

Диверсификация проекта последовательного КИХ-фильтра в базисе ПЛИС Cyclone IV от компании Altera

А. Строгонов, д. т. н.¹, И. Семейкин, к. т. н.²

УДК 621.3.049.774 | ВАК 05.27.01

При разработке сложных проектов на базе отечественных ПЛИС часто возникают трудности, связанные с ограниченными ресурсами и быстродействием. Например, для создания последовательного КИХ-фильтра на основе ПЛИС требуется всего один умножитель и аккумулятор, однако при реализации этой схемы в составе сложного проекта в базисе отечественных ПЛИС разработчик может столкнуться с недостаточным количеством блоков встроенной памяти или ЦОС-блоков, при этом тактовая частота отечественных ПЛИС не превышает 250 МГц. В статье рассматривается диверсификация (разнообразие схемных решений) проекта последовательного КИХ-фильтра в базисе ПЛИС серии Cyclone IV компании Altera на основе демонстрационного примера `fir_filter` в САПР Quartus II в условиях ограничения аппаратных ресурсов, предложены пути преодоления этих ограничений.

Параллельная структура позволяет вычислять результат фильтрации за один такт синхроимпульса. Параллельные фильтры обеспечивают наивысшую производительность, но требуют значительных аппаратных ресурсов ПЛИС. Механизм конвейерной обработки позволяет создавать фильтры с частотой от 120 до 300 МГц и выше.

На рис. 1 показана структура последовательного КИХ-фильтра на четыре отвода с использованием одного блока умножения и накопления (МАС-блока). КИХ-фильтры, использующие в своей структуре блоки умножения и накопления, в зарубежной литературе называют МАС FIR filter (МАС КИХ-фильтры) [1, 2]. В структуре простейшего (single) МАС КИХ-фильтра, в отличие от полного параллельного фильтра, используется один умножитель с аккумулятором. Такой компромиссный вариант снижает аппаратные затраты в N раз, но также снижает и пропускную способность фильтра на тот же коэффициент. В МАС КИХ-фильтрах пропускная способность выборки обратно пропорциональна количеству его отводов, то есть при увеличении длины фильтра частота дискретизации

системы пропорционально уменьшается. МАС-блок может быть реализован как на аппаратных ЦОС-блоках, так и на логических ресурсах ПЛИС.

Отказаться от использования аппаратных умножителей ЦОС-блоков позволяет распределенная арифметика [3]. В этом случае частота дискретизации КИХ-фильтра уже не будет зависеть от его длины. Например, в мегаядре

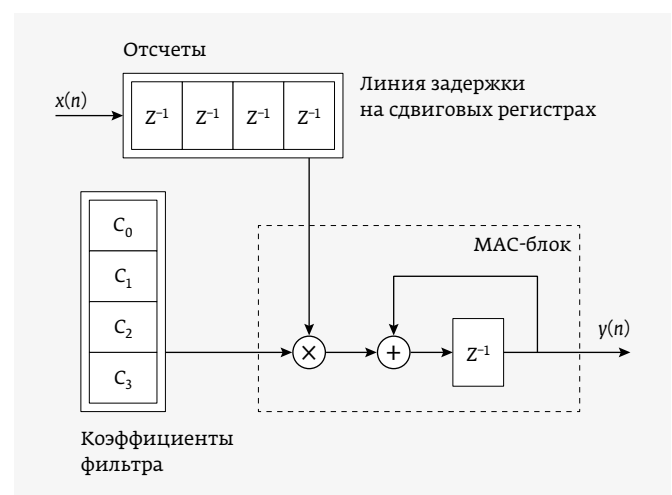


Рис. 1. Блок-схема КИХ-фильтра на четыре отвода с использованием одного МАС-блока

¹ Воронежский государственный технический университет, профессор кафедры полупроводниковой электроники и нанoeлектроники, тел. +7 910 247-14-70, andreistrogonov@mail.ru.

² АО «НИИЭТ», технический директор, igorsemeykin@list.ru.

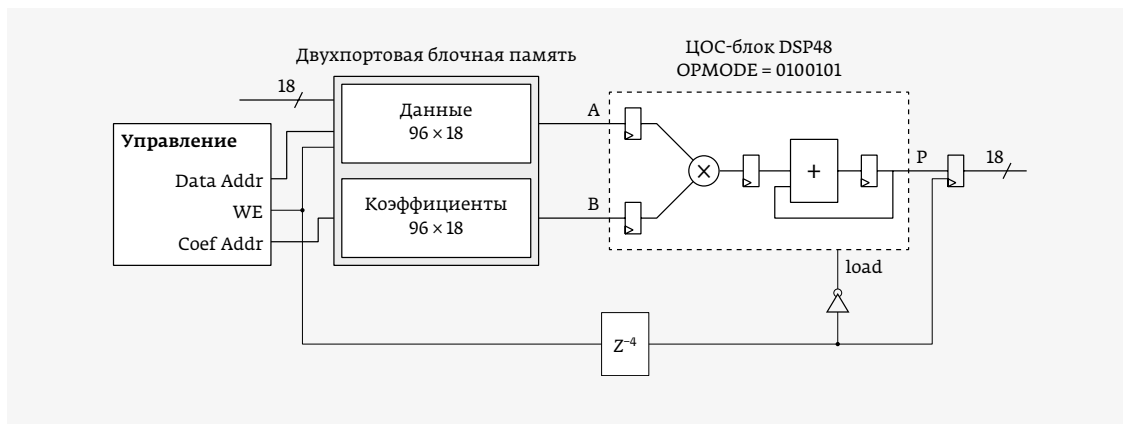


Рис. 2. Блок-схема КИХ-фильтра с использованием ЦОС-блока DSP48 ПЛИС Virtex-4

(MegaCore) Altera FIR Compiler и в генераторе параметризованных ядер Xilinx CORE Generator реализована поддержка параллельной и последовательной распределенной арифметики.

В случае низкой частоты дискретизации сигнала и большого количества коэффициентов КИХ-фильтра последовательная структура MAC КИХ-фильтра является оптимальной, а использование двухпортовой блочной памяти ПЛИС в конфигурации смешанного режима позволяет реализовать линию задержки на ОЗУ (циклический буфер входных данных) и ПЗУ для хранения коэффициентов фильтра.

На рис. 2 показан КИХ-фильтр с использованием ЦОС-блока DSP48 ПЛИС Xilinx Virtex-4 для реализации операций умножения и накопления и блочной памяти ПЛИС [2]. Данные записываются и считываются из порта А (режим ОЗУ), а коэффициенты считываются только из порта В (режим ПЗУ). В отличие от структуры параллельного КИХ-фильтра требуется наличие управляющего автомата (блок «Управление»), который обеспечивает необходимую адресную логику для организации циклического буфера ОЗУ для порта А (линия задержки) и В (ПЗУ), а также вырабатывает сигналы разрешения загрузки данных и коэффициентов в MAC-блок и захвата в регистр результата в случае, если результат накопления не может быть немедленно использован в последующей обработке. Также необходим учет латентности работы основных узлов фильтра. Так для формирования корректных значений на выходе MAC-блока и их последующего захвата регистром результата необходимо сигнал WE управляющего автомата задержать на четыре такта синхроимпульса.

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ SINGLE MAC КИХ-ФИЛЬТРОВ В БАЗИСЕ ПЛИС ALTERA

Рассматриваемый проект fir_filter последовательно КИХ-фильтра на четыре отвода представлен в САПР Altera Quartus II Version 13.1. Проект можно найти по ссылке altera/13.1/quartus/designs/fir_filter. Кроме того, проект

доступен и в более ранних САПР Altera Quartus II, а также в старших версиях САПР Quartus Prime.

В учебном пособии [4] подробно рассматривается этот же проект для реализации в ПЛИС серии APX20K EP20K100QC208-1 с использованием анализа временных характеристик в САПР Quartus II. В оригинальном представлении его можно найти в Quartus Tutorial version 1999.10 Revision 2.

Обратим внимание лишь на некоторые моменты проекта fir_filter. В проекте fir_filter входной сигнал и коэффициенты фильтра представлены с 8- и 3-битной точностью, а умножитель настроен на выполнение операций с числами без знака. Произведение представляется с 11-разрядной точностью, а аккумулятор выдает результат с 8-разрядной точностью.

Умножитель разработан на мегафункции lpm_mult, а аккумулятор описан на языке Verilog. Умножитель представлен символом, а линия задержки, коэффициенты, аккумулятор и управляющий автомат – в виде блок-схем. Умножитель настроен для работы с числами без знака в асинхронном (умножитель без синхронизации), а сумматор в синхронном режиме.

Проект использует многотактовую синхронизацию (два синхросигнала clk и clkx2). Синхросигнал clk используется для тактирования линией задержки и внутреннего регистра аккумулятора, а clkx2 – для тактирования регистром результата. Период синхросигнала clk в два раза больше clkx2.

Линия задержки (taps.v), ПЗУ для хранения коэффициентов (hvalues.v), управляющий автомат (state_m) и аккумулятор (acc.v) в оригинальном проекте представлены на языке Verilog. Verilog-коды функциональных блоков представлены в учебном пособии [4].

Вариант 1

Этот вариант последовательного КИХ-фильтра на четыре отвода с использованием мегафункции умножения LPM_MULT (асинхронный режим) и накопления ALTACCUMULATE.

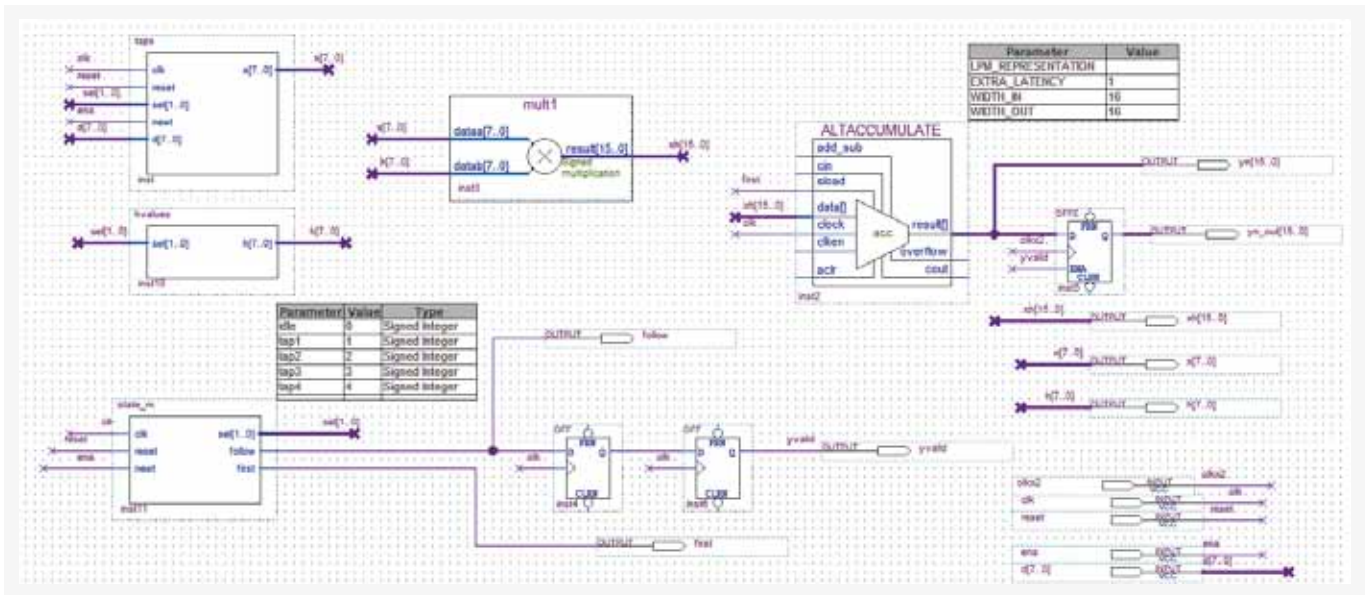


Рис. 3. Проект последовательного КИХ-фильтра на четыре отвода с использованием мегафункции умножения LPM_MULT (асинхронный режим) и накопления ALTACCUMULATE

Адаптируем оригинальный проект fir_filter к задаче проектирования последовательного КИХ-фильтра на четыре отвода [5–7] $y = C_0x_0 + C_1x_1 + C_2x_2 + C_3x_3$ со следующими коэффициентами $C_0 = -2, C_1 = -1, C_2 = 7, C_3 = 6$ в базе ПЛИС Cyclone IV EP4CGX22CF19C6 (содержит 80 умножителей с размерностью операндов 9×9). Входной сигнал и коэффициенты фильтра представим с 8-разрядной точностью.

Для линии задержки (taps.v), ПЗУ для хранения коэффициентов (hvalues.v) и управляющего автомата (state_m.v) используем оригинальный Verilog-код [4]. Умножитель

настроен на работу с числами со знаком в асинхронном режиме. Во всех рассматриваемых вариантах реализации КИХ-фильтра умножитель строится на аппаратном умножителе ЦОС-блока, без использования логических ресурсов ПЛИС.

На рис. 3 показан проект последовательного КИХ-фильтра на четыре отвода, а на рис. 4 демонстрируется функциональное моделирование прохождения сигнала по структуре КИХ-фильтра. На вход фильтра поступает сигнал $-5, 3, 1, 0, 0$ и т. д. Правильные значения на выходе фильтра: $10, -1, -40, -10, 25, 6$ и т. д.

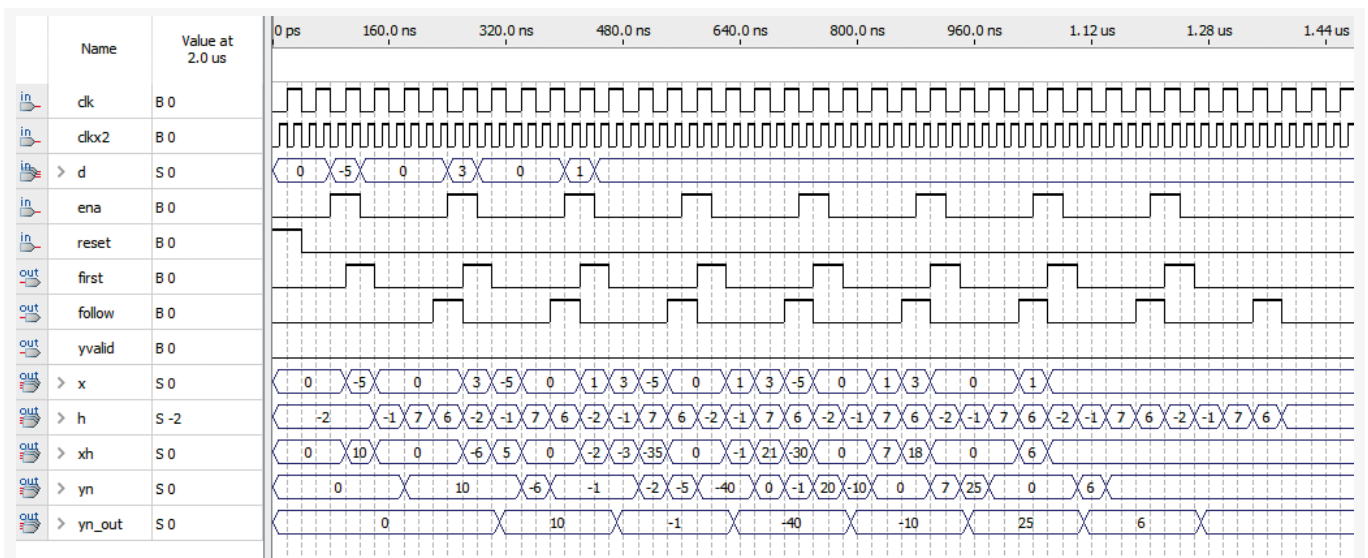


Рис. 4. Функциональное моделирование прохождения сигнала по структуре КИХ-фильтра на четыре отвода (асинхронный режим работы умножителя)



КОМПЛЕКСНЫЕ ПОСТАВКИ ЭКБ

**ВСЯ ПОСТАВЛЯЕМАЯ
ПРОДУКЦИЯ ПРОХОДИТ
«ВХОДНОЙ КОНТРОЛЬ»
И «ИДЕНТИФИКАЦИЮ»**

АО «ТЕСТПРИБОР» осуществляет комплексные поставки электронных компонентов отечественного и иностранного производства для различного применения классов:

- Industrial, ▪ Military, ▪ Space



ПРОГРАММА ПОСТАВОК

- Микросхемы
- ПЛИС
- СБИС
- Источники питания
- Преобразователи
- Конденсаторы
- Резисторы
- Транзисторы
- Диоды
- Реле
- Переключатели
- Разъемы
- Фильтры
- Атенюаторы
- Ответвители

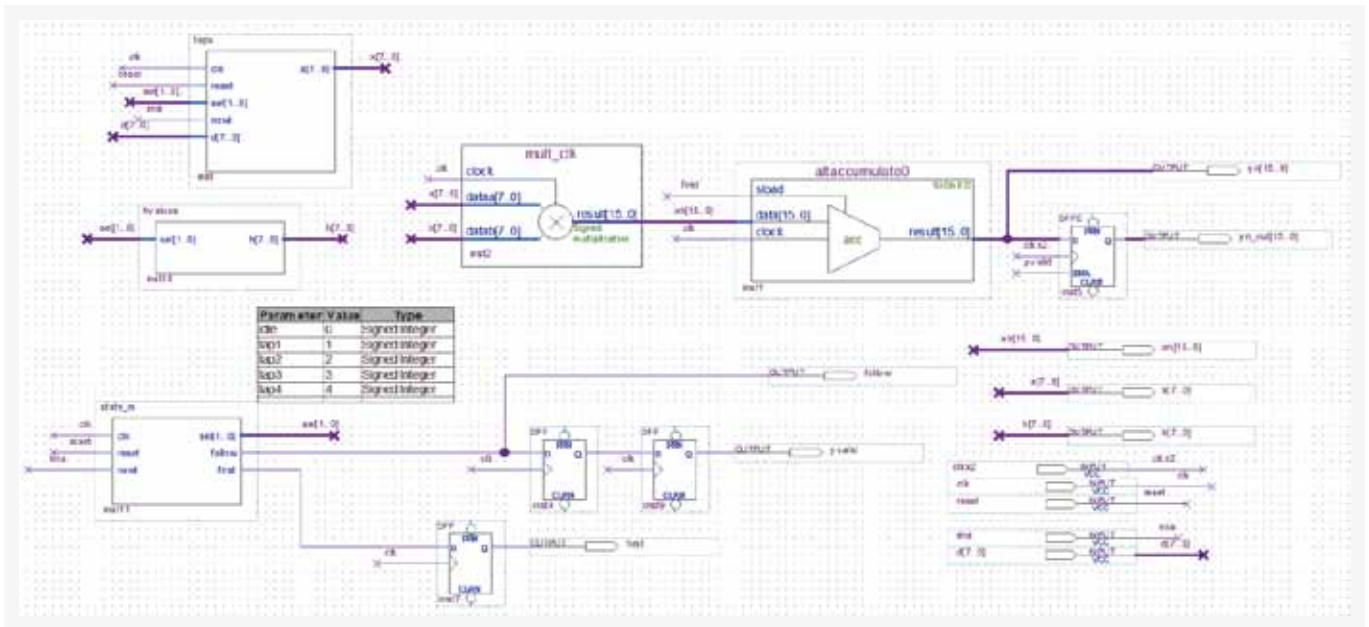


Рис. 5. Проект последовательного КИХ-фильтра на четыре отвода с использованием мегафункции умножения LPM_MULT (синхронный режим) и накопления ALTACCUMULATE

Аккумулятор работает в синхронном режиме, и его латентность составляет один такт синхросигнала. Для согласования работы управляющего автомата и аккумулятора необходимо выходной сигнал first автомата подключить ко входу сигнала синхронной загрузки аккумулятора load.

Результат умножения и вычисления суммы произведений представляются с 16-разрядной точностью. Для получения правильных значений профильтрованного сигнала на выходе yn_out[15..0] необходимо выходной сигнал

follow управляющего автомата задержать на два такта синхросигнала и подключить его ко входу разрешения тактирования ena регистра результата.

Вариант 2

Этот вариант представляет собой модификацию варианта 1, в котором умножитель работает в синхронном режиме (рис. 5, 6). Остальные блоки без изменений. Латентность умножителя и аккумулятора составляет один такт

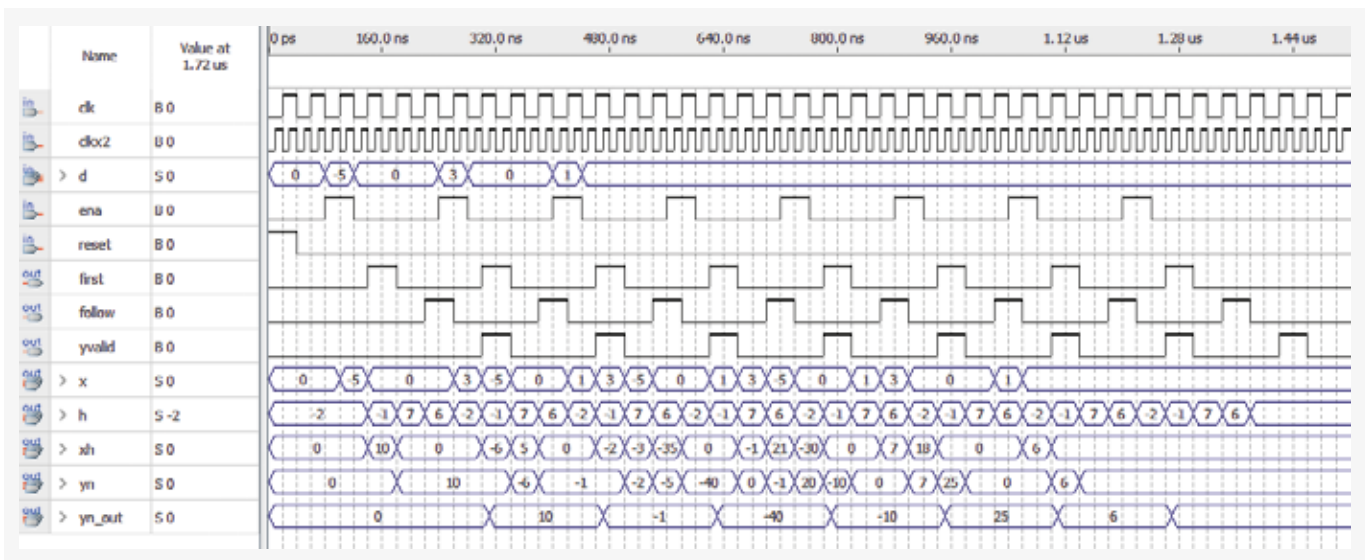


Рис. 6. Функциональное моделирование прохождения сигнала по структуре КИХ-фильтра на четыре отвода (умножитель в синхронном режиме)

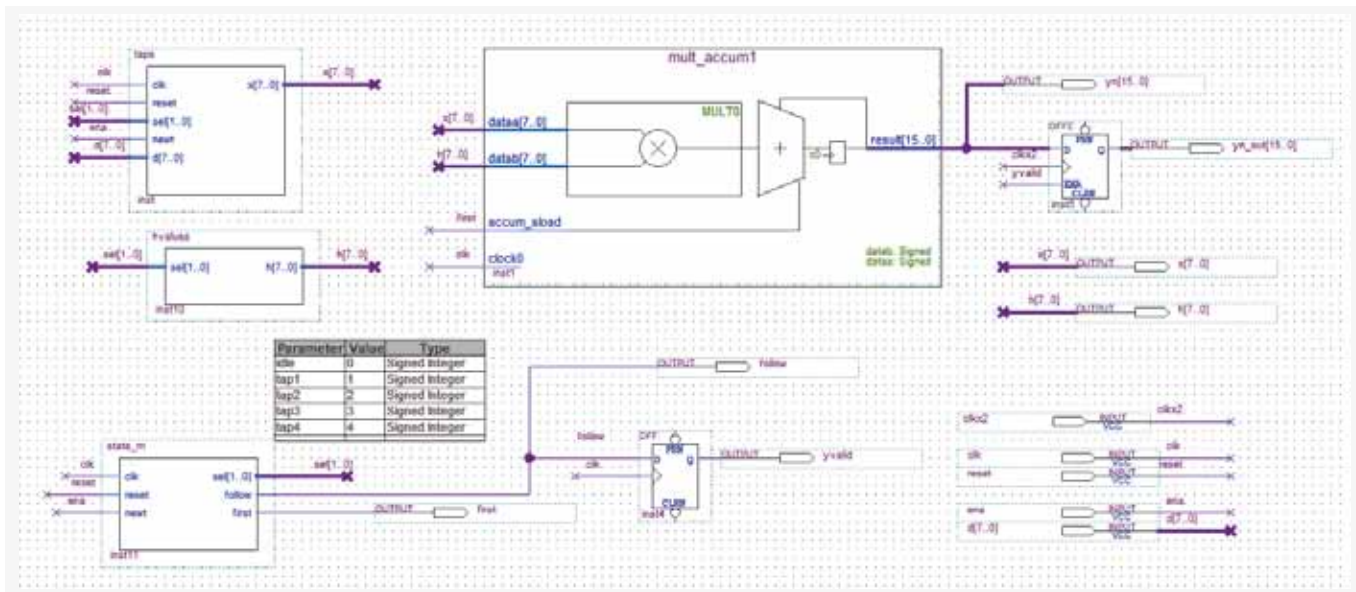


Рис. 7. Проект последовательного КИХ-фильтра на четыре отвода с использованием МАС-блока (мегафункция ALTMULT_ACCUM)

синхросигнала. Для согласования работы управляющего автомата, умножителя и аккумулятора необходимо выходящий сигнал first автомата задержать на один такт синхросигнала с помощью двухтактного триггера, выход которого подключен ко входу сигнала синхронной загрузки аккумулятора sload. А сигнал follow задержать на два такта с помощью двух триггеров.

Выходной сигнал автомата first необходимо подключить ко входу синхронной загрузки аккумулятора accum_sload. Для получения правильных значений профильтрованного сигнала на выходе y_{n_out}[15..0] необходимо выходящий сигнал follow задержать на один такт синхросигнала, поскольку умножитель в мегафункции ALTMULT_ACCUM работает в асинхронном режиме.

Вариант 3

Рассмотрим вариант, когда умножитель и аккумулятор реализованы на мегафункции умножения и накопления ALTMULT_ACCUM без регистров на входах и выходах умножителя (рис. 7). Линия задержки, ПЗУ, управляющий автомат представлены оригинальным Verilog-кодом.

Вариант 4

В этом варианте рассматривается последовательный КИХ-фильтр на четыре отвода с использованием управляющего автомата, разработанного с помощью редактора State Flow. На основе оригинального Verilog-кода управляющего автомата (state_m.v) с помощью

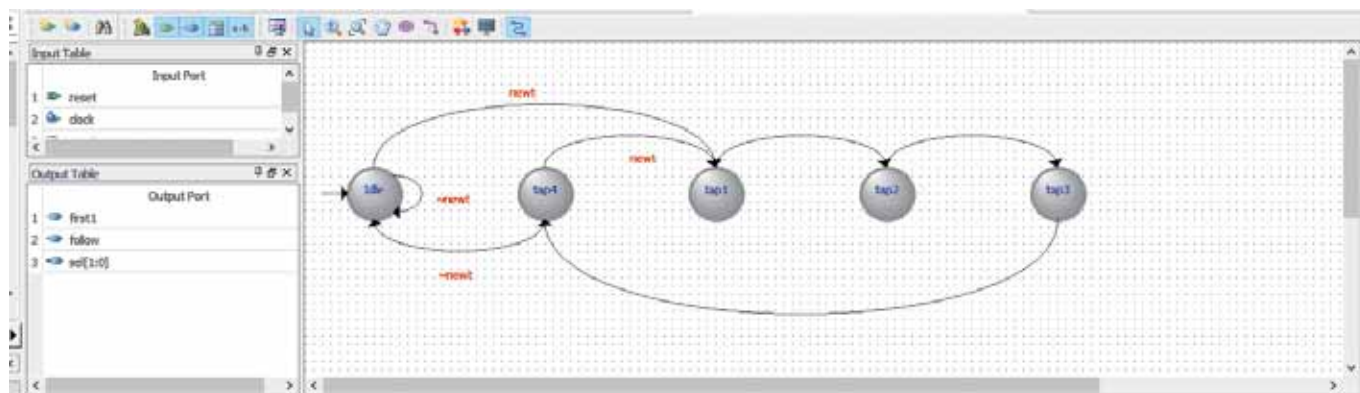


Рис. 8. Графическое представление управляющего автомата на пять состояний, восстановленное из оригинального Verilog-кода

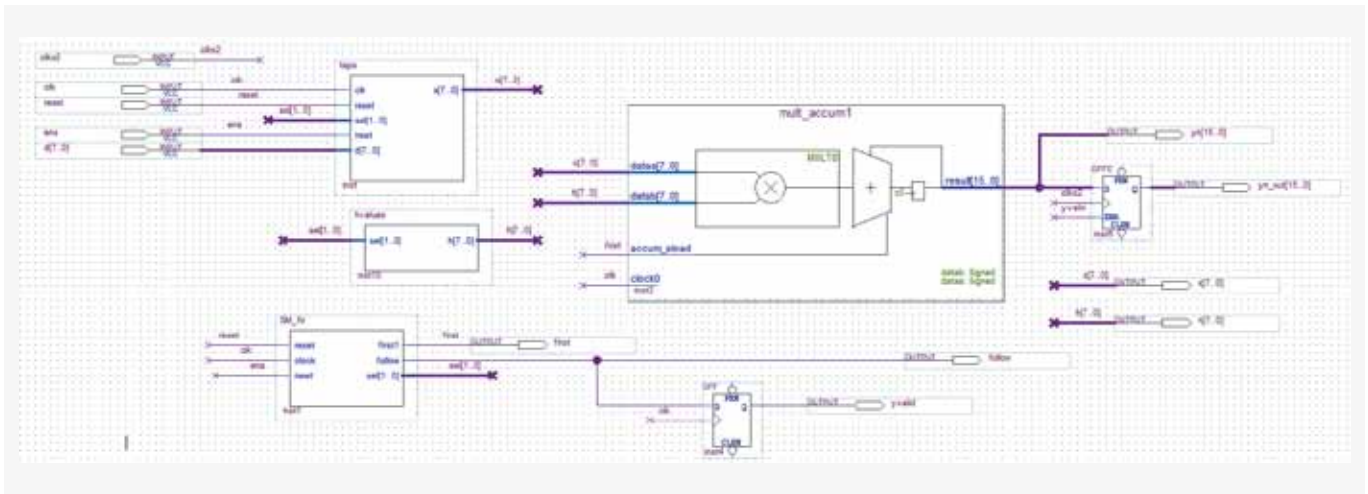


Рис. 9. Проект последовательного КИХ-фильтра на четыре отвода с использованием VHDL-кода, извлеченного из графического представления управляющего автомата

редактора State Flow восстановим его графическое представление (рис. 8) и извлечем в автоматическом режиме из его описания VHDL-код (SM_fir.vhd). Остальные блоки и учет латентности – как в варианте 3. Каких-либо дополнительных согласований работы блоков по латентности не требуется. Выходной сигнал follow необходимо задержать на один такт синхросигнала. Интересен вариант 4 представлением адресного сигнала sel[1..0]. В графическом схемном редакторе используется иное обозначение – sel[1:0]. На рис. 9 показан проект последовательного КИХ-фильтра с использованием VHDL-кода, извлеченного из графического представления. Функциональное

моделирование (рис. 10) подтверждает правильность работы данного варианта.

Вариант 5

Этот вариант представляет собой модификацию варианта 4. Покажем, как можно реализовать аккумулятор без использования мегафункции ALTACCUMULATE. Для этого необходимо использовать аккумулятор на основе синхронного сумматора на мегафункции lpm_add_sub и шинного мультиплексора 2 в 1 на мегафункции lpm_mux в обратной связи (рис. 11, 12). На один из входов мультиплексора необходимо подключить константу – логический

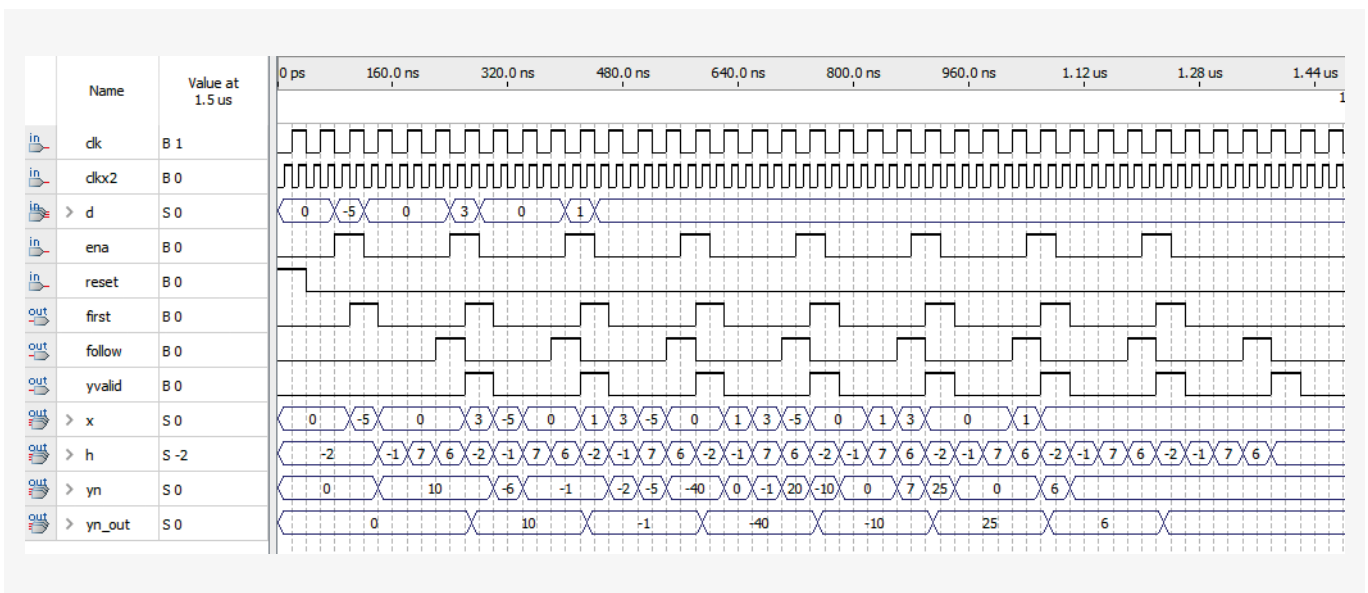


Рис. 10. Функциональное моделирование прохождения сигнала по структуре КИХ-фильтра на четыре отвода (управляющий автомат – VHDL-код)

СПЕЦИАЛЬНОЕ ТЕХНОЛОГИЧЕСКОЕ ОБОРУДОВАНИЕ ДЛЯ РАЗРАБОТКИ И ПРОИЗВОДСТВА ЭКБ

Молекулярно-лучевая эпитаксия

| A3N | A3B5 | A2B6

Плазмохимическое травление и осаждение

| ICP-RIE | RIE | ICP-PECVD | PECVD

Физическое осаждение

| Магнетронное распыление
| Электронно-лучевое напыление

Быстрый термический отжиг и процессинг



ПРОИЗВОДСТВО И РЕИНЖИНИРИНГ КОМПЛЕКТУЮЩИХ И КЛЮЧЕВЫХ УЗЛОВ

| Молекулярные источники
| Вводы вращения
| Ростовые и линейные манипуляторы

в том числе для установок МЛЭ иностранного производства



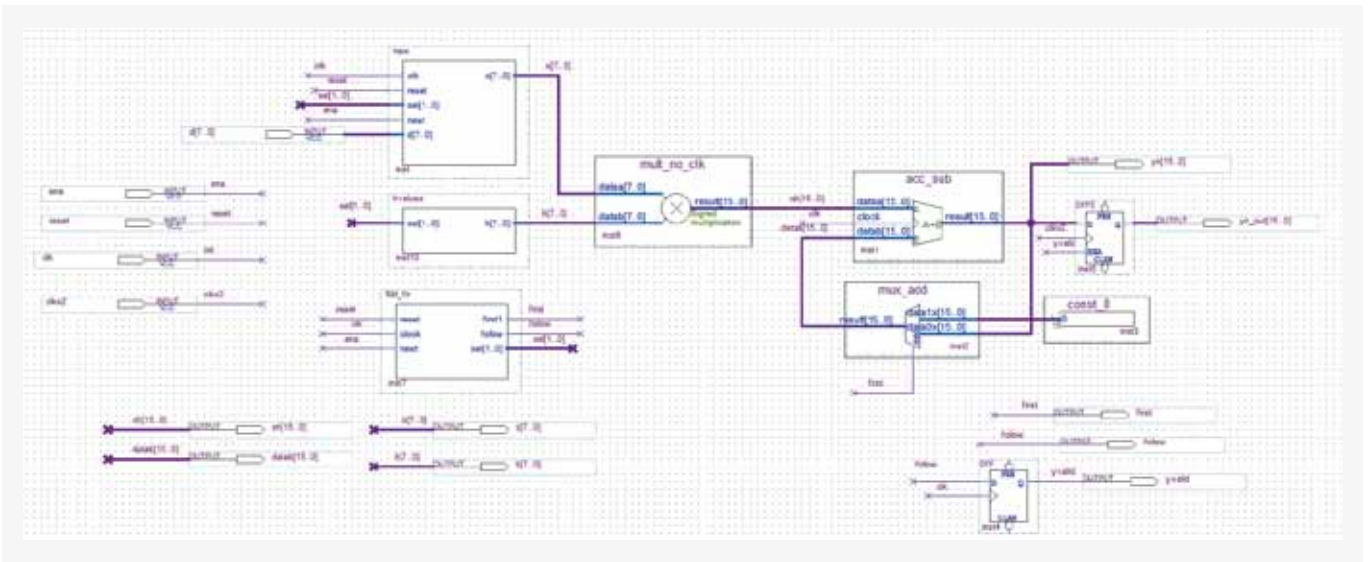


Рис. 11. Проект последовательного КИХ-фильтра на четыре отвода с использованием аккумулятора на основе синхронного сумматора на мегафункции `lpm_add_sub` и шинного мультиплексора 2 в 1 на мегафункции `lpm_mux` в обратной связи

ноль. Выходной сигнал `follow` необходимо задержать на один такт синхронизации.

Вариант 6

Представляет собой вариант 4 с той лишь разницей, что линия задержки организована на регистрах (рис. 13). Из оригинального проекта `fir_filter` используется лишь асинхронное ПЗУ, представленное Verilog-кодом. Отводы

линии задержки коммутируются на один из входов умножителя с помощью шинного мультиплексора 4 в 1. Как и в вариантах 1, 3, 4, 5, в которых умножитель работает в асинхронном режиме, в этом случае сигнал `first` напрямую подключается ко входу `accum_sload` мегафункции `ALTMULT_ACCUM`, а сигнал `follow` задерживается на два такта синхронизации. Результаты моделирования представлены на рис. 14.

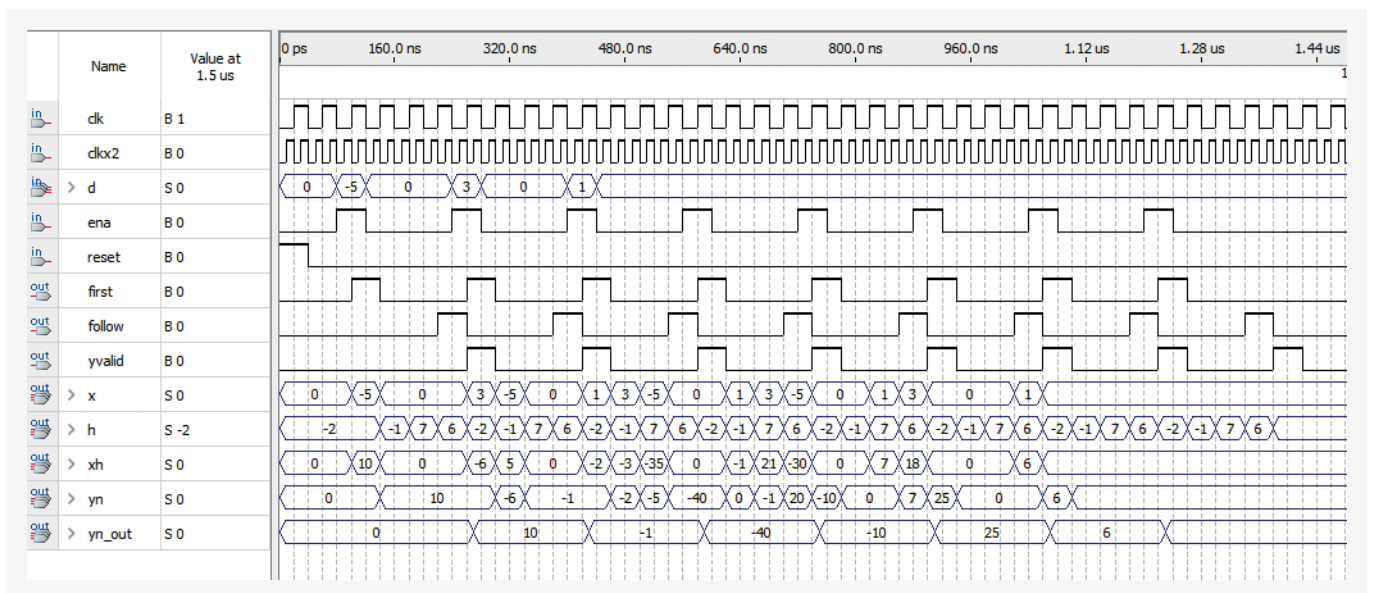


Рис. 12. Функциональное моделирование прохождения сигнала по структуре КИХ-фильтра с использованием аккумулятора на основе синхронного сумматора на мегафункции `lpm_add_sub` и шинного мультиплексора 2 в 1 на мегафункции `lpm_mux` в обратной связи

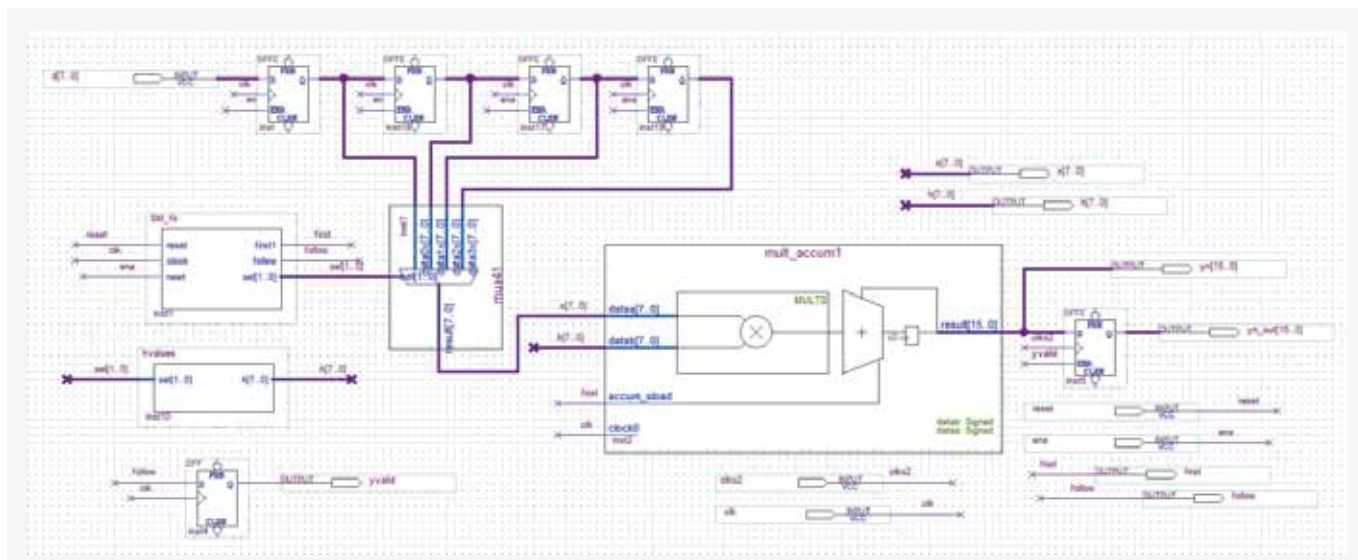


Рис. 13. Проект последовательного КИХ-фильтра на четыре отвода с использованием линии задержки на регистрах

Далее в рассматриваемых проектах используется одностотная синхронизация (варианты 7–9).

Вариант 7

Линию задержки также можно реализовать на основе двухпортовой памяти с использованием мегафункции altshift_taps (Shift Register (RAM-based)) со следующими установками: число отводов – 4; дистанция между отводами – 4 (рис. 15). Коммутация отводов линии задержки осуществляется с помощью шинного мультиплектора 4 в 1, на адресный вход которого подключается выход управляющего автомата sel[1..0]. В проекте используется

однототная синхронизация и все блоки оригинального проекта fir_filter заменены.

ПЗУ для хранения коэффициентов фильтра реализуется на мегафункции ROM: 1 port с инициализацией. Мегафункция настроена так, что к адресной и выходной шине данных подключены дополнительные регистры для организации конвейеризации. Такая настройка приводит к тому, что ПЗУ работает в синхронном режиме.

Мегафункция умножения и накопления ALTMULT_ACCUM представлена с конвейеризующими регистрами на входах и на выходах умножителя, с двумя регистрами на входе синхронной загрузки accum_load.

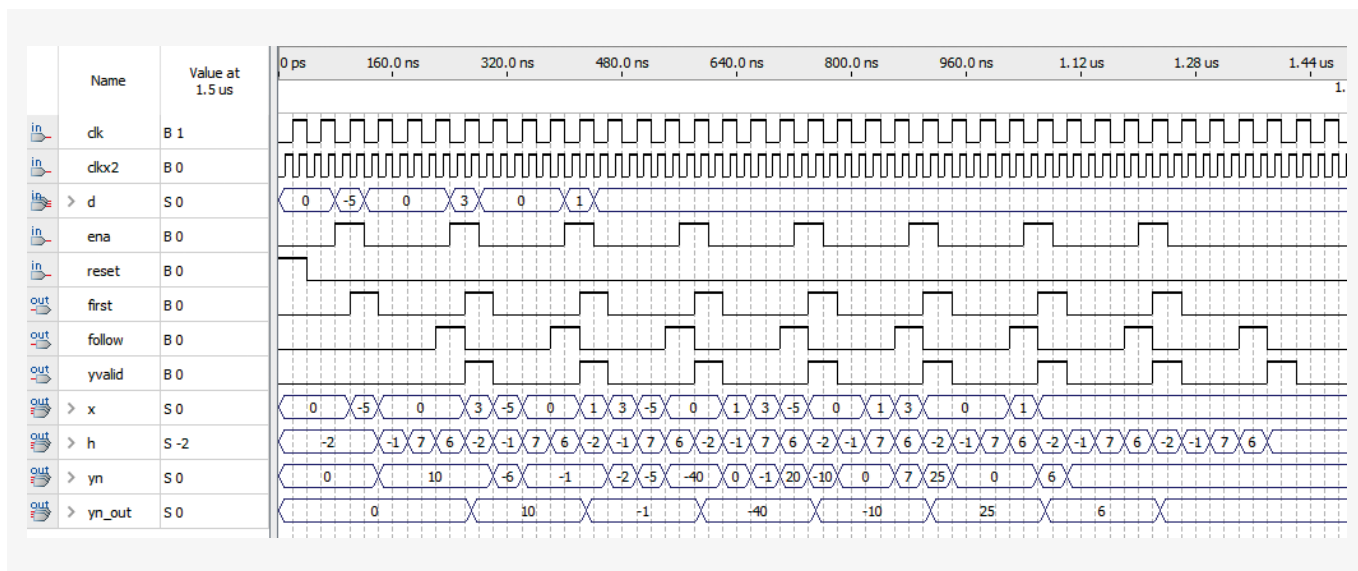


Рис. 14. Функциональное моделирование прохождения сигнала по структуре КИХ-фильтра с использованием линии задержки на регистрах

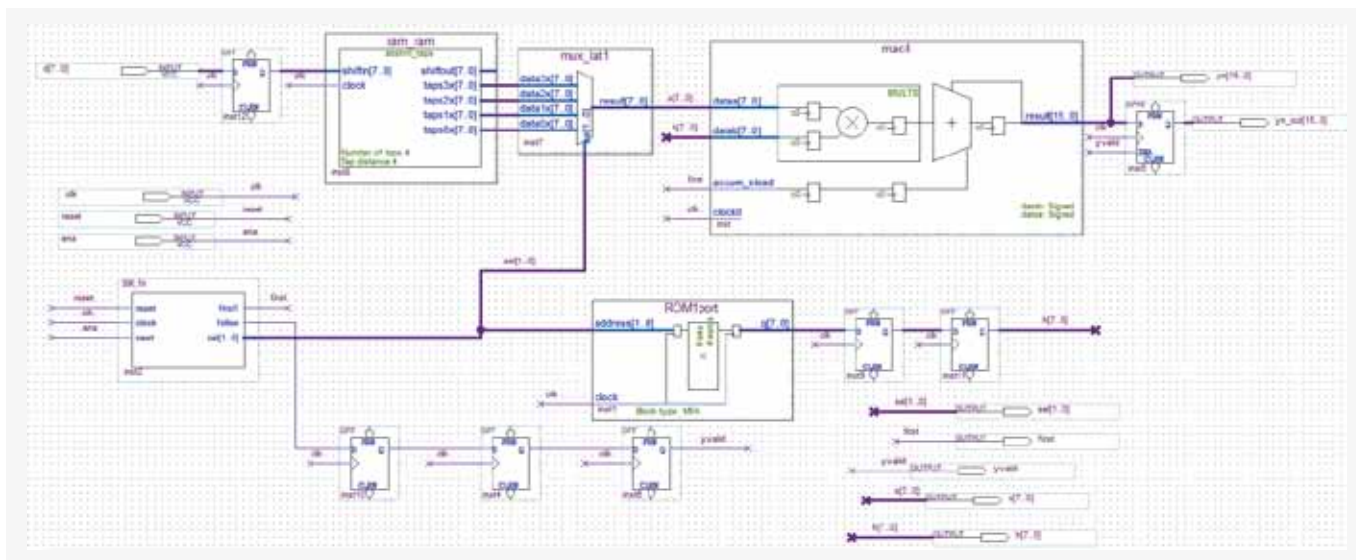


Рис. 15. Управляющий автомат (state flow), линия задержки на двухпортовой памяти (мегафункция Shift Register (RAM-based)), MAC-блок (мегафункция умножения и накопления ALTMULT_ACCUM), синхронное ПЗУ (мегафункция)

Учет латентности может быть осуществлен следующим образом. Сигнал first управляющего автомата напрямую подключается ко входу accum_sload мегафункции ALTMULT_ACCUM. Сигнал follow нужно задержать на три такта синхросигнала, число триггеров определяется по числу уровней конвейеризации. В данных настройках мегафункции умножения и накопления используются три уровня. К выходной шине данных ПЗУ необходимо подключить два регистра и один регистр на вход линии задержки. В этом случае сигналы $x[7..0]$ (номер отвода линии задержки зависит от значения на адресной шине мультиплектора) и $h[7..0]$ (коэффициенты) перемножаются корректно. На рис. 16 показаны временные диаграммы.

Вариант 8

Проект отличается от варианта 7 тем, что в качестве управляющего автомата для генерации адресов используется 2-разрядный счетчик на четыре состояния (мегафункция lpm_counter) для адресации к мультиплектору и ПЗУ (рис. 17, 18). В качестве сигнала first, задержанного на один такт синхронизации, используется выход cout счетчика (сигнал сквозного переноса, который возникает, когда на выходах счетчика $q[1..0]$ устанавливается число 3). Использование асинхронного ПЗУ приводит к тому, что для правильного умножения необходимо коэффициенты задержать на четыре такта с помощью четырех регистров.

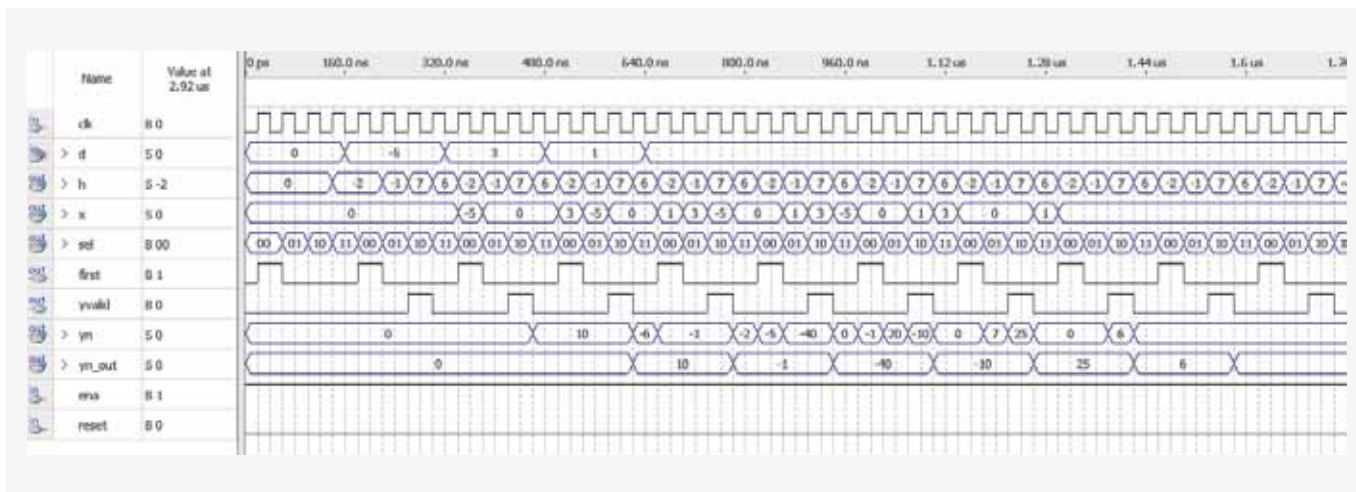


Рис. 16. Функциональное моделирование прохождения сигнала по структуре КИХ-фильтра с использованием линии задержки на двухпортовой памяти (мегафункция Shift Register (RAM-based))

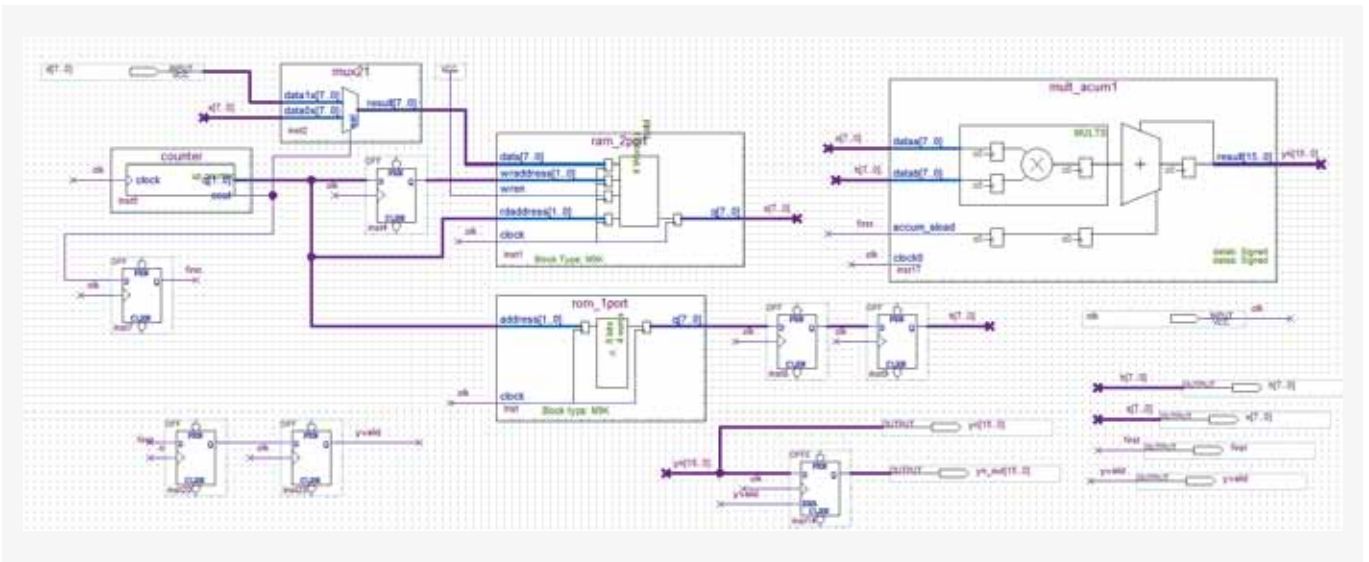


Рис. 19. Функциональная модель последовательного КИХ-фильтра на четыре отвода с использованием линии задержки на основе двухпортовой памяти, мегафункции RAM: 2-PORT, синхронного ПЗУ, мегафункции ROM: 1 port

В режиме new_data новые данные записываются в память и одновременно отображаются на выходе блока памяти. Выбор режимов new_data, old_data и don't_care зависит от типа блока памяти ПЛИС. В данном проекте используются блоки памяти типа M9K.

Как и в варианте 8, сигнал first получен задержкой сигнала с выхода cout. А сигнал yvalid получен задержкой сигнала first на два такта. Функциональное моделирование (рис. 20) подтверждает правильность работы КИХ-фильтра.

РЕЗУЛЬТАТЫ РЕАЛИЗАЦИИ ПРОЕКТОВ В БАЗИСЕ ПЛИС CYCLONE IV

В табл. 1 представлены результаты размещения проектов в базис ПЛИС Cyclone IV EP4CGX22CF19C6 и оценка

быстродействия (максимальная частота переключения триггеров, Fmax), полученная с помощью приложения TimeQuest без учета временных ограничений (без sdc-файла). Каждый из рассмотренных проектов по объему занимает менее 1% логических и аппаратных ресурсов ПЛИС.

Для вариантов 1–6 с двухчастотной (clk и clkx2) и для вариантов 7–9 с одночастотной (clk) синхронизацией без использования заранее созданного пользователем sdc-файла САПР Quartus II автоматически запускает Classic Timing Analyzer и в качестве требования к синхронизации по умолчанию применяет значение частоты Fmax_REQUIREMENT, равное 1000 МГц (типовая полоса обрабатываемого радиосигнала или производительность,

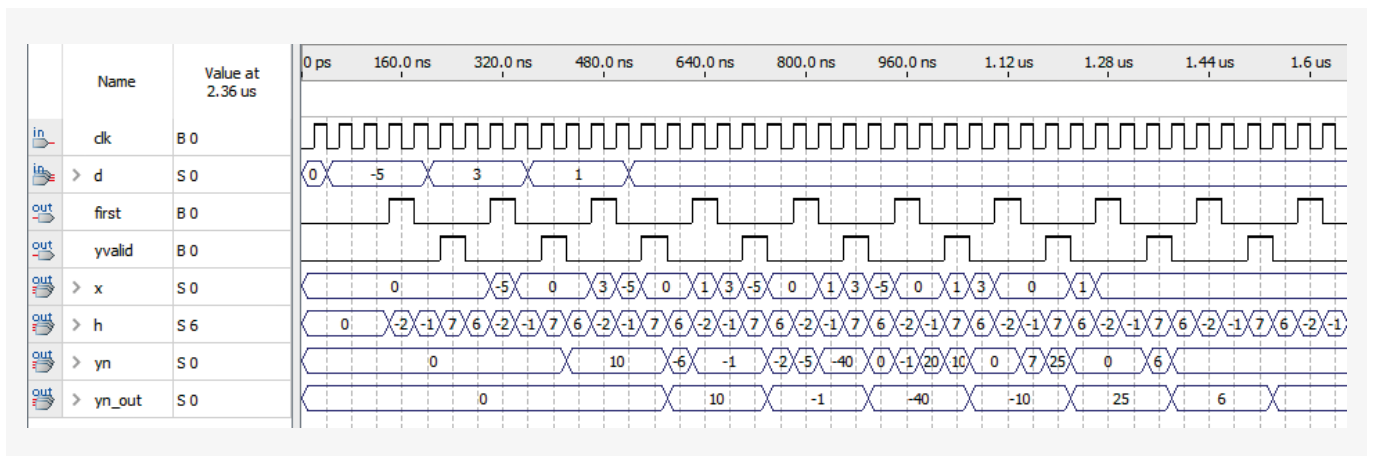


Рис. 20. Функциональное моделирование прохождения сигнала по структуре КИХ-фильтра с использованием линии задержки на основе двух портовой памяти, мегафункции RAM: 2-PORT, синхронного ПЗУ, мегафункции ROM: 1 port

Slow 1200mV 0C Model Minimum Pulse Width: 'clk'							
	Slack	Actual Width	Required Width	Type	Clock	Clock Edge	Target
23	-1.000	1.000	2.000	Min Period	clk	Rise	state_minst11filter.tap4
24	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[0]
25	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[1]
26	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[2]
27	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[3]
28	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[4]
29	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[5]
30	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[6]
31	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn[7]
32	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[0]
33	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[1]
34	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[2]
35	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[3]
36	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[4]
37	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[5]
38	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[6]
39	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_1[7]
40	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[0]
41	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[1]
42	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[2]
43	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[3]
44	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[4]
45	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[5]
46	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[6]
47	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_2[7]
48	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_3[0]
49	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_3[1]
50	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_3[2]
51	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_3[3]
52	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_3[4]
53	-1.000	1.000	2.000	Min Period	clk	Rise	taps:inst\xn_3[5]

Рис. 21. Отрицательные запасы по параметру Minimum Pulse Width „clk“ модели Slow при 1200 мВ и 0 °С для варианта 3

Slow 1200mV 0C Model Minimum Pulse Width: 'clk'							
	Slack	Actual Width	Required Width	Type	Clock	Clock Edge	Target
1	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[0]
2	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[1]
3	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[2]
4	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[3]
5	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[4]
6	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[5]
7	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[6]
8	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn[7]
9	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[0]
10	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[1]
11	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[2]
12	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[3]
13	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[4]
14	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[5]
15	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[6]
16	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_1[7]
17	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[0]
18	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[1]
19	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[2]
20	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[3]
21	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[4]
22	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[5]
23	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[6]
24	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_2[7]
25	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_3[0]
26	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_3[1]
27	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_3[2]
28	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_3[3]
29	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_3[4]
30	1.792	2.008	0.216	High Pulse Width	clk	Rise	taps:inst\xn_3[5]

Рис. 22. Положительные запасы по параметру Minimum Pulse Width „clk“ модели Slow при 1200 мВ и 0 °С для варианта 3

характерная для ЦОС-процессоров: 1 ГГц/4 такта = 250 Мвыб/с).

На рис. 21 представлен отчет с отрицательными запасами по времени (Slack) по параметру Minimum Pulse Width „clk“ модели Slow при 1200 мВ и 0 °С для варианта 3 при работе на заданной частоте 1000 МГц. Следовательно, необходимо ограничить частоты синхронизации до появления положительных запасов по времени.

Для вариантов 1–6 с двухчастотной синхронизацией (clk и clkx2) используем следующий scd-файл с частотами 100 МГц (период 10 нс) для порта clk и 200 МГц (период 5 нс) для порта clkx2:

```
create_clock -name {clk}
-period 10.000
-waveform {0.000 2.000}
[get_ports {clk}],
create_clock -name {clkx2}
-period 5.000
-waveform {0.000 2.000}
[get_ports {clkx2}].
```

Тестирование для вариантов 1–6 с частотами 100 и 200 МГц показало наличие положительных запасов по параметру Minimum Pulse Width „clk“. В качестве примера на рис. 22 представлен отчет с положительными запасами по параметру Minimum Pulse Width „clk“ модели Slow при 1200 мВ и 0 °С для варианта 3. Однако, для двухчастотной синхронизации, в отличие от одночастотной, при использовании sdc-файла приложение TimeQuest не рассчитывает величину Fmax. В табл. 1 для вариантов 7–9 дополнительно приведена оценка быстродействия для одночастотной синхронизации:

```
create_clock -name {clk}
-period 10.000
-waveform {0.000 2.000}
[get_ports {clk}].
```

ЗАКЛЮЧЕНИЕ

По своей структуре последовательные МАС КИХ-фильтры в базисах ПЛИС Xilinx и Altera схожи и отличаются лишь разной латентностью работы основных узлов фильтра, реализуемых на блоках встроенной памяти и ЦОС-блоках. Особенностью последовательных КИХ-фильтров является наличие управляющего автомата, который адресуется к линии задержки и ПЗУ для хранения коэффициентов фильтра, а также управляет МАС-блоком и регистром результата.

В статье представлено девять вариантов реализации проекта последовательного КИХ-фильтра от простого до сложного с использованием циклического буфера в двухпортовой памяти ОЗУ, синхронного ПЗУ и конвейеризованного МАС-блока. Проекты задействуют примерно одинаковое количество логических и аппаратных ресурсов ПЛИС Cyclone IV EP4CGX22CF19C6 (менее 1%), но имеют разное быстродействие. Наивысшее быстродействие показывают проекты 7, 8 и 9 с использованием конвейеризации и блочной памяти ПЛИС для организации линии задержки. Для двухчастотной ($\text{clk}=100$ МГц и $\text{clkx2}=200$ МГц) и одночастотной ($\text{clk}=100$ МГц) синхронизации при использовании файлов временных ограничений обеспечены положительные запасы на рассматриваемых частотах.

Таблица 1. Оценка задействованных ресурсов (один множитель с размерностью операндов 9×9) и быстродействия проектов в ПЛИС Cyclone IV EP4CGX22CF19C6

Варианты проектов	Количество логических элементов / регистров	Оценка быстродействия для модели Slow ($F_{\text{max}}/F_{\text{max}}$ в наихудшем случае, МГц)	
		При 1200 мВ, clk и clkx2 1000 МГц, 0 °С	При 1200 мВ, clk и clkx2 1000 МГц, 85 °С
Оригинальный проект <code>fir_filter</code>	93/77	273/250	242/242
1	99/96	180/180	162/180
2	72/72	206/206	185/185
3	70/70	145/145	129/129
4	72/70	144/144	129/129
5	86/70	142/142	128/128
6	75/70	145/145	130/130
7	85/70, 96 бит встроенной памяти	348/250 (295/250 при clk 100 МГц)	337/250 (267/250 при clk 100 МГц)
8	62/46, 96 бит встроенной памяти	328/250 (305/250 при clk 100 МГц)	296/250 (278/250 при clk 100 МГц)
9	63/55, 96 бит встроенной памяти	348/250 (398/250 при clk 100 МГц)	343/250 (359/250 при clk 100 МГц)

ЛИТЕРАТУРА

1. **Мейер-Баесе У.** Digital Signal Processing with Field Programmable Gate Arrays. Fourth Edition // <http://www.springer.com/series/4748>.
2. Xilinx. DSP: Designing for Optimal Results High-Performance DSP Using Virtex-4 FPGAs // DSP Products Advanced Design Guide. Edition 1.0. March 2005.
3. **Строгонов А. В.** Применение Altera DSP Builder системы Matlab/Simulink для разработки имитационной модели КИХ-фильтра на параллельной распределенной арифметике // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. 2020. № 3. С. 1–9.
4. **Комолов Д. А., Мяльк Р. А., Зобенко А. А., Филиппов А. С.** Системы автоматизированного проектирования фирмы Altera MAX+plus II и Quartus II. Краткое описание и самоучитель. М.: ИП РадиоСофт, 2002. 352 с.
5. **Строгонов А. В.** Реализация алгоритмов цифровой обработки сигналов в базисе программируемых логических интегральных схем: Учебное пособие / 4-е изд., испр. и доп. СПб: Издательство «Лань», 2019. 352 с.
6. **Строгонов А. В., Цыбин С. А., Городков П. С.** Проектирование последовательных КИХ-фильтров в САПР ПЛИС Quartus II // Компоненты и технологии. 2016. № 1. С. 10–15.
7. **Строгонов А. В., Цыбин С. А., Городков П. С.** Особенности использования двухпортовой памяти при проектировании последовательных КИХ-фильтров в САПР ПЛИС Quartus II // Компоненты и технологии. 2016. № 4. С. 40–46.



|| 2022
www.kulon.spb.ru

РАЗРАБОТКА И ПРОИЗВОДСТВО КЕРАМИЧЕСКИХ КОНДЕНСАТОРОВ И ПРОХОДНЫХ ФИЛЬТРОВ

СЕРИЙНАЯ ПРОДУКЦИЯ:

- многослойные конденсаторы:
К10-17, К10-42, К10-47, К10-50,
К10-54, К10-57, К10-79, КМК;
- трубчатые конденсаторы:
ТК, К10-51К, КТП, КТ-1Е;
- фильтры: Б14, Б23А, Б23Б,
Б28, Б29, Б7-2, Б24.

НОВЕЙШИЕ РАЗРАБОТКИ:

- варисторы ВР-18, ВР-19;
- фильтры Б36;
- конденсаторы К10-89, К10-90.