

Использование синтезатора Synplicity Synplify для разработки проектов цифровых устройств в Altera Quartus II

Андрей СТРОГОНОВ, д. т. н.
andreistrogonov@mail.ru
Сергей ЦЫБИН, к. т. н.
tsybin@edc-electronics.ru
Павел ГОРОДКОВ
gorodkoff@gmail.com

Составной частью процесса компиляции проекта в ПЛИС является этап анализа и синтеза. Под термином «синтез логики» следует понимать автоматическое преобразование проекта с уровня регистровых передач (RTL) на уровень вентилей технологического базиса ПЛИС.

Вариант 1. Использование VQM-файлов, созданных синтезатором Synplify Pro в проектах САПР Quartus II

Из работ [1, 2] возьмем проект последовательного КИХ-фильтра на четыре отвода с использованием мегафункции умножения LPM_MULT и накопления ALTACCUMULATE (рис. 1). Линия задержки, ПЗУ и управляющий автомат представлены Verilog-кодом. Проект реализован в ПЛИС Cyclone II EP2C5F256C8 и основан на примере, поставляемом с САПР Quartus II. Его можно найти по адресу `altera\90\qdesigns\fir_filter`, а для версии 13.0 — `altera\13.0\quartus\qdesigns\fir_filter`.

С помощью меню `File/Create/Update/Create HDL File for Current File` преобразуем проект последовательного КИХ-фильтра (`fir_mult_accum.bdf`) в VHDL-файл (`fir_mult_accum.vhd`). При этом файл `fir_mult_accum.bdf` временно необходимо удалить из проекта, а файл `fir_mult_accum.vhd` сделать его верхним уровнем иерархии. Осуществим компиляцию проекта с использованием синтезатора QIS (Quartus II Integrated Synthesis) (рис. 2). Файл `fir_mult_accum.vhd` демонстрирует структурный стиль языка VHDL. На рис. 3 показано RTL-представление проекта КИХ-фильтра в САПР Quartus II.

Дополнительно с помощью опции Start VQM Writer меню Start получим синтезированный нетлист проекта (Verilog Quartus

Mapping). VQM-файл является ограниченным подмножеством формата Verilog и представляет собой технологическое мэппирование (отображение) проекта с помощью сетевых примитивов в уникальный базис ПЛИС (пример 1). Его можно найти в папке `atom_netlist\fir_mult_accum.vqm`.

VQM-файл можно извлечь как на этапе анализа и синтеза, так и после полной компиляции, с учетом размещения и трассировки. В случае его повторного использования в проекте (предварительно файл `fir_mult_accum.vqm` необходимо добавить в проект и сделать верхним уровнем иерархии, а файл `fir_mult_accum.bdf` временно удалить из проекта) размещение, трассировка и временной анализ исключаются.

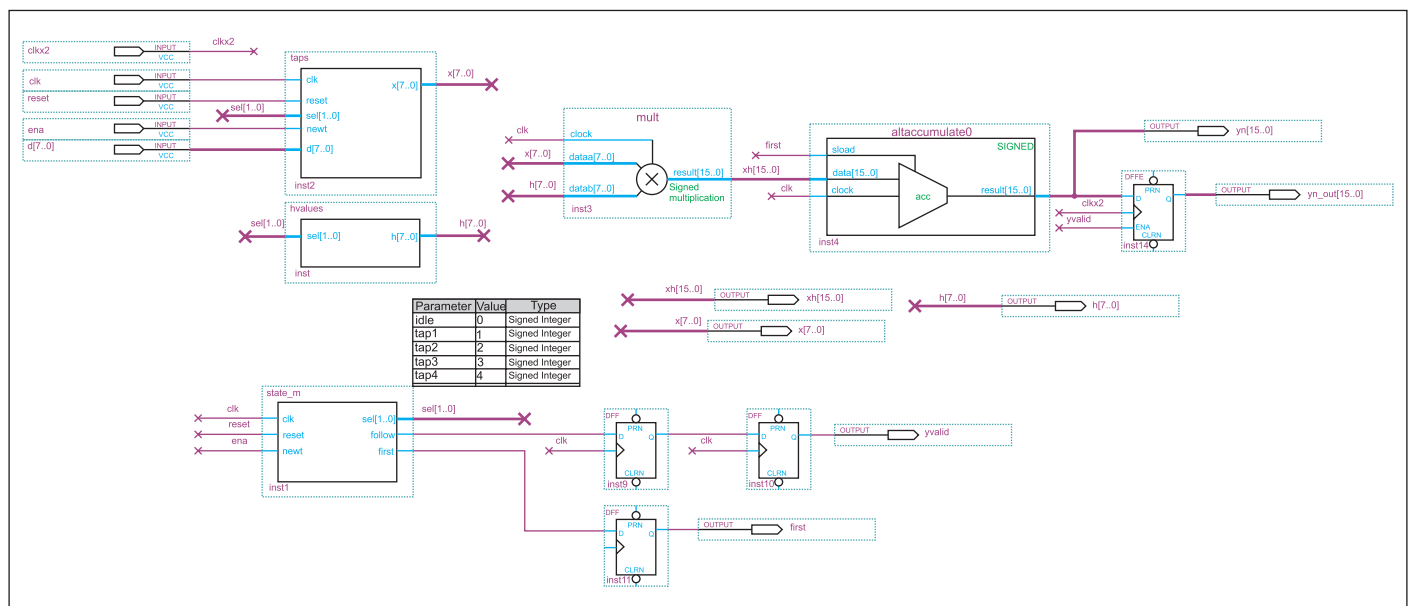


Рис. 1. Проект последовательного КИХ-фильтра на четыре отвода с использованием мегафункции умножения LPM_MULT и накопления ALTACCUMULATE

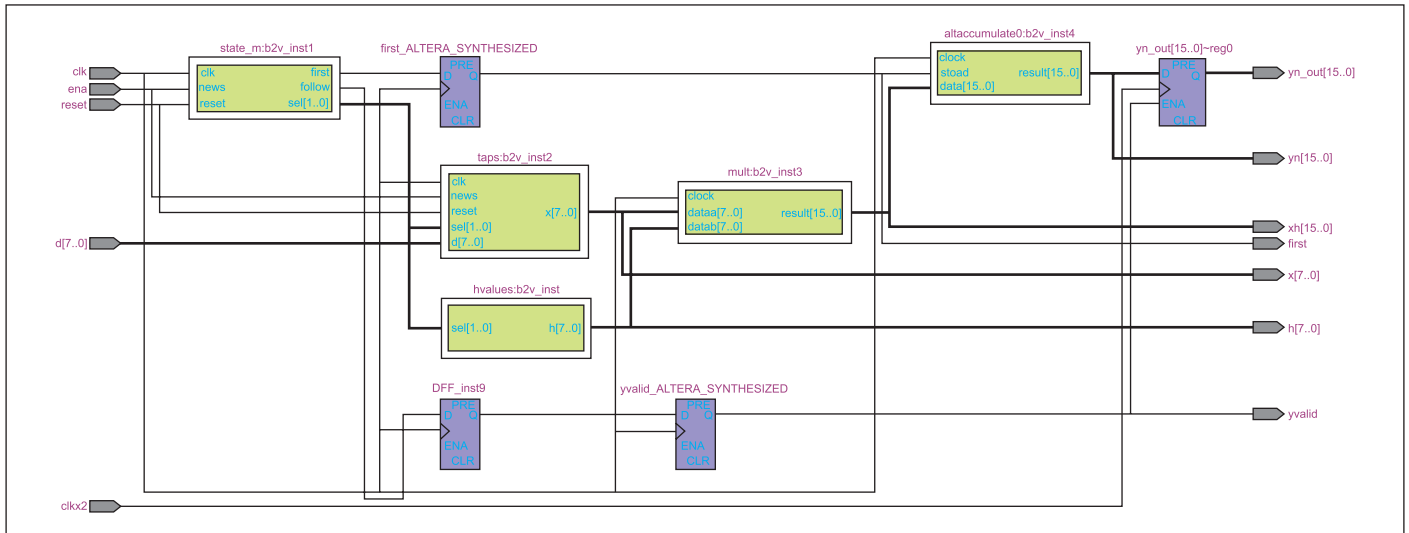


Рис. 3. RTL-представление проекта КИХ-фильтра в САПР Quartus II

```
// VENDOR "Altera"
// PROGRAM "Quartus II"
// VERSION "Version 9.1 Build 350 03/24/2010 Service Pack 2 SJ Web Edition"

// DATE "03/11/2016 23:58:08"

module fir_mult_accum (
    ena,
    reset,
    clk,
    clkx2,
    d,
    yvalid,
    first,
    h,
    x,
    xh,
    yn,
    yn_out);
input ena;
input reset;
input clk;
input clkx2;
input [7:0] d;
output yvalid;
output first;
output [7:0] h;
output [7:0] x;
output [15:0] xh;
output [15:0] yn;
output [15:0] yn_out;
wire \mult_b2v_inst3\lpm_mult:lpm_mult_component\mult_bis:auto_generated\mac_out2-0;
wire \mult_b2v_inst3\lpm_mult:lpm_mult_component\mult_bis:auto_generated\mac_out2-1;
wire \clk-ctrl;
wire \state_mb2v_inst1\Selector0-0;
wire \reset-ctrl;
wire \state_mb2v_inst1\filter.idle;
wire \state_mb2v_inst1\Selector1-0;
wire \state_mb2v_inst1\filter.tap1;
wire \state_mb2v_inst1\filter.tap2;
wire \state_mb2v_inst1\filter.tap3;
wire \state_mb2v_inst1\filter.tap4;
wire DFF_inst9;
```

Пример 1. Фрагмент VQM-файла, созданный САПР Quartus II

Осуществим анализ и синтез проекта с использованием программных средств сторонних разработчиков. Заходим в меню Assignments/Settings/EDA Tool Settings/Design Entry Synthesis и подключаем синтезатор Synplify Pro (рис. 4). В автоматическом режиме из проекта, представленного vhdл-файлом fir_mult_accum.vhd, с помощью Synplify Pro извлекаем VQM-файл, который будет назначен верхним уровнем иерархии (рис. 5,

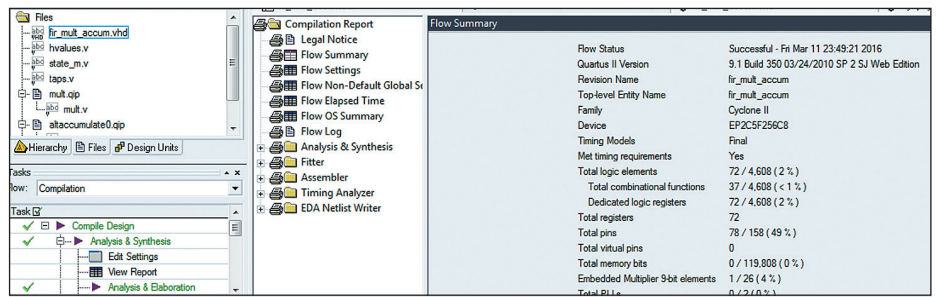


Рис. 2. Проект последовательного КИХ-фильтра и результаты компиляции с использованием QIS

пример 2). При этом автоматически создается проект в Synplify Pro, который можно найти в папке synplify_(имя проекта)_work. В нашем случае имя проекта fir_mult_accum. В папке synplify_fir_mult_accum_work содержится проект fir_mult_accum.prj.

На рис. 6 показано условное представление проектных файлов в Synplify Pro.

Умножитель и аккумулятор, созданные в Quartus II с помощью мегафункций LPM_MULT и ALTACCUMULATE, представляются как черные ящики (сетевые примитивы ПЛИС Cyclone II) и не раскрываются, а линия задержки, управляющий автомат и ПЗУ обрабатываются синтезатором самостоятельно. Верхним уровнем иерархии проекта яв-

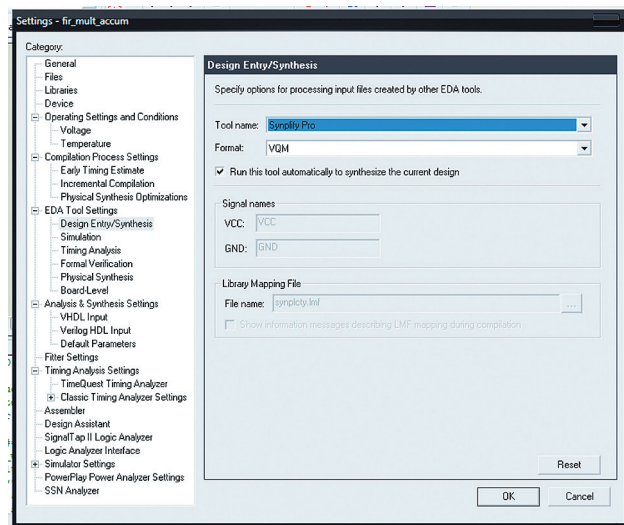


Рис. 4. Подключение синтезатора Synplify Pro с последующим извлечением VQM-файла

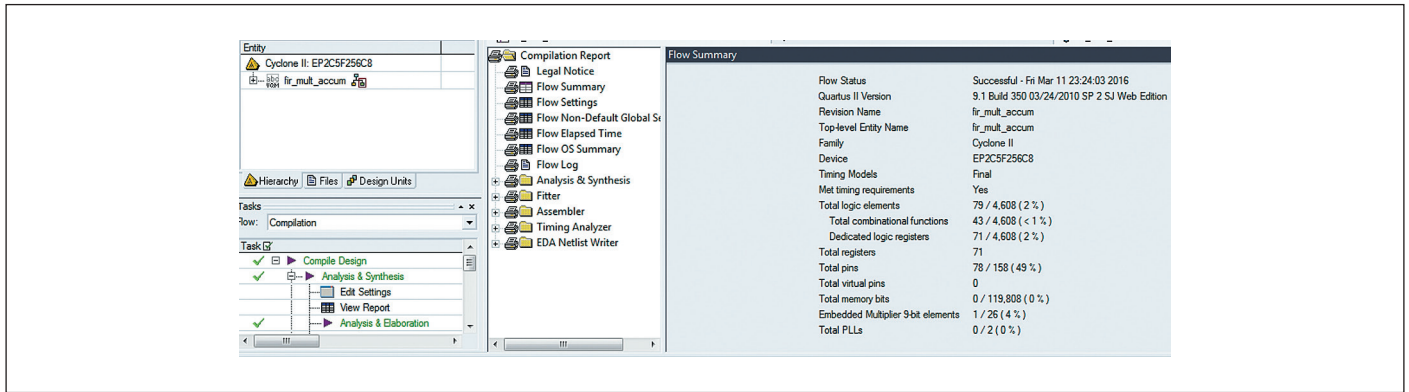


Рис. 5. Автоматически назначается верхним уровнем иерархии проекта VQM-файл, созданный с помощью Synplify Pro, и результаты компиляции

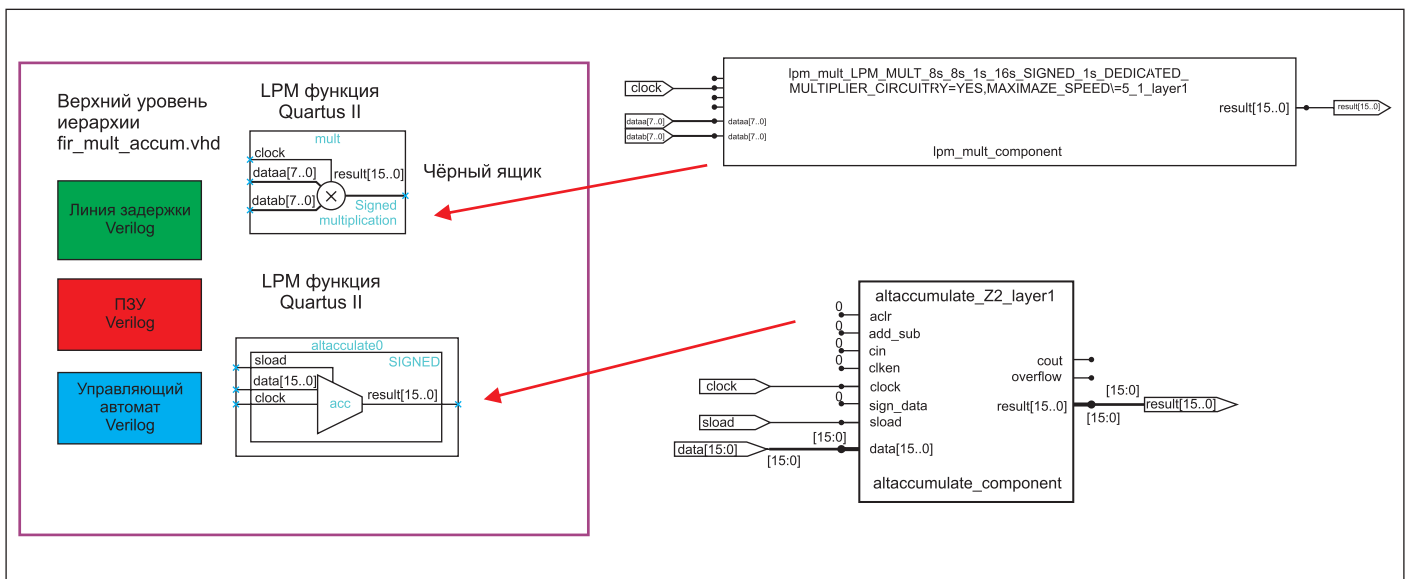


Рис. 6. Представление мегафункций LPM_MULT и ALTACCUMULATE в виде черных ящиков

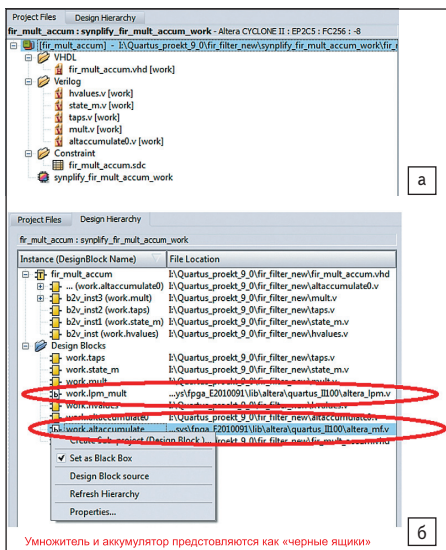


Рис. 7. Проект fir_mult_accum.prj в Synplify Pro: а) закладка «Файлы проекта»; б) закладка «Проектная иерархия»

```
// Written by Synplify
// Product Version "E-2010.09-1"
// Program "Synplify Pro", Mapper "maprc, Build 140R"
...

// VQM4.1+
// @13:53
lpm_mult lpm_mult_component (
    .result({xh_c_15, xh_c_14, xh_c_13, xh_c_12, xh_c_11,
            xh_c_10, xh_c_9, xh_c_8, xh_c_7, xh_c_6, xh_c_5, xh_c_4, xh_c_3,
            xh_c_2, xh_c_1, xh_c_0}),
    .dataa({G_29, G_26, G_23, G_20, G_17, G_14, G_11, G_8}),
    .datab({sel_i_a2_0, sel_i_a2_0, sel_i_a2_0, sel_i_a2_0,
            sel_i_a2_0, VCC, VCC, h_1_0_x2_0_0}),
    .sum(GND),
    .clock(clk_c),
    .clken(VCC),
    .aclr(GND)
);
defparam lpm_mult_component.lpm_hint = "DEDICATED_
MULTIPLIER_CIRCUTTRY=YES,MAXIMIZE_SPEED=5";
defparam lpm_mult_component.lpm_pipeline = 1;
defparam lpm_mult_component.lpm_representation = "SIGNED";
defparam lpm_mult_component.lpm_type = "LPM_MULT";
defparam lpm_mult_component.lpm_widtha = 8;
defparam lpm_mult_component.lpm_widthb = 8;
defparam lpm_mult_component.lpm_widthp = 16;
...
endmodule /* mult */
```

Пример 2. Фрагмент VQM-файла, созданный с помощью Synplify Pro

ляющего автомата (рис. 9). Используется кодирование с одним активным или горячим состоянием (one hot encoding, ONE), то есть в каждый конкретный момент времени активным (hot) может быть только один триггер состояния, поэтому управляющий автомат имеет структуру сдвигового регистра (рис. 10).

Сравним результаты компиляции с помощью синтезаторов QIS (рис. 2) и Synplify Pro (рис. 5). Число логических элементов и триггеров для QIS — 72 и 72 соответственно, а для Synplify Pro — 79 и 71 соответственно. LUT для реализации комбинационных функций 37 против 43 у Synplify Pro. Число триггеров сократилось на единицу.

По аналогичному принципу протестирован проект параллельного КИХ-фильтра на 24 отвода, представленный Verilog-кодом (академический САПР VTR707, тест производительности). Проект можно найти в папке vtr_flow\benchmarks\arithmetic\generated_circuits\FIR_filters\verilog\fir_pipe_24.v. За счет того что число умножителей с размерностью операндов 9×9 уменьшилось с 22 до 18, произошло перераспределение на логические

ляется файл *fir_mult_accum.vhd*, созданный с помощью Quartus II (рис. 7а). На рис. 7б показана проектная иерархия.

На рис. 8 показано RTL-представление проекта КИХ-фильтра в Synplify Pro. В качестве примера раскроем иерархию управ-

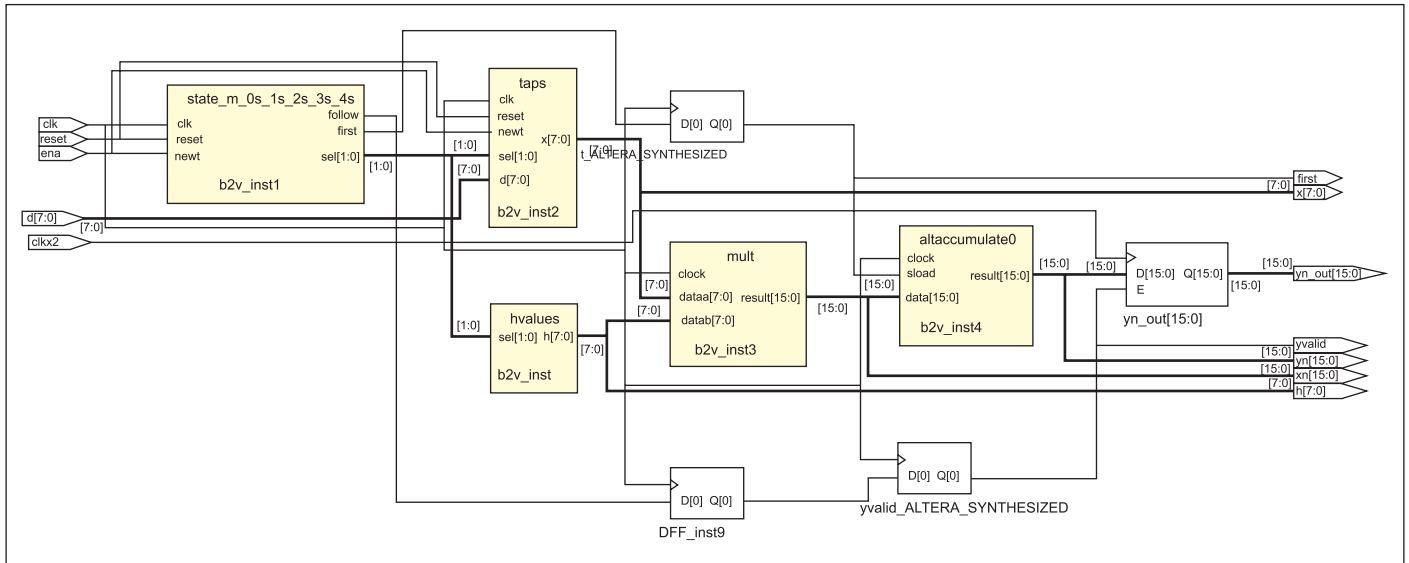


Рис. 8. RTL-представление проекта КИХ-фильтра в Synplify Pro

Таблица. Общие сведения по числу задействованных ресурсов ПЛИС Altera Cyclone II Device EP2C5F256C8

Проект	Число логических элементов (Total logic elements)	Триггеров логических элементов (Dedicated logic registers)	LUT для реализации комбинационных функций	Аппаратный умножитель с разрядностью операндов 9-9 (Embedded Multiplier 9-bit elements)	Рабочая частота F _{max} , МГц в наилучшем случае
Последовательный КИХ-фильтр на четыре отвода					
QIS	72	72	37	1	125,05
Synplify Pro	79	71	43	1	130,41
Параллельный КИХ-фильтр на двадцать четыре отвода					
QIS	1059	1058	367	22	157,11
Synplify Pro	1131	1095	439	18	116,02

ресурсы ПЛИС. Число используемых логических элементов и триггеров в проекте увеличилось, и как следствие, упала рабочая частота (таблица).

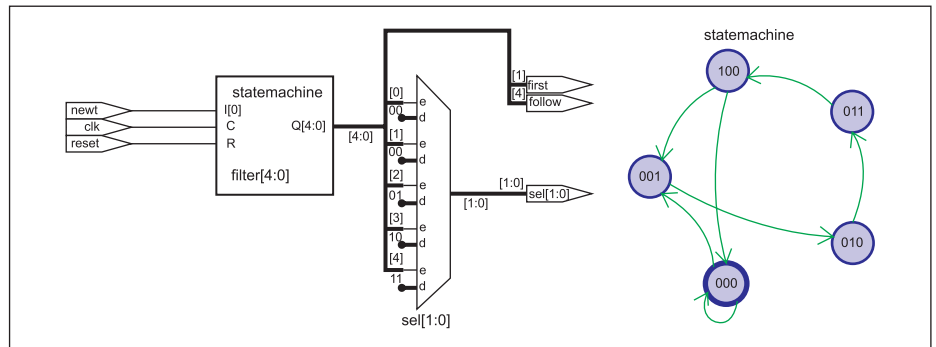


Рис. 9. Цифровой автомат с логикой формирования выхода и диаграмма состояний

Вариант 2. Запуск проектов в Quartus II непосредственно из Synplify Pro

Считается, что ресинтез технологического мэппинга с помощью специального стиля языка Verilog (Technology Independent

Coding Styles) позволяет сократить число используемых логических ресурсов ПЛИС на 4–5%. Применение VQM-файла, созданного с помощью Synplify Pro, позволяет исключить этап синтеза из маршрута проектирования ПЛИС Quartus II. Идея показана на рис. 11.

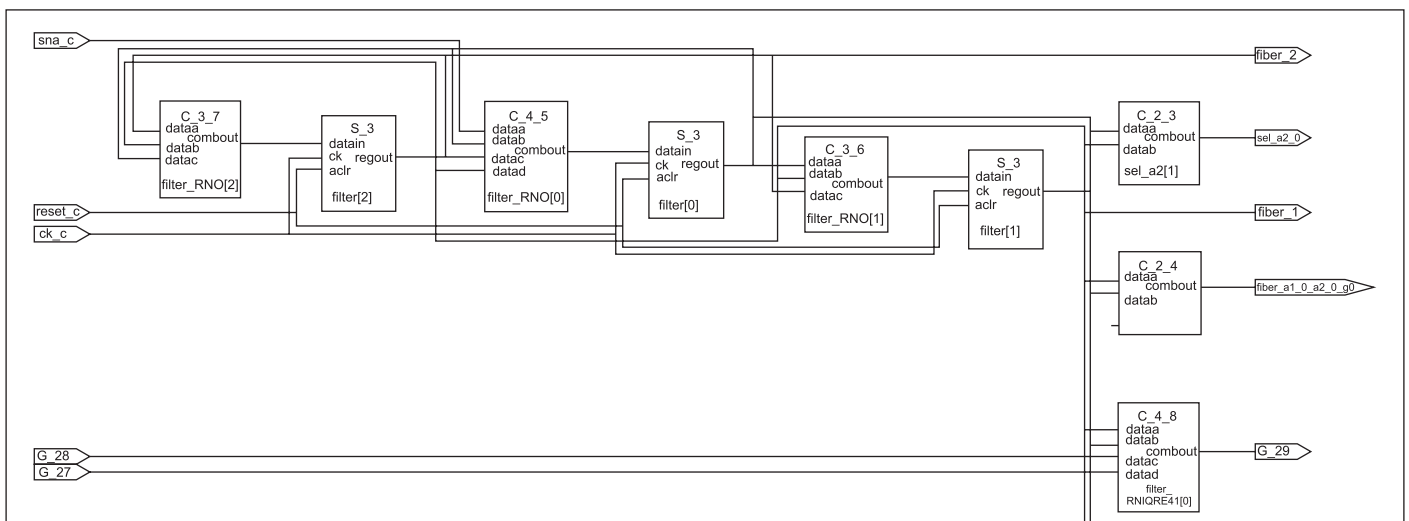


Рис. 10. Представление цифрового автомата на уровне логических элементов (технологическое мэппинг)

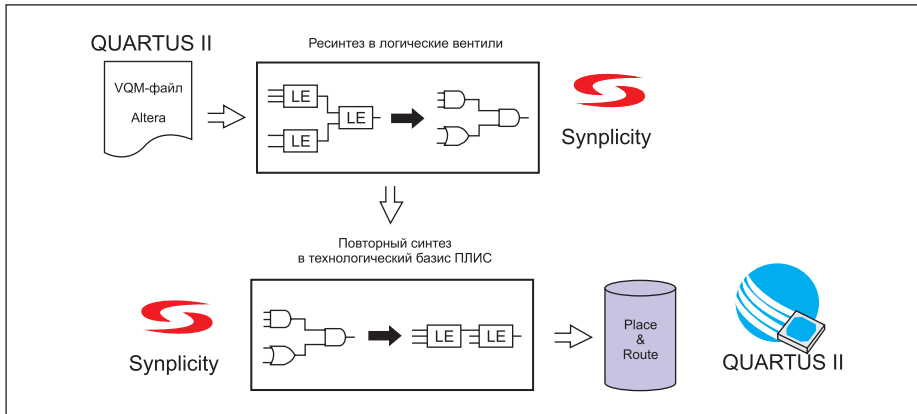


Рис. 11. Из VQM-файла проекта, созданного Quartus II, осуществляется ресинтез технологического мэппинга в логические вентили и последующий синтез логики в технологический базис ПЛИС синтезатором Synplify Pro с получением VQM-файла

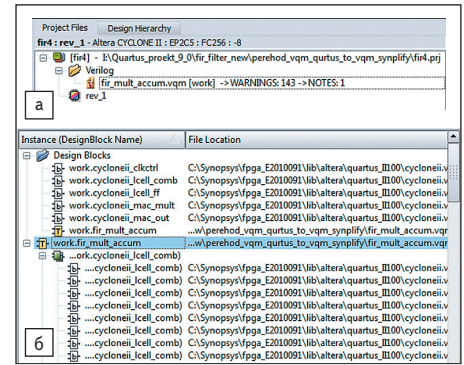


Рис. 12. Проект fir4.prj в Synplify Pro, созданный на основе единственного VQM-файла (fir_mult_accum.vqm):

- а) закладка «Файлы проекта»;
б) закладка «Проектная иерархия»

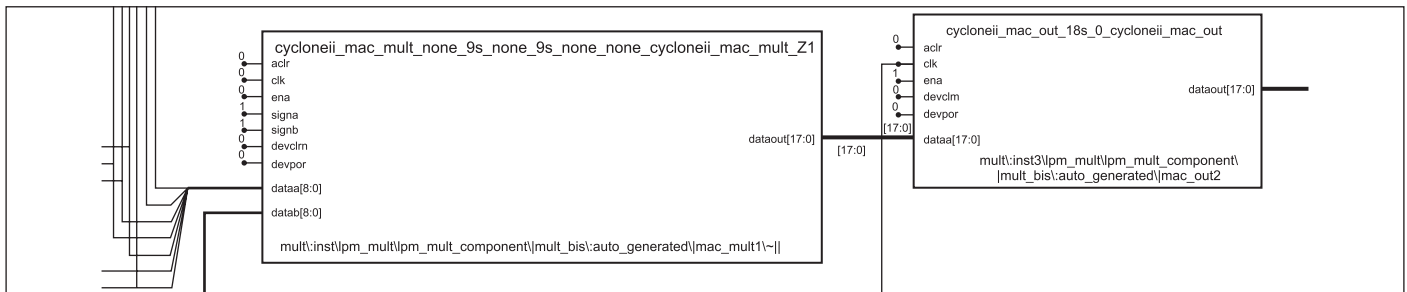


Рис. 13. RTL-представление проекта (фрагмент) КИХ-фильтра в Synplify Pro с использованием сетевых примитивов mac_mult и mac_out ПЛИС Cyclone II

Извлечем VQM-файл из схемного описания проекта (*fir_mult_accum.bdf*) в Quartus II. Далее создадим проект в Synplify Pro. Верхним уровнем иерархии назначим VQM-файл *fir_mult_accum.vqm*, созданный Quartus II (рис. 12а). На рис. 12б приводится список используемых сетевых примитивов ПЛИС Cyclone II, которые представляются как черные ящики. RTL-представление проекта показывает, что он полностью состоит из сетевых примитивов. На рис. 13 дано RTL-представление фрагмента проекта КИХ-фильтра в Synplify Pro с использованием сетевых примитивов *mac_mult* и *mac_out* ПЛИС Cyclone II.

Предъявим VQM-файл, созданный Synplify Pro САПР Quartus II, и осуществим компи-

ляцию проекта (меню **Options/Quartus II/Launch Quartus**). Число логических элементов и триггеров составит 72 и 72 соответственно, LUT для реализации комбинационных функций 37. Таким образом, никакого выигрыша в экономии ресурсов ПЛИС не получено.

Выводы

Синтезатор Synplify Pro осуществляет по VHDL-, Verilog-проектам (с привлечением мегафункций), созданным в Quartus II, технологическое мэппирование в базис ПЛИС фирмы Altera. При этом в некоторых случаях можно получить значительно более высокие рабочие частоты и меньшее число используемых логических элементов и ЦОС-блоков ПЛИС.

На примере простого проекта последовательного КИХ-фильтра на четыре отвода с применением мегафункции умножения *LPM_MULT* и накопления *ALTACCUMULATE* показано, что при использовании синтезатора Synplify Pro число LUT для реализации комбинационных функций увеличилось на шесть, а число триггеров уменьшилось на единицу.

Для параллельного КИХ-фильтра на 24 отвода, представленного Verilog-кодом, удалось уменьшить число аппаратных умножителей с 22 до 18, но число используемых логических элементов и триггеров в проекте увеличилось, и как следствие, упала рабочая частота с 157,11 до 116,02 МГц.

Литература

1. Строгонов А., Цыбин С., Городков П. Проектирование последовательных КИХ-фильтров в САПР ПЛИС Quartus II // Компоненты и технологии. 2016. № 1.
2. Строгонов А., Цыбин С., Городков П. Особенности использования двухпортовой памяти при проектировании последовательных КИХ-фильтров в САПР ПЛИС Quartus II // Компоненты и технологии. 2016. № 4.
3. Altera. 2004. Designing with Synplify Pro & Altera's Quartus II Software. SynplifyPro_QuartusII_Ver5_v4_1.pdf
4. Синтезатор Synplify в связке с Quartus Altera. www.acvarif.info/prvhdl/prvhdl21.html