

РЕАЛИЗАЦИЯ Verilog-ПРОЕКТОВ В БАЗИСЕ ЗАКАЗНЫХ БИС И ПЛИС С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТА СИНТЕЗА Yosys

А.Строгонов, д.т.н.¹, П.Городков²

УДК 621.3.049
ВАК 05.27.00

Большинство современных проектов цифровых схем создается с использованием языков описания аппаратуры (hardware description language – HDL), таких как Verilog или VHDL. Для преобразования HDL-кода в цифровые схемы применяют программные инструменты синтеза. Один из наиболее популярных инструментов – Yosys (Yosys Open SYnthesis Suite) – широко используется при проектировании заказных БИС и ПЛИС в университетских центрах. Этот программный пакет с открытым исходным кодом обрабатывает практически любой проект на языке Verilog-2005, преобразуя его в форматы BLIF, EDIF, VTOR, SMT-LIB и др. При разработке заказных БИС этот инструмент используется в маршруте проектирования Qflow. Рассмотрим особенности применения Yosys при реализации Verilog-проектов в базе заказных БИС и ПЛИС.

При создании цифровых БИС и ПЛИС выделяют следующие уровни проектирования: системный, алгоритмический, поведенческий, уровень регистровых передач (RTL), уровень логических вентилях, физический (библиотеки логических элементов заказных БИС и LUT ПЛИС включая триггеры) и транзисторный (схемотехнический). Программный инструмент Yosys охватывает три уровня синтеза проекта: поведенческий, синтез на уровне регистровых передач и логический (рис.1).

РЕАЛИЗАЦИЯ Verilog-ПРОЕКТОВ В БАЗИСЕ ЗАКАЗНЫХ БИС

В отличие от инструмента синтеза Odin II [1–3], который используется в академическом САПР ПЛИС VTR7, Yosys может синтезировать крупные Verilog-проекты с сайта Open Cores (например, микропроцессорные ядра OpenRISC1200, OpenMSP430, K68 и др.). Yosys содержит встроенные формальные методы проверки свойств и эквивалентности [4]. Он позволяет отображать Verilog-проекты в базе как библиотечных ячеек заказных БИС (в формате liberty), так и ПЛИС Xilinx седьмой серии и Lattice iCE40 [5]. Для логической оптимизации и отображения в базе заказных БИС и ПЛИС используется программный инструмент ABC.

Каждая задача синтеза с использованием программного пакета Yosys выполняется с помощью скриптов. Рассмотрим примеры скриптов для синтеза и после-

1 Воронежский государственный технический университет, профессор кафедры полупроводниковой электроники и нанoeлектроники, тел. (4732) 437695, andreistrogonov@mail.ru.

2 Воронежский государственный технический университет, аспирант кафедры полупроводниковой электроники и нанoeлектроники, тел. (4732) 437695, gorodkoff@gmail.com.



Рис.1. Уровни абстракции и синтеза при проектировании цифровых БИС и область использования Yosys (три уровня синтеза)

дующего отображения в базе библиотеки заказных БИС и в базе ПЛИС Xilinx серии Virtex 7 для Verilog-проекта счетчика (рис.2). По команде `read_verilog counter` выполняется построение по Verilog-коду счетчика абстрактного синтаксического дерева с последующим RTL-представлением. По команде `techmap` крупнозернистые RTL-ячейки (например, сумматоры или умножители) преобразуются в мелкозернистые (например, в вентили 2И-НЕ, 2ИЛИ-НЕ, триггеры и др.). Далее мелкозернистые ячейки в два приема отображаются в аппаратные комбинационные элементы и триггеры, соответствующие какой-либо библиотеке (например, `asic_cells`). В библиотеке `asic_cells` определены как вентили, так и триггеры. Для заказ-

```

# read design
read_verilog counter.v
# high-level synthesis
hierarchy -check -top counter
proc; opt; fsm; opt; techmap; opt
# mapping registers to ASIC cells
dfflibmap -liberty asic_cells.lib
# mapping logic to ASIC cells using
# Berkeley ABC
abc -lut 4; op
# map internal cells to FPGA cells
techmap -map fpga_cells.v; opt
# write netlist
write_verilog asic_synth.v
a)

# read design
read_verilog counter.v
# high-level synthesis
hierarchy -check -top counter
proc; opt; fsm; opt; techmap; opt
# mapping logic to LUTs using
# Berkeley ABC
abc -lut 4; op
# map internal cells to FPGA cells
techmap -map fpga_cells.v; opt
# write netlist
write_verilog fpga_synth.v
b)

```

Рис.2. Примеры скриптов для синтеза и последующего отображения в базе библиотеки заказных БИС (а) и в базе ПЛИС Xilinx серии Virtex 7 (б) на примере Verilog-проекта 3-разрядного суммирующего счетчика

ных БИС триггеры отображаются с помощью команды `dfflibmap` в формате `liberty` с использованием библиотеки `asic_cells`. Логика проекта отображается с помощью программного инструмента ABC и библиотеки `asic_cells` (см. рис.2а). Для ПЛИС (более подробно в следующем разделе) логика отображается непосредственно в 4-входные LUT (см. рис.2б).

На рис.3–6 показаны этапы преобразования Verilog-кода 3-разрядного суммирующего счетчика в мультиплексоры и триггеры с последующим представлением в базе комбинационных логических элементов и регистров библиотеки `asic_cells`.

В настоящее время многие компании, в частности `efabless` (`efabless.com`), сотрудничают с разработчиками в области проектирования и создания БИС с использованием программных продуктов с открытым кодом.

Рассмотрим маршрут проектирования заказных БИС Qflow, который представляет собой полный набор инструментов с открытым кодом для синтеза цифровых схем, начиная с Verilog-кода и заканчивая физической топологией

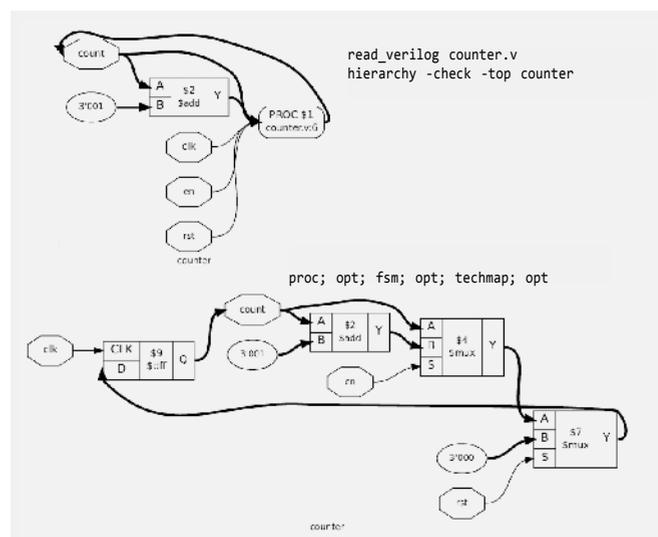


Рис.3. Вверху: RTL-представление проекта с помощью команды `read_verilog hierarchy` с последующим анализом (`hierarchy`); внизу: поведенческое представление проекта с использованием крупнозернистых RTL-ячеек с помощью команд `proc` и `opt`

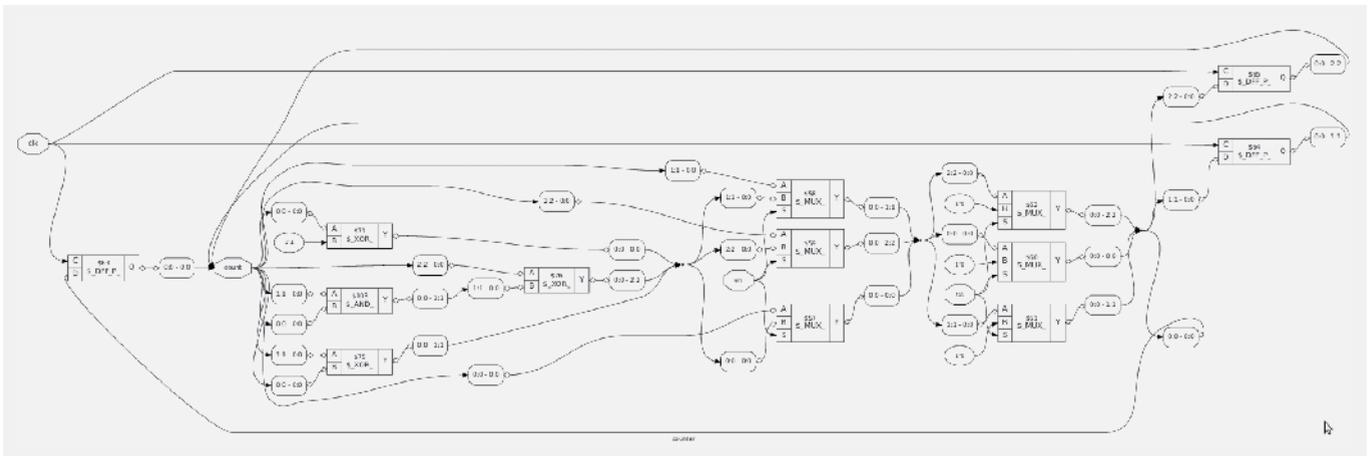


Рис.4. Отображение крупнозернистых RTL-ячеек в мелкозернистые с помощью команд `techmap` и `opt`

для конкретного технологического процесса изготовления. Qflow предполагает использование следующих программных инструментов: Yosys / ABC (синтез и оптимизация), Graywolf (глобальное размещение), Qrouter (детальная трассировка), Magic (топологический редактор) (рис.7).

Qflow поддерживает основные промышленные форматы САПР БИС: LEF (Library Exchange Format, формат библиотечного обмена), DEF (Design Exchange Format, формат конструктивного обмена), GDSII (для топологии) и др. LEF-файл условно делится на библиотечную и технологическую части. Библиотечная часть содержит описание внешней геометрии ячеек (границы ячеек, положение), информацию о расположении входов и выходов и другие обструкции (препятствия) для трассировки. Данная абстракция исполь-

зуется на этапах размещения и разводки элементов на кристалле, а информация о внутренней структуре ячеек для этих стадий не важна. Технологическая часть содержит разнообразные правила проектирования, необходимые для размещения и разводки. Она включает требования к размерам металла, направлениям, спейсингам (расстояниям между соседними блоками, межслойными переходами (via), металлами, ячейками и т.д.) и множество другой технологической информации, которая используется для правильного размещения и разводки в соответствии с технологическими нормами.

Применение инструментов из маршрута Qflow с открытым кодом подобно применению коммерческих САПР при разработке заказных БИС, таких как Cadence, Synopsys, Tanner. По умолчанию основ-

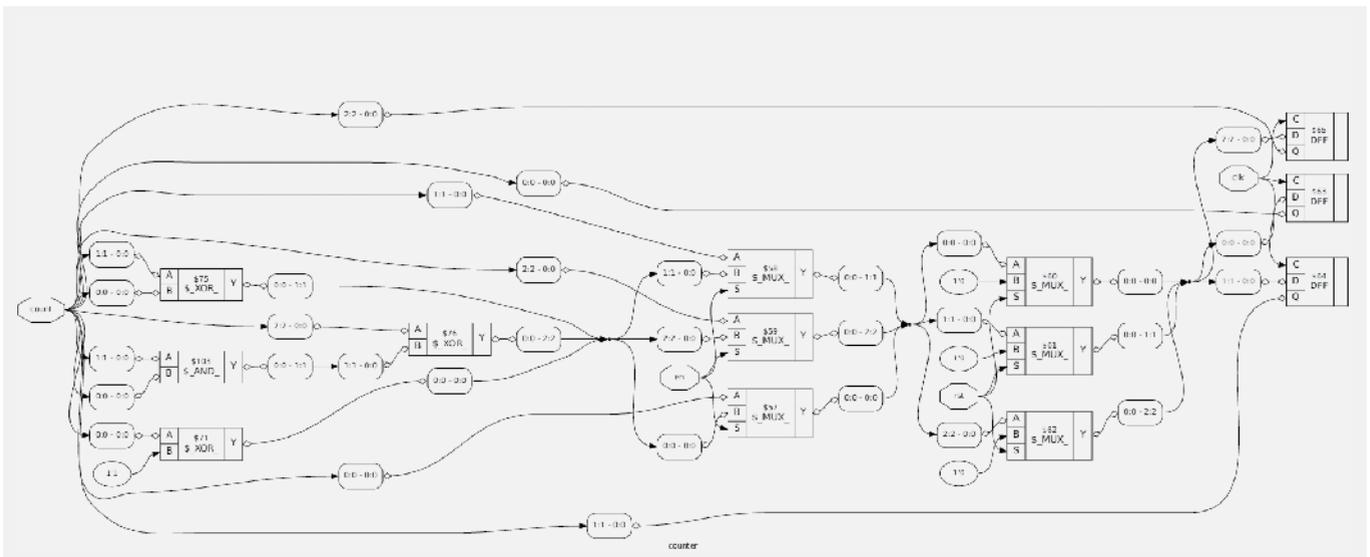


Рис.5. Отображение триггеров с помощью команды `dfflibmap - liberty` в базе библиотеки `asic_cells`

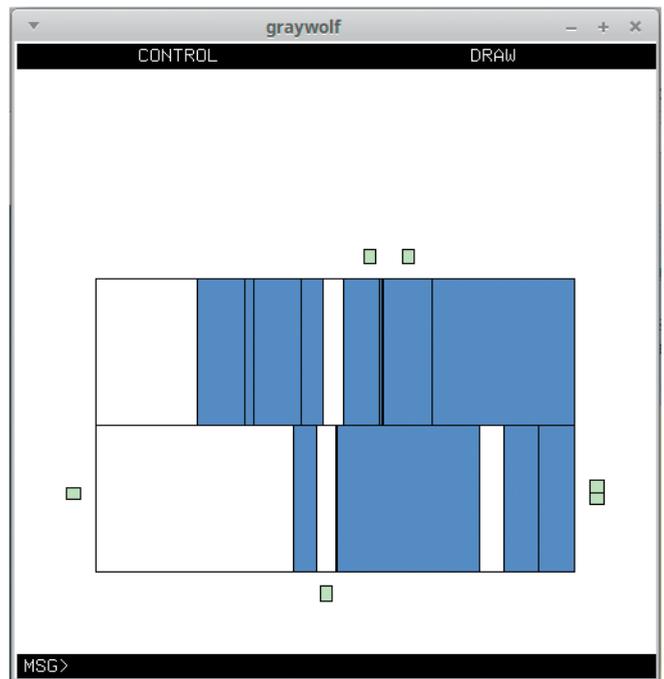
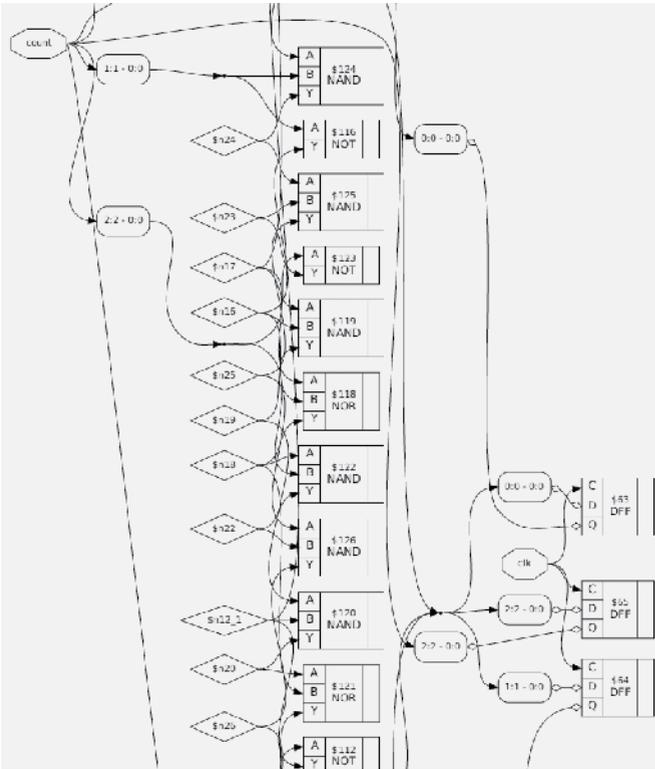


Рис.8. Вид рабочего окна программного инструмента Graywolf в процессе размещения логических элементов 3-разрядного суммирующего счетчика на кристалле

Рис.6. Отображение логических элементов с помощью команды abc - liberty в базе библиотеки asic_cells

ными средствами Verilog-синтеза в Qflow выбраны Yosys/ABC, а Odin-II/ABC могут быть указаны в качестве альтернативного интерфейса.

После стадии отображения логических элементов на библиотеку стандартных ячеек выполняются их размещение и трассировка на кристалле. Этап размещения определяет грубую оценку маршрутизации.

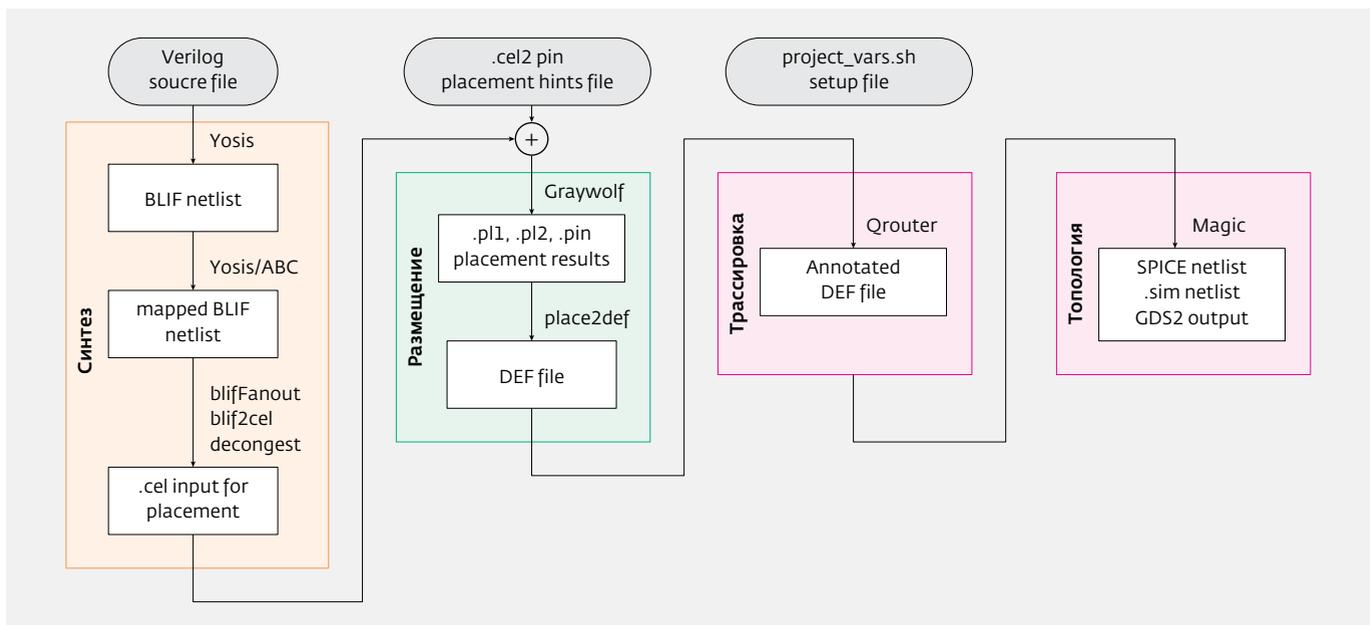


Рис.7. Маршрут проектирования заказных цифровых БИС Qflow с использованием программных инструментов с открытым кодом Yosys/ABC, GrayWolf, Qrouter, Magic

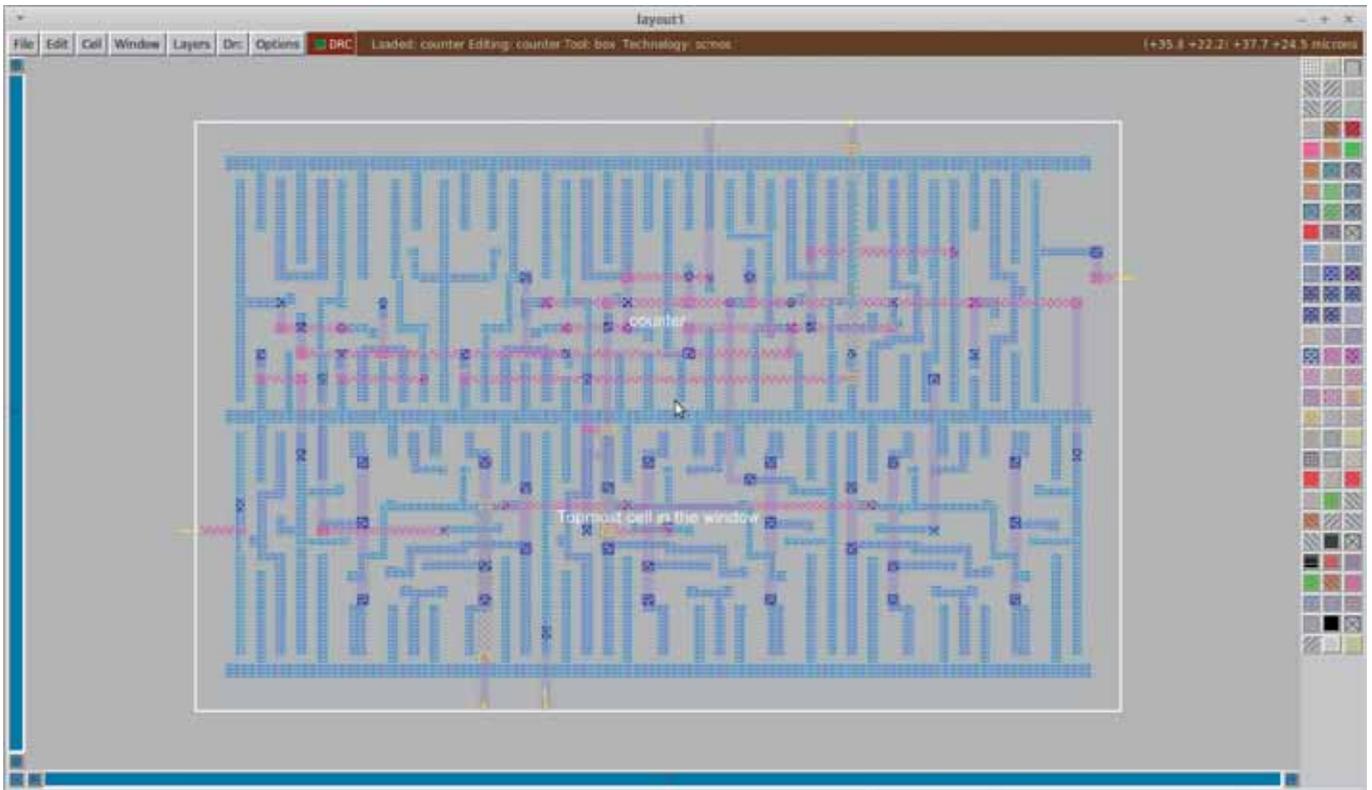


Рис.9. Абстрактная топология 3-разрядного суммирующего счетчика в топологическом редакторе Magic

Ячейки (логические элементы) распределяются в блоке таким способом, чтобы минимизировать суммарную

длину проводников, соединяющих все контакты ячеек. Инструмент размещения Graywolf, разработанный

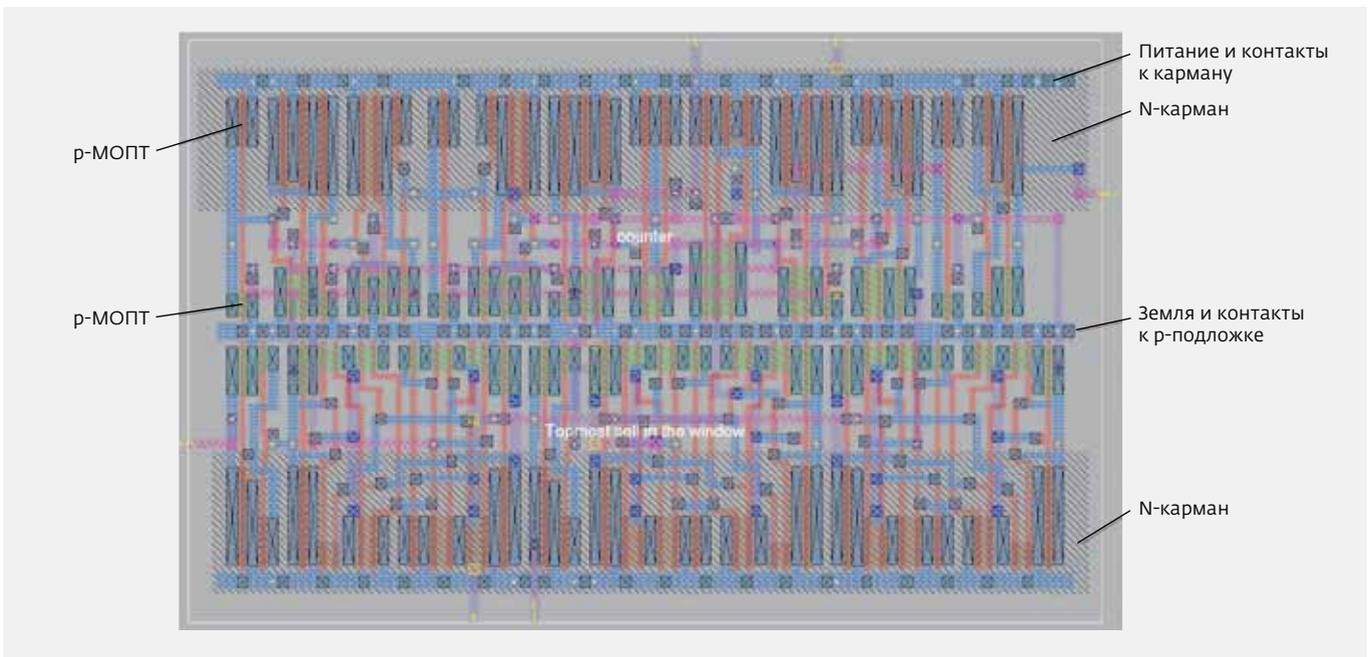


Рис.10. Топология 3-разрядного суммирующего счетчика, созданная по масштабируемой КМОП-технологии OSU0.35 в топологическом редакторе Magic

```

module top(clk, ctrl, led_7, led_6, led_5, led_4, led_3, led_2, led_1, led_0);
input clk, ctrl;
output led_7, led_6, led_5, led_4;
output led_3, led_2, led_1, led_0;
reg [31:0] counter;
always @(posedge clk)
    counter <= counter + (ctrl ? 4 : 1);
assign {led_7, led_6, led_5, led_4, led_3, led_2, led_1, led_0} = counter >> 24;
endmodule

```

Рис.11. Verilog-код проекта счетчика (файл example.v)

в Йельском университете, распространялся как ПО с открытым исходным кодом в течение длительного времени, пока не перешел в разряд коммерческих продуктов. Последняя версия Graywolf с открытым исходным кодом не обеспечивает детальную трассировку, но является инструментом размещения профессионального уровня.

Для детальной трассировки ячеек на кристалле БИС используется программный продукт Qrouter (<http://opencircuitdesign.com/qrouter/>). Qrouter представляет собой инструмент для физического соединения топологических ячеек логических элементов внутри блока согласно списку соединений с помощью сгенерированных металлических слоев и межслойных контактов. Он базируется на использовании модифицированного волнового алгоритма трассировки Ли и представляет собой двухстадийный многослойный трассировщик, который прокладывает трассы поверх топологических ячеек (по принципу "бесканальное море вентиляей"). Qrouter считывает входные форматы LEF и DEF и формирует на выходе аннотируемый DEF-файл.

Библиотеки стандартных ячеек цифровых БИС – основной компонент маршрута проектирования. Инструменты высокоуровневого синтеза коммер-

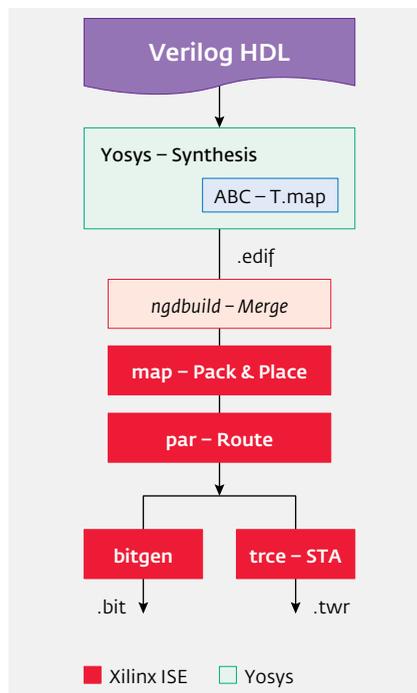


Рис.12. Маршрут проектирования цифровых схем, представленных Verilog-кодом, с использованием синтезатора Yosys и САПР Xilinx ISE в базе ПЛИС

ческих САПР БИС используют патентованные библиотеки, предоставляемые различными кремниевыми фабриками. Проблема с патентованными библиотеками заключается в том, что они не могут быть использованы в качестве примеров проектирования цифровых устройств и размещены на общедоступных веб-сайтах. Однако все же существует несколько комплектов библиотек стандартных ячеек с открытым исходным кодом для популярных технологических процессов. Некоторые из них, основанные на единых правилах проектирования масштабируемой КМОП-технологии от MOSIS (MOSIS Scalable CMOS design

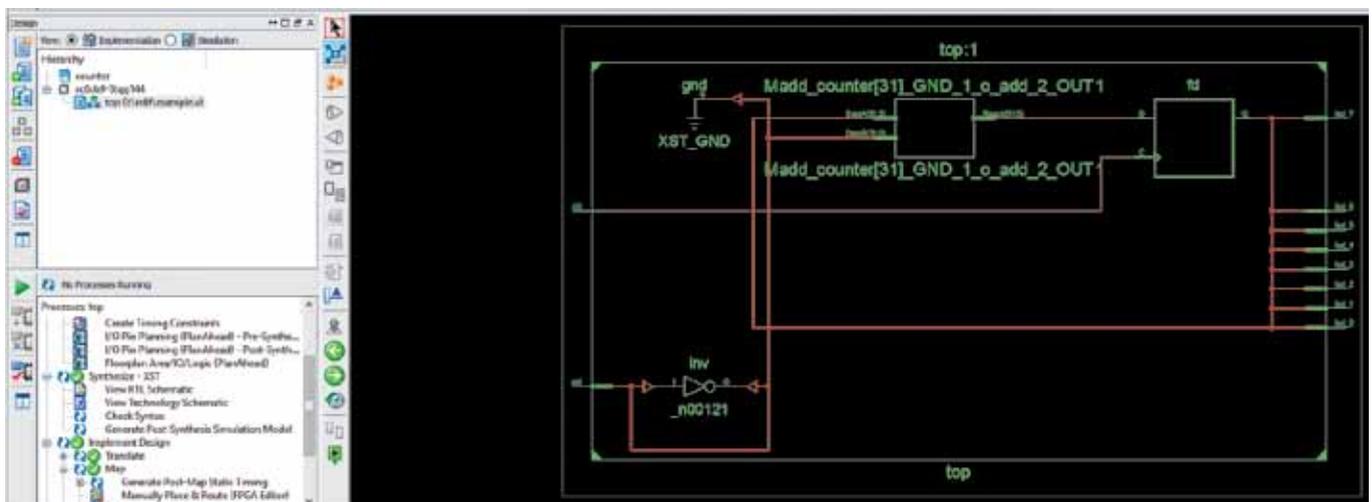


Рис.13. RTL-представление проекта "Бегущие огоньки" в САПР ISE версии 14.2

```
# read design, чтение проекта
read_verilog example.v
# high-level synthesis, построение иерархии
hierarchy -check -top top
proc; opt; fsm; opt; techmap; opt
# mapping logic to LUTs using Berkeley ABC
abc -lut 6; opt
# map internal cells to FPGA cells
techmap -map +/xilinx/cells_map.v; opt
# insert clock buffers
select -set clocks */t:FDRE %x:+FDRE[C] */t:FDRE %d
iopadmap -inpad BUF0P 0:I @clocks
# insert i/o buffers
iopadmap -outpad OBUF I:0 -inpad IBUF 0:I @clocks %n
# write netlist
write_edif synth.edif
```

Рис.14. Скрипт для синтеза Verilog-проекта в Yosys

rules) (см. www.mosis.com, www.isi.edu), имеются в открытом доступе. Набор правил проектирования от MOSIS не столь специфичен, как правила кремниевых фабрик.

Многие кремниевые фабрики, такие как TSMC, IBM, AMI, Orbit, выпускают по правилам проектирования MOSIS Scalable CMOS design rules субмикронные БИС. Единые правила проектирования, поддерживаемые разработчиками САПР БИС, позволяют дизайн-центрам, не имеющим собственных производств, решить задачу вывода на рынок изделий и обеспечить переносимость проектов между ведущими производителями.

В качестве основной единицы измерения расстояния по правилам проектирования выбрана масштабная величина (лямбда). В масштабируемой технологии КМОП (SCMOS) топология схемы создается в соответствии с лямбда-методологией, согласно которой единица измерения масштабируется в соответствии с изменением технологии в сторону уменьшения размеров, что позволяет избежать повторного проектирования топологии кристалла.

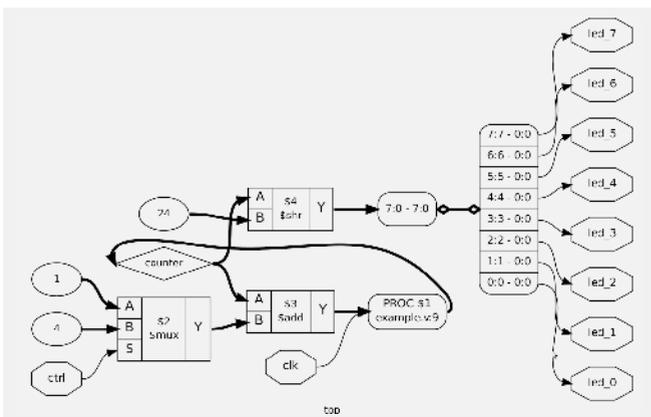


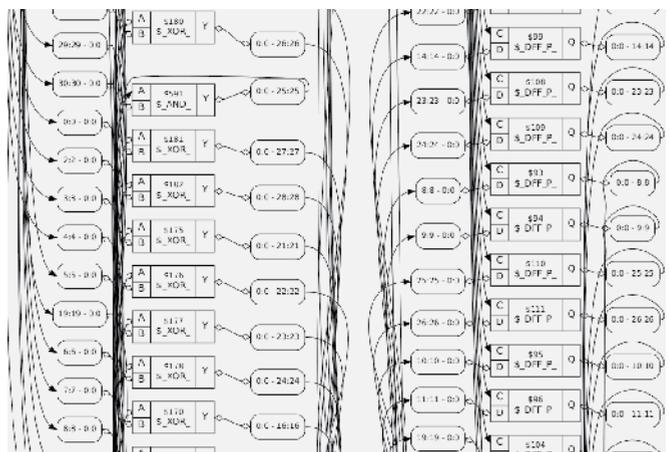
Рис.16. RTL-представление проекта в Yosys

```
NET "clk" TNM_NET = clk;
TIMESPEC TS_clk = PERIOD "clk" 50 MHz HIGH 50%;
NET "clk" LOC = P56;
NET "ctrl" LOC = P1;
NET "led_0" LOC = P134;
NET "led_1" LOC = P133;
NET "led_2" LOC = P132;
NET "led_3" LOC = P131;
NET "led_4" LOC = P127;
NET "led_5" LOC = P126;
NET "led_6" LOC = P124;
NET "led_7" LOC = P123;
```

Рис.15. Файл временных ограничений (ucf-файл) для рассматриваемого проекта

Топологический редактор Magic в маршруте Qflow использует библиотеки стандартных ячеек цифровых БИС, реализованные по масштабируемой КМОП-технологии, совместимые с топологическими библиотеками коммерческих кремниевых фабрик, преимущественно TSMC и AMI 0,18, 0,35 и 0,5 мкм. По умолчанию предлагается библиотека с открытым исходным кодом OSU0.35 (разработана в государственном университете Оклахомы, США) с топологическими проектными нормами 0,35 мкм, не требующая лицензионного соглашения, как, например, в случае применения библиотек кремниевых фабрик (MOSIS AMI, HP и др.). Набор библиотек для проектирования по методу стандартных ячеек доступен для скачивания на сайте www.vlsiarch.ecen.okstate.edu. Библиотека OSU0.35 содержит набор файлов, включая файлы в форматах LEF и GDSII.

На рис.8 показано окно программного инструмента Graywolf в процессе размещения логических элементов 3-разрядного суммирующего счетчика на кристалле. В состав GrayWolf входит дополнительная утилита



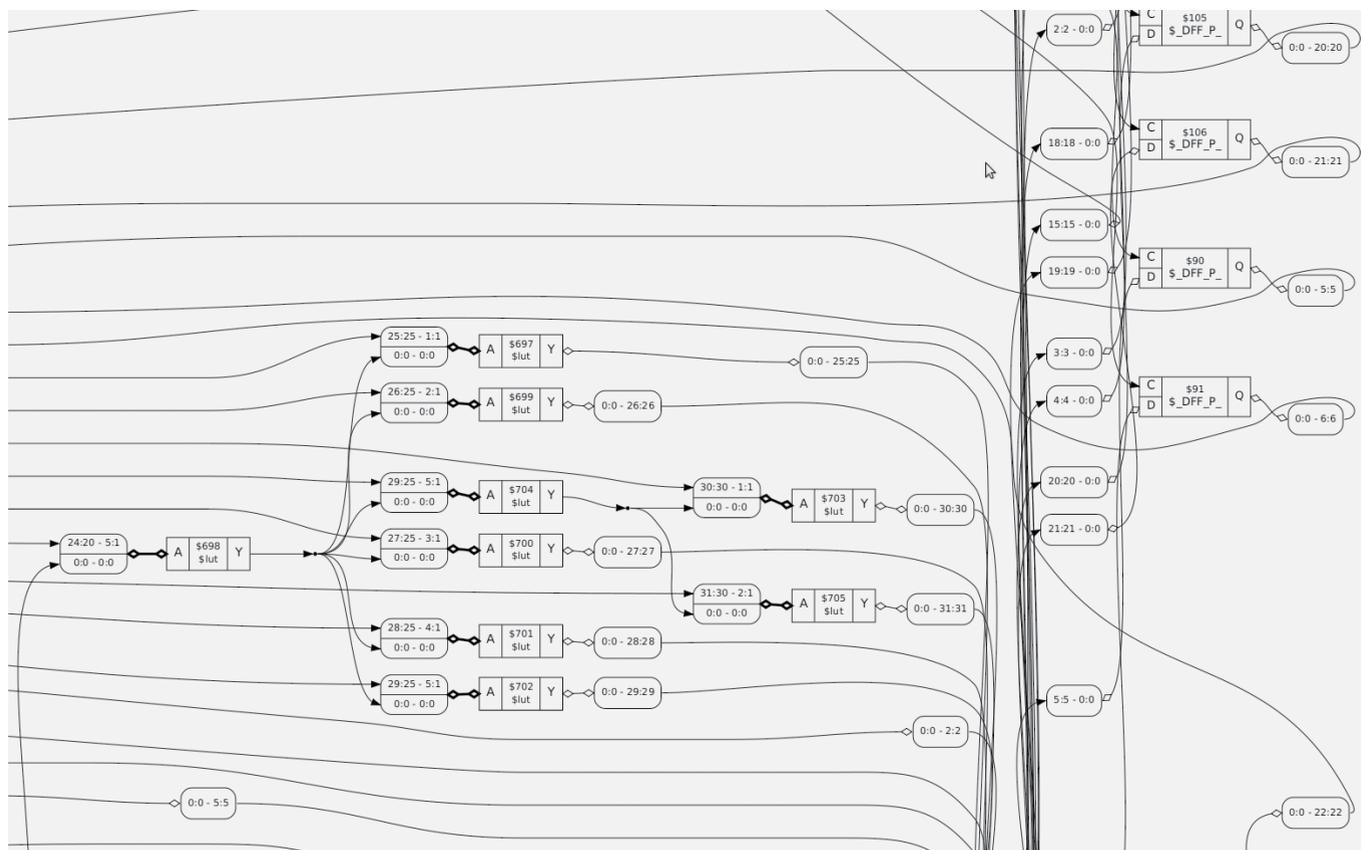


Рис.18. Отображение логики (фрагмент) проекта в базисе 6-входовых LUT (\$lut)

place2def, которая преобразует выходные результаты в формат DEF, содержит информацию о списках цепей и размещенных элементах. Она также записывает дополнительную информацию о количестве используемых слоев металлизации в cfg-файл, что необходимо для маршрутизации.

На рис.9 представлена абстрактная топология счетчика в топологическом редакторе Magic с учетом размещения и детальной трассировки. На рис.10 показана топология счетчика, созданная в соответствии с масштабируемой КМОП-технологией OSU0.35 (модификация технологического процесса кремниевой фабрики TSMC, SCN4M_SUB.20, масштабируемая КМОП-технология по правилам MOSIS, N-карман, четыре слоя металлизации, слой поликремния, субмикронные проектные нормы, лямбда 0.2, <https://www.mosis.com/vendors/view/tsmc/035>).

Для построения физической абстракции нужно загрузить LEF-файл в консоле редактора Magic с помощью команды `lef read /usr/local/share/qflow/tech/osu035/osu035_stdcells.lef`. Для перехода от абстрактного описания к полному описанию топологии (gds), определенной технологическим процессом OSU0.35, необходимо загрузить технологический файл в фор-

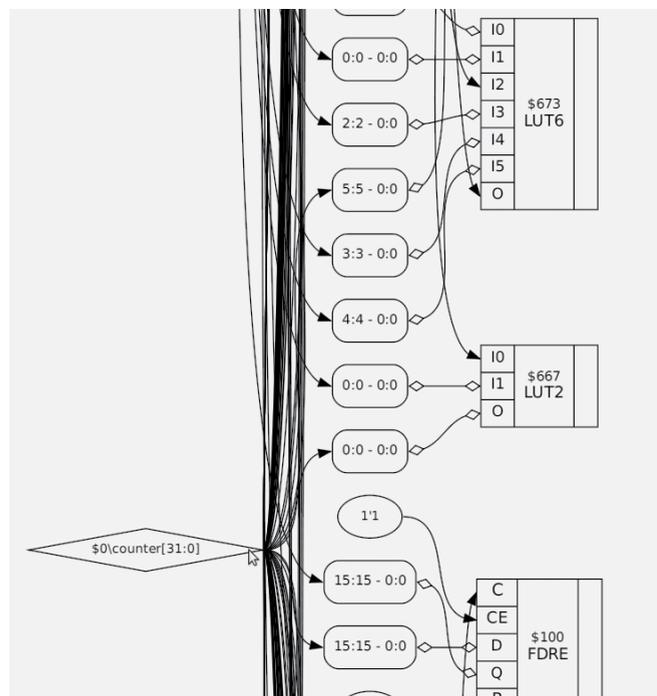


Рис.19. Технологическое отображение (фрагмент) в примитивы ПЛИС Xilinx (LUT2, LUT6, FDRE)

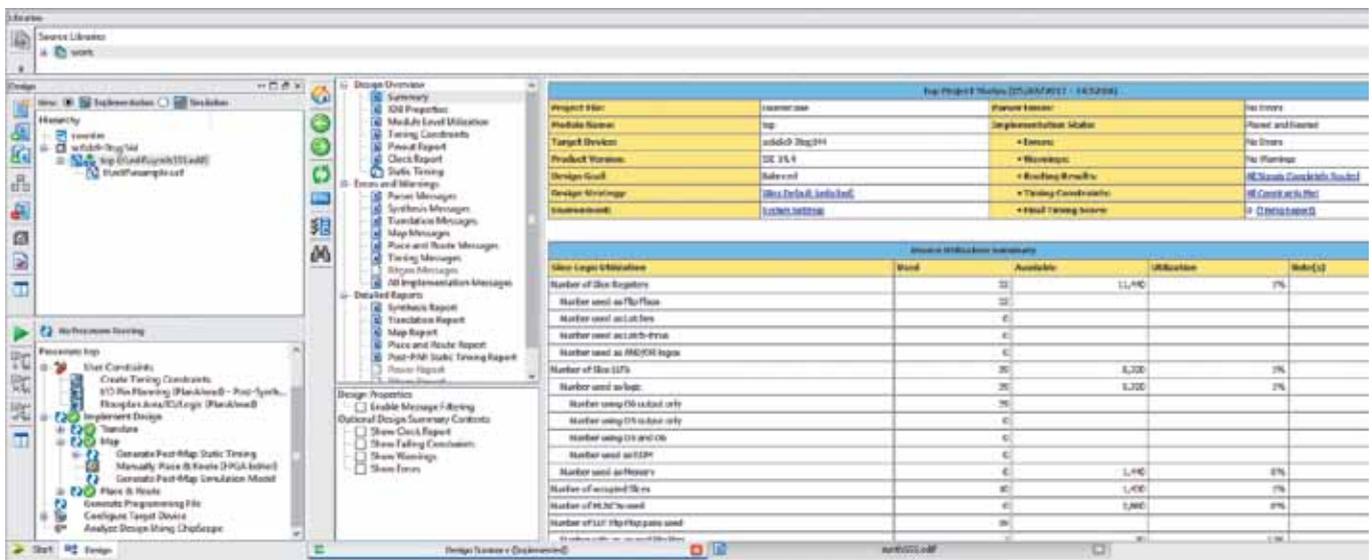


Рис.20. Успешно реализованный проект, представленный edif-файлом и файлом временных ограничений, в базе ПЛИС XC6SLX9-2-TQG144

мате gds2 в консоле редактора Magic с помощью команды gds read osu035_stdcells.gds2.

РЕАЛИЗАЦИЯ Verilog-ПРОЕКТОВ В БАЗИСЕ ИНДУСТРИАЛЬНЫХ ПЛИС XILINX

В качестве примера рассмотрим Verilog-проект "Бегающие огоньки", в котором светодиоды загораются с разной частотой в зависимости от сигнала ctrl. Verilog-

код проекта представляет собой 32-разрядный счетчик на базе сумматора и регистра с обратной связью, к разрядам которого (24–31) подключены восемь светодиодов led_0–led_7 (рис.11). На рис.12 представлен маршрут проектирования цифровых схем в виде Verilog-кода в базе ПЛИС с использованием синтезатора Yosys и САПР Xilinx ISE. На рис.13 показано RTL-представление проекта в САПР ISE ver 14.2.

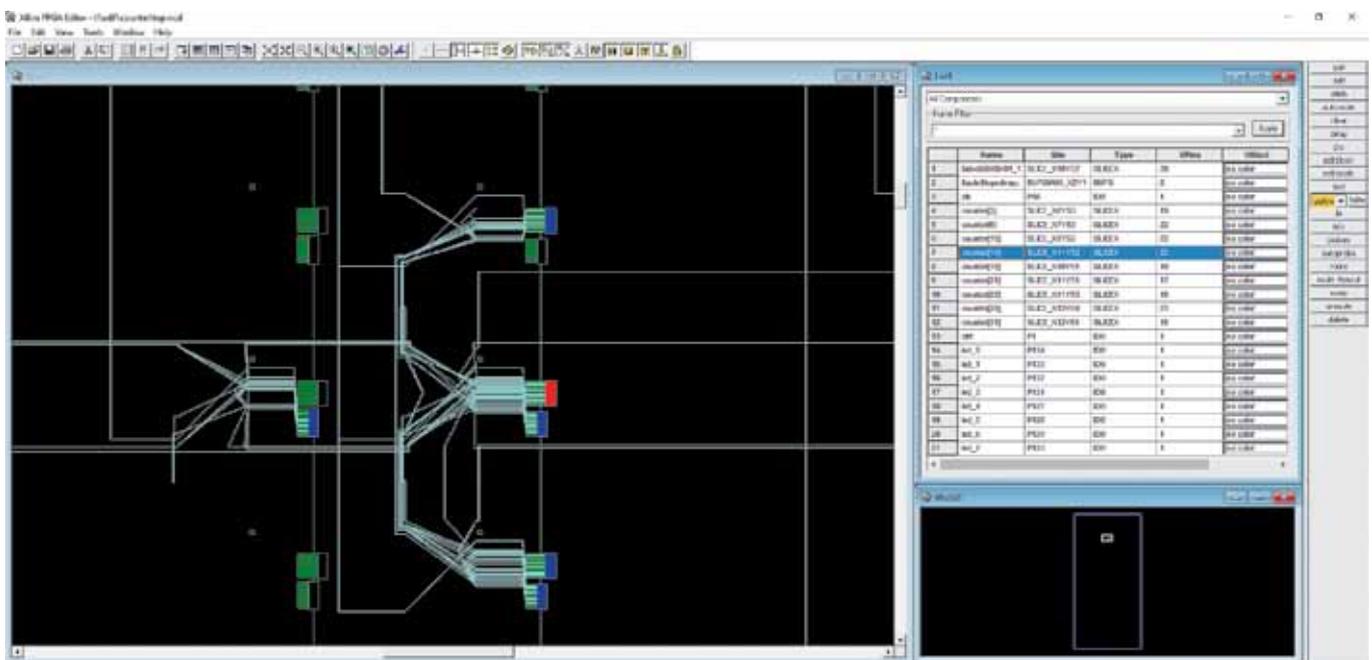


Рис.21. Реализация проекта, представленного edif-файлом, в базе ПЛИС XC6SLX9-2-TQG144 (красным отмечена задействованная секция)

Yosys, преобразующий Verilog-проекты в формат EDIF, позволяет их отображать в базе ПЛИС Xilinx серий Spartan-6 и Virtex-7. Реализуем проект в ПЛИС серии Spartan-6 XC6SLX9-2-TQG144. Для логической оптимизации и отображения в базе ПЛИС в Yosys интегрирован программный инструмент ABC.

На рис.14 показан скрипт для синтеза рассматриваемого Verilog-проекта и отображения в базе ПЛИС Xilinx серии Spartan 6, а на рис.15 – файл временных ограничений (ucf-файл) для этого проекта. Рассмотрим основные команды скрипта. Более подробную информацию о командах для синтеза в базе ПЛИС Xilinx можно найти на сайте Yosys в разделе Synt_xilinx (http://www.clifford.at/yosys/cmd_synth_xilinx.html).

По команде `read_verilog example.v` выполняется построение абстрактного синтаксического дерева с последующим RTL-представлением, которое отличается большей детальностью, чем в ISE (рис.16). Видна особенность использования сигнала `ctrl`, который подключается к адресному входу мультиплексора 2 в 1. По команде `techmap` крупнозернистые RTL-ячейки (сумматоры, регистры) преобразуются в мелкозернистые (вентили, триггеры) (рис.17). Логика проекта с помощью команды `abc -lut 6` отображается непосредственно в 6-входные LUT (рис.18). Команда `techmap -map +/xilinx/cells_map.v` отображает 6-входные LUT и триггеры в технологическом базисе ПЛИС Xilinx Spartan-6 (рис.19). Инструмент Yosys позволяет также синтезировать в MAC-блоки (DSP48) ПЛИС Xilinx.

На рис.20 показаны проект, представленный edif-файлом и файлом временных ограничений, а также сведения об используемых логических ресурсах ПЛИС XC6SLX9-2-TQG144, а на рис.21 – задействованные логические и трассировочные ресурсы ПЛИС. Отображенные на рис.20 и 21 результаты подтверждают, что edif-

файл, сформированный синтезатором Yosys по Verilog-коду, успешно обрабатывается утилитами `ngdbuild`, `map` (упаковка элементарных логических элементов в секции КЛБ, но секции еще не получают реальных мест в ПЛИС) и `par` (для секций, сформированных на этапе `map`, определены места в матрице конфигурируемых логических блоков (`place`) с последующей трассировкой связей между сигналами секций (`route`)) САПР Xilinx ISE.

Yosys применяется и для реализации проектов в базе ПЛИС от Lattice Semiconductor. Для отладки проектов на базе ПЛИС iCE40 создана открытая аппаратная платформа `iceBoard 1.0`. Отличительная особенность данной платформы – использование для программирования ПЛИС открытого и свободного программного обеспечения `iceTC`, основными компонентами которого являются инструмент синтеза Yosys, средство размещения и трассировки `Arachne-pnr`, программные инструменты для формирования битового потока, документация на ПЛИС iCE40 Lattice IceStorm и загрузчик битового потока (программатор) `icaprogram`.

ЛИТЕРАТУРА

1. **Строгонов А., Городков П.** Реализация Verilog-проектов в базе академических ПЛИС с применением САПР VTR7.0 // Компоненты и технологии. 2017. № 5. С. 12–17.
2. **Строгонов А., Городков П.** Программные средства с открытым исходным кодом для проектирования цифровых устройств в базисах БИС и ПЛИС // Компоненты и технологии. 2017. № 3. С. 88–97.
3. **Строгонов А., Цыбин С., Городков П.** САПР VTR7 для проектирования академических ПЛИС // Компоненты и технологии. 2016. № 3. С. 65–73.
4. Yosys: свободный софт для Verilog-синтеза. – <http://ramlanyammambam.livejournal.com/tag/fpga>
5. <https://github.com/cliffordwolf/yosys>