

Импортозамещающая технология ПЛИС-БМК

Часть II. Перевод проектов ПЛИС в полузаказные БИС по технологии ПЛИС-БМК

Владимир Евстигнеев,
д. т. н.,
Александр Кошарновский,
к. т. н.

evstigneev_mniti@pochta.ru

Евгений Дегтярев,
к. т. н.,
Михаил Критенко,
к. т. н.,
Сергей Цыбин,
Алексей Быстрицкий

Состояние разработок в области систем перевода проектов, отложенных на ПЛИС, в проекты БИС

Разработанная авторами в конце 90-х годов система «Транслятор» является, пожалуй, одним из первых в России средств перевода проектов зарубежных ПЛИС в проекты БИС. В системе используется лицензионное программное обеспечение фирмы ALTERA — PLS-FIXEDPC. Программное обеспечение PLS-FIXEDPC состоит из двух независимых ядер: Max+Plus II, которое рассчитано на проектирование изделий емкостью до 250 тыс. вентиляей, и Quartus, которое рассчитано на проектирование изделий емкостью до 1,5 млн вентиляей. Система Max+Plus II полностью перекрывает требования для проектирования по технологии ПЛИС+БМК сложностью до 30 тыс. вентиляей и была взята за основу при создании системы «Транслятор».

Max+Plus II позволяет вводить схему в графическом виде на основе символов библиотечных элементов БМК, компилировать ее и моделировать проект с задержками, получаемыми при реализации на ПЛИС. Max+Plus II поддерживает следующие семейства ПЛИС: FLEX10K, FLEX10KA/B/E, FLEX8000, FLEX6000, MAX9000, MAX3000A и др.

Полный список поддерживаемых ПЛИС содержится в help-файле системы Max+Plus II.

Схема разработки проекта БИС с использованием системы «Транслятор» представлена на рис. 1. В основе подхода лежит использование графического редактора и симулятора Max+Plus II для создания и моделирования проекта БИС на БМК.

Такая реализация стала возможной благодаря использованию подсистемы перевода графической информации, использующей внутренние форматы графического редактора и симулятора. При трансляции проекта одновременно проводится проверка на правила проектирования БИС на БМК.

Входными данными для системы «Транслятор» является проект, введенный в графическом виде в системе MAX+Plus II, с использованием специально разработанной библиотеки элементов, настроенных для конвертирования. В результате работы создается файл для моделирования в системе MAX+Plus II. Используя этот файл, можно провести сравнительное моделирование проектов, реализованных на ПЛИС и БМК.

Основные недостатки системы «Транслятор»:

- Система базируется на САПР ПЛИС MAX+Plus II фирмы ALTERA и не обеспечивает перевод проектов из ПЛИС других фирм. Это является существенным ограничением, так как разработчики РЭА используют ПЛИС и других фирм, таких, как Actel, Xilinx и др.
- В настоящее время разработчики РЭА при создании проектов ПЛИС интенсивно используют языки высокого уровня VHDL, Verilog HDL, AHDL, различные параметризованные модули, IP-блоки и генераторы функций, а система «Транслятор» допускает использование только универсальной библиотеки и позволяет создавать проект только в графическом виде. Это значительно ограничивает возможности разработчиков при создании проекта ПЛИС и БИС.
- Пожалуй, самым существенным ограничением системы «Транслятор» является то, что она ориентирована на конвертирование проектов ПЛИС в полузаказные БИС на основе БМК серии 1578XM. Емкость данных БМК не позволяет переводить в полузаказные БИС проекты ПЛИС сложностью свыше 12 тыс. вентиляей, что не удовлетворяет современным требованиям.

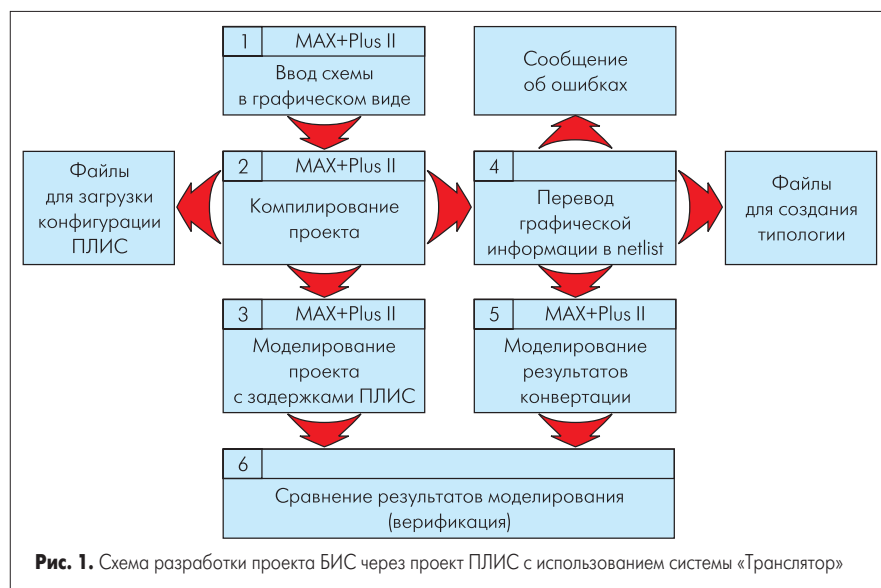


Рис. 1. Схема разработки проекта БИС через проект ПЛИС с использованием системы «Транслятор»

Таким образом, существует острая необходимость создания технологии проектирования ПЛИС-БМК, свободной от этих недостатков и обеспечивающей разработку БИС на БМК до 500 тыс. вентиля и более.

Кроме того, в настоящее время отечественной промышленностью разрабатываются и аттестуются новые технологии проектирования полужаказных БИС военного применения, в частности, на основе аттестованных библиотек стандартных элементов и блоков. Поэтому разрабатываемые элементы технологии проектирования должны предусматривать возможность проведения разработок БИС на основе стандартных элементов и блоков, а также возможность использования IP-блоков.

Для этого необходимо в первую очередь установить порядок взаимодействия разработчиков и изготовителей электронной аппаратуры и разработчиков и изготовителей электронных компонентов и разработать маршрут перевода проектов ПЛИС в БИС по технологии проектирования ПЛИС-БМК с распределением работ по этапам маршрута.

Разработка порядка взаимодействия разработчиков и изготовителей электронной аппаратуры и разработчиков и изготовителей электронных компонентов

Существующий ГОСТ 27394-87 устанавливает порядок разработки и распределения работ между предприятием-заказчиком (далее в тексте — заказчиком) и предприятием-исполнителем (далее в тексте — исполнителем) при разработке заказных и полужаказных интегральных микросхем. Распределение основных работ между заказчиком и исполнителем по этому ГОСТу при разработке и организации производства полужаказных БИС приведено в таблице 1.

Данное распределение не учитывает достижения современной технологии и уровень сложности проектов, поскольку процесс разработки топологии БИС существенно усложнился и требует специальных дорогостоящих аппаратных и программных средств, к тому же может осуществляться только на предприятии-изготовителе БИС или в специализированных центрах проектирования, разрабатывающих проекты для изготовителей БИС (кремниевых фабрик).

По экономическим соображениям нецелесообразно проводить изготовление опытных образцов БИС до проведения их испытаний в составе аппаратуры заказчика. В частности, ГОСТ 29110-91 допускает изготовление макетных образцов БИС до изготовления опытных образцов, тем самым давая возможность сократить затраты на дорогостоящие испытания.

В ГОСТ 27394-87 подразумевается использование унифицированных средств вычислительной техники и САПР как на предприятиях РЭА, так и разработчиками и изготовителями БИС. С точки зрения решаемых задач это неэффективно, поэтому в настоящее время на предприятиях РЭА используются мощные ПК под управлением опера-

Таблица 1

Основной вид работы по созданию полужаказных ИС	Заказчик	Исполнитель	Предложения по редакции
Подготовка заявки на разработку ИС	Ответственный	—	Исключить
Рассмотрение заявки на разработку ИС	Соисполнитель	Ответственный	Сохранить
Предоставление технической документации, необходимой для проектирования ИС	—	Ответственный	Только в части библиотек и РТМ
Разработка функциональных электрических схем	Ответственный	—	Сохранить
Разработка алгоритмов (тестов) проверки функционирования	Ответственный	—	Сохранить
Разработка топологии межсоединений	Ответственный	—	Редактировать
Разработка и изготовление машинных носителей топологии межсоединений на БК	Ответственный	—	Редактировать
Разработка карты заказа	Ответственный	Соисполнитель	Сохранить
Функциональное моделирование ИС (для цифровых ИС)	Ответственный	Соисполнитель	Сохранить
Проверка соответствия разработанной топологии ИС исходной функциональной схеме	Ответственный	Соисполнитель	Редактировать
Контроль информации на машинных носителях, разработанных заказчиком	Соисполнитель	Ответственный	Сохранить
Разработка и отладка программ параметрического и функционального контроля	Соисполнитель	Ответственный	Сохранить
Изготовление фотошаблонов (при использовании электронной литографии — ввод кодированной программы в технологическую установку электронной литографии)	—	Ответственный	Сохранить
Изготовление опытных образцов ИС	—	Ответственный	Сохранить
Исследование опытных образцов ИС (определение возможности их применения в РЭА)	Ответственный	—	Сохранить
Разработка эксплуатационной документации на ИС (при необходимости)	Соисполнитель	Ответственный	Сохранить
Разработка ТУ на ИС	Соисполнитель	Ответственный	Редактировать
Предварительные испытания опытных образцов	Соисполнитель	Ответственный	Редактировать
Изготовление и поставка опытной партии ИС	—	Ответственный	Сохранить
Организация серийного производства и обеспечение поставки ИС	—	Ответственный	Сохранить

ционной системы Windows и программные средства, ориентированные на разработку аппаратуры. Что касается дизайн-центров проектирования БИС и кремниевых фабрик, то здесь используются рабочие станции под управлением UNIX и дорогостоящие программные средства. Обмен данными осуществляется в стандартных форматах.

Разработка маршрута перевода проектов ПЛИС в полужаказные БИС по технологии проектирования ПЛИС-БМК

Одним из требований к технологии проектирования ПЛИС-БМК является возможность конвертирования проектов ПЛИС различных производителей, созданных с помощью различных средств ввода проекта (VHDL, Verilog HDL, AHDL, различные параметризованные модули, IP-блоки, генераторы функций). Если рассмотреть, например, САПР ПЛИС фирм Altera и Xilinx, то обнаружатся существенные отличия как в структуре проектных файлов, так и алгоритмов компиляции проектов. Но что объединяет САПР ПЛИС различных производителей, так это возможность вывода (экспорта) откомпилированного проекта в формате EDIF 2.0.0 (стандарт описания проектов электронных устройств). Таким образом, целесообразно использование именно этого формата как унифицированного средства представления проекта ПЛИС для конвертирования в полужаказные БИС. Кроме того, временная модель проекта ПЛИС может быть представлена на выходе САПР ПЛИС в виде «овремененного» описания на Verilog HDL. На основе такого подхода был разработан маршрут для проектирования по технологии ПЛИС-БМК,

представленный на рис. 2. Данный маршрут разработан на основе программных средств от Cadence, поскольку все ведущие дизайн-центры и производители БИС используют именно эти программные продукты.

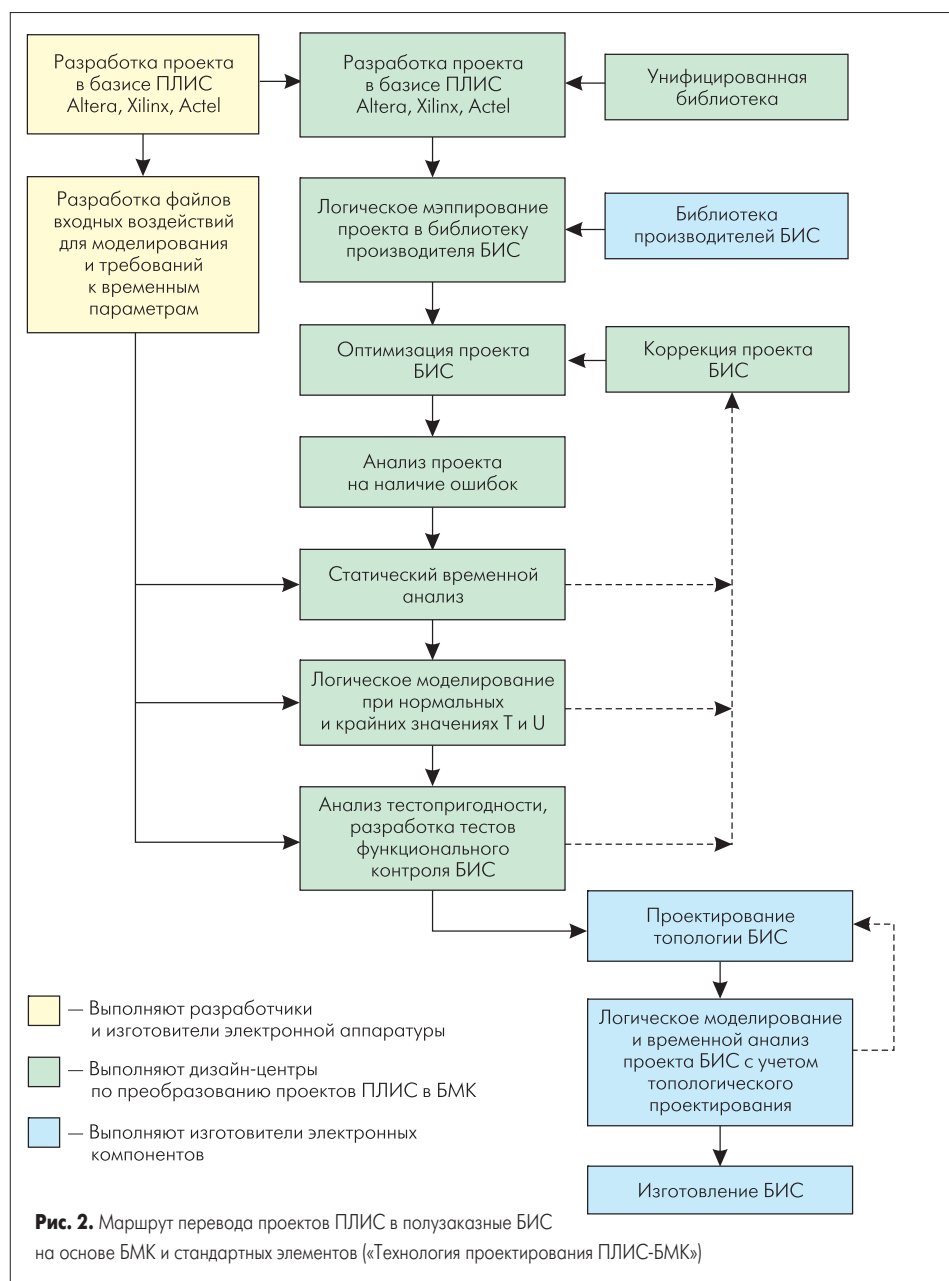
Используемые библиотеки

В основу маршрута положен принцип «мэппирования» или переназначения библиотек. Ядром САПР перевода проектов является «Унифицированная библиотека» в виде стандартного набора комбинационных функций и триггерных (запоминающих) элементов, формируемая в базе данных системы проектирования. Данная библиотека разработана в ходе ОКР и позволяет преобразовать описание проекта ПЛИС EDIF 2.0.0 в целевую библиотеку конкретного производителя БИС с помощью программного продукта EMBIT BuildGates Synthesis.

Для встраивания унифицированной библиотеки в базу данных системы EMBIT BuildGates Synthesis выполнены следующие действия:

- Создана библиотека графических символов Symlib;
- Создана временная библиотека .ctlf;
- Разработаны сценарии на TCL (Tool Command Language), управляющие процедурой линкования (мэппирования) элементов временной базы данных и соответствующих графических символов.

Библиотека Symlib состоит из набора текстовых ASCII-файлов, содержащих описание графических символов и их портов для библиотечных элементов. К одному и тому же символу может линковаться как элемент из унифицированной библиотеки, так и элемент с такой же логической функцией



из библиотеки производителя БИС. ASCII-файл generic.sum содержит описание графических сегментов и портов для каждого символа унифицированной библиотеки.

Для создания временного представления унифицированной библиотеки разработан файл generic.tlf (timing library format). Этот файл содержит информацию о портах, минимальных внутренних задержках и выполняемой логической функции для каждого библиотечного элемента. Tlf-файл средствами Embit BuildGates Synthesis транслируется в базу данных BuildGates.

Линкование временного представления элементов унифицированной библиотеки и их графических символов управляется tcl-сценариями. Разрабатываются сценарии nullmap.tcl и mapping.tcl. Nullmap.tcl представляет собой карту соответствия библиотечных элементов и их графических символов. Mapping.tcl содержит правила, по которым это соответствие осуществляется.

В базу данных САПР были интегрированы целевые библиотеки БМК 1578ХМ производства НИИЭТ (Воронеж). В дальнейшем в базу данных САПР будут интегрированы библио-

теки БМК производства ОАО «НИИМЭ и Микрон», ОАО «Ангстрем», НТЦ «Элинс» (Зеленоград).

Для встраивания библиотеки БМК в базу данных системы BuildGates Synthesis выполняются следующие действия:

- Создается временная библиотека .ctlf для библиотеки производителя БИС;
- Дополняется библиотека графических символов Symlib;
- Разрабатываются сценарии на TCL, выполняющие процедуру линкования (мэппирования) элементов временной библиотеки производителя БИС и соответствующих графических символов.

Для создания временного представления библиотеки производителя БИС в базе данных BuildGates используется исходный TLF-файл (не компилированный) версии 4.3. Текстовый файл .tlf стандартными средствами BuildGates Synthesis должен быть загружен в базу данных. Временное представление библиотеки производителя БИС используется при оптимизации проекта и синтезе.

Унифицированная библиотека уже содержит набор стандартных графических символов, по-

этому нет смысла повторять их для библиотеки производителя БИС, достаточно установить соответствующие линки. Поскольку библиотеки производителя БИС могут содержать элементы, которые не входят в унифицированную библиотеку, библиотека Symlib может быть расширена дополнительными файлами с описанием недостающих символов.

Линкование элементов временной библиотеки производителя БИС и соответствующих графических символов управляется tcl-сценариями. Основная часть библиотеки линкуется к тем же графическим символам, что и унифицированная библиотека, остальные элементы — к дополнительным. Разрабатываются файлы соответствия по именам и по портам и правила, по которым это соответствие устанавливается.

Основные этапы маршрута

1. Трансляция проекта (netlist) в формат САПР БИС

Входные форматы: EDIF2.0.0.

Используемые программы: Embit BuildGates Synthesis.

Средствами Embit BuildGates Synthesis исходный netlist проекта в формате EDIF2.0.0 транслируется в базу данных BuildGates. Логические элементы исходного netlist при этом линкуются к элементам унифицированной библиотеки.

2. Логическое мэппирование проекта в библиотеку производителя БИС

Входные форматы: SDF.

Используемые программы: Embit BuildGates Synthesis.

Линкование (mapping) проекта в библиотеку производителя БИС проводится с использованием средств оптимизации программной системы BuildGates Synthesis и осуществляется следующим образом:

- Встроенная в базу данных BuildGates библиотека производителя БИС назначается как целевая библиотека для проведения операции remapping.
- В базу данных BuildGates Synthesis загружается SDF-файл для управления операцией remapping и оптимизации.
- В результате операции remapping и оптимизации образуется новый netlist, логические элементы которого линкуются к библиотеке производителя БИС.

3. Оптимизация проекта

Используемые программы: Embit BuildGates Synthesis.

Средствами BuildGates Synthesis должны быть установлены необходимые временные ограничения, определены критические пути и кластерные деревья.

Проводится оптимизация полученного netlist с учетом этих установок для улучшения быстродействия критических цепей.

4. Анализ проекта на наличие ошибок

Используемые программы: Embit BuildGates Synthesis, Envisia PKS.

Проводятся проверки check_netlist полученного проекта на наличие обрывов связей, включение рекурсивно повторяющихся модулей, обратных связей, неуправляемых pins и nets, pins

и nets, которые управляются несколькими источниками, неподключенных портов. В netlist вносятся соответствующие исправления.

5. Статический временной анализ

Используемые форматы: Verilog HDL, SDF, ctf.

Используемые программы: Veritime, Embit BuildGates Synthesis, CDC.

Проводится статический временной анализ проекта, переведенного на библиотеку производителя БИС (контроль времен установки T_{SU} и удержания данных T_H на входе триггеров, задержки появления сигналов относительно тактовых импульсов, задержки сигналов между регистрами и т. д.). Для проведения статического временного анализа с использованием программного продукта Veritime проект транслируется из базы данных BuildGates в формат Verilog HDL (gate level). Должен быть получен SDF-файл для проекта, переведенного в библиотеку производителя БИС.

6. Логическое моделирование

Используемые форматы: Verilog HDL, SDF, TLF, STL.

Используемые программы: Verilog-XL, CDC, STL.

Для логического моделирования должны быть разработаны тесты на STL-языке и средствами stl-интерпретатора перетранслированы в модули testfixture, представляющие собой наборы тестовых векторов, с помощью которых задаются воздействия в Verilog-XL-программной среде.

Для создания «эталонной» базы данных используются описание на Verilog HDL и SDF-файл проекта ПЛИС, а также модули testfixture. Моделирование проекта ПЛИС проводится при разных температурах и напряжениях, результаты моделирования должны сохраняться каждый раз в отдельном файле.

Логическое моделирование проекта, переведенного на библиотеку производителя БИС, проходит в несколько этапов:

- Проводится логическое моделирование проекта с использованием тех же, что и для ПЛИС, модулей testfixture. Результаты моделирования сравниваются с аналогичными результатами в «эталонной» базе данных с помощью программы simCompare;
- Проводится логическое моделирование проекта с использованием SDF-файла и Back Annotation. Используются те же, что и для ПЛИС, модули testfixture, температурные режимы и напряжения. По окончании каждого моделирования при заданных температурах и напряжениях результаты моделирования должны сравниваться с аналогичными результатами в «эталонной» базе данных с помощью программы simCompare.

7. Проектирование топологии БИС

Используемые форматы: Verilog HDL, TLF, SDF.

Используемые программы: Preview Gate Ensemble, VerilogIn, Verilog-XL.

- Проект из описания Verilog HDL (gate level) транслируется в базу данных DFII программной среды Cadence с помощью транслятора VerilogIn.
- В Verilog-XL проводится создание «эталонной» базы данных для проекта с использованием SDF-файла и модулей testfixture.

- В Preview проводится планировка проекта, то есть формирование иерархии, создание групп объектов, определение регионов для групп и отдельных объектов, анализ полученной планировки с точки зрения насыщения площади и межсоединений.
- Проводится разработка топологии для макроблоков. Для улучшения качества размещения и трассировки используются средства временного анализа, встроенные в Preview и SDF-файл.
- Проводится размещение проекта в целом, анализ этого размещения, трассировка.
- Для моделирования проекта в Verilog-XL с учетом паразитных емкостей и сопротивлений средствами Preview генерируется RSPF-файл (Reduced Standard Parasitic Format) и новый SDF-файл. Для получения нового SDF-файла необходимо наличие ctf-представления для библиотеки БМК.
- В Verilog-XL проводится моделирование проекта с использованием нового SDF-файла и Back Annotation. В качестве входных воздействий применяются те же модули testfixture, которые использовались при создании «эталонной» базы данных.
- С помощью программы simCompare проводится сравнение результатов моделирования для двух баз данных — «эталонной» и полученной при моделировании проекта с учетом паразитных сопротивлений и емкостей.

Унификация библиотек отечественных БМК

Унификация по составу библиотечных элементов, макро- и мегафункций в библиотеках

Как известно, проект, выполненный в САПР ПЛИС, на каком бы уровне он не был представлен, после компиляции разбивается на функции, реализуемые в пределах одного логического элемента ПЛИС, и может быть экспортирован в формате EDIF 2.0.0. Исходя из маршрута перевода проектов по технологии проектирования ПЛИС-БМК унификация библиотек достигается использованием «Унифицированной библиотеки» в виде стандартного набора комбинационных функций и триггерных (запоминающих) элементов, формируемого в базе данных, и обеспечивается «перенацеливанием» в целевую библиотеку конкретного производителя БИС средствами Embit BuildGates Synthesis.

Унификация по временным и частотным параметрам библиотечных элементов, макро- и мегафункций

Для решения проблем временной совместимости были разработаны и исследованы временные модели задержек для ПЛИС и полужаказных БИС, представленные на рис. 3–5.

Как было отмечено ранее, библиотеки, макро- и мегафункции или текстовые описания на языке высокого уровня, которые используют в своих проектах разработчики ПЛИС после компиляции проекта, разбиваются на функции, реализуемые в пределах одного логического элемента (как правило, функции четырех переменных). Это поясняет пример,

представленный на рис. 6. Исходная схема, представленная на рис. 6а, после компиляции проекта преобразуется в схему, изображенную на рис. 6б, элементы которой соответствующим образом размещаются в логических элементах (ЛЭ) ПЛИС. То же самое происходит, когда проект введен в текстовом виде или с использованием параметрических функций (библиотек). Тем самым нивелируется понятие задержек библиотечных элементов ПЛИС. Таким образом, задержки в проекте ПЛИС зависят от того, каким образом проект «откомпилировался», или иначе, разместился в ячейках, и как организованы связи между ними. Задержка распространения сигнала по шинам межсоединений (трассировки) в ПЛИС Altera Flex 10K в зависимости от размещения приведена в таблице 2.

При проектировании БИС на БМК в конечном виде проект раскрывается до уровня библиотеки первого уровня (топологических примитивов). Задержки не зависят от места расположения логических элементов, влияя на задержки только длина связи и количество подключенных элементов. Поэтому сравнение задержек следует проводить на уровне этих элементов.

В таблице 3 в качестве примера приведены значения задержек для схемы на рис. 6. Задержки рассчитаны с использованием вышеприведенных моделей для наиболее распространенных ПЛИС фирм Altera и Actel в сравнении с задержками БМК семейств 1592 XM (семейство БМК емкостью до 100 тыс. вентилях производства «Ангстрем»), семейств 1578 XM (семейство БМК емкостью до 30 тыс. вентилях от НИИЭТ).

При проведении анализа принималось во внимание, что значения для элементов ПЛИС приводятся для наихудшего случая, поэтому задержки БМК были соответственно скорректированы.

Как следует из таблицы 3, задержки БИС, изготавливаемой по технологии 2 мкм (1578XM), сравнимы с задержками ПЛИС, изготавливаемой по технологии 0,8 мкм (Actel Act), а задержки БИС, изготавливаемой по технологии 1,2 мкм (1592XM), сравнимы с задержками ПЛИС, изготавливаемой по технологии 0,6 мкм (Altera Flex10K). Подтверждается и тот факт, что задержки сигналов зависят от того, как логика распределяется по ячейкам ПЛИС (см. отличие задержек по сигналам B0–B1 и C1 для ПЛИС Actel и Altera).

После суммирования имеющихся отличий в элементах ПЛИС и БИС получены следующие результаты:

- В проектах ПЛИС и БИС при расчете задержек используются разные временные модели, имеющие существенные отличия.
- Принципиально отличается физическая реализация логики в кристаллах — в БИС проект раскрывается до уровня «топологических примитивов», в ПЛИС — до уровня логических блоков, включающих комбинаторную часть и D-триггер. В ПЛИС логика сворачивается, оптимизируется и разбивается на функции в соответствии с возможностями комбинаторной части логического блока.

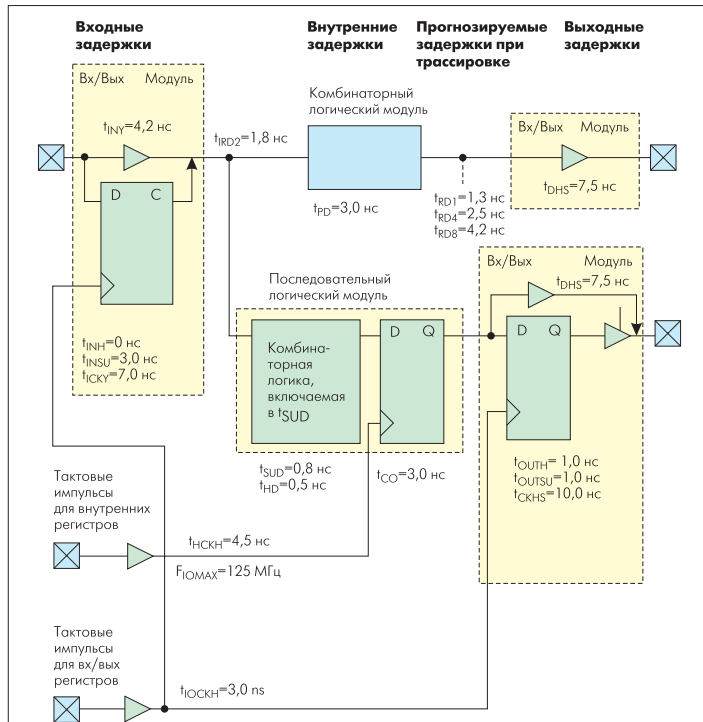


Рис. 3. Временная модель ПЛИС Actel Act3

Наименование и описание основных параметров временной модели:

- t_{INY}** — время задержки входного буфера;
- t_{INH}** — время удержания данных на входном регистре модуля входа-выхода;
- t_{INSU}** — время установления данных на входном регистре модуля входа-выхода;
- t_{ICKY}** — время задержки распространения сигнала данных от тактового импульса во входных-выходных модулях;
- t_{IRD2}** — прогнозируемое время задержки при трассировке при коэффициенте разветвления по выходу, равному 2;
- t_{PD}** — время задержки распространения сигнала в комбинаторном модуле;
- t_{RD1}** — прогнозируемое время задержки при трассировке для наилучших рабочих условий при коэффициенте разветвления, равном единице (t_{RD4} — при коэффициенте разветвления по выходу, равном четырем, t_{RD8} — восьми);
- t_{DHS}** — время задержки сигнала в выходном буфере (при емкости нагрузки 35 пФ);
- t_{ENZHS}** — время задержки при переключении выходного буфера из состояния Z в активный режим;
- t_{SUD} , t_{OUTSU}** — время установки данных на входе соответствующего триггера;
- t_{CO}** — время задержки распространения сигнала в триггере логического модуля относительно тактового сигнала;
- t_{HD}** — время удержания данных на входе триггера;
- t_{OUTH}** — время удержания данных на выходном регистре модуля входа-выхода;
- t_{HCKN}** — время задержки распространения сигнала встроенного тактового входа до входов триггера S модуля;
- t_{IOCKN}** — время задержки распространения сигнала встроенного тактового входа до входов триггера модуля входа-выхода.

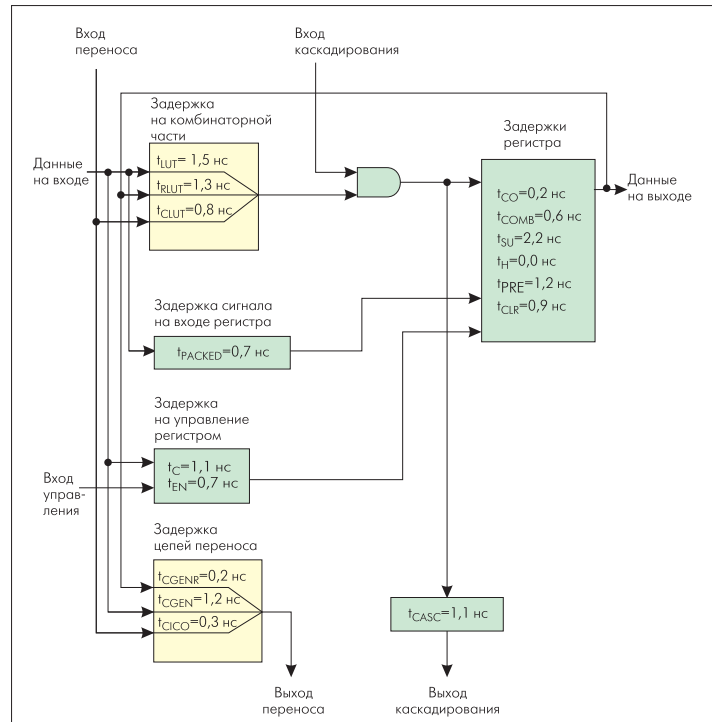


Рис. 4. Временная модель для логических элементов Altera Flex 10K

(задержки приведены для градации быстродействия — 3)

Наименование и описание основных параметров временной модели:

- t_{LUT}** — время задержки распространения входных данных на комбинаторной части логического элемента;
- t_{RLUT}** — время задержки сигнала обратной связи на комбинаторной части логического элемента;
- t_{CLUT}** — время задержки цепи ускоренного переноса на комбинаторной части логического элемента;
- t_{PACKED}** — время задержки сигнала на входе регистра;
- t_C** — время задержки сигнала до тактового входа регистра;
- t_{EN}** — время задержки сигнала управления регистром;
- t_{CGENR}** — время задержки прохождения сигнала цепи обратной связи до выхода сигнала переноса;
- t_{CGEN}** — время задержки прохождения сигнала от входа до выхода сигнала переноса;
- t_{CICO}** — время задержки прохождения сигнала цепочки переноса от входа до выхода;
- t_{CO}** — время задержки, равное времени между появлением тактового сигнала на входе регистра логического элемента и появлением данных на его выходе — задержка срабатывания триггера;
- t_{COMB}** — комбинаторная задержка;
- t_{SU}** — время установки данных на входе регистра до прихода тактового сигнала;
- t_H** — время удержания данных на входе регистра до прихода тактового сигнала;
- t_{PRE}** — время задержки по сигналу установки регистра;
- t_{CLR}** — время задержки по сигналу сброса регистра;
- t_{CASC}** — время задержки по сигналу каскадирования.

- В ПЛИС задержки любых логических элементов с количеством входов до 4 (для некоторых архитектур — до 5) одинаковы и определяются задержкой комбинаторной части логических элементов. В БИС на БМК наоборот — задержки элементов сильно отличаются по переднему и заднему фронтам сигнала, а на задержку переключения значительно влияет коэффициент разветвления по выходу (количество подключенных входов других элементов или, иначе, емкостная нагрузка).
- В ПЛИС очень неэффективно реализуются многовходовые логические функции (более 5 переменных). Чем больше входов, тем непропорционально сильнее растет задержка в элементах ПЛИС, так как добавляются задержки по шинам связи логических элементов.
- Значительно отличаются времена установки T_{SU} и удержания данных T_H на входе триггеров для ПЛИС и БИС на БМК. По сути, триггеры в ПЛИС физически реализуются одинаково, а конфигурирование их, например, как JK, D или T-триггера осуществляется с использованием комбинационной логики логического элемента, задержка которого учитывается в этих временных параметрах. Для БИС эти времена отличаются для различных типов триггеров в пределах одного семейства БМК и эти триггеры физически реализуются по-разному. Поэтому эти важные временные параметры невозможно унифицировать для ПЛИС и библиотечных элементов БМК.
- Цепи синхронизации отличаются, особенно если используются глобальные такто-

вые цепи — в ПЛИС они могут быть быстрее. В БИС необходимо формировать дерево тактовых импульсов и обеспечивать необходимую буферизацию.

- Для других цепей и связей — наоборот, для ПЛИС они значительно медленнее. Основной вклад в задержку вносят задержки связей через проходной транзистор (pass-transistor) или переключки типа антифьюз.
- В ПЛИС отсутствуют элементы и шины с третьим состоянием, поэтому объединение по выходу осуществляется мультиплексированием. В БИС на БМК элементы и шины с третьим состоянием активно используются.

В любом случае, проект БИС, представленный в виде списка связей библиотечных элементов (netlist), значительно отличается от «компилированного» netlist ПЛИС.

Таблица 2

Наименование параметра	Значение задержки, нс
$t_{SAMELAB}$ — задержка на трассировку для логических элементов, управляемых другими логическими элементами в том же логическом блоке	0,4
$t_{SAMEROW}$ — задержка на трассировку для ряда вх/вых элементов, логических элементов или встроенного блока того же ряда	2,6
$t_{SAMECOLUMN}$ — задержка на трассировку для логических элементов, управляемых вх/вых элементами того же столбца	3,5
$t_{DIFFROW}$ — задержка на трассировку для различных элементов, расположенных в разных столбцах	6,1
$t_{TWOROWS}$ — задержка на трассировку между вх/вых элементами или встроенными блоками и логическими элементами, или встроенными блоками другого ряда	8,7
$t_{LEPERIPH}$ — задержка на трассировку сигналов, управляющих вх/вых логическими элементами через периферийную управляющую шину	5,6
$t_{LABCARRY}$ — задержка на трассировку для сигнала ускоренного переноса от выхода логического элемента до входа другого логического элемента, в различных блоках	2,0
$t_{LABCASC}$ — задержка на трассировку для сигнала каскадирования от выхода логического элемента до входа другого логического элемента в различных блоках	2,6
$t_{DIN2IOE}$ — задержка от входа выделенного внешнего сигнала до входа управления вх/вых элементов	5,2
t_{DIN2LE} — задержка от входа выделенного внешнего сигнала до входа управления логического элемента	3,1
$t_{DCLK2IOE}$ — задержка от входа выделенного внешнего тактового сигнала до поступления тактового сигнала на вх/вых элемент	2,0
$t_{DCLK2LE}$ — задержка от входа выделенного внешнего тактового сигнала до поступления тактового сигнала на логический элемент	3,1

Таким образом, как следует из результатов анализа, согласование и унификация по временным и частотным параметрам библиотечных элементов, макро- и мегафункций ПЛИС и БИС сильно затруднена и нецелесообразна. Поэтому необходимо согласование временных и частотных характеристик полузаказных БИС с характеристиками, полученными на ПЛИС на уровне проектов, с учетом задержек связей, а не на уровне библиотечных элементов, при этом главными являются требования к временным и частотным характеристикам проекта.

Согласование временных и частотных характеристик проектов полузаказных БИС с характеристиками, полученными на ПЛИС, осуществляется путем проведения статического временного анализа (контроль времен установки T_{SU} и удержания данных T_H на входе триггеров, задержки появления сигналов относительно тактовых импульсов, задержки сигналов между регистрами и т. д.), а также сравнительного логического моделирования на соответствие временным и частотным характеристикам при нормальных и крайних значениях температур и напряжений (при проведении сравнительного моделирования исходная эталонная временная модель проекта ПЛИС формируется средствами САПР ПЛИС на выходе, в виде «овремененного» Verilog HDL описания и SDF файла проекта). Это очень важные элементы «Технологии проектирования

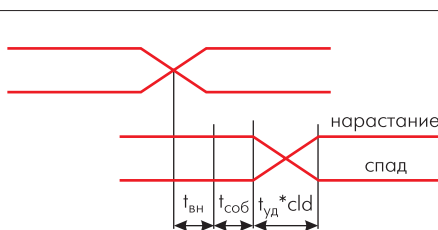


Рис. 5. Временная модель логических элементов БИС на БМК (или на стандартных элементах)

Задержка элементов рассчитывается по формуле

$$t_{\text{общ}} = t_{\text{вн}} + t_{\text{соеб}} + t_{\text{уд}} \cdot S \cdot Cld$$

$t_{\text{общ}}$ — общая задержка (нс) от определенного входа до определенного выхода;

$t_{\text{вн}}$ — внутренняя задержка (нс) ячейки без учета собственной выходной емкости;

$t_{\text{соеб}}$ — задержка (нс) на собственной выходной емкости;

$t_{\text{уд}}$ — удельная задержка на единицу нагрузочной емкости в (нс/пФ);

Cld — емкость нагрузки на выходе ячейки в пФ.

Таблица 3

	Задержка распространения сигнала для схемы на рис. 6, нс					
	В БИС на БМК 1578ХМ	В ПЛИС Actel Act3	В БИС на БМК 1592ХМ	В ПЛИС ф. Altera Flex10K		
A0→Y	6,1	5,7	7,3	3,0	2,6	3,4
A1→Y	6,1	5,7	7,3	3,0	2,6	3,4
B0→Y	3,7	3,8	3,0	1,4	1,5	3,4
B1→Y	3,7	3,8	3,0	1,4	1,5	3,4
B2→Y	3,7	3,8	3,0	1,4	1,5	3,4
C0→Y	4,8	6,3	7,3	2,0	2,9	3,4
C1→Y	4,8	6,3	7,3	2,0	2,9	1,5
C2→Y	4,8	6,3	7,3	2,0	2,9	3,4

ПЛИС-БМК», и они учтены в маршруте перевода проектов ПЛИС в полузаказные БИС.

Другим важным аспектом является необходимость анализа переведенного проекта на тестопригодность и разработка тестов для контроля БИС при производстве и испытаниях, поскольку всегда существует вероятность, что созданную БИС просто невозможно будет протестировать. Создание тестов может занимать значительную часть процесса перевода, поскольку практика показывает, что большинство разработчиков ПЛИС не проводят глубокого моделирования и не обеспечивают необходимого тестового покрытия проектов, а просто многочисленными перезагрузками ПЛИС добиваются физической работоспособности [1].

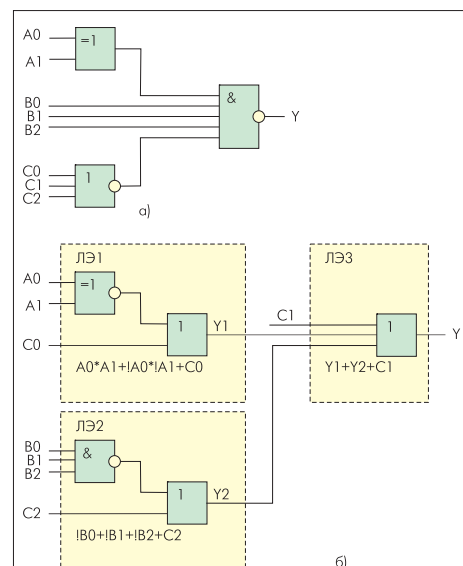


Рис. 6. Библиотечный элемент на восемь переменных: а) до компиляции; б) после компиляции

переменных: а) до компиляции; б) после компиляции

Важно отметить, что в процессе работы по переводу проектов ПЛИС в полузаказные БИС выяснилась целесообразность разработки экспертной системы, подсказывающей, какую библиотеку и какой тип БМК подключать в САПР БИС при «конвертировании» того или иного проекта. Выбор той или иной библиотеки БМК осуществляется на основе требований к временным и частотным характеристикам, определяемым разработчиком ПЛИС проекта, и не зависит от типа используемой ПЛИС.

Литература

1. В. Лобанов Об особенностях проектирования цифровых систем на отечественных БМК // Chip News. 2004. № 2.