

# Проектирование систолических КИХ-фильтров в базисе ПЛИС с помощью системы моделирования ModelSim-Altera

Андрей СТРОГОНОВ,  
д. т. н.  
andreistrogonov@mail.ru  
Алексей БЫСТРИЦКИЙ,  
к. т. н.  
bystritsky@bk.ru

В статье рассмотрены основные особенности проектирования цифровых фильтров на примере систолического КИХ-фильтра в САПР ПЛИС Quartus II версии 11.1 Web Edition. Начиная с версии 10.0 из САПР Quartus II исключен векторный редактор, а моделирование предлагается вести с помощью различных симуляторов высокоуровневых языков описания аппаратных средств, например Active-HDL, Riviera-Pro, ModelSim и др. В качестве свободно распространяемого симулятора с ограниченными возможностями пользователю предлагается использовать систему моделирования ModelSim-Altera Free. Применение системы моделирования ModelSim активно обсуждалось не только на страницах журнала «Компоненты и технологии» [1, 2], но и на форумах разработчиков РЭА на ПЛИС [3–5]. Отдельного внимания заслуживает открытый проект «Марсход» [4].

Рассмотрим уравнение КИХ-фильтра (нерекурсивного цифрового фильтра с конечно-импульсной характеристикой), которое представляется как арифметическая сумма произведений:

$$y = \sum_{k=0}^{K-1} c_k x_k, \quad (1)$$

где  $y$  — отклик цепи;  $x_k$  —  $k$ -я входная переменная;  $c_k$  — весовой коэффициент  $k$ -й входной переменной, который является постоянным для всех  $n$ ;  $K$  — число отводов фильтра.

Разработаем модель систолического фильтра на четыре отвода  $y = C_0x_0 + C_1x_1 + C_2x_2 + C_3x_3$  в САПР ПЛИС Quartus II версии 11.1 в базисе ПЛИС серии Cyclone II с однотипными процессорными элементами (рис. 1 и 2). ПЛИС серии Cyclone II выбраны из тех соображений, что в САПР ПЛИС Quartus II Web Edition не поддерживаются ПЛИС серий Cyclone V и Stratix V.

Дадим произвольное имя файлу верхнего уровня проектной иерархии — *poly\_syst\_main.bdf*. Коэффициенты фильтра и входные отсчеты возьмем такие же, как

в работах [6–8]. Предположим, что коэффициенты фильтра — целочисленные, со знаком, и они известны:  $C_0 = -2$ ,  $C_1 = -1$ ,  $C_2 = 7$  и  $C_3 = 6$ . На вход КИХ-фильтра поступают входные отсчеты  $-5, 3, 1$  и  $0$ . Правильные значения на выходе фильтра:  $10, -1, -40, -10, 26, 6$  и т. д., то есть согласно формуле (1). Дополнительную информацию можно получить в работе [8].

На рис. 3 показаны временные диаграммы работы систолического КИХ-фильтра на четыре отвода, реализованного в базисе ПЛИС Stratix III, с однотипными процес-

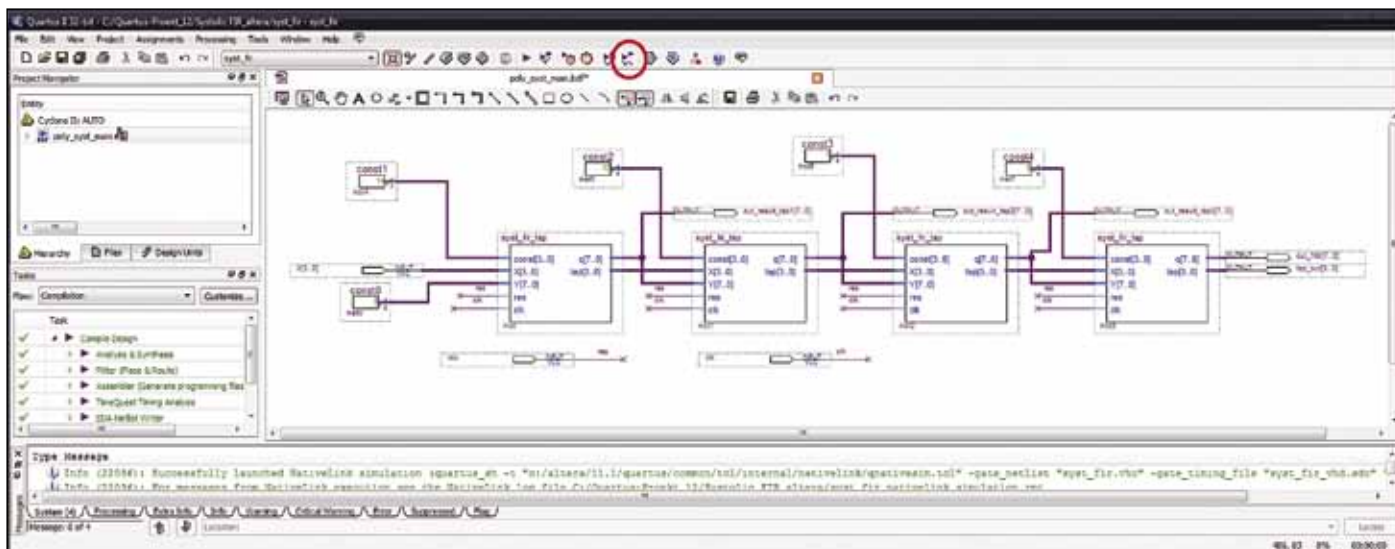


Рис. 1. Систолический КИХ-фильтр на четыре отвода в САПР ПЛИС Quartus II версии 11.1 с однотипными процессорными элементами

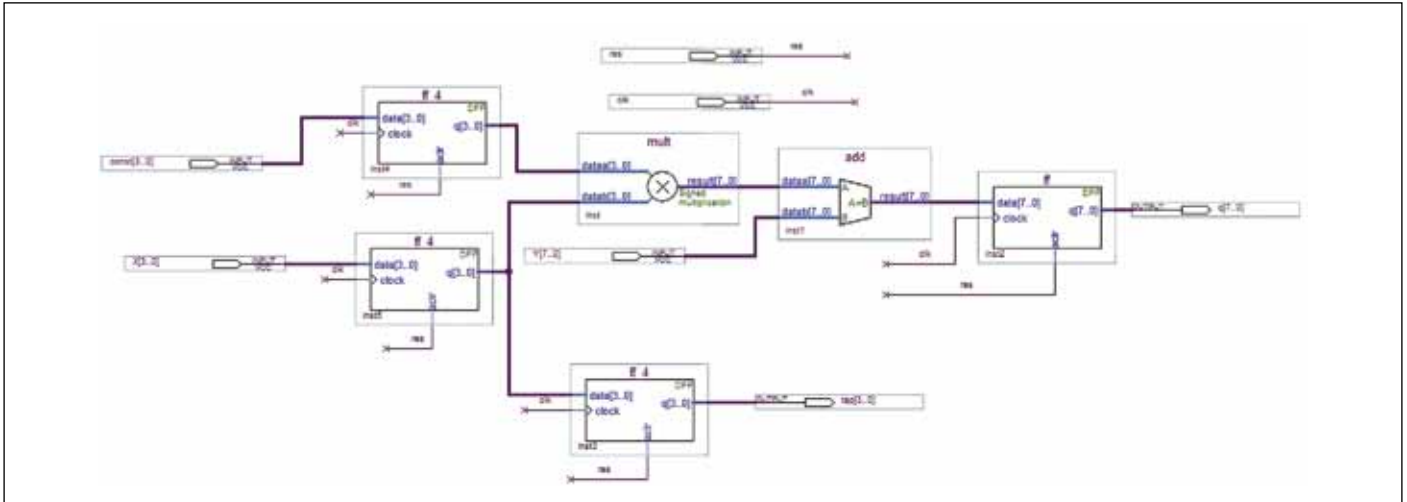


Рис. 2. Процессорный элемент

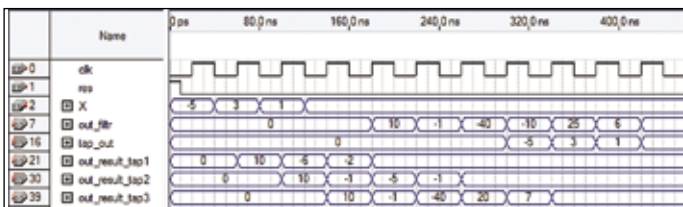


Рис. 3. Временные диаграммы работы систолического КИХ-фильтра на четыре отвода с однотипными процессорными элементами в САПР ПЛИС Quartus II версии 9.1

сорными элементами в САПР ПЛИС Quartus II версии 9.1 из работы [8].

Рассмотрим моделирование КИХ-фильтра с использованием симулятора ModelSim-Altera Starter Edition. Предварительно его необходимо подключить — с помощью меню **Assignments/settings/EDA Tool settings**. В поле **Tool name** САПР ПЛИС Quartus II версии 11.1 нужно выбрать симулятор ModelSim-Altera, а в поле **Format Netlist Writer settings** указать выходной формат «нетлиста» — язык VHDL (рис. 4).

Предварительно следует создать текстовый сценарий функционирования систолического КИХ-фильтра на основе структурного стиля языка VHDL («тестбенч»). В качестве промежуточного шаблона, ко-

торый нужно отредактировать, можно взять файл *poly\_syst\_main.vht*. Для этого необходимо его сформировать: меню **Processing/Start/Start Test Bench Template Writer** (пример 1).

Отредактируем объект *poly\_syst\_main\_vhd\_tst*, который основан на компоненте *poly\_syst\_main*, и сохраним его под именем *test\_poly\_syst\_main.vhd* (пример 2).

После компиляции в САПР Quartus II при нажатии кнопки EDA Gate Level Simulation (моделирование на уровне вентилях, временная модель Slow Model) автоматически должен запуститься симулятор ModelSim-Altera (на рис. 1 пиктограмма кнопки отмечена красным кружком).

Возможны два варианта. Рассмотрим первый вариант с созданием «тестбенча». Для этого необходимо откомпилировать файл *test\_poly\_syst\_main.vhd* с помощью меню Compile симулятора ModelSim. После компиляции в рабочей библиотеке work должны появиться два объекта: *poly\_syst\_main* и *test\_poly\_syst\_main* (рис. 5). Двойным щелчком мыши по объекту *test\_poly\_syst\_main* автоматически запускаются различные вспомогательные окна (рис. 6).

Помещаем курсор в окно **Objects**, нажимаем на правую кнопку мыши меню **Add/To Wave/Signals in Region**, и в окне **Wave** появляется список сигналов проекта. Далее целесообразно настроить окно **Wave**, в котором отображаются временные диаграммы рабо-

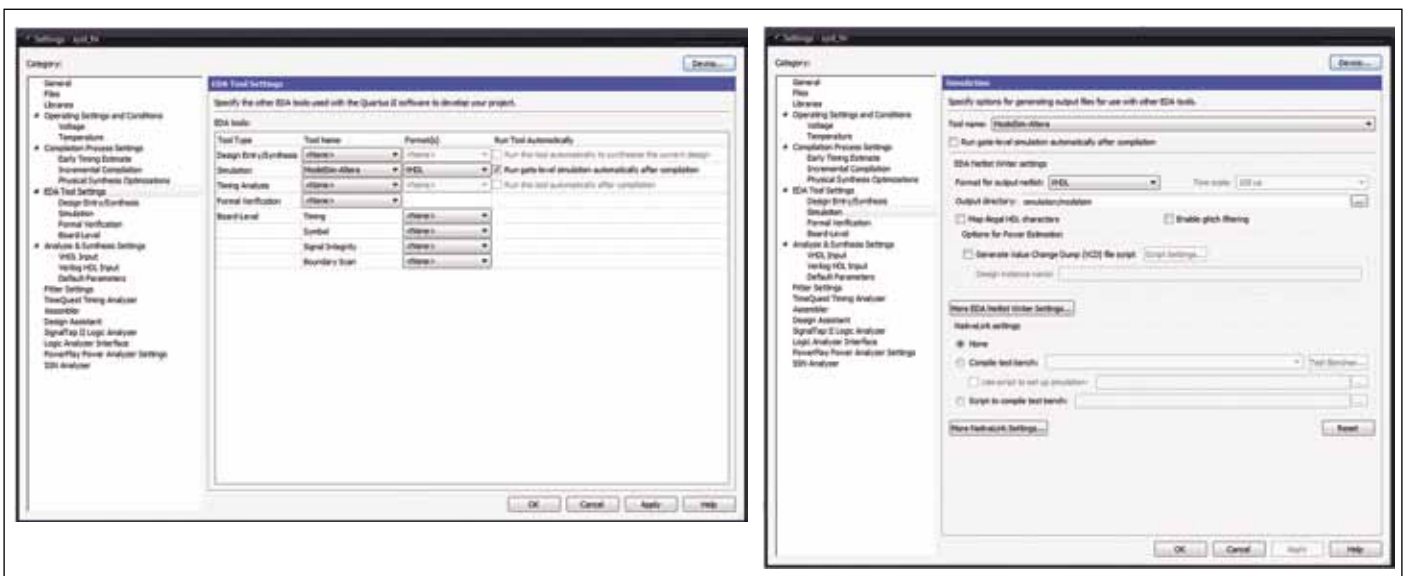


Рис. 4. Настройки закладки EDA Tool Settings САПР ПЛИС Quartus II версии 11.1

```

-- Vhdl Test Bench template for design : poly_syst_main
--
-- Simulation tool : ModelSim-Altera (VHDL)

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY poly_syst_main_vhd_tst IS
END poly_syst_main_vhd_tst;
ARCHITECTURE poly_syst_main_arch OF poly_syst_main_vhd_tst IS
-- constants
-- signals
SIGNAL clk : STD_LOGIC;
SIGNAL out_filt1 : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL out_result_tap1 : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL out_result_tap2 : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL out_result_tap3 : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL res : STD_LOGIC;
SIGNAL tap_out : STD_LOGIC_VECTOR(3 DOWNTO 0);
SIGNAL X : STD_LOGIC_VECTOR(3 DOWNTO 0);
COMPONENT poly_syst_main
PORT (
clk : IN STD_LOGIC;
out_filt1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
out_result_tap1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
out_result_tap2 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
out_result_tap3 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
res : IN STD_LOGIC;
tap_out : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
X : IN STD_LOGIC_VECTOR(3 DOWNTO 0)
);
END COMPONENT;
BEGIN
i1 : poly_syst_main
PORT MAP (
-- list connections between master ports and signals
clk => clk,
out_filt1 => out_filt1,
out_result_tap1 => out_result_tap1,
out_result_tap2 => out_result_tap2,
out_result_tap3 => out_result_tap3,
res => res,
tap_out => tap_out,
X => X
);
init : PROCESS
-- variable declarations
BEGIN
-- code that executes only once
WAIT;
END PROCESS init;
always : PROCESS
-- optional sensitivity list
-- ( )
-- variable declarations
BEGIN
-- code executes for every event on sensitivity list
WAIT;
END PROCESS always;
END poly_syst_main_arch;

```

**Пример 1.** Шаблон теста систолического КИХ-фильтра на языке VHDL (poly\_syst\_main.vht)

ты фильтра. Нажимаем **Simulate\Runtime Options**. В поле **Default Radix** нажимаем

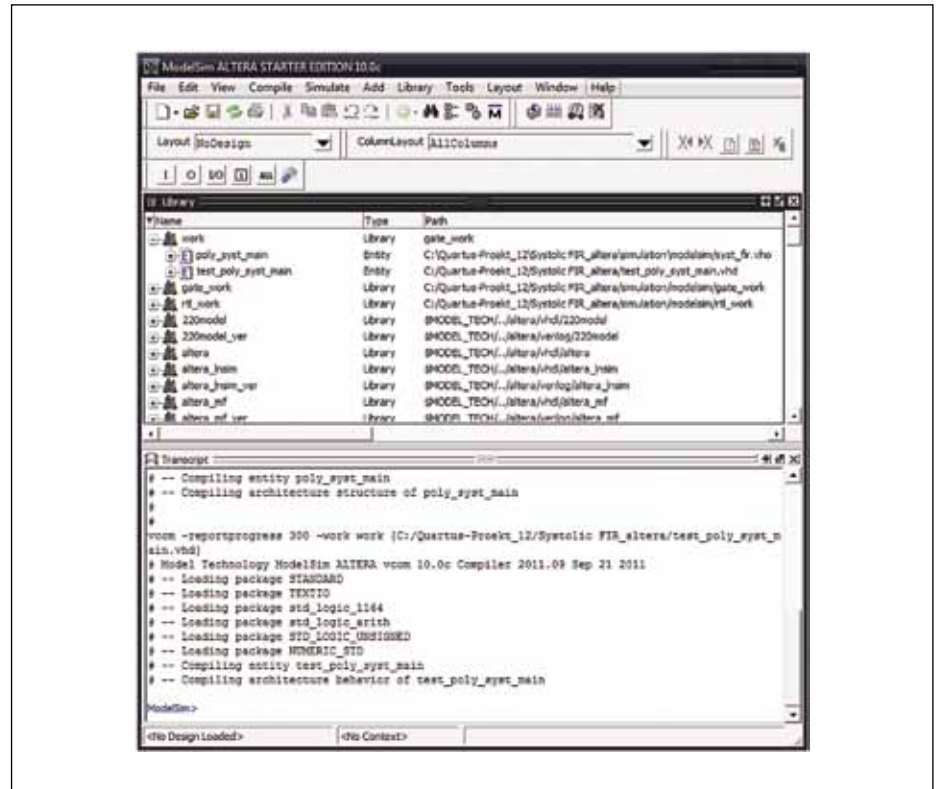


Рис. 5. Создание рабочей библиотеки work, в которую помещается два объекта: poly\_syst\_main и test\_poly\_syst\_main

кнопку **Decimal**, что позволяет перейти от двоичной системы счисления, представленной в «тестбенче», к десятичной со знаком. В поле **Default Run** задаем шаг моделирования 100 нс. Последовательно нажимая на пиктограмму кнопки **Run**, получим временные диаграммы работы КИХ-фильтра с шагом 100 нс (рис. 6). Сравниваем полученные результаты с временными диаграммами на рис. 3 и убеждаемся в правильности работы систолического КИХ-фильтра на четыре отвода в базе ПЛИС Cyclone II.

Рассмотрим второй вариант, без использования «тестбенча». Для этого нам нужен файл *poly\_syst\_main.vhd*, а задание на моделирование сформируем непосредственно

в векторном редакторе (окно **Wave**) с помощью специальных инструментов **Clock** и **Force**. Двойным щелчком мыши по объекту *poly\_syst\_main* автоматически запускаются различные окна. Помещаем курсор в окно **Objects**, как и в первом варианте, нажимаем на правую кнопку мыши меню **Add/To Wave/Selected Signals** и выбираем только интересующие нас сигналы.

В окне **Wave** выбираем синхросигнал *clk* и нажимаем на правую кнопку мыши, выбираем меню **Clock**. В окне **Define Clock** в полях задается период синхросигнала — 100 нс, коэффициент заполнения — 50, уровни «лог. 1» и «лог. 0». Радиокнопкой активизируем передний фронт синхросигнала (рис. 7).



Рис. 6. Временные диаграммы работы систолического КИХ-фильтра на четыре отвода с однотипными процессорными элементами в симуляторе ModelSim-Altera версии 10.0

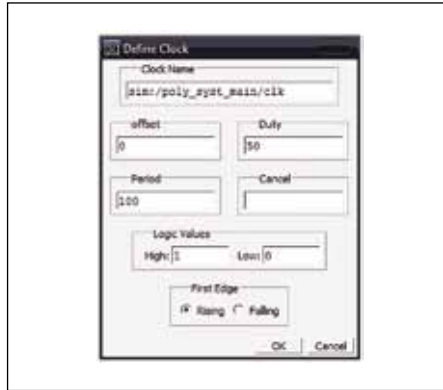
```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;
ENTITY test_poly_syst_main IS
END test_poly_syst_main;

ARCHITECTURE behavior OF test_poly_syst_main IS
COMPONENT poly_syst_main
PORT(
res : IN STD_LOGIC;
clk : IN STD_LOGIC;
X : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
out_filt1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
out_result_tap1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
out_result_tap2 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
out_result_tap3 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
tap_out : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END COMPONENT;
--Inputs
SIGNAL clk : std_logic := '0';
SIGNAL res : std_logic := '1';
SIGNAL X_in : STD_LOGIC_VECTOR(3 DOWNTO 0) := "1011";
--Outputs
SIGNAL out_systol_filt1 : std_logic_VECTOR(7 DOWNTO 0);
SIGNAL out_result_tap1 : std_logic_vector(7 downto 0);
SIGNAL out_result_tap2 : std_logic_vector(7 downto 0);
SIGNAL out_result_tap3 : std_logic_vector(7 downto 0);
SIGNAL tap_out : std_logic_vector(3 downto 0);
BEGIN
    uut: poly_syst_main PORT MAP(
        clk => clk,
        res => res,
        X => X_in,
        out_filt1 => out_systol_filt1,
        out_result_tap1 => out_result_tap1,
        out_result_tap2 => out_result_tap2,
        out_result_tap3 => out_result_tap3,
        tap_out => tap_out
    );
    process
    begin
        clk <= '0';
        wait for 50 ns;
        clk <= '1';
        wait for 50 ns;
    end process;
    process
    begin
        wait for 125 ns;
        res <= '0';
    end process;
    tb : process
    begin
        wait for 100 ns;
        X_in <= "1011";
        wait for 100 ns;
        X_in <= "0011";
        wait for 100 ns;
        X_in <= "0001";
        wait for 100 ns;
        X_in <= "0000";
        wait;
    END process;
END;

```

**Пример 2.** Тест систолического КИХ-фильтра на языке VHDL (test\_poly\_syst\_main.vhd)



**Рис. 7.** Окно настройки тактового синхросигнала clk



**Рис. 8.** Настройка сигнала сброса res



**Рис. 9.** Настройка сигнала X

Сигнал асинхронного сброса res (активный — высокий уровень) внутренних регистров процессорных элементов зададим равным нулю с помощью меню **Force**. Радиокнопкой отмечаем вид действия над сигналом — **Freeze** («замороженный»), в поле **Value** зададим «лог. 0», то есть на всем временном промежутке моделирования сигнал сброса res будет неактивным (рис. 8).

Выбираем сигнал X и с помощью меню **Force** задаем десятичное число -5. Затем нажимаем на пиктограмму кнопки **Run** для осуществления моделирования с шагом 100 нс. Далее будем изменять только значения, поступающие на вход X (рис. 9), с помощью меню **Force** и последовательно нажимать на кнопку **Run** до получения нужных откликов (пример 3 и рис. 10). Задание для моделирования (пример 3) можно использовать, повторно сохранив его в текстовый файл.

В заключение отметим, что в версии 13.0 вновь появился встроенный векторный редактор с собственной системой моделирования [4]. Создать векторный файл можно с помощью меню **File/New/Verification/Debugging Files/University Program VWF**. Запуск моделирования осуществляется непосредственно из окна **Simulation Waveform Editor** с по-

мощью меню **Simulation/Run Functional Simulation**.

На рис. 11 показаны временные диаграммы работы систолического КИХ-фильтра на четыре отвода в САПР ПЛИС Quartus II версии 13.0. Проект размещен в ПЛИС серии MAX II.

```

force -freeze sim:/poly_syst_main/res 0 0
force -freeze sim:/poly_syst_main/clk 1 0, 0 {50000 ps} -r {100 ns}
force -freeze sim:/poly_syst_main/X -5 0
run
force -freeze sim:/poly_syst_main/X 3 0
run
force -freeze sim:/poly_syst_main/X 1 0
run
force -freeze sim:/poly_syst_main/X 0 0

```

**Пример 3.** Задание для моделирования из окна Transcript



**Рис. 10.** Окно Wave с результатами моделирования КИХ-фильтра

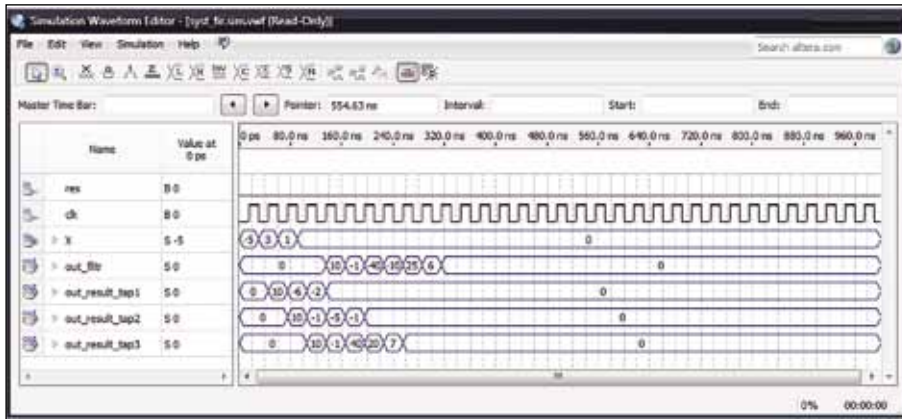


Рис. 11. Временные диаграммы работы систолического КИХ-фильтра на четыре отвода с однотипными процессорными элементами в САПР ПЛИС Quartus II версии 13.0

## Выводы

Использование текстового сценария на языке VHDL совместно с симулятором ModelSim-Altera Free позволяет отлаживать сложные проекты в кратчайшие сроки. ■

## Литература

1. Зотов В. ModelSim — система HDL-моделирования цифровых устройств // Компоненты и технологии. 2002. № 6.
2. Каршенбойм И. Краткий курс HDL. Ч. 8. Моделирование в ModelSim SE // Компоненты и технологии. 2008. № 11.
3. Сообщество easyelectronics.ru. Quartus + ModelSim. Интеграция в картинках и создание TestBench — <http://we.easyelectronics.ru/plis/quartus-modelsim-integraciya-v-kartinkah-i-sozdanie-testbench.html>
4. Блог о плате-конструкторе «Марсоход», САПР Altera Quartus II, Verilog — [www.marsohod.org/index.php/projects/plata1](http://www.marsohod.org/index.php/projects/plata1)
5. Библиотека Антона Егорова. Quartus II по-русски — <http://naliwator.narod.ru/links.html>
6. Строгонов А., Быстрицкий А. КИХ-фильтр на распределенной арифметике: проектируем сами // Компоненты и технологии. 2013. № 3.
7. Строгонов А., Быстрицкий А. Проектирование параллельных КИХ-фильтров в базе ПЛИС // Компоненты и технологии. 2013. № 6.
8. Строгонов А., Быстрицкий А. Систолические КИХ-фильтры в базе ПЛИС // Компоненты и технологии. 2013. № 8.