



УДК 681.3

А. М. Сергиенко, канд. техн. наук,

В. П. Симоненко, д-р техн. наук

Национальный технический университет Украины «КПИ»

(Украина, 03056, Киев, пр-т Победы, 37,

тел.: (044) 4549337, E-mail: aser@comsys.ntu-kpi.kiev.ua)

Отображение периодических алгоритмов в программируемые логические интегральные схемы

Рассмотрен метод отображения периодических алгоритмов, представленных графом синхронных потоков данных, в конвейерный вычислитель, реализованный в программируемой логической интегральной схеме. Метод заключается в размещении графа алгоритма в многомерном индексном пространстве и отображении его в подпространства структур и времени. Ограничения на процесс отображения позволяют минимизировать как тактовый интервал, так и аппаратные затраты, включая мультиплексоры.

Розглянуто метод відображення періодичних алгоритмів, зображених графом синхронних потоків даних, в конвейерний обчислювач, реалізований у програмованій логічній інтегральній схемі. Метод полягає в розміщенні графа алгоритму у багатовимірному індексному просторі та його відображенні у підпростори структур і часу. Обмеження на процес відображення дають змогу мінімізувати як тактовий інтервал, так і апаратні витрати, включаючи мультиплексори.

К л ю ч е в ы е с л о в а: VHDL, ПЛИС, FPGA, SDF, отображение алгоритма, структурный синтез.

Современные программируемые логические интегральные схемы (ПЛИС) имеют емкости, составляющие миллионы вентилях и сотни аппаратных умножителей, и работают с частотами в сотни мегагерц. Поэтому их заслуженно считают перспективной элементной базой для высокопроизводительных расчетов, обработки данных и моделирования, а также цифровой обработки сигналов (ЦОС). Однако широкому внедрению ПЛИС препятствуют трудности, связанные с отображением в них алгоритмов. Разработка спецвычислителя на базе ПЛИС остается достаточно трудоемкой работой и требует высокой квалификации.

При высокоуровневом (системном) синтезе вычислительных устройств (ВУ) периодический алгоритм, такой как алгоритм ЦОС, обычно представляют графом синхронных потоков данных (ГСПД). В нем вершины — акторы — представляют собой операторы, а дуги — передачи переменных

между акторами с задержкой на заданное число циклов. Причем каждый актор в течение цикла генерирует и использует переменные, число которых остается неизменным от цикла к циклу [1,2].

Синтез ВУ обычно выполняется в три этапа: 1) формирование множества аппаратных ресурсов; 2) составление расписания выполнения акторов; 3) назначение акторов на ресурсы. В методах синтеза используется различный порядок подэтапов: назначение переменных регистрам или шинам, назначение операций процессорным элементам (ПЭ), составление расписания выполнения акторов, пересылок переменных, а также различные алгоритмы реализации этих подэтапов. Наиболее ответственный и сложный этап — составление расписания, так как от него зависит производительность ВУ и загруженность его ресурсов. Этот этап представляет собой решение сложной комбинаторной задачи.

ГСПД является графом одной итерации (периода) периодического алгоритма. ГСПД бывают как циклическими, так и ациклическими. Дуги, замыкающие ациклический граф в циклический, описываются межитерационными зависимостями. Положительное число, нагружающее замыкающую дугу, равно дистанции передачи операнда и измеряется в итерациях. Составление расписания для циклических ГСПД — наиболее сложная задача. Распространенным подходом при этом является поиск расписания периода алгоритма с планированием вычислений для его ациклической части. Обычно используют алгоритмы списочного планирования, силового планирования, раскраски графа, метод планирования с выравниванием акторов по левому краю пространства ресурсы — время (left edge scheduling). Однако прямое их применение приводит к низкой загруженности ВУ в начале и в конце цикла. Для учета цикличности алгоритма анализируют граф конфликтов ресурсов и операций, а также граф интервалов существования переменных [2, 3].

Оптимальность синтезированного ВУ зависит от всех этапов синтеза, которые, как правило, выполняются независимо один от другого. Каждый из них имеет локальную цель, которая часто противоречит цели других этапов. Так, минимизация ресурсов при формировании их множества неизбежно приводит к увеличению длительности расписания. Поэтому большинство методов системного синтеза не обеспечивает высокое качество ВС, реализующих сложные алгоритмы.

В последнее время получают распространение такие программные средства, как AccelDSP и System Generator, помогающие перевести в прошивку ПЛИС периодический алгоритм, представленный в виде ГСПД с помощью среды Matlab Simulink [4]. Однако эти средства эффективны только для алгоритмов с ациклическими ГСПД, например для фильтров с

конечной импульсной характеристикой или при единичном отображении ГСПД, или для алгоритмов, операторы которых соответствуют крупным библиотечным компонентам.

В работах [5, 6] предложены методы отображения алгоритмов, представленных ГСПД, в структуры параллельных ВУ. Методы основаны на размещении графа алгоритма в многомерном индексном пространстве и отображении его в подпространства структур и времени. В методах структурного проектирования цифровых фильтров (ЦФ), описанных в [6, 7], также используется этот подход. При таком подходе этапы синтеза выполняются одновременно, а ВУ оптимизируется с удовлетворением противоречивых требований к нему. В данном случае этот подход применяется для синтеза ВУ, реализующего периодические алгоритмы в ПЛИС.

Отображение ГСПД в подпространства структур и времени. Исходными данными для проектирования ВУ является периодический алгоритм, представленный ГСПД, и критерии оптимизации. Рассмотрим ГСПД, в которых число групп переменных, потребляемых актором, равно числу результатов, выдаваемых этим же актором в течение одного цикла. Это означает, что ПЭ будут работать в одинаковом темпе, как, например, в конвейере. К подклассу таких алгоритмов относятся алгоритмы ЦФ с постоянной частотой дискретизации. Предположим, что ВУ состоит из простых ПЭ, связанных между собой согласно графу структуры ВУ. ПЭ включает в себя арифметико-логическое устройство (АЛУ) определенного типа (умножитель, сумматор) с регистром результата на его выходе (или с буфером FIFO) и мультиплексорами входных данных на его входе. ПЭ выполняет операцию над входными данными с записью результата в регистр не более, чем за один такт. Если ПЭ не имеет регистра, то считается, что операция выполняется без задержки.

В качестве критерия оптимизации примем период выполнения алгоритма $Q_T = Lt_C$ и суммарную цену ПЭ различного типа $Q_S = \sum_p C_p q^p$.

Здесь введены следующие обозначения: L — число тактов периода; t_C — длительность такта; C_p — аппаратная сложность ПЭ p -го типа; q^p — число ПЭ p -го типа.

При конвейерной обработке данных в ПЛИС особенно важно достижение минимальной длины критического пути, поскольку в ПЛИС на маршруты прохождения сигнала между логическими элементами приходится в среднем от 50 до 90 % задержки критического пути. Поэтому ПЛИС плохо конкурируют по быстродействию с заказными микросхемами. При отображении ГСПД в ПЛИС с $L > 1$ к задержке критического пути неизбежно добавляются задержки мультиплексоров. Кроме того, в

ПЛИС аппаратные затраты мультиплексоров могут существенно превосходить затраты регистров и сумматоров. Так, двухвходовый мультиплексор в ПЛИС фирм Altera, Xilinx занимает столько же ресурсов, как сумматор такой же разрядности. В ПЛИС фирмы Xilinx количество ресурсов, занимаемых n -входовым мультиплексором, при $n \leq 8$ приблизительно пропорционально n . Только пяти- и семивходовый мультиплексоры реализуются соответственно как шести- и восьмивходовый. Таким образом, с учетом частоты использования различных мультиплексоров в реальных проектах среднюю сложность одного входа мультиплексора ПЛИС можно оценить в 0,57 сложности регистра или АЛУ. Поэтому следует минимизировать как число ПЭ, так и число входов их мультиплексоров, отражающее сложность линий связи.

ГСПД представляется в трехмерном целочисленном пространстве в виде конфигурации алгоритма (КА) $K_G = (K, D, A)$, где K — матрица векторов-вершин \mathbf{K}_i , соответствующих операторам алгоритма; D — матрица векторов-дуг \mathbf{D}_j , соответствующих непосредственным информационным связям между операторами; A — матрица инцидентности ГСПД. В векторе-вершине $\mathbf{K}_i = \langle k_i, s_i, t_i \rangle$ координаты k_i, s_i, t_i соответствуют типу оператора (например, $k = 1$ — умножение), номеру процессорного элемента, где выполняется данный оператор, и такту, в котором записывается в регистр результат этого оператора.

Различным эквивалентным структурным решениям ВУ соответствуют эквивалентные КА, которые отличаются лишь разными матрицами K и D . Матрица K кодирует некоторое допустимое решение, так как матрица D вычисляется из уравнения $D = KA$. Поиск оптимального структурного решения заключается в нахождении такой матрицы K , которая минимизирует заданный критерий качества. Если применяется генетический метод оптимизации, то матрица K может быть использована в качестве гена представителя популяции. При поиске эффективных структурных решений необходимо руководствоваться следующими закономерностями.

КА является корректной, если в матрице K нет двух одинаковых векторов:

$$\forall \mathbf{K}_i, \mathbf{K}_j (\mathbf{K}_i \neq \mathbf{K}_j, i \neq j).$$

Расписание выполнения алгоритма корректно, если операторы, отображаемые в один и тот же ПЭ, выполняются в различных тактах:

$$\forall \mathbf{K}_i, \mathbf{K}_j (k_i = k_j, s_i = s_j) \Rightarrow t_i \neq t_j \pmod{L}.$$

Однотипные операторы следует отображать в ПЭ того же типа, т. е.

$$\mathbf{K}_i, \mathbf{K}_j \in K_{p,q} (k_i = k_j = p, s_i = s_j = q), |K_{p,q}| \leq L,$$

где $K_{p,q}$ — множество векторов-вершин операторов p -го типа, отображаемых в q -й ПЭ p -го типа ($q = 1, 2, \dots, q_{\max}^p$) [5].

Если ГСПД — циклический, т. е. в нем есть циклы межитерационной зависимости, то должна быть равна нулю сумма векторов-дуг \mathbf{D}_j , входящих в любой из циклов графа. Для i -го цикла $\sum b_{i,j} \mathbf{D}_j = 0$, где $b_{i,j}$ — элемент i -й строки цикломатической матрицы ГСПД. Такие циклы существуют, например, в алгоритмах рекурсивной фильтрации. Обратные векторы-дуги $\mathbf{D}_{Bj} = \langle 0, 0, -iL \rangle$ означают задержку, равную i циклов (итераций), где i — число, нагружающее соответствующую замыкающую дугу в ГСПД. Таким образом, КА равна объединению ациклической конфигурации, выполняющей вычисления одной итерации и множества обратных дуг \mathbf{D}_{Bj} , представляющих межитерационную задержку переменных на iL тактов. Эти векторы направлены в противоположную сторону относительно оси времени ot и называются замыкающими векторами.

Эффективная КА получается в два этапа. На первом этапе вершины ГСПД вместе с дугами размещаются в трехмерном пространстве как множества векторов \mathbf{K}_i и \mathbf{D}_j с учетом всех указанных выше условий, т. е. формируется начальная КА. Минимизируется число ПЭ в искомой структуре выполнением требования $|K_{p,q}| \rightarrow L$, т. е. число вершин, отображаемых в один ПЭ, стремится к L .

На втором этапе выполняется уравнивание КА. Для этого рассматривается ациклический подграф ГСПД без замыкающих векторов — дуг \mathbf{D}_{Bj} . Во все его дуги включаются промежуточные вершины операторов задержки (регистров). В результирующей уравновешенной КА все векторы-дуги, кроме замыкающих, имеют вид $\mathbf{D}_j = \langle a_j, b_j, 1 \rangle$ или $\mathbf{D}_j = \langle a_j, b_j, 0 \rangle$. При этом вершины-операторы образуют ярусы, расстояние между которыми по координате времени t равно одному такту.

Уравновешенная КА оптимизируется путем взаимных перестановок векторов-вершин из одного яруса с минимизацией числа регистров и входов мультиплексоров. Применяются и другие известные стратегии, например ресинхронизация или алгоритм левого края, минимизирующие число регистров.

Структуру ВУ и расписание выполнения алгоритма можно получить, расщепив КА K_G на конфигурацию структуры (КС) K_S и конфигурацию предшествования (КП), которые имеют ту же матрицу A , а векторы матрицы K_S координат ПЭ и матрицы моментов срабатывания K_T равны координатам векторов матрицы K , т. е. $\langle k_i, s_i \rangle$ и $\langle t_i \rangle$. КС и КП представляют собой отображение КА в подпространства структур и времени, которое выполняется элементарным образом.

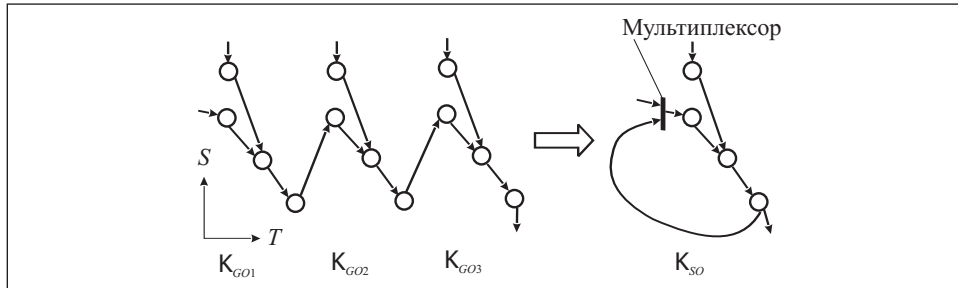


Рис. 1. Отображение КА согласно утверждению 2

Метод минимизации входов мультиплексоров. Рассмотрим метод минимизации числа мультиплексоров в конвейерных ВУ, наиболее эффективный для проектов ПЛИС. Очень часто ГСПД алгоритмов имеют изоморфные подграфы. Так, алгоритмы больших рекурсивных фильтров — это цепочки из одинаковых ступеней второго порядка, а в графах нерекурсивных фильтров можно выделить периодические фрагменты. Для минимизации входов мультиплексоров используем следующие утверждения.

Утверждение 1. Для корректной КА число входов N_{Mi} мультиплексоров i -й вершины ПЭ не больше числа неодинаковых векторов $\mathbf{D}_{i,j}$, инцидентных вершинам $\mathbf{K}_{i,k}$, которые отображаются в вершину ПЭ.

Доказательство утверждения 1 очевидно. Выражение «не больше» означает, что, не более чем с одного входа ПЭ подается операнд на один вход АЛУ этого ПЭ, и для такого входа мультиплексор не нужен.

Утверждение 2. Пусть ГСПД корректной конфигурации алгоритма K_G имеет до L изоморфных подграфов, которые представлены эквивалентными подконфигурациями $K_{GOi} = (K_{Oi}, D_{Oi}, A_{Oi})$. Тогда, для того чтобы при отображении КА K_G в структуру с конфигурацией K_S и периодом вычислений L получалось минимальное число ПЭ и входов их мультиплексоров, необходимо и достаточно, чтобы все подконфигурации K_{GOi} отображались в одну подконфигурацию структуры K_{SO} , причем

$$\begin{aligned} \forall \mathbf{K}_{i,j} \in K_{Oi} (\mathbf{K}_{i,j} = \langle c_{j,k}, c_{j,l}, t_{i,j} \rangle), \\ \forall \mathbf{K}_{i,j} \in K_{Oi}, \end{aligned} \quad (1)$$

($\mathbf{D}_{i,l}$ концом инцидентный $\mathbf{K}_{i,j} \Rightarrow \mathbf{D}_{i,l} = \langle c_k, c_l, 1 \rangle, c_l \neq 0$).

Следовательно, подграф структуры изоморфен подграфам алгоритма. На рис. 1 показан пример отображения КА с периодом $L = 3$ в соответствующую КС.

Доказательство достаточности. Нетрудно показать что при условиях (1) ПЭ с одноходовым АЛУ будет иметь один вход, а с двуххо-

довым АЛУ — соответственно, два входа, т. е. каждый ПЭ, кроме ПЭ ввода-вывода этой подструктуры, не имеет мультиплексоров.

Доказательство необходимости. Пусть одна или две вершины-операторов переставлены, т. е. не выполняются условия (1). Тогда возможны два случая. В первом — одна переставленная вершина отобразится в новую вершину ПЭ, т.е. увеличатся аппаратные затраты на один ПЭ. Кроме того, вектор $\mathbf{D}_{i,l}$, выходящая из этой вершины, изменит свои координаты и не будет равен вектору $\mathbf{D}_{i,k}$, который до этого вместе с ним отображался в одну дугу графа структуры. Поэтому следует добавить мультиплексор в ПЭ, в который поступает результат с нового ПЭ.

Во втором случае вершина будет отображена в другой, не полностью занятый, ПЭ или две вершины будут взаимно переставлены. Тогда на входах АЛУ этих ПЭ появятся двухходовые мультиплексоры. Таким образом, любые изменения в подконфигурациях, нарушающие требование (1), приводят к увеличению числа ПЭ или входов мультиплексоров.

Методика синтеза цифровых фильтров с кратными задержками.

Рассмотрим применение описанного выше метода для проектирования высокоэффективных цифровых фильтров, реализуемых в ПЛИС. Цифровые фильтры принято описывать передаточной характеристикой $H_0(Z)$ от комплексной переменной Z . Она однозначно соответствует алгоритму вычисления фильтра. При этом каждому члену Z^k в формуле характеристики соответствует задержка на k циклов или цепочка из k регистров задержки. Если в фильтре число регистров задержки увеличить в n раз, то получим фильтр с характеристикой $H_n(z) = H_0(Z^n)$. Такой фильтр с кратными задержками имеет следующую особенность: его амплитудно-частотная характеристика (АЧХ) по форме такая же, как у фильтра-прототипа $H_0(Z)$, но в диапазоне $0 — f_s$ она повторяется n раз, где f_s — частота дискретизации.

При последовательном соединении ступеней фильтров результирующая АЧХ — это пересечение АЧХ ступеней, что называют маскированием АЧХ. Также маскирование и использование фильтров с кратными задержками позволяет получать высокодобротные фильтры с минимальными аппаратными затратами [8]. Кроме того, если не экономить регистры задержки, то фильтр с кратными задержками — это эквивалент фильтра с несколькими частотами дискретизации, например системы из фильтров-дециматоров и интерполяторов. В ПЛИС фирмы Xilinx кратные задержки выполняют на блоках FIFO типа SRL16, занимающих такой же объем, как и отдельные регистры. Для синтеза таких фильтров на базе ПЛИС предлагается следующая методика.

На первом этапе выбираем алгоритм, ГСПД которого имеет изоморфные подграфы. Дуги графа нагружены задержками, которые кратны и рав-

ны K . Пусть число таких подграфов равно L . При этом условии достигается максимальная загруженность результирующей структуры. Подграф алгоритма, для которого $K = 1$, представляем в виде конфигурации периода алгоритма (КПА). В ней дополнительно выделяем пограничные вершины, в которые заходят и из которых выходят замыкающие векторы, соответствующие задержкам на k циклов. Эти пограничные вершины далее будут отображены в итерационные входы и выходы конфигурации структуры. КПА оптимизируется при условии, что она отображается в структуру с периодом вычислений, равным одному такту, т. е. с максимальным параллелизмом.

На втором этапе от двух до L КПА соединяются между собой в соответствии с алгоритмом фильтрации. При этом соблюдаются условия утверждения 2. Поскольку в некоторой ступени используются задержки на k тактов, к ее КПА добавляются подконфигурации задержек на $k-1$ периодов, которые через замыкающие векторы длиной k_L подсоединяются к соответствующим пограничным вершинам. Также добавляются вершины операторов и дуги, дополняющие КПА, подконфигурации задержек до полного ГСПД.

На третьем этапе КА отображается в конфигурацию структуры с периодом вычислений L . Результирующая структура включает в себя структуру, соответствующую КПА, к итерационным входам которой подключены мультиплексоры, а к их входам через задержки на $k-1$ периодов подаются задерживаемые данные.

Аппаратные затраты ВУ включают затраты на построение конвейерной структуры, в которую отображается КПА, мультиплексоры на итерационных входах с числом входов до L и задержки на различное число периодов (от 0 до $k-1$). При этом конвейерная структура мультиплексоров не содержит. Аппаратные затраты, выраженные в сложности сумматора, не считая числа регистров, можно определить по формуле $\Theta'_S = n_A + 10n_M + 0,57Ln_D$, где n_A и n_M — число сумматоров и блоков умножения, равное числу сложений и умножений в алгоритме ступени; n_D — число переменных, по которым выполняются межитерационные связи, т.е. число линий задержек в ступени фильтра. Здесь сложность умножителя принята равной сложности десятка сумматоров, так как в современных ПЛИС при их максимальной заполненности на один аппаратный умножитель приходится от шести до двадцати синтезируемых сумматоров такой же разрядности, т.е. в среднем — десять (для серии Xilinx Virtex 4) [4].

Число регистров в структуре зависит от топологии ГСПД ступени фильтра, распределения длины задержек по ступеням и пропорционально L . Результирующая структура при числе ступеней фильтра, равном L , имеет минимальное число регистров, сумматоров и блоков умножения. Это

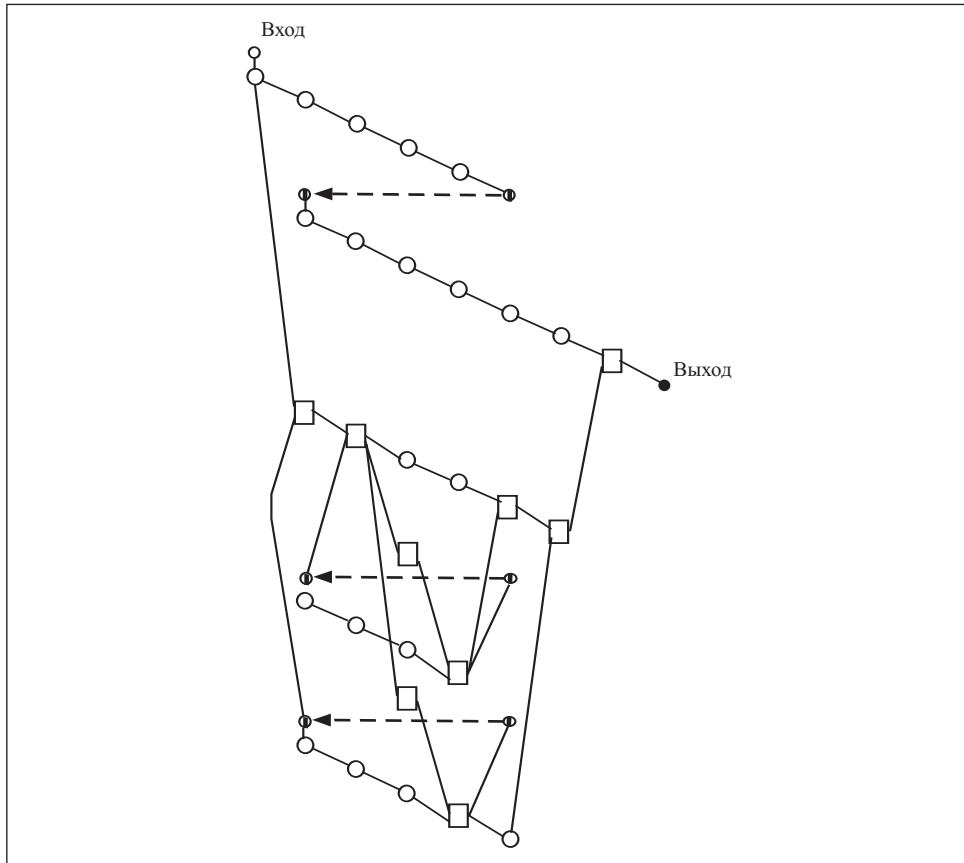


Рис. 2. Результирующая КА: ● — пограничная вершина; ○ — задержка на один такт или регистр; □ — сложение; ← - - — обратная дуга

утверждение основано на том, что при условии (1) ровно L вершин операторов задержки, операторов сложения и умножения отображаются соответственно в одну вершину регистра, сумматора и блока умножения. Следовательно, все ресурсы структуры асимптотически загружены на 100 %.

Пример синтеза цифрового фильтра. Рассмотрим пример синтеза фильтра нижних частот, состоящего из четырех ступеней с задержками кратности 1, 2, 4 и 8. Первые три ступени — это фильтры-псевдодециматоры, а последняя ступень — фильтр-формирователь. Фильтр-псевдодециматор выполняет фильтрацию нижних частот с такой же АЧХ, как у фильтра-дециматора, но не выполняет собственно децимацию. Вследствие того, что фильтры-псевдодециматоры имеют кратные задержки и выполняется маскирование, результирующая АЧХ имеет низкую частоту

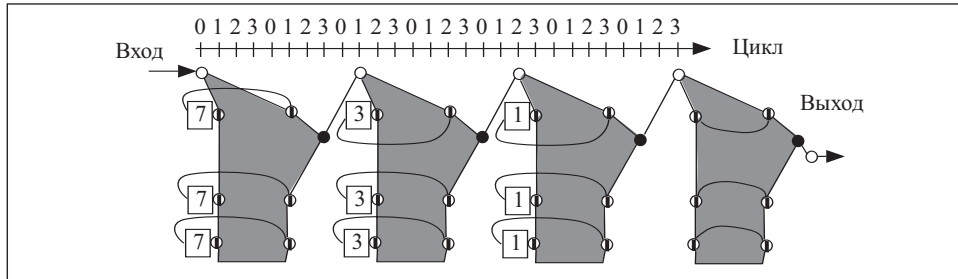


Рис. 3. Конфигурация периода алгоритма одной ступени фильтра: Φ — пограничная вершина; \square — задержка на один, три и семь тактов

среза (менее $f_s/16$) и узкую переходную полосу [8]. Каждая ступень выполнена на основе всепропускающего фильтра, что обеспечивает высокие и стабильные характеристики фильтра, а также возможность регулирования его характеристик. Передаточная функция одной (первой) ступени фильтра имеет вид

$$H(z) = Z^{-1} + \frac{Z^{-2} + b(1+a)Z^{-1} + a}{1 + b(1+a)Z^{-1} + aZ^{-2}}, \quad (2)$$

где b — коэффициент, регулирующий частоту среза; a — коэффициент, регулирующий крутизну спада [8].

На первом этапе синтеза фильтра формируется уравновешенная КПА (рис. 2), соответствующая волновому фильтру второго порядка с передаточной характеристикой (2).

На втором этапе четыре конфигурации периода алгоритма КПА соединяются между собой последовательно. На рис. 3 КПА представлены затемненными многоугольниками. На третьем этапе получаем структуру фильтра (рис. 4) и синтезируем его управляющий автомат. Реализация фильтра состоит в его описании на языке VHDL и дальнейшем отображении в ПЛИС. При этом можно использовать методику отображения КА в ПЛИС, описанную в [6].

Оценим аппаратные затраты и производительность полученного фильтра. При построении базовой структуры фильтра традиционным способом она имеет четыре последовательно соединенные ступени. Каждая ступень имеет семь сумматоров, два блока умножения и три регистра, которые предназначены для реализации конвейерного режима вычислений и приема входного операнда. Кроме того, ступени должны иметь по три регистровые задержки длиной 1, 2, 4 и 8 регистров, которые можно реализовать как блоки SRL16. Результирующие аппаратные затраты сос-

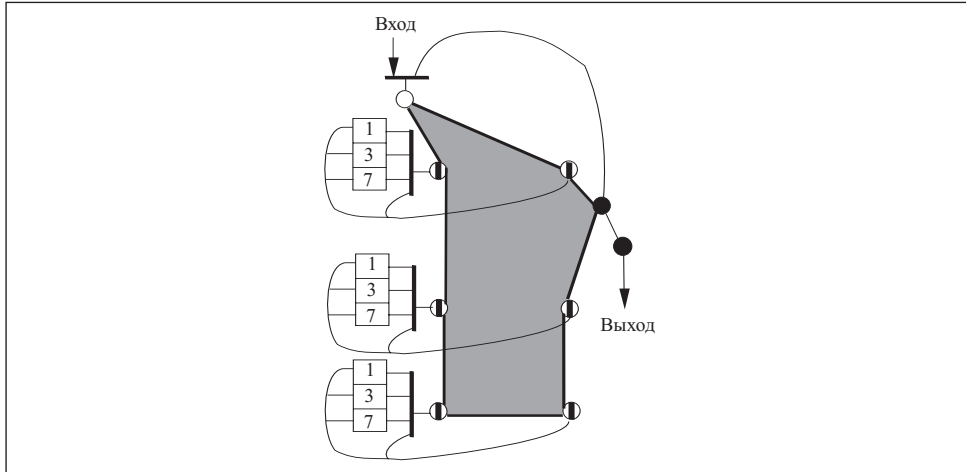


Рис. 4. Конфигурация структуры фильтра: Φ — пограничный вход-выход; \square — задержка на один, три и семь тактов; \mathbf{I} — мультиплексор

тавляют $\Theta_{SB} = 132$. Для такой структуры в критический путь входят четыре сумматора, т.е. период тактового интервала составляет $\Theta_{TB} = 4T_S$.

Синтезированная структура имеет два блока умножения, семь сумматоров, 26 регистров и регистровых задержек, а также один двухходовый и шесть четырехходовых мультиплексоров. В структуре задействовано 22 входа мультиплексора и результирующие аппаратные затраты составляют $\Theta_S = 65,5$. Период тактового интервала составляет $\Theta_T = T_S$, т. е. в критический путь входит лишь задержка одного сумматора или умножителя, а длительность цикла составляет $LT_S = 4T_S$. Таким образом, полученная структура фильтра обеспечивает вдвое меньшие аппаратные затраты, чем традиционная, при такой же производительности.

Цифровой фильтр с динамической перестройкой частоты среза. На основе описанных алгоритмов был разработан проект цифрового фильтра с динамической перестройкой частоты среза. При этом для улучшения фильтрации в приведенном выше примере порядок всепропускающего фильтра повышен до трех. Частота среза в фильтре перестраивается скачками — отводом сигнала от той или иной ступени, а также плавно — изменением коэффициентов ступени, от которой выполняется отвод, т. е. коэффициентов фильтра-формирователя. В результате частота среза изменяется плавно в пределах от $0,015$ до $0,4 f_S$. Поскольку ступени образуют пары одинаковых фильтров, а число ступеней увеличено до $L = 8$, при любой заданной частоте среза уровень подавления составляет не менее 75—80 дБ, а наклон АЧХ в переходной полосе — не менее 100 дБ/октава.

Для задания частоты среза 12-битным кодом применен аппаратный калькулятор, пересчитывающий код частоты в значение коэффициентов фильтра-формирователя. Поскольку алгоритм волнового фильтра принципиально не допускает таких явлений, как нестабильность работы, возбуждение, при всех возможных кодах частоты и входных данных фильтр работает стабильно.

Фильтр был реализован в ПЛИС Xilinx Virtex2P. Его аппаратные затраты без затрат на реализацию калькулятора коэффициентов составляют 706 эквивалентных конфигурируемых логических блоков (CLB slices) и три блока умножения. Вследствие высокой степени конвейеризации структуры тактовая частота достигает 190 МГц. Таким образом, фильтр может обрабатывать в реальном времени сигналы с частотой дискретизации до 24 МГц.

Выводы. Описанный метод проектирования конвейеризованных ВУ для реализации периодических алгоритмов, обеспечивает минимизацию как числа АЛУ, блоков умножения, так и числа входов мультиплексоров в результате использования особенностей алгоритмов и структурных свойств современных ПЛИС. При этом синтезированное ВУ имеет минимальный период тактового интервала при выполнении алгоритма в конвейерном режиме с заданным периодом вычислений L .

Разработанные методы и методика проектирования многоступенчатых цифровых фильтров с кратными задержками обеспечивает при заданных ограничениях (период L , элементная база, ГСПД с изоморфными подграфами) минимальные период синхросерии и аппаратные затраты. Проверка методики при проектировании многоступенчатых волновых фильтров показала, что можно уменьшить аппаратные затраты в два раза при такой же производительности по сравнению с традиционной структурой.

A method of mapping periodical algorithms into pipeline processor based on FPGA is considered. The algorithm is represented by the synchronous dataflow graph. The method lies in placing the algorithm graph in multidimensional index space and then in its mapping into subspaces of structures and time. The limitations to the mapping process allow to minimize both the clock cycle and hardware volume including multiplexers.

1. *Bhattacharyya S. S., Leupers R., Marwedel P.* Software Synthesis and Code Generation for Signal Processing Systems // IEEE Trans. on Circuits and Systems—II: Analog and Digital Signal Processing.— 2000. — Vol 47, № 9. — P. 849—875.
2. *The Synthesis Approach to Digital System Design* / Ed. P. Michel, U. Lauther, P. Duzy. — NY: Kluwer Academic Pub, 1992. — 415 p.
3. *Eles P., Kuchinski K., Zebo P.* System Synthesis with VHDL. — NY : Kluwer Academic Pub, 1998. — 370 p.
4. *Hill T.* The benefits of FPGA Coprocessing // Xcell journal: Xilinx. — 2006. — Vol. 58, № 3. — P. 29—31.

5. Каневский Ю. С., Овраменко С. Г., Сергиенко А. М. Отображение регулярных алгоритмов в структуры специализированных процессоров // Электрон. моделирование. — 2002. — 24. № 2. — С. 46—59.
6. Сергиенко А. М. VHDL для проектирования вычислительных устройств. — Киев : ДИАСофт, 2003. — 208 с.
7. Каневский Ю. С., Логинова Л. М., Сергиенко А. М. Структурное проектирование рекурсивных цифровых фильтров // Электрон. моделирование. — 1995. — 17, № 3. — С. 18—22.
8. Chung J. G., Parhi K. K. Pipelined wave digital filter design for narrow-band sharp-transition digital filters // Proc. IEEE Workshop VLSI Signal Processing. CA, La Jolla. — 1994. — P. 501—510.

Поступила 18.09.06;
после доработки 07.11.06

СЕРГИЕНКО Анатолий Михайлович, канд. техн. наук, ст. науч. сотр. Национального технического университета Украины «КПИ», который окончил в 1981 г. Область научных исследований — отображение алгоритмов в структуры вычислительных средств, цифровая обработка сигналов.

СИМОНЕНКО Валерий Павлович, д-р техн. наук, профессор Национального технического университета Украины «КПИ», который окончил в 1965 г. Область научных исследований — организация вычислительных процессов в вычислительных системах.