Зб. праць ювілейної міжн. наук.-практ. конф. 50-річчя ств. каф. ОТ. –Київ, 6-8 квітня 2010. – Київ: НТУУ «КПІ». –с.125-127.

ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ, РЕАЛИЗУЕМЫХ АППАРАТНО

Сергиенко А.М., Лепеха В.Л., Лесик Т.М. НТУУ «КПИ»

Введение

Персональные компьютеры, средства цифровой обработки сигналов (ЦОС) и многие другие вычислительные системы (ВС) реального времени выполняют периодические алгоритмы, обрабатывающие потоки данных и/или команд. Область ЦОС основана именно на обработке потоков данных и поэтому она наиболее ярко отображает задачи, алгоритмы и устройства обработки таких потоков. Для их реализации алгоритмов ЦОС необходимо использовать специальные ВС на основе заказных СБИС, многопроцессорных сигнальных микропроцессоров, а также програмируемые логические интегральные схемы (ПЛИС).

Сейчас для программирования многопроцессорных параллельных ВС применяется парадигма многопотоковой обработки (multithreading), которая поддерживается как на уровне матобеспечения, так и на аппаратном уровне современных компьютеров. Но многопотоковая модель не гарантирована от блокировок. Поэтому вместо нее предложено использовать модель графа синхронных потоков данных (ГСПД) и его модификации, гарантирующие безопасность алгоритма [1].

Современные задачи ЦОС не могут быть решены без разработки методов и средств отображения алгоритмов в СБИС и ПЛИС, программирования многопроцессорных ВС. Такая разработка возможна, прежде всего, при условии нового видения природы алгоритмов обработки потоков данных. В докладе предлагается модель пространственного ГСПД, которая дает возможность проектировать эффективные аппаратные устройства для реализации периодических алгоритмов.

Модели потоков данных

По определению Поста и Тьюринга, *алгоритм* – это вычислительный процесс, который выполняется моделью вычислителя, которая сконструирована в рамках точных математических понятий [2]. Модель вычислителя, которая реализует функциональный алгоритм, принято представлять в виде графа. В такой модели вершины графа означают операторы или локальные процессы алгоритма, а дуги – каналы передачи данных, зависимости по данным и управлению. Реализация алгоритма на такой модели представляет собой передачу данных в направлении дуг и обход вершин в сответствии с их инцидентностью, наличием данных на их входах или по другим правилам. На рис.1 показан граф классификации наиболее известных графовых вычислительных моделей.

Анализ различных графовых моделей с точки зрения детерминизма алгоритма, возможностей и удобства задания алгоритма для обработки потоков данных и отображения его в аппаратные средства приведен в [3,4]. Далее будет рассмотрена модель ГСПД, как модель, наиболее приближенная к заданию аппаратно реализованных алгоритмов.

Граф потоков данных (ГПД) – это направленный граф, вершины которого – акторы – представляют операции, а дуги –потоки передачи данных. Акторы потребляют со своих входов данные, называемые метками или токенами и выдают метки на свои выходы. Производными от ГПД (рис.1) являются ГПД Денниса, граф Карпа и Миллера, сеть Петри, сеть процессов Кана, ГСПД и др.

Корректность задания алгоритма на модели вычислителя может быть проверена или аналитически (если это возможно), или путем составления расписания выполнения вычислительного процесса на ней. Расписание также составляется при выполнении алгоритма в определенной вычислительной системе (ВС). Если структура ВС отвечает графу модели, то составление расписания состоит в определении порядка выполнения операторов (акторов, процессов). Если структура ВС произвольная, то предварительно выполняют назначение операторов на процессоры этой ВС.

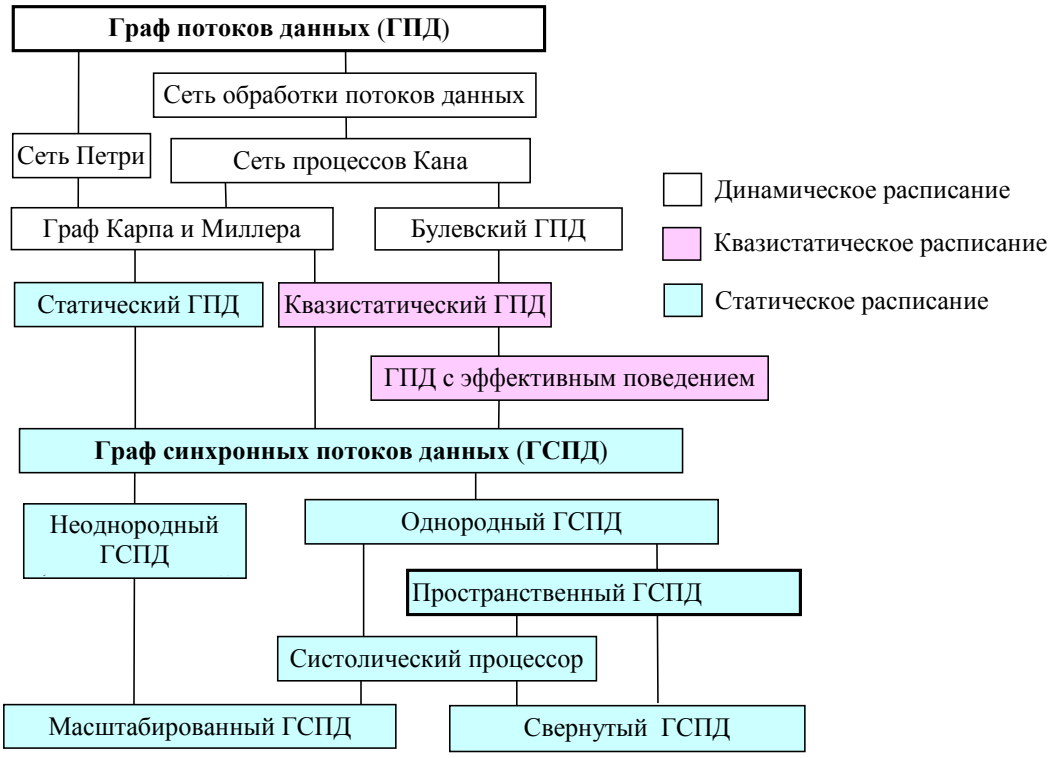

Рис.1. Классификация графов потоков данных

Статическое расписание составляется до выполнения алгоритма в ВС, т.е. во время компиляции, а при *динамическое расписание* можно определить только при исполнении алгоритма в ВС при обработке конкретных данных. Согласно этим расписаниям, различают *статические и динамические модели*. Если поведение и параметры модели можно предвидеть на стадии компиляции (максимальный период вычислений, размер буферов потоков и т.п.), то такую модель называют моделью с *квазистатическим расписанием*. Следует отметить, что при отображении алгоритма в аппаратуру СБИС и ПЛИС допускаются алгоритмы только со статическим расписанием.

Таким образом, ГПД представляют собой естественные модели для задания алгоритмов обработки потоков данных. Они имеют широкие возможности выразительного задания большого множества алгоритмов на высоком уровне абстракции. Платой за это является необходимость динамического составление расписания, и в результате − недетерминированность поведения, которая угрожает наличием блокировок [1,3]. Из-за этого они используются, в основном, в ВС с динамическим расписанием и не приспособлены для синтеза аппаратных средств вычислительной техники.

Поток данных – это средство передачи упорядоченных данных между компонентами модели. Данные в потоке, упорядоченные по номерам ассоциированных с ними тегов, например, временных меток, называют *сигналом*. Сигналы и соответствующие им потоки считаются *синхронными*, если теги пар сигналов имеют взаимно-однозначное соответствие. Модель считается *синхронной*, если все сигналы в ней – синхронные, иначе – это *асинхронная модель*.

Как правило, синхронный сигнал производится *регулярным актором*, т.е. актором, который использует и производит некоторое количество меток, которое стабильно при выполнении алгоритма и может быть определено во время компиляции. В отличие от него, *динамический актор* выдает различное число данных в зависимости от данных в его входных потоках, т.е. генерирует асинхронный сигнал.

Граф Карпа и Миллера, в котором все акторы — регулярные, получил название *графа синхронных потоков данных* (ГСПД, synchronous data flow, SDF). Так как метки в потоках ГСПД можно перенумеровать номерами циклов, то потоки являются синхронными. Если количество меток,

использованных каждым входом и сгенерированных каждой вершиной в одном цикле, одинаково и равно константе, то такой граф называют *однородным* ГСПД (homoheneous SDF). А если это количество различно, то это *неоднородный* (многоскоростной) ГСПД (multirate SDF).

Неоднородный ГСПД – это удачная модель для компактного задания периодических алгоритмов. Но его сложнее анализировать в сравнении с однородным ГСПД. Кроме того, его моделирование усложняется из-за возможности блокировок. Поэтому для своего анализа и синтеза на его основе ВС неоднородный ГСПД часто преобразуют в эквивалентный однородный ГСПД. Модель неоднородного ГСПД получила большое распространение и служит, например, основой пакета Matlab-Simulink.

ГСПД – это статическая модель. Для неоднородного ГСПД отсутствие блокировок несложно определяется при решении системы уравнений баланса $\Gamma \boldsymbol{r} = 0$, где Γ – топологическая матрица графа, $\boldsymbol{r}$ – вектор повторений срабатывания акторов. Однородный ГСПД никогда не блокируется при наличии меток начального состояния в буферах, входящих в циклы графа и поэтому блокировки в нем не проверяются. Таким образом, среди большого разнообразия потоковых моделей алгоритмов, ГСПД является наиболее подходящей для задания алгоритмов, отображаемых в аппаратные средства.

Применение модели ГСПД для проектирования аппаратных средств

ГСПД всегда ассоциировались с алгоритмами ЦОС. Однородный ГСПД взаимно однозначно отвечает сигнальному графу алгоритма ЦОС или вычислительной схеме с периодом вычислений один такт. Существуют его такие разновидности, как *масштабированный* ГСПД (scalable SDF), *многомерный* ГСПД, *цикло-статический* ГСПД (cyclo-static dataflow), *параметрический* ГСПД, блоковый ГПД (Blocked Data Flow), предназначенные для задания алгоритмов решения современных задач ЦОС. В [3] дан обзор методов отображения ГСПД в многопроцессорные ВС на основе программной реализации акторов.

Согласно общепринятой методологии, отображение ГСПД в ВС выполняют в три этапа. На первом этапе выбирают множество аппаратных ресурсов синтезируемого вычислителя, ассортимент и количество которых отвечают набору операторов алгоритма и требуемой производительности. На втором – составляется расписание исполнения акторов на выбранных ресурсах. И на третьем этапе – этапе назначения − акторы распределяются среди ресурсов, данным назначаются элементы памяти, определяется структура ВС, планируются пересылки данных, синтезируется управляющий автомат. Оптимизация ВС состоит в формировании множества альтернативных решений и выборе наиболее предпочтительной ВС по заданному критерию качества.

Такое отображение ГСПД часто приводит к решениям, далеким от оптимальных. Это объясняется тем, что этапы отображения преследуют противоречивые цели − выбор ресурсов минимизирует аппаратные затраты, а составление расписания – минимизирует время вычислений за счет увеличения аппаратных ресурсов. Окончательная оценка аппаратных затрат выполняется на последнем этапе, когда уже готово расписание. Поэтому для оптимизации необходимо повторять этапы отображения итерационно.

Составление расписания сводится к построению на основе ГСПД графа зависимости по данным (ГЗД) и отображению его топологической сортировки в пространство событий. При этом точное решение этой задачи (целочисленное линейное программирование, метод ветвей и границ) возможно только для числа вершин не более двух-трех десятков. Такой эффективный эвристический метод, как силовое планирование, также не учитывает факторы, возникающие на этапе размещения.

Таким образом, модель ГСПД, обладая выразительностью, отсутствием блокировок, имеет трудности с отображения в аппаратные ресурсы. Во многом, это связано с тем, что в ВС отображается не сам ГСПД, а соответствующий ему ГЗД (составление расписания) и множество его вершин (назначение на ресурсы), а межпроцессорные связи ВС, внутренняя структура процессорных элементов (ПЭ), управляющий автомат формируются искусственно.

Поэтому предлагается вершинам и дугам ГСПД назначить теги, которые содержат многомерные векторы целочисленных координат, благодаря которым ГСПД можно отображать непосредственно и одновременно как в структуру ВС, так и ее расписание, что будет рассмотрено ниже.

Важно отметить, что граф систолического процесора, по существу, является однородным ГСПД. Причем, его вершины и дуги также имеют теги из многомерных целочисленных координат. Однако, в отличие от отображения ГСПД, сами систолические процессоры синтезируются путем отображения ГЗД. Таким образом, ГСПД можно строить, используя эффективные и формальные методы синтеза систолических процессоров [5].

Пространственный ГСПД

Любой граф G_A, в том числе ГСПД и соответствующий ему ГЗД, задается матрицей инцидентности

$$A = ||a_{i,j}||_{N*M},$$

где $a_{i,j} = 1$, если дуга j графа G_A концом инцидентна i-й вершине или $a_{i,j} = -1$, если началом инцидентна i-й вершине оператора, и $a_{i,j} = 0$ − при отсутствии инцидентности, N и M − число вершин и дуг графа G_A.

Данная матрица инцидентности − нетрадиционна. В большинстве случаев задания графов, если дуга концом инцидентна вершине, то $a_{i,j} = -1$, что соответствует направлению потока через дугу. В нашем случае − матрица инцидентности имеет противоположный знак. Это связано с тем, что разность координат дуги, умноженных на $a_{i,j}$, указывает на положительную задержку операнда.

В общем случае, ГСПД можно представить в n-мерном целочисленном пространстве $\mathbf{Z}^n$. Векторы $\boldsymbol{K}_i \in \mathbf{Z}^n$ отождествляются с вершинами графа G_A, т.е. играют роль тегов. Координаты векторов $\boldsymbol{K}_i$, могут иметь разную интерпретацию. Часть координат векторов $\boldsymbol{K}_i$ представляют собой координаты $\boldsymbol{K}_{Si}$ ПЭ, в котором выполняется i-й оператор, а другая часть $\boldsymbol{K}_{Ti}$ кодирует момент времени выполнения этого оператора $\boldsymbol{K}_i = \left(\boldsymbol{K}_{Si}^{\mathrm{T}}, \boldsymbol{K}_{Ti}^{\mathrm{T}}\right)^{\mathrm{T}}$. В другой интерпретации, при отображении гнезда циклов с ядром в виде одного оператора, вектор $\boldsymbol{K}_i$ является набором индексов переменных.

По другим координатам векторов $\boldsymbol{K}_i$ может вестись отсчет, например, уровней иерархии алгоритма и структуры, типа оператора и ПЭ, задержки выполнения оператора, кодироваться информация об особенностях операторов и ПЭ, таких, как необходимость конвейерной реализации, заданный момент выполнения и т.п. Для простоты изложения, но без потери смысла далее будем рассматривать размерность пространства $n = 3$. При изображении ГСПД алгоритма в 3-мерном целочисленном пространстве $\mathbf{Z}^3$ i-й оператор отождествляется с радиусом-вектором $\boldsymbol{K}_i = (s, p, t)^{\mathrm{T}}$, ($i = 1...N$), где p – тип операции, s − пространственная координата, t – временная координата. Более подробно о семантике координат векторов будет изложено при рассмотрении отображения ГСПД.

В простом случае, вершины ГСПД генерируют метки, которые отвечают одиночным переменным x_j и по всем выходным дугам одной вершины передается эта же переменная x_j. В других случаях, когда вершина выдает различные переменные в разные дуги, следует расщепить оператор вершины на несколько операторов, которые генерируют одиночные результаты и которые эквивалентны исходному оператору.

Аналогично, каждая переменная x_j из M переменных алгоритма отождествляется с n-мерным вектором $\boldsymbol{D}_j$; ($j = 1...M$) $\in \mathbf{Z}^3$, причем

$$\boldsymbol{D}_j = \boldsymbol{K}_l - \boldsymbol{K}_i, \qquad (2.5)$$

где $\boldsymbol{K}_i$ отвечает оператору с результатом x_j, а $\boldsymbol{K}_l$ − оператор, для которого x_j – исходный операнд, т.е. $\boldsymbol{K}_i$ непосредственно предшествует $\boldsymbol{K}_l$.

Множество векторов $K = \{\boldsymbol{K}_i\}$ и $D = \{\boldsymbol{D}_j\}$ взаимосвязаны точно так же, как множества вершин и дуг алгоритма. Таким образом, представление алгоритма в многомерном пространстве означает его кодирование в виде матриц векторов $\boldsymbol{K}_i$, $\boldsymbol{D}_j$, а также матрицы инцидентности алгоритма. Для того, чтобы точно задать ГСПД с помощью этих множеств векторов, необходимы следующие определения:

Конфигурацией алгоритма (КА) является тройка $C_A = (K,D,A)$, где $K = (\boldsymbol{K}_1, ..., \boldsymbol{K}_N)$ — матрица координат векторов-вершин $\boldsymbol{K}_i = (s_i, p_i, t_i)^{\mathrm{T}} \in \mathbf{Z}^3$; $D = (\boldsymbol{D}_1, ..., \boldsymbol{D}_{M+1})$ — матрица координат векторов-дуг $\boldsymbol{D}_j = (s_j, p_j, t_j)^{\mathrm{T}} \in \mathbf{Z}^3$; $A = ||a_{i,j}||_{N*(M+1)}$ – матрица инцидентности графа G_A, которая дополнена M+1-м столбцом с элементом $a_{p,M+1} = 1$, которому отвечает базисный вектор $\boldsymbol{D}_{M+1} = \boldsymbol{D}_B$,

соединяющий начало системы координат пространства $\mathbf{Z}^3$ с некоторым произвольным вектором-вершиной $\boldsymbol{D}_B = \boldsymbol{K}_p \in K$ [6].

Название „конфигурация" происходит от понятия комбинаторной конфигурации, которая представляет собой один из многих элементов множества, которое составлено из групп элементов другого множества. Так же, как и комбинаторная конфигурация, КА задается элементом множества возможных представлений алгоритма в многомерном пространстве, т.е. фактически элементом множества решений задачи отображения алгоритма.

Все векторы-операторы $\boldsymbol{K}_i$ конфигурации алгоритма занимают в пространстве ресурсы − время (с координатами *s,t*) площадь некоторой фигуры, покрываемой прямоугольником со сторонами S_m и T_m. При максимальной загруженности ПЭ структуры, площадь прямоугольника $Q = S_m T_m$ принимает минимальное значение. При различных вариантах отображения можно получить разное время выполнения алгоритма T_m и количество ПЭ S_m. Причем, при условии максимальной загруженности ПЭ, площадь прямоугольника Q остается приблизительно постоянной, т.е. в определенной степени можно получить масштабируемую результирующую структуру ВУ .

На рис. 2 показаны примеры двух вариантов размещения некоторого алгоритма в двумерном пространстве, которые иллюстрируют масштабируемость отображения КА.

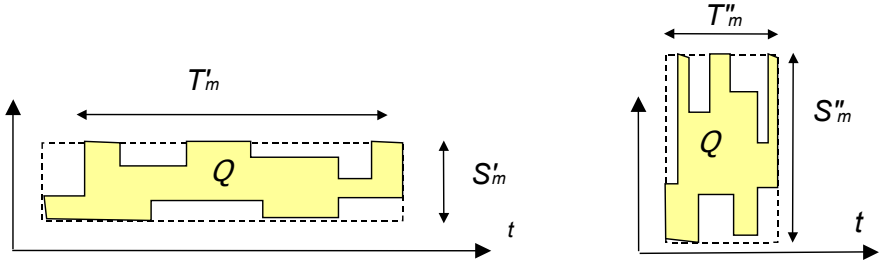

Рис.2 − Варианты размещения алгоритма в пространстве структура-время

Применение пространственного ГСПД для синтеза конвейерных ВС

Конвейерная ВС представляет собой многоступенчатый аппаратный вычислитель, который обрабатывает потоки данных. Его структура может быть получена путем отображения ГСПД. Пространственный ГСПД дает возможность формального построения конвейерных вычислителей на основе метода его отображения, описанного ниже.

Если ГСПД алгоритма представлен в пространстве $\mathbf{Z}^n$, то в соответствии с условием инъективности отображения алгоритма, граф структуры ВС и моменты выполнения операторов должны быть представлены в подпространствах $\mathbf{Z}_S$ и $\mathbf{Z}_T$, прямая сумма которых равна исходному пространству $\mathbf{Z}^n = \mathbf{Z}_S \oplus \mathbf{Z}_T$. Для того, чтобы упростить процесс выделения этих подпространств, целесообразно выбирать их с ортонормированными базисами, которые совпадают с осями ординат, т.е. $\mathbf{Z}_S = \mathbf{Z}^m$ и $\mathbf{Z}_T = \mathbf{Z}^{n-m}$.

Конфигурацией структуры (КС) является тройка $C_S = (K_S, D_S, A)$, где $K_S = (\boldsymbol{K}_{S1},...,\boldsymbol{K}_{SN})$, и $D_S = (\boldsymbol{D}_{S1}, ..., \boldsymbol{D}_{S(M+1)})$ − матрицы координат векторов-вершин $\boldsymbol{K}_{Si} = (k_{Si,1},...,k_{Si,m})^{\mathrm{T}} \in \mathbf{Z}^m$ и векторов-дуг $\boldsymbol{D}_{Sj} = (d_{Sj,1},...,d_{Sj,m}) \in \mathbf{Z}^m$, которые равняются координатам ПЭ, в которых выполняется *i*-й оператор и относительным координатам линий связи структуры, по которой передается *j*-я переменная, соответственно; A − матрица инцидентности ГСПД. В простом случае, $\boldsymbol{K}_{Si} = (s_i, p_i)^{\mathrm{T}} \in \mathbf{Z}^2$.

Конфигурацией предшествования (КП) является тройка $C_T = (K_T, D_T, A)$, где $K_T = (\boldsymbol{K}_{T1},...,\boldsymbol{K}_{TN})$, и $D_T = (\boldsymbol{D}_{T1}, ..., \boldsymbol{D}_{T(M+1)})$ −матрицы координат векторов-вершин $\boldsymbol{K}_{Ti} = (k_{Ti,1},...,k_{Ti,(n-m)})^{\mathrm{T}} \in \mathbf{Z}^{n-m}$, $\boldsymbol{D}_{Tj} = (d_{Tj,1},...,d_{Tj,(n-m)})^{\mathrm{T}} \in \mathbf{Z}^{n-m}$, которые отвечают моментам срабатывания *i*-го оператора и относительной задержке передачи *j*-й переменной, A − матрица инцидентности ГСПД или ГЗД.

В простейшем случае, $\boldsymbol{K}_{Ti} = (t_i) \in \mathbf{Z}$. Тогда матрице K_T отвечает *вектор временной развертки* $\boldsymbol{T} = (t_1,...,t_N)^{\mathrm{T}}$, который указывает, в какие моменты времени срабатывают определенные операторные вершины.

КА является *корректной*, если в матрице K нет двух одинаковых векторов, т.е.

$$\forall\, \boldsymbol{K}_{\mathrm{i}},\boldsymbol{K}_{\mathrm{j}}\ (\boldsymbol{K}_{\mathrm{i}}\neq \boldsymbol{K}_{\mathrm{j}},\ i\neq j\ ). \tag{1}$$

Если в ГСПД есть циклы, то должна быть равна нулю сумма векторов-дуг $\boldsymbol{D}_{\mathrm{j}}$, входящих в любой из циклов графа, т.е. для i-го цикла

$$\sum b_{\mathrm{i,j}}\, \boldsymbol{D}_{\mathrm{j}} = 0, \tag{2}$$

где $b_{\mathrm{i,j}}$ – элемент i-й строки цикломатической матрицы ГСПД.

КП C_T для ГСПД является (*строго*) *корректной* относительно временной функции R, если

$$\boldsymbol{K}_{Tr} - \boldsymbol{K}_{Ti} = \boldsymbol{D}_{Tj} \in D_T \Rightarrow \quad \begin{cases} R\,(\boldsymbol{K}_{Tr}) \geq R\,(\boldsymbol{K}_{Ti});\ - \text{для ненагруженных дуг} \\ R(\boldsymbol{K}_{Tr}) < R(\boldsymbol{K}_{Ti}).\ - \text{для дуг, нагруженных задержками} \end{cases} \tag{3}$$

$$i,r = 1,...,N;\ \ j = 1,...,M+1;$$

(для строгой корректности − неравенство строгое), т. е. если две вершины связаны дугой, то момент срабатывания оператора предшествующей вершины всегда не больше момента срабатывания следующей.

Расписание для КА $C_A = (K,D,A)$ ГСПД, выполняемого с периодом L тактов является корректным, если операторы, отображаемые в один и тот же ПЭ, выполняются в различных тактах, т.е.

$$\forall \boldsymbol{K}_i,\ \boldsymbol{K}_j \in K^0(k_{i,r} = k_{j,r};\ r=1,...,n-1) \Rightarrow \begin{cases} k_{i,n} \equiv k_{j,n} \bmod L - \text{если } (\boldsymbol{K}_i \boldsymbol{K}_j) - \text{дуга, нагруженная задержками} \\ k_{i,n} \not\equiv k_{j,n} \bmod L\ - \text{в остальных случаях} \end{cases} \tag{4}$$

КА линейным отображением своих компонентов может быть преобразованная в КС и КП, так что:

$$K = \begin{pmatrix} K_S \\ K_T \end{pmatrix};\ \ D = \begin{pmatrix} D_S \\ D_T \end{pmatrix}, \tag{5}$$

если КА, КП и расписание являются корректными, т.е. удовлетворяют условиям (1) − (4), причем полученная модель ВС будет выполнять заданный алгоритм в конвейерном режиме с периодом L тактов.

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

$$\boldsymbol{K}_{\mathrm{i}},\boldsymbol{K}_{\mathrm{j}} \in K_{\mathrm{p,q}}(k_{\mathrm{i}}=k_{\mathrm{j}}=p,\ s_{\mathrm{i}}=s_{\mathrm{j}}=q),\ |K_{\mathrm{p,q}}| \leq L,$$

где $K_{\mathrm{p,q}}$ – множество векторов-вершин операторов p-го типа, отображаемых в q-й ПЭ p-го типа (q=1,2,… $q^{\mathrm{p}}_{\mathrm{max}}$).

Эффективную КА ищут в два этапа. На первом этапе вершины ГСПД вместе с дугами размещаются в трехмерном пространстве как множества векторов $\boldsymbol{K}_{\mathrm{i}}$ и $\boldsymbol{D}_{\mathrm{j}}$ с учетом условий, приведенных выше, т.е. формируется начальная КА. Минимизируется число ПЭ в искомой структуре путем выполнения требования $|K_{\mathrm{p,q}}| \rightarrow L$, т.е. число вершин, отображаемых в один ПЭ, стремится к L. Также возможна перестановка вершин относительно оси времени *ot*, которая отвечает ресинхронизации ГСПД.

На втором этапе КА уравновешивается. Рассматривается ациклический подграф ГСПД без обратных дуг $\boldsymbol{D}_{\mathrm{Bj}}$. Во все его дуги включаются промежуточные вершины операторов задержки (регистров). В результирующей уравновешенной КА все векторы-дуги, кроме обратных дуг, равны $\boldsymbol{D}_{\mathrm{j}} = \langle a_{\mathrm{j}},b_{\mathrm{j}},1\rangle$ или $\boldsymbol{D}_{\mathrm{j}} = \langle a_{\mathrm{j}},b_{\mathrm{j}},0\rangle$. При этом вершины-операторы образуют ярусы, расстояние между которыми по координате времени *ot* равно 1 такт. Уравновешенная КА оптимизируется путем взаимных перестановок векторов-вершин, принадлежащих одному ярусу с минимизацией числа регистров и входов мультиплексоров. Применяются и другие стратегии, например, алгоритм левого края.

Структуру ВС и расписание выполнения алгоритма можно получить, расщепив КА K_G на КС K_S и 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. $\langle k_{\mathrm{i}},s_{\mathrm{i}}\rangle$ и $\langle t_i\rangle$, т.е. согласно (5). КС и

КП представляют собой отображение КА в подпространства структур и времени, которое выполняется элементарным образом.

Данный метод дает возможность целенаправленно выполнять поиск как структуры ВС, так и расписания выполнения алгоритма в ней. В [7] метод усовершенствован с целью минимизации аппаратных затрат ПЛИС, в которой реализуется конвейерная ВС. При этом благодаря тому, что результатом отображения выступает описание ВС на языке VHDL, метод дает возможность не строить собственно структуру ВС и расписание выполнения алгоритма, а переложить это задание на компилятор-синтезатор проекта для ПЛИС. Пространственный ДГПД с эффективным поведением отображается в конвейерную ВС методом, предложенным в [8]. Этот метод дает возможность формально отображать алгоритмы с операторами управления в структуру конвейерных ВС с заданным периодом вычислений, которые имеют минимизированные аппаратные затраты и высокую тактовую частоту. Методы проверены при синтезе ряда конвейерных устройств ЦОС, таких как цифровые фильтры, процессоры быстрого преобразования Фурье, дискретного косинусного преобразования и многих других, которые имеют оптимизированное отношение производительность − аппаратные затраты. Методы также могут быть эффективно использованы для программирования многопроцессорных ВС [9], а также такой SIMD-архитектуры, как Intel MMX [10].

Выводы

Граф потоков данных (ГПД) представляет собой естественную модель для задания алгоритмов обработки таких потоков. Динамические ГПД имеют широкие возможности для задания алгоритмов, но их анализ усложнен и они могут иметь блокировки. Из-за этого они используются, в основном, в системах с динамическим расписанием и не приспособлены для синтеза конвейерных вычислительных систем.

Конвейерные ВС следует проектировать путем отображения графов синхронных потоков данных (ГСПД) или ГПД с эффективным поведением и квазистатических ГПД, которые имеют ряд свойств, таких же, как у ГСПД. Меньшая выразительность и большая трудоемкость представления алгоритма на модели однородного ГСПД компенсируется тем, что при представлении такого ГСПД в многомерном пространстве в виде пространственного ГСПД его отображение в конвейерную структуру выполняется формально с получением минимизированных аппаратных затрат.

1. Lee E. A. The Problem with Threads // IEEE Computer. –2006. —V39. −№5. –P.33-42.
2. Котов В.Е. Введение в теорию схем программ. –Новосибирск: Наука. —1978. –258 с.
3. Bhattacharya S.S., Leupers R. Marwedel P. Software Synthesis and Code Generation for Signal Processing Systems // IEEE Trans. on Circuits and Systems, Part II, Analog and Digital Signal Processing. – 2000. –V.47. −№ 9. –p. 849−875.
4. Сергиенко А.М., Симоненко В.П. Алгоритмические модели обработки потоков данных // Электрон. моделирование.–2008. − Т.30, № 6.−С. 49−60.
5. Каневский Ю.С. Систолические процессоры. −Киев:Техніка. —1991. —172 с.
6. Каневский Ю.С., Овраменко С.Г., Сергиенко А.М. Отображение регулярных алгоритмов в структуры специализированных процессоров //Электрон. моделирование.−2002.−Т.24.−№2.−С. 46−59.
7. Симоненко В.П., Сергиенко А.М. Отображение периодических алгоритмов в программируемые логические интегральные схемы //Электрон. моделирование. – 2007. – Т.29. №2. –С.49−61.
8. Сергиенко А.М. Синтез структур для выполнения периодических алгоритмов с операторами управления // Вістн. Націон. Техн. Університету України „КПІ”. Сер. Інформатика, управління і обчислювальна техніка. —2008. №47. —С.62 —68.
9. Сергиенко А.М. Отображение алгоритма QR-разложения Гивенса в многопроцессорную систему // Электрон. моделирование. − 2004. – Т.26. −№5. − С.43−53.
10. Sergiyenko A., Kaniewski J., Arefjev A., Kortshev D. A Method for Mapping DSP Algorithms into Pentium MMX™ Architecture. //Proc 3-d Int. Conf. On Parallel Processing and Applied Mathematics, PPAM'99. −Kazimierz Dolny, Poland. −Sept. 14-17. −1999. −p.348−356.