов. Целевая ПЛИС должна поддерживать применяемые в проекте интерфейсы.
3. Количество доступных для назначения контактов. Их должно быть достаточно, чтобы разместить необходимые интерфейсы, но при этом необходимо учитывать их взаимное расположение, так как это напрямую влияет на процесс трассировки печатной платы модуля.

Затем для выбранной ПЛИС создается базовый проект. Порядок при этом следующий. Сначала необходимо создать корневой модуль проекта. Он содержит в себе объявления внешних портов, которые будут размещены на матрице выводов целевой ПЛИС. Помимо этого в нем впоследствии располагаются два основных блока проекта: модуль логики и модуль интерфейсов, а также необходимые примитивы для конфигурации сигналов, зависящие от применяемой САПР. Например, в проекте САПР Xilinx Vivado необходимо явно задавать примитивы буферов дифференциальных сигналов и буферов сигналов тактирования трансиверов.

Для размещения сигналов на матрице выводов ПЛИС необходимо осуществить конфигурацию IP-блоков используемых интерфейсов прототипа или микропроцессора. Так средствами САПР можно проверить корректность размещения сигналов, подключенных к IP-ядрам. Для выполнения этой задачи необязательно реализовывать всю иерархию. Определение назначения сигналов на матрице выводов даст возможность разработать плату модуля параллельно с разработкой проекта, что значительно сократит время разработки прототипа в целом. Только после этого можно начинать реализовывать остальные уровни иерархии. Стоит помнить, что на конечном этапе разработки топологии многослойных печатных плат возникает необходимость обратной связи с проектом ПЛИС для проверки сделанных с целью удобства трассировки взаимных перемещений сигналов внутри групп на матрице выводов микросхемы.

Модуль интерфейсов содержит в себе модуль синхронизации, управляющий частотой эмуляции, и модули всех доступных в проекте данной микросхемы линий связи, составляющих основу интерфейсов прототипа в ПЛИС. Линии связи предназначены для передачи состояний логических связей между блоками RTL-модели МП и состоят из совокупности каналов связи. Каждый канал определяется набором физических интерфейсов одного типа. В общем случае линия связи может состоять из каналов, основанных на разных типах физических уровней. Но для простоты отладки и скорости разработки следует использовать каналы одного типа. Состав в случае модуля КУБ-ПРО на основе обеих ПЛИС следующий: 4 модуля линий связи интерфейса прототипа типа А, 4 модуля линий связи интерфейса прототипа Б и 2 IP-ядра DDR4[2]. При этом на одну линию приходится один канал связи.

Из-за большой ширины логических связей между двумя блоками логики, состояния которых необходимо передавать между ПЛИС, на уровне канала требуется двойное параллельное-последовательное преобразование. Поэтому модуль канала связи состоит из двух частей: модуля сериализации-десериализации (SERDES) и модуля физического уровня, включающего IP-блоки ПЛИС. Это обусловлено ограниченным коэффициентом сериализации последних. Модуль SERDES универсален для применяемых каналов типа А и типа Б и выполняет

последовательно-параллельное преобразование для принятых данных и параллельно-последовательное преобразование для отправляемых.

Модуль физического уровня отвечает за работу уже с IP-ядрами конкретных интерфейсов, применяемых в канале. Данный модуль состоит из блока тренировки, выполняющий функции первоначальной настройки интерфейсов по включению питания или сбросу путем реализации алгоритма автокалибровки уникального для каждого интерфейса (как следствие данный модуль разный для разных интерфейсов) и IP-блоков применяемого интерфейса. В общем случае это приемник и передатчик, однако, могут быть вариации в зависимости от логики работы интерфейса. Например, для высокоскоростных приемопередатчиков в Intel Stratix 10 GX предусмотрен один блок, объединяющий приёмник и передатчик, а также требуется отдельные IP-блоки для генерации синхросигналов и сбросов.

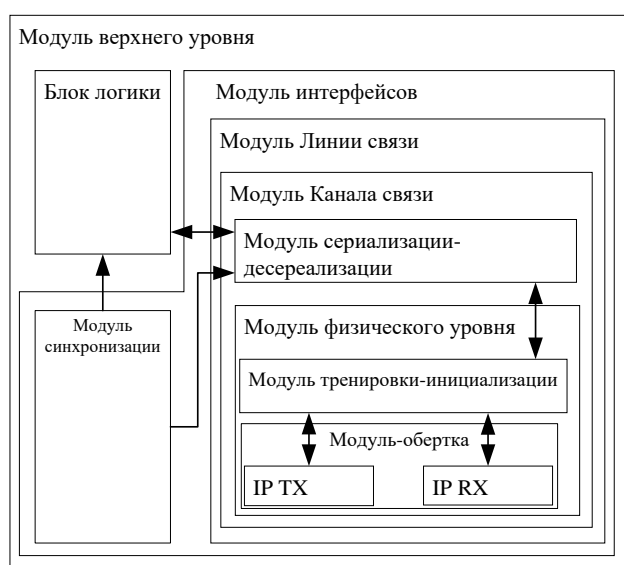


Рис. 1. Иерархия модулей проекта

В приведённой структуре следует выделить три домена синхронизации: домен эмуляции, домен приемника и домен передатчика. Последние два индивидуальны для каждого канала. Домен эмуляции один для всего проекта ПЛИС. Модуль SERDES соединяет в себе все домены синхронизации линии связи, так как именно он преобразует данные от всех источников данных: модуля логики, приемников и передатчиков. Важно отметить, что у модуля есть возможность тактировать блок логики синхросигналом с другого модуля. Таким образом, домен эмуляции может быть один для блоков логики, расположенных в разных ПЛИС на разных модулях.

III. ИНТЕРФЕЙСЫ МОДУЛЯ ПРОТОТИПА

Для связи между ПЛИС необходимо применять специальные интерфейсы. В форм-факторе КУБ-ПРО они разделены по применению на два типа: тип А и тип Б. В модуле КУБ-ПРО для интерфейса типа А

применяется физический интерфейс LVDS с шириной шины x16 и скоростью 1,2 Гб/с по каждой линии, а для интерфейса типа Б применяются высокоскоростные приемопередатчики (ВСПП) с шириной шины x4. Скорость линий у последних может изменяться в широких диапазонах. Но с учётом необходимости длительной бесперебойной работы непосредственно физических уровней ВСПП без развёрнутых логических уровней, обеспечивающих переповторы и коррекцию ошибок, разумно ограничить скорость передачи. При значении в 5 Гб/с пропускная способность линий интерфейсов обоих типов становится сопоставима. Однако ВСПП имеют существенно большую транспортную задержку из-за сложности аналоговой аппаратной составляющей. По этой причине для передачи данных между блоками логики в качестве основного применяется интерфейс типа А. Тип Б – для реализации пакетных интерфейсов или при передаче данных, где транспортная задержка допустима [2]. Так на основе интерфейса типа Б можно реализовывать высокоскоростные интерфейсы типа PCIe, SATA, IPLink и другие.

Для работы с интерфейсами микропроцессора используются контроллеры, реализуемые непосредственно в RTL-модели МП. Их представление должно быть функционально верифицировано в составе прототипа. В отличие от них контроллеры интерфейсов прототипа описываются целиком в модуле интерфейсов. Принципы их работы стоит описать подробнее.

В предыдущем разделе было сказано, что алгоритм автокалибровки зависит от применяемого интерфейса и, как следствие, от этого зависит наполнение блока тренировки. Проиллюстрируем алгоритм тренировки интерфейса типа А, применяемого в КУБ-ПРО. В IP-ядрах LVDS используется механизм автоподстройки частоты, в приёмнике ещё и механизм цифрового выравнивания фаз для входящего сигнала. Алгоритм тренировки делится на 3 фазы: тренировка блока фазовой автоподстройки частоты (PLL), блока цифрового выравнивания фаз (DPA) и блока скользящего битового окна (Bitslip).

Основная задача блока PLL – тактирование приемника и передатчика. Первая фаза проходит без участия модуля, он ожидает сигнала готовности, после чего переходит на вторую фазу.

Основная задача блока DPA – обнаружение битового потока на интерфейсных линиях. Данный блок необходим так как для передатчика и приемника используются разные источники синхросигнала, вследствие чего может возникать разность фаз. В течение тренировки DPA передатчик передает первую тестовую последовательность, состоящую из чередующихся нулей и единиц на вход соседнего модуля. Соседний модуль выполняет ту же операцию. Приёмник при этом считает количество принятых последовательностей, совпадающих с эталонной. В случае успешного приема счетчик увеличивается на единицу и по принятию порогового количества

последовательностей тренировка считается успешной, после чего модуль переходит в третью фазу. Если последовательность не совпала с эталоном, то счетчик не увеличивается. Если же время тренировки истекло или произошла рассинхронизация блока PLL, то процесс тренировки останавливается, и модуль не выполняет дальнейших действий.

Основная задача блока BitSlip – выделять логические слова в потоке бит. В ходе третьей фазы передатчик отправляет вторую тестовую последовательность. Для тренировки обнаружения логических слов необходимо, чтобы тестовая последовательность бит не преобразовывалась сама в себя при сдвиге, иначе говоря, любую аномалию в потоке можно было зафиксировать (например, повторяющиеся 6 бит 101001). В остальном эта фаза совпадает с второй фазой. Алгоритм приведен на рис. 2.

Алгоритм тренировки интерфейса типа Б практически идентичен алгоритму тренировки интерфейса типа А за исключением отличий в работе и возможностях разных IP-блоков. Также различаются верхние параметры фаз алгоритма, так как они имеют разную опорную частоту.

Важно отметить, что в процессе разработки базового проекта для ПЛИС FPGA Xilinx Virtex Ultrascale пришлось вводить дополнительный модуль-обертку IP-блоков. Данное решение продиктовано особенностями IP-ядер Xilinx LVDS и ВСПП: эти IP-ядра являются контроллерами банков ПЛИС, которые реализуют выбранный интерфейс. Получается, что каждому каналу необходимо свое IP-ядро. Модуль-обертка указывает, какое именно ядро или пару ядер выбрать в зависимости от того, в каком канале находится данный модуль. В проекте для Intel Stratix 10 GX данный модуль не требуется, так как IP-ядра не привязаны к конкретным банкам. Структура модуля-обертки приведена на рис. 3 для канала интерфейса типа А и на рис. 4 для канала интерфейса типа Б.

Для верификации контроллеров интерфейсов микропроцессора необходимо чтобы эти интерфейсы физически присутствовали на модуле прототипа и реализованы в проекте. В отличие от интерфейсов прототипа контроллеры интерфейсов содержатся в RTL-модели микропроцессора, следовательно, в модуле интерфейсов должны располагаться только физические уровни данных интерфейсов. Задача базового проекта – тестировать физический уровень ПЛИС, используемый для реализации интерфейсов микропроцессора, и полученную топологию линий интерфейсов на модулях эмуляции.

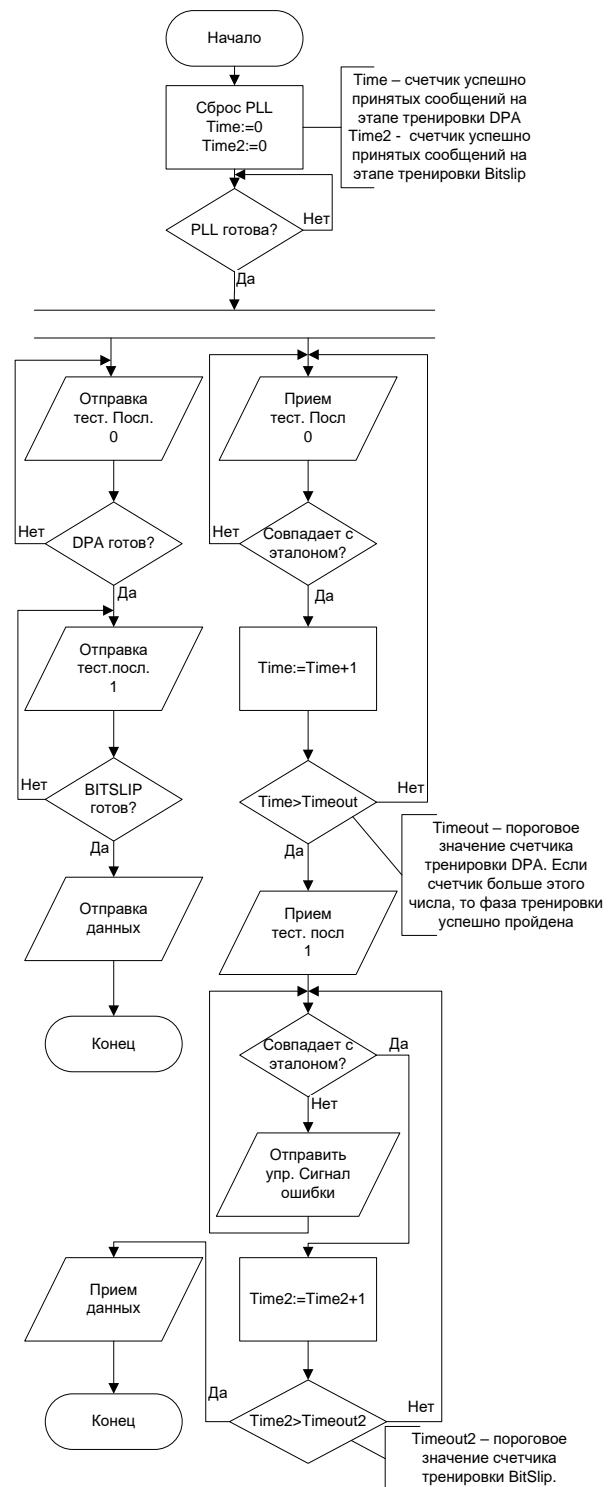


Рис. 2. Алгоритм тренировки IP-ядра LVDS

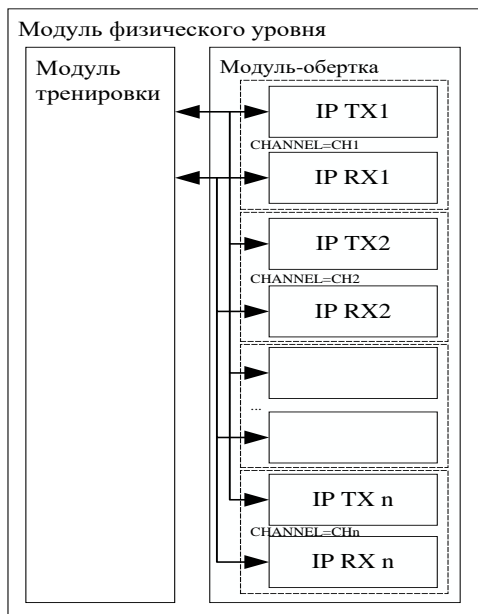


Рис. 3. Модуль-обертка IP-ядер LVDS

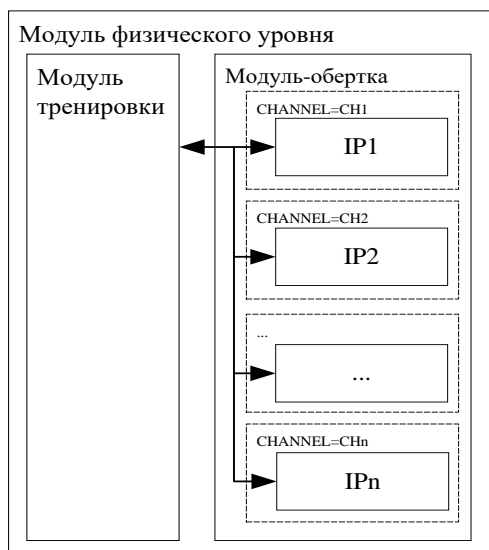


Рис. 4. Модуль-обертка IP-ядер ВСПП

Например, на модуле КУБ-ПРО есть два слота DDR4. Необходимые IP-блоки физического уровня в этом случае расположены в модуле интерфейсов, но для них не реализуются модули линий и каналов связи. В отличие от интерфейсов типа А и Б здесь есть встроенный алгоритм калибровки физического уровня. Он интегрально проверяет работоспособность интерфейса взаимодействия с модулем памяти, установленным в слот. При успешном прохождении данного теста выдаётся соответствующий сигнал. В случае наличия проблем для интерфейсов такого уровня в САПР есть диагностический инструмент для поэтапной отладки.

Такие интерфейсы как RS-232, HDA, USB3.0 и EZ-USB SX2, не требующие использования в ПЛИС специализированных физических уровней, не имеют встроенных средств тестирования. Проверка их

реализации в модулях возможна либо при эмуляции RTL-модели МП с его контроллерами интерфейсов, либо путём создания отдельных контроллеров, которые будут удаляться при переходе от базовых проектов к результирующим.

IV. НАЗНАЧЕНИЕ СИГНАЛОВ НА МАТРИЦЕ ВЫВОДОВ

Задача размещения сигналов на матрице выводов ПЛИС необходима для проектирования печатной платы модуля прототипа. В общем случае эта нетривиальная задача так как она зависит от выбранной ПЛИС, применяемой САПР, набора и параметров интерфейсов. Несмотря на эту неоднозначность можно выделить общий маршрут размещения сигналов:

1. Разместить сигналы интерфейсов с жесткими ограничениями по расположению, использующих несколько банков ПЛИС (например, DDR4).

2. Разместить сигналы интерфейсов с жесткими ограничениями в рамках одного банка (например, LVDS, ВСПП).

3. Разместить сигналы интерфейсов с ограничениями по напряжению питания (RS-232, HDA, SPI, EZ-USB SX2).

4. Разместить все остальные сигналы.

На каждом этапе нужно учитывать геометрические требования модуля – относительное расположение и размер соединителей соответствующих интерфейсов, возможности вывода из под BGA корпусов ПЛИС.

Интерфейс DDR4 имеет различные требования в зависимости от применяемой ПЛИС. Например, ПЛИС FPGA Xilinx Virtex UltraScale XCVU440 состоит из трех кристаллов. Для размещения DDR4 на этой ПЛИС необходимо, чтобы все сигналы находились в пределах одной половины кристалла [4]. Также необходимо учитывать, что в пределах одного банка можно располагать интерфейсы только с одинаковым напряжением питания [5]. Для шин DDR4 данное напряжение равно 1.2 В. А, например, для LVDS – 1.8 В. Это накладывает дополнительные ограничения на размещение.

Интерфейсы типа А и Б состоят из следующих сигналов: шины сериализованных данных, линий тактирования, сигналы сброса логики и интерфейсов прототипа, сигнала присутствия и сигналов интерфейса JTAG. Шины сериализованных данных и сигналы синхронизации - дифференциальные, поэтому они занимают в два раза больше выводов, чем собственная указанная ранее ширина. Сигналы одного IP-блока LVDS, применяемого в интерфейсе типа А, должны находиться в пределах одного банка, но могут располагаться практически в любом из банков общего назначения ПЛИС. Сигналы одного IP-блока ВСПП, применяемого в интерфейсе типа Б, должны располагаться в области трансиверов FPGA, но не обязаны находиться в одном банке. На практике один канал редко размещают за пределами одного банка из-

за возможных проблем тактирования, вызванных транспортными задержками сигналов, передаваемых между банками. Особенно это актуально при высокой плотности сигналов или реализации высокоуровневых интерфейсов.

Размещение высоковольтных сигналов зачастую реализовано в современных ПЛИС при помощи выделения специальных банков, поддерживающих данную возможность. Под высоковольтными сигналами подразумеваются сигналы, имеющие уровень логической единицы выше напряжения периферийного питания общих банков ПЛИС. Например, для внешних физических уровней интерфейсов RS-232, HDA, SPI и EZ-USB SX2, расположенные на модуле КУБ-ПРО, уровень логической единицы составляет 2,0 В. У XCVU440 всего два банка поддерживающих опорные напряжения выше 1.8. Очень часто высоковольтные банки меньше обычных. Следствие данного факта – возможная нехватка места для высоковольтных интерфейсов. Одно из возможных решений данной проблемы – реализация интерфейса в обычном банке на стандартном уровне логической единицы ПЛИС с последующим преобразованием напряжения сигнала к нужному значению аппаратными средствами, расположенными на модуле. Разумеется, данный подход негативно скажется на задержке распространения сигналов данного интерфейса и потенциально снижает надёжность системы.

V. ЗАКЛЮЧЕНИЕ

В статье рассмотрены основные проблемы разработки проекта ПЛИС FPGA модуля специализированного прототипа микропроцессора. Основные проблемы данного подхода заключаются в большой размерности логических связей между блоками логики, необходимость передавать данные в пределах одного такта эмуляции и необходимость разработки нескольких проектов для набора ПЛИС.

Способы их решения сильно зависят от устройства специализированного прототипа. В данной статье приведены решения для системы матричной масштабируемой эмуляции, устроенной по модульному принципу. Согласно данному принципу описание процессора разбивается на блоки логики, каждый из которых размещается на одном модуле системы. Модуль применяемой в рамках обозначенной выше системы эмуляции имеет форм-фактор КУБ-ПРО, согласно которому на модуле располагается одна ПЛИС FPGA с подключенными к ней интерфейсами и подсистемами.

Проблема большой размерности логических связей между блоками решена путем двойной сериализации-десериализации данных, что позволяет передавать данные большой размерности через интерфейсы сравнительно небольшой размерности.

Проблема передачи данных решена применением для основного интерфейса передачи данных между модулями стандарта LVDS с разрядностью данных 16

бит, аппаратная реализация которого имеет малые задержки и имеет приемлемую пропускную способность. Это позволяет не только передавать данные за один такт эмуляции, но и снизить длительность такта, что ведет к росту скорости эмуляции.

Проблема разработки нескольких проектов для набора ПЛИС решена посредством введения понятия базового проекта, который единственен для модуля одного типа. Данный проект имеет отдельный модуль, в котором выделены все общие части результирующих проектов, что позволяет создавать результирующие проекты путем замены генератора данных на блок логики.

С учетом решения данных проблем составлен маршрут разработки проекта для специализированного модульного прототипа форм-фактора КУБ-ПРО. Согласно этому маршруту сначала создается корневой модуль проекта, в котором объявляются все сигналы и размещаются все IP-блоки, потом IP-блоки выделяются в модуль интерфейсов, на верхнем уровне которого находятся IP-ядра физических уровней интерфейсов микропроцессора и блоки линий связи модулей. Линии связи в свою очередь наполняются каналами связи, каналы – модулями физических уровней, в которых размещаются IP-блоки соответствующих интерфейсов.

Помимо этого рассмотрена задача размещения сигналов модуля прототипа на матрице выводов ПЛИС FPGA, решаемая на первом этапе маршрута проектирования. Составлен общий алгоритм размещения сигналов и приведен пример применения данного алгоритма для модуля КУБ-ПРО на основе ПЛИС FPGA Xilinx Virtex UltraScale и Intel Stratix 10 Intel Stratix 10.

ЛИТЕРАТУРА

- [1] Amos D., Lesea A., Richter R. FPGA-Based Prototyping Methodology Manual ISBN: 978-1617300042
- [2] С.В. Юрлин. Универсальный подход к построению масштабируемых прототипов многоядерных микропроцессоров (КУБ-ПРО). // Вопросы радиоэлектроники. –2018. – №3. – Сер. ЭВТ. – Выпуск 2. – С.93-98.
- [3] Khalid M. Routing architecture and layout synthesis for multi-FPGA systems // Торонто, 1999. URL: http://www.collectionscanada.gc.ca/obj/s4/f2/dsk1/tape8/PQDD_0004/NQ41187.pdf
- [4] Xilinx Inc. UltraScale Arcgutecture-Based FPGAs Memory IP v1.4 // Xilinx Inc. 2017. URL: https://www.xilinx.com/support/documentation/ip_documentation/ultrascale_memory_ip/v1_4/pg150-ultrascale-memory-ip.pdf (дата обращения: 20.03.2018)
- [5] Xilinx Inc. UltraScale and UltraScale+ FPGAs Packaging and Pinouts // Xilinx Inc., 2017. URL: https://www.xilinx.com/support/documentation/user_guides/ug575-ultrascale-pkg-pinout.pdf (дата обращения: 20.03.2018)

Development of FPGA Project for Microprocessor Prototype Module

A.N. Petrov¹, S.V. Yurlin^{1,2}

¹JSC «MCST», ²PJSC «INEUM named after I.S. Bruk», Sergey.V.Yurlin@mcst.ru

Abstract — This article is an overview of an FPGA project development for a module of distributed hardware emulation system used by JSC «MCST» with use of Intel Stratix 10 GX and Xilinx Virtex UltraScale architectures. The employment of multi-FPGA hardware emulation approach raises several problems: the necessity to transfer significant volumes of data between modules, the requirement to finish the transfer within one emulation cycle and need to create several FPGA projects for different module functionality. Usage of binary serialization of data solves the problem of big volume transfer: in designated SerDes of one inter-FPGA line and in transmitter IP-core of an interface. The requirement of one emulation cycle transfer is adhered by using 16-bit LVDS interface, the hardware implementation of which has low processing latency and acceptable bandwidth. The multi-project problem is solved via the introduction of the base project, which is universal for all modules with the same FPGA and has all common modules for every project: interface IP-cores and inter-FPGA line controllers. This project acts as a template for further development of projects for modules of desired functionality. The task of creating pin assignments is overviewed with creating a general workflow and examples for FPGAs mentioned above.

Keywords — emulation, prototype, FPGA, microprocessor, KUB-PRO.

REFERENCES

- [1] Amos D., Lesea A., Richter R. FPGA-Based Prototyping Methodology Manual ISBN: 978-1617300042
- [2] S.V. Yurlin Universal'nyj podhod k postroeniyu masshtabiruemyh prototipov mnogoyadernyh mikroprocessorov (KUB-PRO). (A universal approach to building scalable prototypes of multi-core microprocessors (KUB-PRO)) // Voprosy radioelektroniki / № 3, ser. EVT, M., 2018, pp. 93-98. (in Russian).// Вопросы радиоэлектроники. –2018. – №3. – Сер. ЭВТ. – Выпуск 2. – С.93-98.
- [3] Khalid M. Routing architecture and layout synthesis for multi-FPGA systems // Toronto, 1999. URL: http://www.collectionscanada.gc.ca/obj/s4/f2/dsk1/tape8/PQDD_0004/NQ41187.pdf
- [4] Xilinx Inc. UltraScale Arcgutecture-Based FPGAs Memory IP v1.4 // Xilinx Inc. 2017. URL: https://www.xilinx.com/support/documentation/ip_documentation/ultrascale_memory_ip/v1_4/pg150-ultrascale-memory-ip.pdf (дата обращения: 20.03.2018)
- [5] Xilinx Inc. UltraScale and UltraScale+ FPGAs Packaging and Pinouts // Xilinx Inc., 2017. URL: https://www.xilinx.com/support/documentation/user_guides/ug575-ultrascale-pkg-pinout.pdf (дата обращения: 20.03.2018)