

СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНО-ОРИЕНТИРОВАННЫХ ПРОЦЕССОРОВ ДЛЯ БОРТОВЫХ КОРРЕЛЯЦИОННО-ЭКСТРЕМАЛЬНЫХ НАВИГАЦИОННЫХ СИСТЕМ

© 2009 Н. А. Лукин

ИМаш УрО РАН, ФГУП “НПО автоматики”, г. Екатеринбург

Приводятся результаты исследования по оптимальной реализации алгоритма привязки, являющегося одним из основных в комплексе алгоритмов корреляционно-экстремальных навигационных систем, использующих датчики геофизических полей. Основным результатом является методика ДН-синтеза алгоритмов, позволяющего синтезировать рациональные архитектуры функционально-ориентированных процессоров (ФОП). Описывается возможная реализация алгоритма привязки на базе двумерного массива процессорных элементов в варианте СБИС-технологии. Излагаются результаты ДН-синтеза в применении к ФОП на базе стандартных микропроцессоров. Приводятся результаты разработки двух поколений реальных бортовых ФОП КЭНС, которые были созданы с использованием предлагаемой методики.

Корреляционно-экстремальные навигационные системы, функционально-ориентированные процессоры, сложность вычислений, параллельная обработка данных.

Введение

Современные системы управления летательными аппаратами (СУ ЛА) должны обеспечивать высокую точность решения основных задач при высокой динамике движения. Средствами только инерциальной навигации этого обеспечить не удаётся, поэтому в состав интегрированных навигационных систем вводятся подсистемы коррекции различного типа, назначение которых состоит в выработке поправок к навигационным параметрам. Среди подобных систем выделяются корреляционно-экстремальные навигационные системы (КЭНС), принцип действия которых основан на использовании датчиков внешних геофизических полей, реализации корреляционных алгоритмов и коррекции траектории движения ЛА. Алгоритмы КЭНС отличаются повышенной вычислительной сложностью и должны реализовываться в течение минимальных временных интервалов. Как правило, для реализации алгоритмов КЭНС используются специализированные бортовые вычислители, производительность которых должна быть максимальной при жёстких ограничениях на массу, габариты и потребляемую мощность. Это требует применения системного подхода к реализации таких вычислителей. Одним из

основных системных принципов их построения является ориентация архитектуры на эффективное выполнение основных функциональных преобразований, содержащихся в алгоритмах КЭНС. Как известно, реализация такого принципа приводит к созданию функционально-ориентированных процессоров.

В настоящей работе кратко излагаются элементы общей теории архитектурного проектирования ФОП и раскрываются основные этапы системного проектирования ФОП для КЭНС ЛА. Кроме того, описываются реальные ФОП КЭНС, при разработке которых были применены предлагаемые методы.

Краткая характеристика основной задачи КЭНС как составной части СУ ЛА

В составе высокоманёвренных ЛА КЭНС предназначены для обеспечения высокой точности решения основной задачи бортовой СУ за счет оперативной коррекции траектории движения аппаратов. Сама коррекция в большинстве случаев основана на решении задачи привязки местоположения аппарата к заданному участку местности и последующего уточнения траектории его движения. В свою очередь привязка формулируется как двухэтапная задача.

На первом этапе определяют корреляционную меру между двумя группами параметров геофизических полей: полученными с помощью датчиков и хранящимися в оперативной памяти бортовой цифровой вычислительной системы (БЦВС). В качестве геофизических полей используются различные характеристики (параметры) пространства, окружающего аппарат: тепловые поля, поля яркостей отраженного сигнала и т.д. [1]. Одними из наиболее распространённых полей являются поля рельефа подстилающей поверхности. Этому есть множество причин, основными из которых являются высокая информативность рельефа и его сезонная устойчивость [2]. Кроме того, информация о наблюдаемом рельефе конкретной местности, как правило, однозначно связана с её географическими координатами, что позволяет повысить эффективность определения поправки к траектории движения аппарата. В качестве рельефометрических датчиков в КЭНС используются одно- и многоканальные радиовысотометры, реализующие измерения высот рельефа в некоторой области пространства под ЛА. Поэтому результатом каждого сеанса измерения является в общем случае одномерный массив измеренных значений высот рельефа. Определение координат ЛА по дальности (X) и по нормали к вектору дальности (Z) производится в инерциальной навигационной системе (ИНС) ЛА с определёнными погрешностями. Поэтому измеренному в k -ом сеансе значению высоты рельефа y_k соответствует окрестность некоторой точки ($X_{ок}, Z_{ок}$). Без существенного ухудшения точности решения задачи привязки предполагают, что данная окрестность представляет собой в общем случае прямоугольник с координатами вершин, равными соответственно ($X_{ок} \pm \delta X, Z_{ок} \pm \delta Z$), где параметры δX и δZ задают размер прямоугольника и зависят от погрешностей ИНС и параметров движения ЛА. Координаты “внутренних” точек прямоугольника в географической системе координат однозначно связаны с координатами измеренных высот подстилающей поверхности. В практике проектирования КЭНС описываемый прямоугольник получил название матрицы гипотез, так как

среди множества элементов матрицы есть лишь один, наиболее достоверный, и нужно каким-либо способом, основанным на вычислении корреляции между y_k и всеми элементами матрицы, найти координаты этого элемента. Для многоканальных радиовысотометров каждому значению y_{ki} соответствует отдельная матрица (μ - число каналов высотометра). Необходимо также учесть, что центр пятна засветки каждого луча высотометра практически всегда находится внутри некоторого квадрата, образованного смежными точками прямоугольника. Это требует уточнения положения y_k с помощью билинейной интерполяции относительно точек указанного квадрата.

Размерность матрицы зависит от многих факторов (ошибок ИНС, требуемой точности определения местоположения ЛА, динамики движения ЛА и т.д.). Для высокоманевренных ЛА размерность матрицы может достигать порядка десятков или сотен, что требует на каждом сеансе измерений порядка сотен или тысяч сравнений для каждого канала высотометра.

Описанная процедура носит циклический характер, она повторяется в каждом цикле приема сигналов с высотометра.

На втором этапе решения задачи привязки реализуется нахождение экстремума корреляционной меры как функционала на множестве гипотез для каждой матрицы. Комбинаторная сложность алгоритма (в худшем случае в процессе сравнения необходимо перебрать все элементы матрицы гипотез) в задаче привязки увеличивается из-за того, что в случае многих одинаковых по величине экстремумов требуется определять расстояние каждого экстремума от центра гипотезной матрицы с учетом направления вектора продольной скорости движения ЛА. Определение экстремума корреляционной меры производится после заключительного сеанса визирования, данный алгоритм является разовым.

Координаты X^* и Z^* найденного экстремума корреляционной меры принимаются за наиболее достоверные, после чего выдаётся соответствующее управление на исполнительные органы ЛА, совершается его ма-

невр, и реализуется следующий сеанс измерения. Обобщённо принцип работы СУ ЛА со встроенной КЭНС изображен на рис. 1.

Совместный анализ основных задач, решаемых на борту ЛА (инерциальная навигация, стабилизация, управление, контроль состояния подсистем), и задачи привязки местоположения ЛА к заданному участку местности показывает, что она имеет следующие особенности:

- задача привязки должна решаться одновременно с остальными задачами СУ ЛА;
- решение задачи привязки в течение интервала коррекции траектории имеет циклический характер;
- жёсткое ограничение на время решения задачи (как правило, порядка единиц миллисекунд);
- малое количество входных и выходных параметров (для современных рельефометрических КЭНС – порядка десятка), что обуславливает сравнительно малую связность задачи привязки с остальными задачами СУ ЛА;
- преимущественная работа с большими массивами переменных (число элементов эталонной карты местности, обрабатываемых в одном цикле решения задачи, может быть порядка тысяч), т.е. значительное количество промежуточных (внутренних) параметров.

Таким образом, с точки зрения построения всей системы алгоритмического обеспечения СУ задача привязки может быть характеризована следующими отличительны-

ми признаками – существенная автономность, ориентация математического аппарата на векторно-матричную обработку, минимальное время решения. Системный анализ задачи привязки с точки зрения её как объекта алгоритмической реализации является первым этапом системного проектирования ФОР КЭНС.

Отмеченные отличительные признаки задачи привязки непосредственно связаны с информативными характеристиками возможных алгоритмов её решения. Анализ характеристик алгоритмов важен для направленного выбора архитектур вычислительных систем реального времени.

Проектирование архитектур ФОР на основе методов ДН-синтеза алгоритмов

Анализ вычислительных алгоритмов для систем реального времени показывает, что графы большинства из них обладают сходными признаками:

- 1) малое число входных/выходных и значительное число промежуточных вершин;
- 2) сильная связность по одному из направляющих векторов графа [3] и слабая по другим;
- 3) циклы в алгоритмах обусловлены в большинстве случаев не столько анализом результатов выполнения арифметико-логических операций, сколько итеративным выполнением последовательности преобразований, что связано либо с общим принципом функционирования систем реального времени (циклической работой алгоритмов),

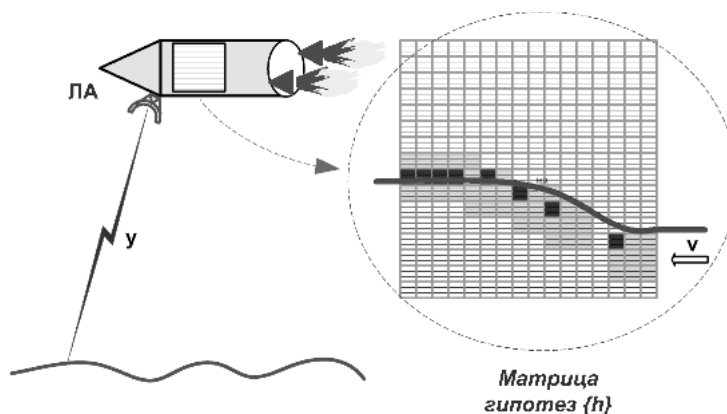


Рис. 1. Принцип коррекции траектории ЛА при работе КЭНС

либо с анализом внешнего признака (например, с содержимым счетчика числа событий).

Это даёт основание отнести данные алгоритмы к классу потоковых и применить к ним методики направленного преобразования графов, основанные на выявлении внутреннего параллелизма (параллелизма смежных операций, параллелизма ветвей и т.д.). Одной из таких методик является *DH*-синтез потоковых алгоритмов [4], который целесообразно считать *вторым этапом* системного проектирования ФОП КЭНС.

Основными шагами *DH*-синтеза произвольных потоковых алгоритмов являются:

1. Преобразование исходного графа алгоритма к виду максимальной канонической ярусно-параллельной формы (ЯПФ). На этом шаге целесообразно применять процедуры эквивалентного преобразования исходных графов, например, описанных в [4]. Они позволяют максимально вскрыть внутренний параллелизм алгоритма и минимизировать число перекрёстных связей между вершинами графа в направлении, ортогональном главному направляющему вектору графа.

2. Построение верхних оценок ширины (D) и высоты (H) максимальной ЯПФ графа как функций от параметров $\{\alpha_k\}_{k=0,1,2,\dots}$ исходного алгоритма. В качестве параметров могут выступать число функциональных преобразований в максимальном ярусе, число ярусов, разрядность переменных, ёмкость памяти для хранения констант и коэффициентов и т.п. Функции $D(\alpha)$ и $H(\alpha)$ всегда имеют экстремумы (за исключением случаев, когда D и H – константы), и на этом этапе синтеза производится поиск локальной окрестности минимумов $D(\alpha)$ и $H(\alpha)$. Минимум $D(\alpha)$ однозначно связан с конкретным графом алгоритма и соответствует минимуму аппаратных затрат на вычисления, а минимум $H(\alpha)$ – минимуму временных затрат. При этом могут существовать варианты:

а. Функции $D(\alpha)$ и $H(\alpha)$ не монотонны. Это означает, что существуют алгоритмы, оптимальные по критерию сложности вычислений, такие что $\exists \alpha^*$; $D(\alpha^*) = \text{Min}\{D(\alpha)\}$; $\forall \alpha_* \neq \alpha^* \Rightarrow D(\alpha_*) \geq D(\alpha^*)$ либо $\exists \alpha^{**}$; $H(\alpha^{**}) = \text{Min}\{H(\alpha)\}$; $\forall \alpha_{**} \neq \alpha^{**} \Rightarrow H(\alpha_{**}) \geq H(\alpha^{**})$. Данный вариант означает существо-

вание точки оптимума внутри интервала изменения ширины и высоты графа алгоритма. Всякая вариация параметров $\{\alpha_k\}$ относительно α^* (α^{**}) приводит только к увеличению затрат. Поэтому для данного варианта характерно наличие минимума одних затрат при немаксимальном значении других. Например, возможно достичь минимума времени вычисления при немаксимальном числе процессоров в системе.

б. Функции $D(\alpha)$ и $H(\alpha)$ монотонны. Это означает, что $\exists \alpha^*$; $D(\alpha^*) = \text{Min}\{D(\alpha)\}$; $\forall \alpha_* < \alpha^* \Rightarrow D(\alpha_*) \geq D(\alpha^*) \oplus \forall \alpha_* > \alpha^* \Rightarrow D(\alpha_*) \geq D(\alpha^*)$ либо $\exists \alpha^{**}$; $H(\alpha^{**}) = \text{Min}\{H(\alpha)\}$; $\forall \alpha_{**} < \alpha^{**} \Rightarrow H(\alpha_{**}) \geq H(\alpha^{**}) \oplus \forall \alpha_{**} > \alpha^{**} \Rightarrow H(\alpha_{**}) \geq H(\alpha^{**})$. Точка оптимума находится на границе интервала изменения ширины и высоты графа. Поэтому минимум одних затрат соответствует максимуму других. Например, минимальное время вычисления возможно только при максимуме числа процессоров.

Данный шаг синтеза алгоритма заканчивается построением графа, соответствующего минимуму $D(\alpha)$ либо минимуму $H(\alpha)$ в зависимости от исходной постановки задачи проектирования ФОП. Назовём такой граф D_{\min} -графом либо H_{\min} -графом.

3. Взвешивание вершин D_{\min} -графа (либо H_{\min} -графа) верхними оценками аппаратной и временной сложности функциональных преобразований, приписанных данным вершинам. При этом сами оценки сложности строятся в базисе схем из функциональных элементов [5]. Эти оценки служат исходным материалом для построения также верхних оценок, но уже для аппаратной L_h или временной L_t сложности вычислений алгоритма в целом. Оценки L_h (L_t), однозначно связанные с $D(\alpha)$ и $H(\alpha)$, представляют собой функции от параметров алгоритмов, но они соответствуют определённому типу архитектуры ФОП – базовой архитектуре [3]. Как известно, именно базовая архитектура, обеспечивая минимум времени реализации любого алгоритма, служит отправным пунктом для процессов последующей их оптимизации и синтеза.

4. Направленное изменение ширины (D) и высоты (H) ЯПФ D_{\min} -графов либо H_{\min} -графов с целью достижения одного из двух

условий: $Min L_i$ при $L_h = const$ или $Min L_h$ при $L_i = const$. Направленность изменения данных графов обеспечивается последовательными преобразованиями типа $(L_h - \Delta L_h)$ или $(L_i - \Delta L_i)$ с одновременным получением оценок $D(\alpha)$ либо $H(\alpha)$. Процесс изменения графа алгоритма завершается получением окончательных оценок $L_h, L_i, D(\alpha), H(\alpha)$, которые в совокупности соответствуют определенной архитектуре.

Таким образом, последовательность шагов DH -синтеза приводит к построению архитектуры ФОР, оптимизированной по критерию сложности вычислений. Сама архитектура ориентирована на рациональную (в ряде случаев оптимальную) реализацию выбранных алгоритмов. Если в качестве исходного алгоритма взят некоторый обобщенный алгоритм для класса задач, то архитектура ориентирована уже на эффективное решение задач из этого класса.

тура ориентирована уже на эффективное решение задач из этого класса.

Применим методикку DH -синтеза алгоритмов к задаче привязки.

Системное проектирование архитектуры ФОР КЭНС

На рис. 2 приведён обобщенный граф алгоритма задачи привязки для рельефометрической КЭНС.

Обозначения на рис. 2: i_{max}, j_{max} – размерности матрицы гипотез; μ_{max} – число каналов радиовысотомера; k – количество сеансов связи с высотомером (количество циклов решения задачи привязки); $\Delta x, \Delta z$ – величины дискретов перебора гипотез в матрице.

Анализ возможности применения DH -синтеза к задаче привязки показал, что её алгоритм является потоковым, а граф при-

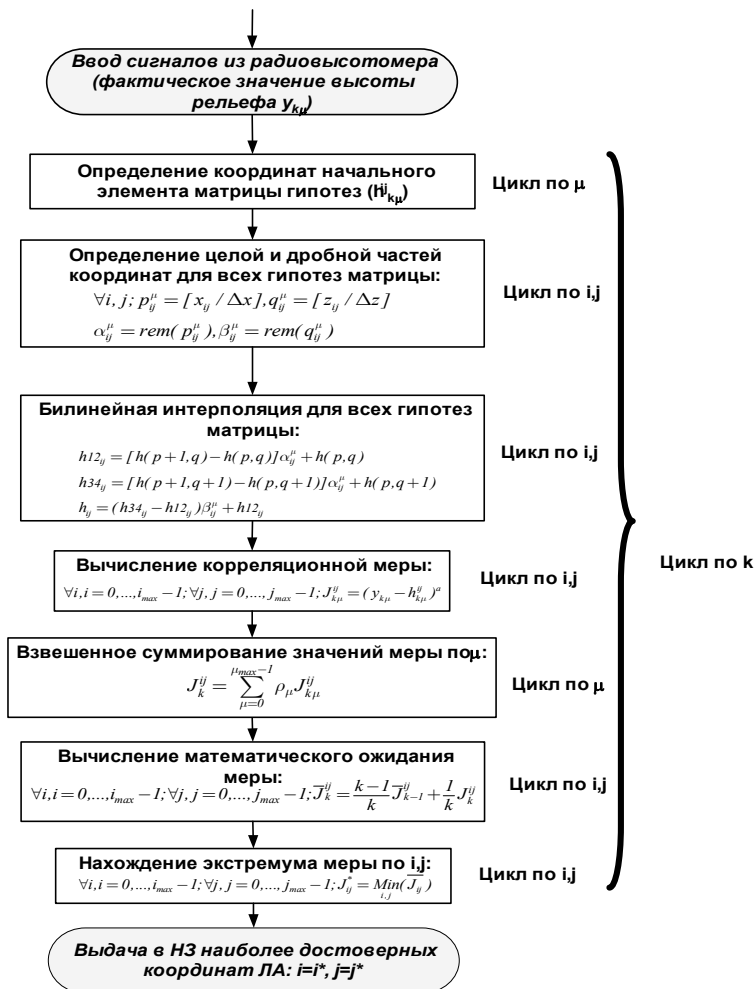


Рис. 2. Обобщенный алгоритм решения задачи привязки для рельефометрической КЭНС

надлежит к классу гамаков. Это означает, что алгоритм обладает, прежде всего, значительным внутренним параллелизмом. Например, для самого насыщенного этапа вычислений – вычисления корреляционной меры $J_{k\mu}^{ij}$ – можно выполнять одновременно $\mu \cdot i \cdot j$ независимых вычислений.

Построение максимальной ЯПФ графа начинается с анализа основных математических соотношений и их упрощения и преобразования в параллельную форму. Это относится, прежде всего, к билинейной интерполяции и вычислению математического ожидания корреляционной меры.

Для параллельной реализации интерполяции применим следующее преобразование. Исходные зависимости имеют вид:

$$\begin{aligned} h_{12_{ij}} &= [h(p+1, q) - h(p, q)]\alpha_{ij}^\mu + h(p, q), \\ h_{34_{ij}} &= [h(p+1, q+1) - h(p, q+1)]\alpha_{ij}^\mu + \\ &\quad + h(p, q+1), \\ h_{ij} &= (h_{34_{ij}} - h_{12_{ij}})\beta_{ij}^\mu + h_{12_{ij}}, \end{aligned} \quad (1)$$

где p, q представляют целые частные от деления $\frac{x}{\Delta x}$ и $\frac{y}{\Delta y}$, а α и β – соответствующие остатки. Граф для (1) приведён на рис. 3а. Высота графа равна 6, ширина – 2.

Следуя методике *DH*-синтеза, преобразуем (1) к максимально параллельной форме:

$$h_{ij} = h(p+1, q)\gamma_1 + h(p, q)\gamma_2 + h(p+1, q+1)\gamma_3 + h(p, q+1)\gamma_4, \quad (2)$$

$$\begin{aligned} \text{где } \gamma_1 &= \alpha_{ij}^\mu \beta_{ij}^\mu, \gamma_2 = (1 - \alpha_{ij}^\mu) \beta_{ij}^\mu, \\ \gamma_3 &= \alpha_{ij}^\mu (1 - \beta_{ij}^\mu), \gamma_4 = (1 - \alpha_{ij}^\mu) (1 - \beta_{ij}^\mu). \end{aligned}$$

Как видно, для обеспечения максимального возможного параллелизма необходимо предварительно один раз вычислить четыре алгебраические формы от остатков α_{ij}^μ и β_{ij}^μ и применять их для интерполяции всех гипотез матрицы. Граф, соответствующий (2), приведен на рис. 3б, из которого видно, что высота графа уменьшилась в два раза и равна 6, а ширина увеличилась и стала равна 4.

Для вычисления математического ожидания корреляционной меры применим следующие преобразования:

$$\begin{aligned} \bar{J}_k^{ij} &= \frac{k-1}{k} \bar{J}_{(k-1)}^{ij} + \frac{1}{k} J_k^{ij} = \\ &= \frac{1}{k} \sum_{p=1}^{k-1} J_p^{ij} + \frac{1}{k} J_k^{ij}. \end{aligned}$$

$$\text{Тогда } k \bar{J}_k^{ij} = \overline{k J_k^{ij}} = \sum_{p=1}^{k-1} J_p^{ij} + J_k^{ij}. \quad (3)$$

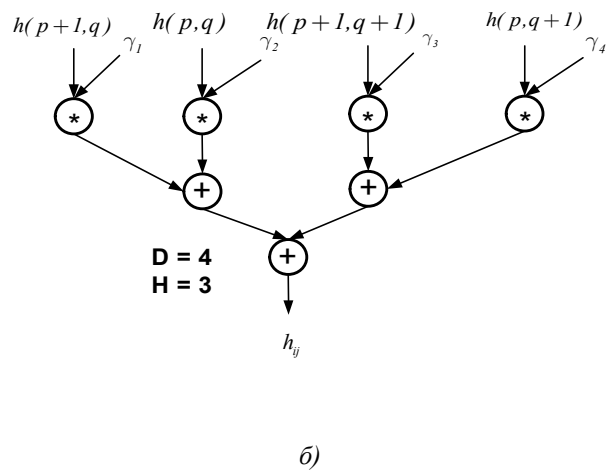
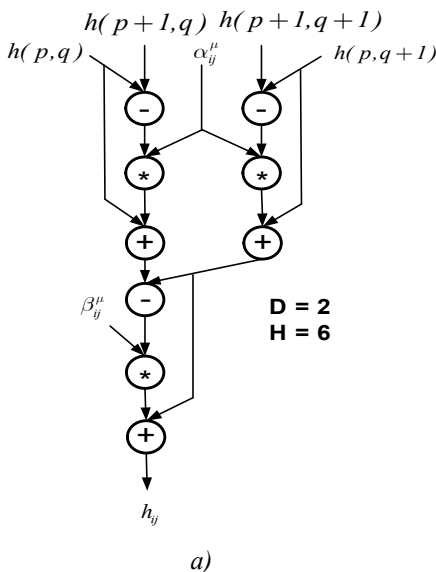


Рис. 3. Графы алгоритма билинейной интерполяции:
а – исходный алгоритм, б – максимально параллельный алгоритм

В этом случае нахождение экстремума меры на заключительном шаге алгоритма будет производиться по формуле

$$\forall i, i = 0, \dots, i_{max} - 1; \forall j, j = 0, \dots, j_{max} - 1; kJ_{ij}^* = \text{Min}(\overline{kJ_{ij}}). \quad (4)$$

Таким образом, уменьшена сложность вычислений для каждого элемента матрицы гипотез.

Следуя методике *DH*-синтеза, преобразуем исходный граф в максимальную каноническую ЯПФ. Этот граф изображен на рис. 4.

Анализ информационной структуры ЯПФ алгоритма задачи привязки показывает следующее:

- Вид графа – струнный гамак [4].
- Количество уровней параллелизма – два. На первом уровне имеется μ независимых потоков обрабатываемых данных, внутри каждого из которых также имеется $i \cdot j$ независимых потоков данных. На втором уровне обрабатывается $i \cdot j$ потоков данных.
- Количество ярусов – 7.

Определим ширину и высоту максимальной ЯПФ:

$$D_{max} = i_{max} \cdot j_{max} \cdot \mu_{max}, H_{min} = 7.$$

Высота максимальной ЯПФ есть константа, что отражает факт максимального распараллеливания вычислений.

После построения максимальной ЯПФ необходимо произвести взвешивание вершин графа верхними оценками сложности в базе схем из функциональных элементов. Такими элементами будем считать простейшие одноразрядные сумматоры и логические элементы, реализующие произвольные булевы функции двух переменных [4]. Характеристики используемых элементов базиса приведены в табл. 1.

Основными преобразованиями в алгоритме привязки являются μ -разрядное суммирование и вычитание, возведение в квадрат μ -разрядных чисел и поиск минимума в массиве μ -разрядных чисел. Верхние оценки сложности реализации указанных преобразований в базе функциональных элементов приведены в табл. 2.

Теперь найдём ширину и высоту графа как функции параметров алгоритма привязки с учётом функциональных преобразований предполагаемого базиса реализации. Тогда ширина и высота графа определяются следующим образом:

$$D_{basis} = i_{max} \cdot j_{max} \cdot \mu_{max}, H_{basis} = 9 + \lceil \log_2 D \rceil. \quad (5)$$

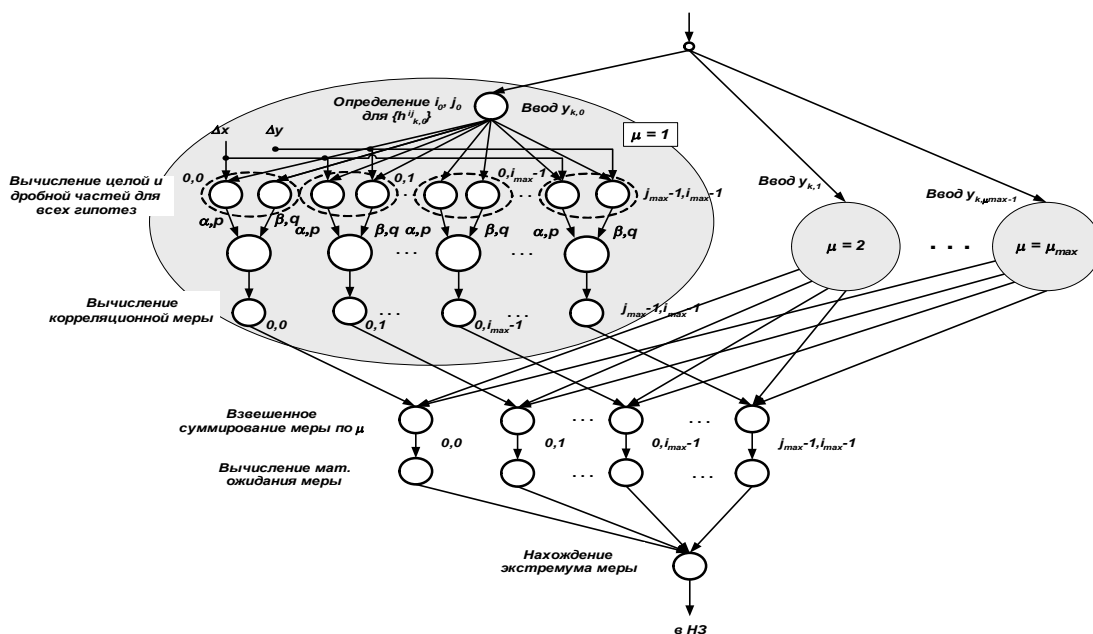
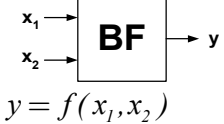


Рис. 4. Максимальная ЯПФ графа задачи привязки

Таблица 1. Базис функциональных элементов и их характеристики

Наименование элемента	Реализуемые функции	L_t , тактов	L_h , вентиляей	Примечание
Одноразрядный сумматор	$c_{i+1} = a_i b_i \vee c_i (a_i \vee b_i)$ $S_i = a_i \oplus b_i \oplus c_i$	$L_t^{S_i} = 2\tau$ $L_t^{c_i} = 3\tau$	$L_h^\Sigma = 6$	
BF(2,1)	 $y = f(x_1, x_2)$	$L_t^f = \tau$	$L_h^f = 1$	Элемент реализует любую БФ двух переменных

Видно, что алгоритм привязки обладает логарифмической высотой. Это минимальная высота для алгоритмов подобного типа, она обеспечивается за счет максимальной ширины. Оценки высоты вида $O(\log_2 N)$, где N – число входных переменных алгоритма, как известно, уменьшить не удаётся. Поэтому высота алгоритма привязки, определённая по формуле (3), является минимальной в классе базиса реализации. Таким образом, если удаётся одновременно выполнять D двухместных операций, то реализация всего алгоритма привязки потребует не более $O(\log_2 D)$ тактов. Например, для размерности матрицы гипотез $i_{max} = j_{max} = 13$ и числа каналов высотомера $\mu_{max} = 5$ можно решить задачу привязки (для одного цикла) всего за 18 тактов работы микропроцессора (МП), но

для этого потребуется около тысячи таких МП.

Оценки D и H в виде (3) представляют собой монотонные функции от i, j и μ , что делает возможным вариант оптимизации затрат согласно п. 2b. Это означает, что возможно сокращать D (аппаратные затраты) простым уменьшением i, j или μ . Это приводит к пропорциональному увеличению H , при этом минимум D (один процессор или процессорный элемент) достигается при максимуме H . Таким образом, предварительный анализ характеристик графа алгоритма привязки показывает, что оптимизация вычислений носит достаточно простой характер.

На основе табл. 2 получим верхние оценки сложности вычислений для алгоритма привязки. Верхние оценки аппаратной сложности вычислений для циклической и разовых частей алгоритма привязки имеют вид:

Таблица 2. Верхние оценки сложности вычислений в базисе схем из функциональных элементов для основных преобразований алгоритма привязки

Преобразование	Формула	Сложность	Примечание
r -разрядное сложение (вычитание) (SM)	$C = A \pm B$	$L_t^{SM} = 3r\tau$ $L_h^{SM} = 6r$	Суммирование с последовательным переносом
Возведение в квадрат r -разрядного числа (SQ)	$C = A^2$	$L_t^{SQ} = 3(2r - 1)\tau$ $L_h^{SQ} = (7r - 6)r$	Аппаратный умножитель матричного типа
Минимум в массиве чисел (MIN)	$C = \text{Min}_{i=1, \dots, N} \{ A_i \}$	$L_t^{MIN} = (3r + 2) \lceil \log N \rceil \tau$ $L_h^{MIN} = 9(N - 1)r$	Древовидный параллельный поиск на основе схем сравнения двух чисел

$$\begin{aligned}
L_h^{(1)} &< 60r^2 i_{\max} j_{\max} \mu_{\max}, \\
L_t^{(1)} &< 4r(14 + \lceil \log_2 \mu_{\max} \rceil), \\
L_h^{(2)} &< 10r i_{\max} j_{\max}, \\
L_t^{(2)} &< 4r \lceil \log_2(i_{\max} j_{\max}) \rceil.
\end{aligned} \tag{6}$$

Анализ соотношений “циклической” и “разовой” составляющих сложности вычислений показывает, что

$$\begin{aligned}
\delta L_h &= L_h^{(1)} / L_h^{(2)} \cong 6r \mu_{\max}, \\
\delta L_t &= L_t^{(1)} / L_t^{(2)} \cong \frac{14 + \lceil \log_2 \mu_{\max} \rceil}{\lceil \log_2(i_{\max} j_{\max}) \rceil}.
\end{aligned}$$

Кроме того,

$$\delta L_t \geq 1 \text{ при } \max\{i_{\max}, j_{\max}\} \geq 287.$$

Видно, что циклическая часть алгоритма привязки полностью определяет архитектуру аппаратной части ФОР КЭНС, причём для размерностей матрицы гипотез, не превышающей 287×287 элементов, эта часть алгоритма будет определять ещё и принципы распараллеливания вычислений. В случае больших матриц ($>287 \times 287$) при распараллеливании необходимо учитывать свойства ЯПФ графа “разовой” части алгоритма.

Данные оценки соответствуют максимальной ЯПФ графа, т.е. максимальному параллелизму вычислений. Этому варианту, в свою очередь, соответствует аппаратная реализация ФОР КЭНС. Его архитектура представляет собой двумерный массив функциональных элементов, реализующих систолический принцип обработки данных. Данный ФОР обладает максимальной производительностью для выбранного алгоритма привязки. Вид оценок подтверждает монотонность аппаратной и временной сложности как функций от i, j, μ . Представляют интерес асимптотические оценки сложности вычислений для этого варианта: $L_h = O(r^2 I)$, где

$I = i_{\max} j_{\max} \mu_{\max}$, $L_t = O(r \log I)$. Видно, что с увеличением размерности матрицы гипотез аппаратная сложность вычислений для максимально параллельной реализации алгоритма привязки растёт значительно быстрее, чем

временная. В случае больших размерностей матрицы это может существенно ограничить применение прямого распараллеливания вычислений для алгоритма КЭНС. В силу монотонности функций сложности $L_h(i, j, \mu)$ и $L_t(i, j, \mu)$ целесообразно данный вариант реализации считать исходным для последующей оптимизации. Это и есть базовая архитектура ФОР в смысле [3].

Следующим шагом ДН-синтеза алгоритмов является направленное изменение его графа (п.4). Этот шаг необходим тогда, когда сложность реализации максимально параллельного варианта алгоритма привязки превосходит допустимые ограничения. Это может быть в случае использования менее совершенной технологии изготовления СБИС, для варианта реализации алгоритма на уровне стандартных МП и т.д. Для алгоритма привязки направленное изменение графа необходимо начинать с максимальной ЯПФ (п.4). Каждый шаг уменьшения D сопровождается некоторым пропорциональным увеличением H , причём для различных ярусов графа это увеличение будет различным. Существует множество стратегий направленного изменения графа, которые приводят к одному и тому же значению параметра H . Поэтому на каждом шаге изменения графа производится оценка параметров L_t и L_h , что существенно уменьшает перебор вариантов. Математически задача направленного изменения графа алгоритма может быть сформулирована следующим образом:

$$\begin{aligned}
\forall n \in N; N \in \{ \hat{N} \}; D_n &= \\
= \text{Min}_{l=1, \dots, N} (D_l - d_l); H_n &= \\
= H + \text{Min}_l \left(\sum_{l=1}^N k_l d_l \right);
\end{aligned} \tag{7}$$

$$L_h = \text{Min}_M \{ L_{hi} \}, L_t = \text{const}, \tag{8}$$

где l – номер яруса графа, N – количество ярусов графа, k_l – коэффициент пропорционального увеличения высоты графа в l -ом ярусе при уменьшении ширины на d_l вершин, M – число вариантов ЯПФ.

В общем случае $\forall v_p, v_q; p \neq q; L_h^p \neq L_h^q$, что обуславливает переборный характер при построении стратегии направленного изменения D и H , которая одновременно обеспечивает достижение всех условий, сформулированных в (7), (8). С другой стороны, монотонность $D(\alpha)$ и $H(\alpha)$, характерная для рассматриваемого алгоритма, обеспечивает сходимость процесса оптимизации к условию (8). По отношению к ФОП КЭНС это означает существование однозначного решения задачи построения оптимального графа алгоритма привязки и, соответственно, оптимальной архитектуры ФОП.

Описанная процедура DH -синтеза алгоритма позволяет однозначно синтезировать архитектуру параллельного ФОП КЭНС в базисе элементарных арифметико-логических преобразований. Эта архитектура представляет собой двумерный массив функциональных элементов, ориентирована на проектирование в технологии “система-на-кристалле” и на реализацию в виде одного кристалла СБИС. При этом в рамках ограничений на реализацию, обусловленных степенью интеграции СБИС, предельным тепловыделением, числом внешних контактов и т.д., DH -синтез позволяет получить архитектуры ФОП с максимальной производительностью.

Использование DH -синтеза алгоритмов на основе анализа и направленного изменения его графа, вершины которого взвешены оценками временной и аппаратной сложности вычислений в базисе схем из функциональных элементов, даёт возможность реализовать процессорные архитектуры, оптимальные по критериям аппаратной либо временной сложности вычислений. Данный подход может служить основой системного проектирования ФОП КЭНС.

Подход к проектированию ФОП основан на взаимосвязи характеристик алгоритмов и параметров процессорных архитектур, и поэтому он может быть использован на любом уровне их реализации – от СБИС до стандартных МП. Выше были рассмотрены проблемы системного проектирования ФОП КЭНС для варианта СБИС-реализации, актуального для бортовых систем ближайше-

го будущего. В то же время практический интерес представляет реализация алгоритма КЭНС с помощью ФОП, который построен на базе стандартных МП, т.к. это наиболее распространённый случай в практике разработки современных КЭНС. Рассмотрим применение метода DH -синтеза алгоритмов к проектированию конкретного ФОП КЭНС на базе стандартных МП.

Системное проектирование ФОП КЭНС на базе стандартных МП

Применение универсальных программируемых микропроцессоров для разработки бортовых вычислительных средств вносит специфику в распараллеливание алгоритмов. Прежде всего, это касается уровня параллелизма обработки данных. По сравнению с вариантом СБИС он существенно повышается, а это отражается на процессах направленного изменения ЯПФ, т.е. собственно на DH -синтезе. Так как любой стандартный МП представляет собой сложную микроэлектронную систему, то реализация мелкозернистого параллелизма, когда функциональное преобразование каждой вершины графа реализуется на отдельном МП, приводит к значительным величинам L_h и L_f .

Рассмотрим пример. Пусть имеются две смежных вершины ЯПФ графа v_i и v_{i+1} , что соответствует последовательному выполнению преобразований вида

$$v_i = f_i(a, b); v_{i+1} = f_{i+1}(v_i, b).$$

Реализуем эти преобразования на отдельном МП. Интерфейсы большинства стандартных МП могут обеспечить лишь синхронную передачу данных. При этом частота обмена информацией по внешним выводам СБИС всегда существенно ниже, чем по внутренним межсоединениям. Поэтому последовательное задействование двух МП будет занимать значительно больше времени, чем последовательное выполнение двух микрокоманд в одном МП.

Рассмотрение различных подходов к реализации ЯПФ на основе стандартных МП с учётом особенностей обмена по их внешним интерфейсам показывает, что одним из главных принципов “покрытия” графов алгоритмов архитектурами МП является ре-

ализация всех вершин ЯПФ, расположенных на одной дуге, в одном МП. Направленное изменение графа в таком случае должно проводиться следующим образом:

1. Уменьшение высоты максимальной ЯПФ за счет объединения всех вершин, соединённых “вертикальными” дугами. Такое уменьшение H в отличие от варианта СБИС-реализации не приводит к увеличению ширины D .

2. Уменьшение ширины максимальной ЯПФ D за счёт объединения дуг с пропорциональным увеличением высоты графа H .

Математическая формулировка задачи направленного изменения ЯПФ для рассматриваемого варианта реализации соответствует (7), (8). При этом параметр D однозначно связан с числом МП (аппаратурные затраты или L_n), а H – с числом тактов микропрограммы в МП (временные затраты или L_t). Наличие ограничений на число МП или время реализации алгоритма так же, как и в случае с СБИС, обеспечивает однозначное определение минимума числа МП при заданном времени реализации алгоритма привязки либо минимума этого времени при заданной величине аппаратурных затрат. В то же время учет топологии архитектур параллельных ФОП как МП-систем может приводить к появлению локальных экстремумов в зависимостях $D(\alpha)$, $H(\alpha)$. В результате, архитектуры ФОП, оптимальные по критерию сложности вычислений для варианта СБИС-реализации, могут быть не эффективными для варианта МП, и наоборот. Таким образом, если системное проектирование архитектур ФОП КЭНС на базе СБИС требует учёта свойств графа алгоритма (связность вершин, ярусность и т.п.), то системное проектирование этих ФОП на базе стандартных МП требует учёта топологии соединений между ними (магистраль, гнездо, кольцо и т.д.). Вот почему оценка временной сложности вычислений для любого алгоритма вообще и алгоритма привязки, в частности, в случае применения МП должна включать в качестве составляющей совокупное время обменов данными, необходимыми для реализации алгоритма [3]. Кроме того, эта оценка должна включать время ввода входных

данных и вывода выходных. В случае необходимости уменьшения совокупной временной сложности вычислений время обменов требуется предельно минимизировать, для чего всю ЯПФ графа целесообразно реализовать в отдельном модуле БЦВС. Этот вариант далее будем рассматривать в качестве основного.

Итак, базовыми положениями для исследования принципов реализации алгоритмов привязки с помощью ФОП на базе стандартных МП являются следующие:

- архитектура ФОП состоит из одинаковых стандартных МП, содержащих локальную память;

- каждый МП “покрывает” целиком дугу ЯПФ (т.е. реализует все преобразования в этой дуге);

- внешний интерфейс каждого МП содержит группу входных и выходных шин, обмены между различными МП возможны только по этим шинам;

- длительность выполнения арифметико-логических операций в каждом МП равны: $L_t^\pm = 1, L_t^x = 1, L_t' = L_t^{comp2,1} = 2$. Длительности приведены в числе тактов.

Это простейший тип архитектуры, он соответствует варианту реализации ФОП на базе простых контроллеров. Такой ФОП отличается однородность архитектуры, что позволяет увеличивать производительность ФОП путём простого наращивания числа МП, и сравнительно малая гибкость интерфейсов.

Взяв за исходное представление алгоритма привязки в виде максимальной ЯПФ, применим пошаговый DH -синтез с учётом особенностей направленного изменения. В результате получается значительное число вариантов алгоритма, из которых интерес представляют следующие.

А. Максимально параллельная реализация алгоритма привязки. Граф алгоритма представлен в виде максимальной ЯПФ (рис. 4). Каждая дуга ЯПФ реализуется на отдельном МП. Обобщённый вид архитектуры ФОП представлен на рис. 5.

Данная архитектура состоит из μ_{max} кластеров, каждый из которых состоит из

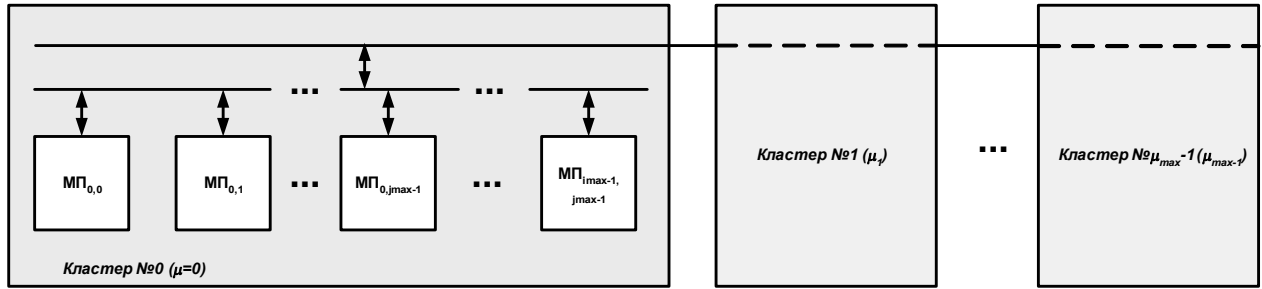


Рис. 5. Архитектура ФОП КЭНС на базе стандартных МП с максимальным уровнем параллелизма обработки данных

$i_{max} j_{max}$ МП, параллельно обрабатывающих данные для одного соответствующего канала высотомера. Видно, что число дуг ЯПФ соответствует числу МП. Получим верхние оценки L_t^A и L_h^A , причём для L_t произведём нормирование на величину τ , что даст возможность измерять временную сложность в числе тактов.

Верхняя оценка аппаратной сложности вычислений L_h получается непосредственно из вида ЯПФ:

$$L_h = i_{max} j_{max} \mu_{max} \cdot \quad (9)$$

Верхняя оценка временной сложности вычислений L_t имеет вид для циклической части алгоритма привязки:

$$L_t^{A(1)} = L_t^{i_0, j_0} + 2L_t^{p, q} + L_t^h + L_t^{(y-h)} + L_t^{(y-h)^2} + L_t^{j_{ij}^k} + L_t^{dJ}, \quad (10)$$

где $L_t^{i_0, j_0}$ - время выполнения процедуры вида $ax+b$, с помощью которой определяется адрес крайнего элемента матрицы гипотез; $L_t^{i_0, j_0} = L_t^x = 1$; $2L_t^{p, q}$ - время выполнения операций деления $x / \Delta x$ и $z / \Delta z$, L_t^h - время выполнения билинейной интерполяции для одной гипотезы; $L_t^{y-h} + L_t^{(y-h)^2}$ - время вычисления корреляционной меры, оно равно двум тактам; $L_t^{j_{ij}^k}$ - время выполнения суммирова-

ния значений меры, полученной по всем каналам высотомера; L_t^{dJ} - время вычисления текущего значения $k \bar{J}_{k\mu}^{ij}$ (3). С учётом конкретных значений верхних оценок получим:

$L_t^{A(1)} \cong \mu_{max} + 12$. При получении оценки (10) не учитывается время обращения к запоминающему устройству, где хранится эталонная карта местности, так как эта процедура является общей для всех рассматриваемых вариантов архитектуры ФОП и общее время реализации всех вариантов алгоритмов увеличивается на постоянную величину.

Как видно, для максимально параллельной реализации алгоритма привязки размерность матрицы гипотез не определяет время вычислений, и оно зависит только от числа каналов высотомера. В то же время аппаратные затраты на реализацию максимально параллельного алгоритма весьма велики. Например, для $i_{max} = j_{max} = 13, \mu_{max} = 5$ может потребоваться 845 МП, что неприемлемо для любых бортовых КЭНС. Поэтому целесообразно рассмотреть другой вариант алгоритма привязки.

В. Параллельная обработка данных по каналам высотомера и последовательная по элементам матрицы гипотез. Граф алгоритма представляет собой результат ДН-синтеза ЯПФ в части максимального уменьшения ширины графа для каждого из μ_{max} каналов. Это приводит к соответствующему увеличению высоты H . Взвешивая вершины графа верхними оценками аппаратной и временной

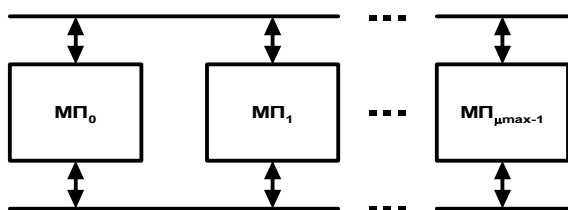


Рис. 6. Архитектура ФОП на базе стандартных МП с параллельной обработкой данных каналов высотомера

сложности отдельных функциональных преобразований, получим соответствующие оценки алгоритма в целом. Они приведены в табл. 3. Данный вариант соответствует архитектуре ФОП, в которой каждый МП реализует алгоритм привязки для “своего” канала высотомера. Можно видеть, что уменьшение числа МП в $i_{max}j_{max}$ раз приводит к пропорциональному увеличению времени решения задачи КЭНС. Для современных КЭНС это даёт уменьшение аппаратных затрат более чем в 150 раз. Поэтому данный вариант является вполне практичным. Архитектура ФОП варианта В приведена на рис. 6.

С. Параллельная обработка данных по всем элементам матрицы гипотез и последовательная по каналам высотомера. Для построения графа алгоритма, соответствующего данному варианту, применим *DH*-синтез, в результате которого получим максимальную ЯПФ для каждого канала и последовательное соединение всех $\mu_{max}-1$ каналов. Взвесим вершины графа верхними оценка-

ми временной и аппаратной сложности. Суммарные оценки приведены в табл. 3. Архитектура ФОП, реализующего данный алгоритм привязки, аналогична приведенной на рис. 6 с той разницей, что в данном случае число МП равно $i_{max}j_{max}$.

Сравнение данного варианта с максимально параллельным по критерию эффективности показывает, что уменьшение числа МП в μ_{max} раз не приводит к пропорциональному увеличению времени реализации алгоритма. Это говорит о большей эффективности описываемого варианта реализации по сравнению с максимально параллельным алгоритмом. Анализ процессов направленного изменения графа алгоритма в ходе *DH*-синтеза приводит к выводу о существовании баланса временных затрат, при котором затраты на вычисления равны затратам на обмен. Подробный анализ структуры временной сложности вычислений выходит за рамки данной статьи, и поэтому ограничимся рассмотрением одного из вариантов параллельной реализации алгоритма привязки, для которого достигим этот баланс.

Д. Параллельная обработка данных по одной строке матрицы гипотез и последовательная по каналам высотомера. Граф данного варианта параллельного алгоритма привязки представляет собой результат промежуточного этапа *DH*-синтеза максимальной ЯПФ. Параллельно обрабатываются все элементы одной строки матрицы гипотез, а затем осуществляется переход к следующей строке. После обработки всех строк матрицы осуществляется переход к обработке элементов матрицы гипотез для следующего

Таблица 3. Верхние оценки аппаратной и временной сложности вычислений для различных модификаций алгоритма привязки при их реализации на стандартных МП

Вариант алгоритма	L_h	L_t
<i>A</i>	$i_{max}j_{max}\mu_{max}$	$\cong \mu_{max} + 12$
<i>B</i>	μ_{max}	$\cong i_{max}j_{max}(\mu_{max} + 13)$
<i>C</i>	$i_{max}j_{max}$	$\cong 16\mu_{max}$
<i>D</i>	i_{max}	$\cong 14i_{max}\mu_{max}$

канала высотомера. Если матрица гипотез квадратная, то все выводы справедливы для варианта параллельной обработки по элементам столбца. Если же данная матрица прямоугольная, то в зависимости от отношения

$$\frac{i_{max}}{j_{max}}$$

следует распараллеливать обработку элементов строки либо столбца. После построения ЯПФ необходимо взвесить все вершины графа верхними оценками сложности вычислений. Для рассматриваемого варианта алгоритма суммарные оценки сложности приведены в табл. 3. Видно, что среди рассматриваемых вариантов параллельная обработка всех элементов одной строки матрицы гипотез также обеспечивает непропорциональный рост временной сложности по отношению к максимально параллельному алгоритму. Это обусловлено достижением упомянутого баланса составляющих временных затрат. Архитектура ФОП, реализующего этот алгоритм, аналогична приведённой на рис. 6, число МП при этом равно i_{max} .

Анализ оценок временной и аппаратной составляющих сложности вычислений показывает, что максимально быстрыми являются варианты максимального распараллеливания, то есть A и C . Но значительная аппаратная сложность (соответственно $i_{max}j_{max}\mu_{max}$ и $i_{max}j_{max}$ МП) делает эти варианты нецелесообразными для практической реализации. Поэтому варианты B и C являются наиболее предпочтительными для разработки архитектуры ФОП. Произведём сравнение этих вариантов алгоритмов по критерию L_i , для чего составим неравенство

$$\delta L_i = L_i^B / L_i^D > 1 \Rightarrow \frac{(\mu_{max} + 13)i_{max}}{14\mu_{max}} > 1. \quad (11)$$

Решение неравенства имеет следующий

вид: $j_{max} > \frac{14\mu_{max}}{\mu_{max} + 13}$, т.е. параллельная обра-

ботка всех элементов одной строки матрицы гипотез практически всегда даёт выигрыш во времени решения задачи привязки.

Значительный теоретический интерес

представляет исследование эффективности рассмотренных вариантов параллельной реализации алгоритма привязки. Как известно [4], одним из критериев системной эффективности вычислений может служить удельный показатель $L_{ht} = L_h \times L_i$. При этом большей эффективности реализации соответствует меньшее значение L_{ht} . Сравнение рассмотренных вариантов по критерию L_{ht} для квадратной матрицы гипотез показывает следующее:

- Для любой конфигурации матрицы гипотез $L_{ht}^D < L_{ht}^A$ при $\mu_{max} > 2(7 \frac{i_{max}}{j_{max}} - 6)$.

Если для алгоритма D принять вариант распараллеливания по меньшему размеру матрицы, то алгоритм D является всегда более эффективным, чем A для любого размера матрицы гипотез и числа каналов высотомера.

- Для алгоритма B справедливо такое же условие, решение неравенства при этом принимает вид:

$$L_{ht}^D < L_{ht}^B \text{ при } \mu_{max} > 14 \frac{i_{max}}{j_{max}} - 13.$$

- Сравнение алгоритмов C и D приводит к следующему очевидному решению:

$L_{ht}^D < L_{ht}^C$ при $8j_{max} > 7i_{max}$. Если учесть возможность выбора параметра распараллеливания в алгоритме D , то также можно сформулировать вывод о его более высокой эффективности по сравнению с алгоритмом C .

Таким образом, по критерию эффективности вычислений следует признать алгоритм параллельной обработки всех элементов одной строки матрицы гипотез наиболее предпочтительным для практической реализации. Он обеспечивает максимальную скорость обработки данных в практически важных случаях.

В течение 80-х и 90-х годов описываемый в работе подход к разработке ФОП, основанный на DH -синтезе, был использован для разработки реальных ФОП КЭНС. При этом был использован алгоритм варианта D .

Результаты разработки ФОП КЭНС

Подход к проектированию ФОП, основанный на *ДН*-синтезе, был использован при разработке двух поколений реальных бортовых ФОП КЭНС, представляющих собой различные поколения.

- ФОП первого поколения (80-е годы).

Этот ФОП разработан на основе стандартных МП. Архитектура ФОП представляет собой систему из 14 процессорных элементов (ПЭ) с общей шиной, общим ЗУ и общим микропрограммным устройством управления. В состав каждого ПЭ входит локальная сверхоперативная память на основе регистров общего назначения, число которых равно 16, что даёт возможность хранить матрицу сигналов размерностью 14×14 внутри процессоров и не затрачивать машинного времени на обращение к общему ЗУ.

В данном ФОП реализован параллелизм двух уровней – в части обработки данных (14 потоков данных) и в части формирования адресов общего ЗУ. С этой целью в архитектуру введен ещё один ПЭ для вычисления текущих адресов общего ЗУ. Адресный ПЭ преобразует адреса переменных в двумерном массиве в адреса переменных в линейном массиве по формуле вида $a + bc$. Ра-

ботой вычислительных ПЭ и адресного ПЭ управляет общее микропрограммное устройство управления, построенное на основе ЗУ микропрограмм. Архитектура ФОП приведена на рис. 7, технические характеристики – в таблице 4.

Описываемый ФОП под названием С-6118 был разработан, изготовлен и отработан сначала на макете, а затем на этапе экспериментального образца в составе функционирующей КЭНС.

Аппаратная реализация С-6118 основана на применении КМОП БИС серий 588,537 и КМОП СИС серии 564. Параллельная архитектура дала возможность весьма эффективно сочетать применение низкочастотных КМОП - схем, что привело к низкой потребляемой мощности ФОП (6 Вт) с достаточно высокой производительностью (4 млн. эквивалентных оп/сек). Это, в свою очередь, дало возможность применить обычные конструктивно-технологические решения для бортовой аппаратуры – двухслойные печатные платы, наклеенные на каркасы из сплава *Al*.

Однокристалльные микропроцессоры серии 588BC2, являясь лучшими из КМОП БИС в нашей стране начала 80-х годов, по-

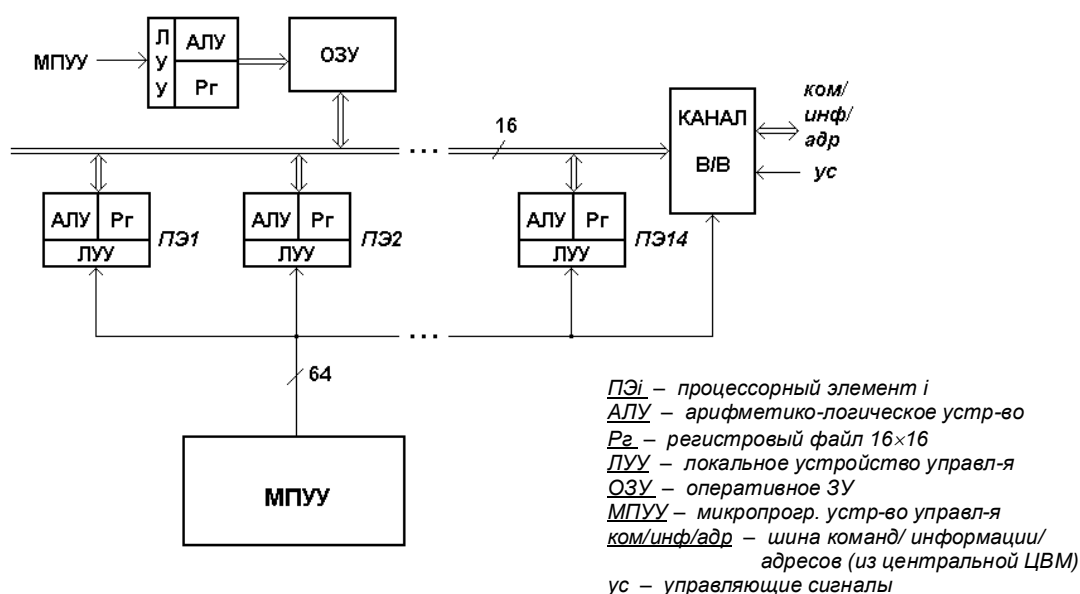


Рис. 7. Архитектура ФОП КЭНС С-6118

Таблица 4. Характеристики параллельных ФОП КЭНС

Характеристики	Модели ФОП	
	С-6118	С-9118
Разрядность, бит	16,32	16,32
Производительность, 10^6 экв. оп/сек	4	8
Емкость ОЗУ, Кбайт	72	112
Габариты, мм×мм×мм	295×182×84	295×182×42
Масса, кг	2,6	1,5
Мощность, Вт	6	15
Элементная база: процессор,	1824	1824
память	537	Б1617
Программное обеспечение	Специализированный язык	Ассемблер
Средства контроля	Тесты	Тесты
Средства обеспечения надежности	Дублирование	Дублирование
Год разработки	1983	1989

зволили реализовать внутреннюю общую шину с минимумом затрат. Система микрокоманд 588 ВС2 была одной из наиболее развитых для схем такого класса, что позволило достичь минимальных по длине программ вычислений.

- ФОП 2-го поколения (80-90-е годы).

Усложнение алгоритмов КЭНС и ужесточение требований на объемно-массовые характеристики бортовой аппаратуры привело

к появлению нового поколения ФОП, представляющего собой развитие архитектуры С-6118.

Анализ путей совершенствования архитектур многопроцессорных ФОП показал, что основными путями являются:

1. Введение в состав ПЭ блоков аппаратной реализации операций, более сложных, чем сложение, и, в первую очередь, умножение.

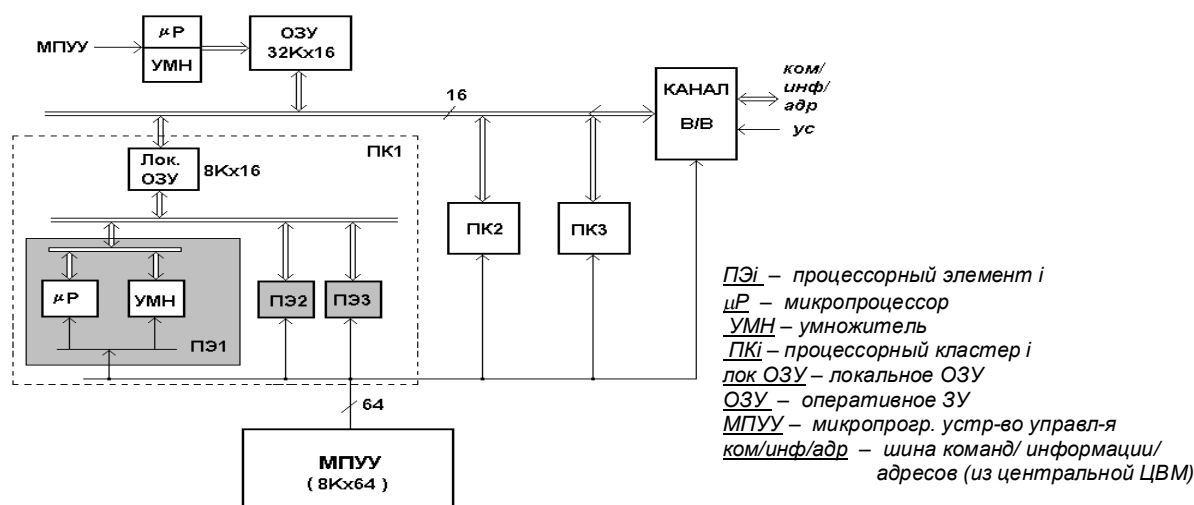


Рис. 8. Архитектура ФОП КЭНС С – 9118

2. Увеличение ёмкости локальных ЗУ.

3. Реализация прямой адресации локальных ЗУ как со стороны адресного ПЭ, так и со стороны “своих” ПЭ.

В результате исследования влияния алгоритмов КЭНС и решений, совершенствующих архитектуру С-6118, была смоделирована, оценена и разработана SIMD-архитектура многопроцессорного ФОП с тройной иерархией ЗУ. В состав каждого ПЭ входили не только универсальные МП, но и умножители, реализующие операцию умножения за такт короткой операции (сложения). Численные эксперименты с моделью ФОП показали, что для выбранного класса задач, а также исходя из требования минимума аппаратных затрат на параллельную работу, все ПЭ должны быть объединены в группы из трёх ПЭ, соединённые с одним локальным ЗУ каждая. Все остальные архитектурные решения были подобны С-6118, в том числе вариант параллельной обработки данных.

Наличие умножителей позволило уменьшить общее число ПЭ с 14 до 9, что сократило массу всего ФОП, а производительность увеличить до 8 млн. оп/сек. Архитектура этого ФОП приведена на рис.8. Данный ФОП, разработанный в 1989 году, получил название С-9118.

Результаты разработки и испытаний бортовых ФОП КЭНС полностью подтвердили правильность принятых технических решений, особенно в части обеспечения высокой производительности при реализации алгоритмов привязки. Кроме того, сравнительный анализ возможных вариантов распараллеливания алгоритма и информации о близких по назначению процессоров показали высокую эффективность применения *DH*-синтеза алгоритмов на этапе выбора основных архитектурных решений бортовых спецвычислителей.

Выводы

1. Алгоритм привязки является основным в системе алгоритмов КЭНС. Его отличает высокая вычислительная сложность, обусловленная большим количеством простых преобразований, которые необходимо выполнить за минимальное время.

2. Граф алгоритма привязки принадлежит к классу потоковых, топология графа относится к типу струнных гамаков. Поэтому целесообразно возложить реализацию алгоритма на автономный ФОП, который является модулем БЦВС.

3. Системное проектирование ФОП предполагает следующие этапы:

- Анализ алгоритма, включающий выявление внутреннего параллелизма и определение наиболее весомых с точки зрения вычислительной сложности преобразований.

- Направленное изменение графа на основе *DH*-синтеза алгоритмов, начиная с максимальной ЯПФ, и построение верхних оценок её ширины и высоты.

- Взвешивание вершин графа верхними оценками аппаратной и временной сложности преобразований в базисе схем из функциональных элементов СБИС.

- Построение (синтез) взвешенного графа с минимальной высотой H (в случае требований на реализацию ФОП с максимальной производительностью) либо с минимальной шириной D (при требованиях на реализацию ФОП с минимальными аппаратными затратами).

- Синтез архитектуры ФОП как двумерного массива процессорных элементов, каждый из которых настроен на реализацию отдельного функционального преобразования (для варианта реализации ФОП в виде заказной СБИС), и как многопроцессорной архитектуры, состоящей из МП, реализующих в совокупности алгоритмы КЭНС (для варианта реализации ФОП на базе стандартных МП).

4. Процедуры *DH*-синтеза алгоритмов обеспечивают формирование архитектур ФОП с максимальной производительностью реализации алгоритма привязки при заданных ограничениях на аппаратные затраты либо с минимальными затратами при заданных требованиях на время решения задачи.

5. Среди возможных результатов *DH*-синтеза алгоритмов КЭНС выделяется вариант алгоритма параллельной обработки всех элементов одной строки матрицы гипотез и последовательной обработки данных для ка-

налов высотомера. Этот вариант обеспечивает минимальное время реализации алгоритма привязки для многопроцессорных архитектур ФОП.

Работа поддержана Программой Президиума РАН №2 “Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация” (Направление 3. Вычислительные технологии и системы).

Библиографический список

1. Белоглазов И. Н., Тарасенко В. П. Корреляционно-экстремальные системы. - М.: Сов. Радио, 1974. – 392 с.
2. Бердышев В. И., Костоусов В. Б. Экстремальные задачи и модели навигации по геофизическим полям. - Екатеринбург : УрО РАН, 2007. - 270 с.
3. Воеводин В. В. Математические модели и методы в параллельных процессах. – М.: Наука, 1986. – 296 с.
4. Лукин Н. А. Функционально-ориентированные процессоры для систем реального времени: основные понятия, история развития, элементы теории проектирования. // Высокопроизводительные вычислитель-

ные системы// Материалы Четвертой молодежной школы. – Таганрог, Изд-во ТТИ ЮФУ, 2007. – С. 98 – 124.

5. Сэвидж Д. Э. Сложность вычислений: Пер. с англ. – М.: Факториал, 1998. – 368 с.

References

1. Beloglazov I.N., Tarasenko V.P. Correlation extreme systems. Moscow: Soviet Radio, 1974. (in Russian).
2. Berdyshev V.I., Kostousov V.B. Extreme problems and models of navigation on geophysical fields. Yekaterinburg, Urals branch of RAS, 2007. (in Russian).
3. Voevodin V.V. Mathematical models and methods for parallel processes. – Moscow, Science, 1986. (in Russian).
4. Lookin N.A. Functional-oriented processors for real-time systems: main categories, history, elements of design theory. // High-performance computer systems// Proceedings of IVth science school. – Taganrog, TTI Publishers, 2007. – 98 – 124 pp. (in Russian).
5. Savage J.E. The Complexity of Computing. – New York, London, Sydney, Toronto: John Wiley & Sons, 1976.

SYSTEM DESIGNING OF FUNCTION-ORIENTED PROCESSORS FOR ONBOARD CORRELATION EXTREME NAVIGATION SYSTEMS

© 2009 N. A. Lukin

Institute of Machine Science, Ural department of the Russian Academy of Sciences,
Science-and-Production Association of Automation, Yekaterinburg

The paper presents the results of investigating optimal realization of the conjunction algorithm, one of the main ones in the complex of algorithms of correlation extreme navigation systems using sensors of geophysical fields. The main result of the investigation is the procedure of algorithm DH-synthesis which makes it possible to synthesize rational architectures of function-oriented processors (FOP). Possible implementation of the conjunction algorithm on the basis of a two-dimensional array of processor elements in VLSI technology is described. The results of DH-synthesis as applied to FOP based on standard microprocessors are presented. The results of developing two generations of actual onboard FOPs for CENSs produced with the use of the procedure proposed are given.

Correlation extreme navigation systems, function-oriented processor, complexity of computations, parallel data processing.

Информация об авторе

Лукин Николай Алексеевич, кандидат технических наук, старший научный сотрудник, заведующий лабораторией. Институт машиноведения Уральского отделения РАН. Область научных интересов: функционально-ориентированные процессоры, параллельные вычисления, сверхбольшие интегральные схемы, компьютерная арифметика, сложность вычислений. E-mail: nicklookin@mail.ru.

Lukin, Nikolay Alexeyevitch, head of laboratory, candidate of technical science, senior researcher, Institute of Machine Science, Ural department of the Russian Academy of Sciences. Area of research: function-oriented processors, parallel computations, very large-scale integrated circuits, computer arithmetic, complexity of computations. E-mail: nicklookin@mail.ru.