

Н. С. КОПОРСКИЙ, Б. В. ВИДИН, И. О. ЖАРИНОВ

ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В МНОГОМАШИННОМ БОРТОВОМ ВЫЧИСЛИТЕЛЬНОМ КОМПЛЕКСЕ

В работе рассматривается подход к разработке эффективных методов организации вычислительных процессов на борту летательных аппаратов. Предлагаются: 1) методика декомпозиции вычислительной системы на функциональные элементы 2) методика определения вектора структурных переменных, существенных для сравнения различных вариантов построения систем.

Ключевые слова: бортовое оборудование, вычислительные процессы, распределение ресурсов.

Введение. В настоящее время оборудование на борту летательных аппаратов непрерывно усложняется, в связи с чем встает вопрос оптимальной централизации вычислительных средств [1]. Традиционно в бортовых комплексах гражданской и военной авиации вычислительные функции выполнялись специализированной бортовой цифровой вычислительной системой (БЦВС), как правило, состоящей из нескольких резервированных вычислителей, выполненных на базе однокристалльной микроЭВМ (ядро БЦВС).

Современная практика построения аппаратуры БРЭО показывает, что в большинстве случаев отдельные компоненты бортовых систем — электронные блоки — уже включают в свой состав микропроцессорные платформы, использующиеся, например, в интеллектуальных периферийных устройствах ввода/вывода, графоускорителях (графических контроллерах систем индикации), в источниках бортового электропитания и т.д.

Как следствие этого, под бортовыми вычислительными системами сегодня следует понимать уже не отдельную, пусть даже и резервированную, бортовую цифровую вычислительную машину (БЦВМ), а более сложный многомашинный комплекс модульных средств вычислительной техники с периферийными устройствами и средствами индивидуального математического и программного обеспечения.

Поскольку современные бортовые комплексы имеют в большинстве случаев общую структуру, а также аналогичную элементную базу, можно ставить задачу создания обобщенной математической модели вычислительной системы БРЭО. Применение современных методов поиска оптимальных вариантов позволит эффективно использовать обобщенную модель и исключить субъективный подход при выборе структуры и элементной базы проектируемых бортовых вычислительных систем. Для построения обобщенной математической модели системы БЦВС необходимо осуществить разбиение системы на элементы. От метода разбиения существенно зависит структура обобщенной математической модели и возможность эффективного использования поисковых процедур при работе с моделью.

Сущность метода разбиения заключается в следующем: исходя из назначения проектируемой системы, получают необходимые условия наличия в ее структуре ряда элементов, тип которых может быть различным. В качестве элементов могут выступать как собственно технические устройства (датчики, исполнительные устройства, усилители и т.д.), так и логические операции, которые должны быть выполнены при решении задачи.

Не снижая общности рассуждений, будем полагать, что синтезу подлежат две независимых математические модели составляющих БЦВС различных уровней:

- модель многомашинной вычислительной системы со связностью на уровне функциональных задач всего комплекса БРЭО (межмашинные логические и электрические связи медленного цикла);

- модель многопроцессорной вычислительной машины со связностью на уровне конкретной задачи отдельной БЦВМ (межпроцессорные логические и электрические связи быстрого цикла).

Математические модели БЦВС различных уровней. Модель многомашинной вычислительной системы. Пусть имеется n информационно связанных задач, которые необходимо решать с помощью многомашинной БЦВС, и соответствующие этим задачам алгоритмы. Каждый алгоритм может быть представлен как некоторая последовательность функциональных операторов.

Граф, соответствующий результирующему вычислительному алгоритму, образуется следующим образом. Каждому, например, i -му функциональному оператору Φ_i ставится в соответствие вершина графа, возле которой записывается ее вес — время выполнения данного функционального оператора в относительных единицах. Вершины Φ_i и Φ_j соединяются линией со стрелкой (дугой графа), направленной в Φ_i только в том случае, если результат, полученный после выполнения Φ_j , является одним из аргументов для Φ_i . Всякая дуга выражает либо вычислительную зависимость между соответствующими функциональными операторами, либо требования порядка выполнения соединяемых вершин, либо то и другое.

Совокупность графов n задач составляют метаграф G_{BC} процесса решения n задач всей вычислительной системы. Так как рассматриваемые задачи информационно связаны, то и граф G_{BC} является связным. Проектирование модели БЦВС состоит в выборе числа вычислительных машин и в определении электрических и логических связей между ними в соответствии с графом G_{BC} , ограничениями (время, надежность и т.д.) и некоторым функционалом оптимальности.

Назовем вершины графа G_{BC} , в которые входят дуги исходных данных, начальными, а вершины, из которых выходят дуги результата решения задач или фрагментов вычисления, — конечными (так, на рис.1 вершины v_1, v_5, v_9 являются начальными, а вершины v_6, v_7, v_{11} - конечными). Назовем также подграф G_{BCi} графа G_{BC} независимым, если ни в одну его вершину не входит дуга другого подграфа.

Нетрудно видеть, что для организации вычислительного процесса необходимо разложить граф G_{BC} на независимые подграфы G_{BCi} . Действительно, для независимого подграфа не нужны промежуточные результаты операторов, не входящих в этот подграф, и, следовательно, каждый из них можно реализовать на отдельных вычислительных машинах, обменивающихся информацией по «медленному» интерфейсу. Очевидно, число БЦВМ будет зависеть от числа независимых подграфов.

Поскольку все БЦВМ будут работать параллельно, время выполнения всех n задач t_p при такой структуре БЦВС будет минимальным, т.е. дальнейшее увеличение числа процессоров не уменьшит t_p . Рассмотрим следующие основные положения, на которых строится алгоритм разложения графа на независимые подграфы.

Если существует множество вершин $v_{i_1}, v_{i_2}, \dots, v_{i_k}$, из которых дуги выходят и входят в вершину v_i , то это множество принадлежит независимым подграфам, в которые входит вершина v_i . В то же время вершина v_i будет входить в независимые подграфы, строящиеся на базе вершин, в которые входят дуги, выходящие из вершины v_i .

Определим матрицу размерностью $m \times m$ (m – число вершин графа), у которой элемент на пересечении i -й строки и j -ого столбца равен 1, если имеется дуга, направленная от i -й вершины к j -й вершине. Если дуга имеет противоположное направление, соответствующий элемент равен -1 . При отсутствии такой дуги рассматриваемый элемент определяется как 0.

На первом этапе алгоритма рассматриваются строки матрицы графа G_{BC} с целью выявления строки, имеющей только отрицательные единицы (тем самым определяется конечная вершина). Номер вершины данной строки будет составлять первый элемент образуемого массива.

На втором этапе по полученной строке определяются столбцы, которые имеют на пересечении с ней отрицательные элементы. Тем самым выявляются вершины, входящие в независимый подграф найденной конечной вершины. Номера этих вершин вводятся в массив.

На третьем этапе просматриваются элементы найденных выше столбцов, и отрицательные из них обнуляются. Этим исключается возможность последующего просмотра уже найденных вершин при наличии в графе контуров. Далее повторяются манипуляции второго этапа со строками, соответствующими указанным выше столбцам, и т.д. После того как исчерпаны все строки, имеющие отрицательные элементы, массив, образованный в результате реализации алгоритма, выводится на печать. Этот массив дает перечисление всех вершин, входящих в первый независимый подграф.

На четвертом этапе происходит подготовка к реализации следующего цикла алгоритма. С этой целью стертые ранее отрицательные элементы восстанавливаются, а полученные на первом этапе строки и соответствующие им столбцы обнуляются. Далее процедура повторяется (начиная с первого этапа).

Алгоритм прекратит реализацию, после того как все строки, имеющие только отрицательные элементы, будут исчерпаны. Если каждый независимый i -й подграф графа G_{BC} реализуется на отдельном процессоре за время t_{p_i} , то $t_p = \max \{t_{p_1}, t_{p_2}, \dots, t_{p_N}\}$.

Если имеются ограничения вида $\sum_{j=1}^l t_{p_{ij}} \leq t_p$, тогда l подграфов, которым принадлежит

суммарное время решения, могут быть реализованы последовательно на одном бортовом вычислителе, при этом общее время t_p решения n задач не увеличится. Конструктивная реализация многомашинной вычислительной системы дает оптимум по критерию минимума числа БЦВМ в БЦВС.

Применение алгоритма функционального разделения графа, показанного на рис.1, позволяет получить три его независимых подграфа (см. рис.2), каждый из которых соответствует своему алгоритму, исполняемому на отдельной БЦВМ.

Модель многопроцессорной вычислительной машины. Как уже отмечалось, вычислительный алгоритм задачи БЦВМ может быть представлен в виде функционального графа операторов, имеющего ярусно-параллельную форму (рис.3). Ветвям графа соответствует выполнение операторов Φ_1, \dots, Φ_{13} ; вершинам — окончания выполнения этих операторов.

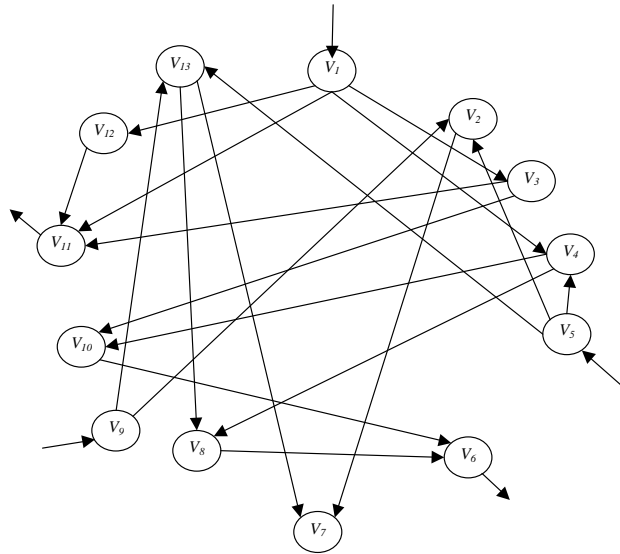


Рис.1. Граф вычислительной системы G_{BC} .

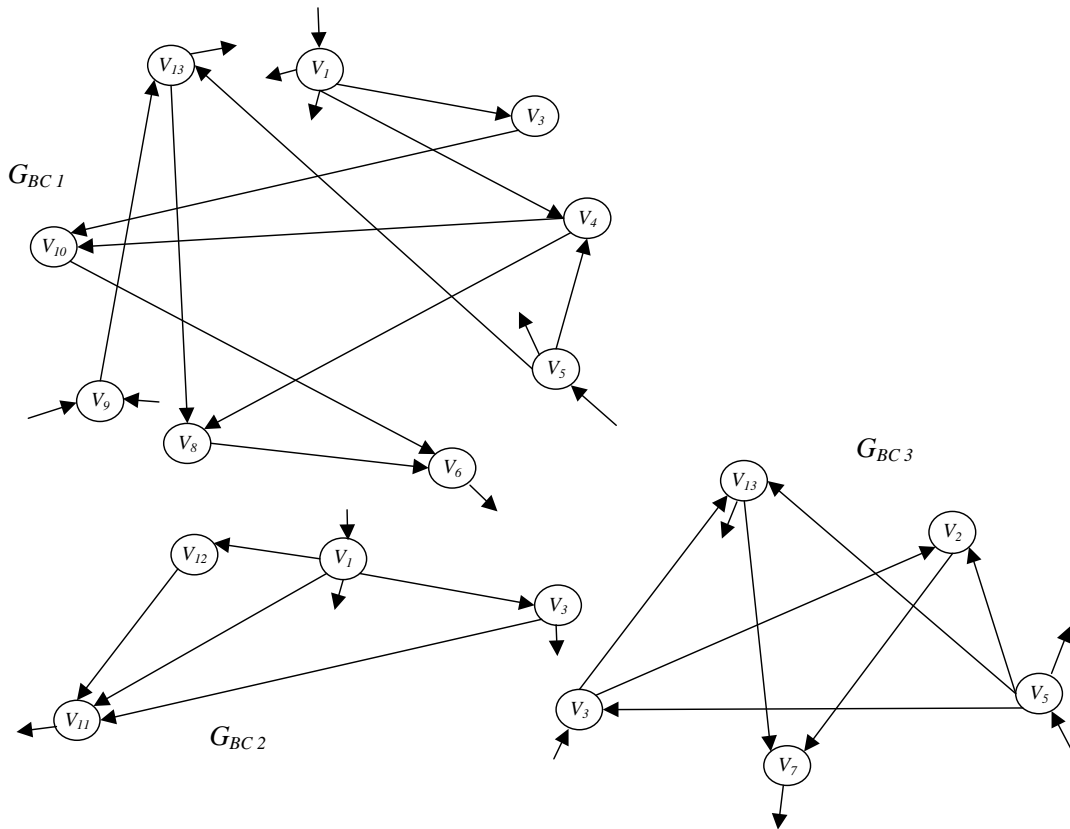


Рис. 2. Независимые подграфы G_{BC1} , G_{BC2} , G_{BC3} графа G_{BC} .

Решение задачи начинается с реализации оператора Φ_1 , затем могут начать выполняться (независимо друг от друга) операторы Φ_2 и Φ_3 . После реализации последнего из операторов Φ_4 , Φ_5 , Φ_6 начинается выполнение операторов Φ_9 и Φ_{11} . Оператор Φ_{12} выполняется после реализации Φ_7 и Φ_8 , Φ_{10} и Φ_9 . Последний из операторов Φ_{13} выполняется после реализации Φ_{11} и Φ_{12} .

Данному графу при решении задачи на многопроцессорной вычислительной машине может быть поставлена в соответствие матрица соединения микропроцессоров. Размерность матрицы определяется числом микропроцессоров, используемых при работе.

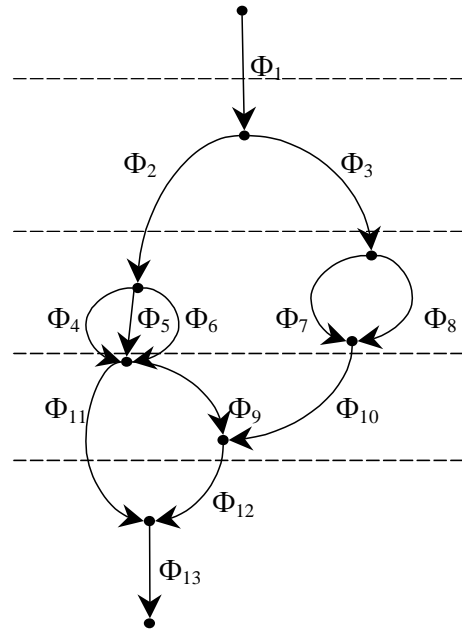


Рис. 4. Матрица соединения процессоров.

Наибольшее число микропроцессоров, необходимых для одновременной работы, определяется по наибольшему числу параллельно выполненных операторов одного яруса. В данном случае этими операторами являются Φ_4 , Φ_5 , Φ_6 , Φ_7 и Φ_8 . Матрица соединения S для данного случая показана на рис. 4.

	1	2	3	4	5
1	1	1	1	1	0
2	1	1	1	0	1
3	1	0	1	1	0
4	1	0	0	1	0
5	0	1	0	0	1

Рис. 4. Матрица соединения процессоров.

Правило заполнения матрицы соединения микропроцессоров следующее. Диагональные элементы соответствуют использованию при решении операторов первого, второго, третьего, четвертого и пятого процессоров.

Результаты выполнения оператора Φ_1 первым процессором служат исходными для выполнения оператора Φ_2 первым и Φ_3 вторым процессорами. Следовательно, элемент матрицы с индексом 1, 2, соответствующий наличию связи между первым и вторым процессорами, будет равен 1. После выполнения оператора Φ_2 первым процессором полученная информация передается на первый, третий, четвертый процессоры. Следовательно, элементы с индексами 1, 3 и 1, 4 будут также равны единицам. Операторы Φ_7 и Φ_8 выполняются вторым и пятым процессорами после выполнения оператора Φ_3 вторым процессором. Следовательно, элемент с индексом 2, 5 заполняется единицей.

После выполнения операторов Φ_4 , Φ_5 и Φ_6 результаты поступают на первый процессор. Элементы матрицы с индексами 3, 1 и 4, 1 заполняются единицами. Оператор Φ_9 выполняется третьим процессором, и, следовательно, элементы с индексами 1, 3 и 4, 3 —

единицы. Оператор Φ_{12} выполняется вторым процессором после выполнения Φ_9 и Φ_{10} , следовательно, элемент с индексом 3, 2, как и элемент с индексом 2, 1, будет равен единице. Полученная таким образом матрица соединения состоит из 0 и 1: единицы соответствуют наличию связи между процессорами во время решения задачи, а нули — ее отсутствию.

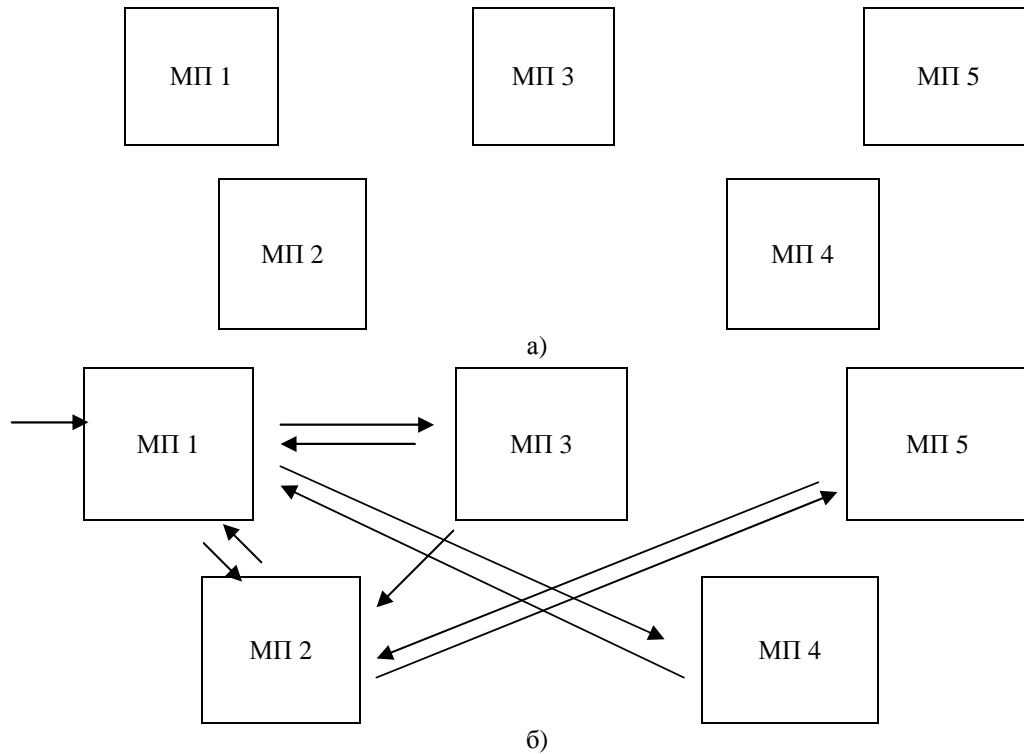


Рис.5. Исходная а) и полученная б) модели вычислительной системы БЦВМ.

В исходной модели (рис.5, а), состоящей в данном случае из пяти микропроцессоров, связи между ними отсутствуют. Матрица соединения, отражающая логику вычислительного процесса, позволяет получить структуру вычислительного устройства (рис.5, б), наилучшим образом реализующего данный алгоритм. Связи между микропроцессорами отражают логику конкретного вычислительного алгоритма и электрически представляют собой каналы информационного обмена по «быстрому» интерфейсу.

Тактико-технические характеристики БЦВС. Модель вектора структурных переменных БЦВС. Полагая, что каждая задача, решаемая БЦВС, может иметь некоторое число различных вариантов реализации, и, соответственно, графов ярусно-параллельной формы, каждому графу решения можно поставить в соответствие свою матрицу соединения. При этом количество вариантов реализации решений задач цикла N очевидно равно

$$N = \prod_{i=1}^n a_i,$$

где n – число задач цикла; a_i – число вариантов решения i -й задачи.

В качестве i -й структурной переменной используются матрицы соединения, соответствующие способу решения каждой i -й задачи. Таким образом, число переменных будет равно n , а число принимаемых ими значений — a_i . Совокупность значений структурных переменных будет определять результирующую структуру вычислительного устройства, предназначенного для решения n задач по определенным вариантам. Элементы суммарной матрицы соединения определяются логическим суммированием элементов матриц для каждой задачи.

Введение вектора структурных переменных позволяет формализовать задачу выбора оптимальной структуры БЦВС на уровне переменных, что позволяет применить для ее решения эффективные вычислительные методы.

Модель производительности БЦВС. Пусть средняя производительность каждого процессора равна $1/W_i$ оп/с, а оператор Φ_k ($k = \overline{1, 13}$) содержит f_k операций. Из структуры графа ясно, что последующий оператор начинает выполняться лишь в том случае, когда выполнены все предыдущие, связанные с началом данного. Поэтому, когда оператору Φ_k предшествуют операторы Φ_{k-1} и Φ_{k-2} , выполненные параллельно на двух процессорах, время выполнения Φ_{k-1} , Φ_{k-2} и Φ_k определяется как $T = \max(W_1 f_{k-1}; W_2 f_{k-2}) + W_3 f_k$.

Правило вычисления T сводится к следующему. В исходном графе выполняются параллельные ветви, ведущие от последнего общего оператора к последнему оператору задачи. Время T определяется как сумма времени последовательного выполнения операторов, общих для всех путей, и максимального времени параллельного выполнения операторов. Так, для графа, представленного на рис.2, значение T будет равно $T = W_1 f_1 + \max\{A; [\max(B; C) + W_2 f_{12}]\} + W_1 f_{13}$, где

$$A = [W_1 f_2 + \max(W_1 f_4; W_3 f_3; W_4 f_6) + W_1 f_{11}];$$

$$B = [W_2 f_3 + \max(W_2 f_7; W_5 f_8) + W_2 f_{10}]; \quad C = [W_1 f_2 + \max(W_1 f_4; W_3 f_3; W_4 f_6) + W_3 f_9].$$

Аналогичным образом могут быть получены выражения для времени решения задачи, имеющей другой вид графа ярусно-параллельной формы. Выражения для T , полученные на основе предварительного анализа вариантов решения задач, могут использоваться при решении задачи оптимального проектирования. Эти выражения справедливы при различных значениях W_i и f_k .

Для определения времени счета необходимо указать число ветвей графа nb , а также количество вершин kv . Необходимо также описать все ветви графа и свести их в весовую матрицу $|a|$ размерностью $(nb \times kv)$. Описание ветвей графа производится по следующему правилу: каждая ветвь графа характеризуется четырьмя величинами $\{a_1, a_2, a_3, a_4\}$, образующими строку матрицы $|a|$, где a_1 - номер вершины, из которой данная ветвь выходит; a_2 - номер вершины, в которую данная ветвь входит; a_3 и a_4 - параметры, характеризующие вес данной ветви графа или, в данном случае, производительность и количество операций, выполняемых i -м процессором на i -м шаге вычислений. Совокупность описаний ветвей графа и образует полную матрицу $|a|$.

Первым этапом определения времени счета графа является вычисление длительности выполнения каждого оператора Φ_k ($k=1, \dots, nb$) графа i -м процессором (для графа, представленного на рис.2, $nb=13$). Для этого элементы третьего и четвертого столбцов матрицы $|a|$ построчно перемножаются, и полученный результат записывается в одноименную строку третьего столбца матрицы $|a|$: $a[i,3] = a[i,3] \times a[i,4]$ ($i = 1, \dots, nb$).

Второй этап заключается в последовательном «свертывании» графа, для чего в матрице $|a|$ ищутся ветви Φ_k ($k=1, \dots, nb$), которые включены между парами одних и тех же узлов. Далее определяется максимальный оператор (из Φ_k), т.е. оператор с наибольшим весом, а операторы с меньшими весами обнуляются. Для этого просматриваются элементы первого столбца матрицы $|a|$ и, если встречаются два элемента столбца

$a[i,1] = a[l,1]$ ($i = 1, \dots, nb - 1; l = i + 1, \dots, nb$), проверяется равенство соответствующих элементов из второго столбца $a[i,2] = a[l,2]$.

Если равенство выполняется, то определяется максимальный из двух элементов $a[i,3]$ и $a[l,3]$. Если $a[i,3] \geq a[l,3]$, то $a[l,1] = 0$; если $a[i,3] < a[l,3]$, то $a[i,1] = 0$.

Третий этап заключается в преобразовании матрицы $|a|$ размерностью $(nb \times 4)$ в матрицу $[S]$ размерностью $(kv \times kv)$. Для этого последовательно отыскиваются ненулевые элементы первого столбца матрицы $|a|$. Если встречается такой элемент $a[i,1] \neq 0; i = 1, \dots, nb$, то элементу матрицы $[S]$, стоящему на пересечении строки с номером $a[i,1]$ и столбца с номером $a[i,2]$, присваивается значение $a[i,3]$ матрицы $|a|: S[a(i,1), a(i,2)] = a[i,3], i = 1, \dots, nb$.

Четвертый этап определения времени счета заключается в преобразовании квадратной матрицы $[S]$ размерностью $(kv \times kv)$ в одномерную матрицу $|B|$ размерностью (kv) . Размерность матрицы $|B|$ определяется числом вершин графа, и каждая ячейка $b[i]$ соответствует i -й вершине графа. В ячейке матрицы $|B|$ записывается фактическое время счета, соответствующее данному шагу вычислений. Например, в ячейке $b[i]$ будет записано время, потраченное на вычисление всех предыдущих операторов ($b[i-1]$), плюс время вычисления оператора, стоящего между $i-1$ и i -й вершиной графа, или, в общем виде, $b[j] = b[i] + S[i, j]$, где $S[i, j]$ - время вычисления оператора Φ_k , стоящего между i -й и j -й вершинами; i - вершина, из которой выходит ветвь Φ_k ; j - вершина, в которую входит ветвь Φ_k .

Если в вершину j входит несколько ветвей (операторов), то время свершения события, или результат вычисления нескольких процессоров, будет определяться работой процессора, для которого $b[i] + S[i, j]$ максимально, т.е. того процессора, который последним закончит работу. Поэтому в ячейку с номером j запишется максимальное из $b[j] = S[i, j] + b[i]$, если $b[j] - [S(i, j) + b(i)] < 0$, где $b[j]$ - результат, предварительно записанный в ячейку с номером j .

Для всех ветвей $S[i, j]$, входящих в вершину j , первым будет записано время $b[j]$ для ветви с наименьшим i . Если результат вычисления времени для всех других ветвей $b[i] + S[i, j]$ окажется меньшим уже имеющегося в ячейке $b[j]$, то значение $b[j]$ сохраняется. Если же вновь вычисленное время $b[i] + S[i, j]$ будет больше $b[j]$, то в ячейку заносится полученная сумма. Матрица S размерностью $(kv \times kv)$ преобразуется в матрицу $|B|$ размерностью (kv) следующим образом: последовательно просматриваются все элементы $S[i, j]$ матрицы $[S]$ ($i=1, \dots, kv; j=1, \dots, kv$), и, если встречается ненулевой элемент $S[i, j] \neq 0$, время записывается в ячейку $b[j]$ по выше описанному правилу. Последний ненулевой элемент $b[i, j]$ определит заполнение ячейки $b[kv]$, в которой будет записано время вычисления последнего оператора плюс время, предшествующее его вычислению, т.е. искомое время счета всего графа.

Модель надежности БЦВС. При построении модели надежности следует исходить из того, что поток отказов элементов распределен по экспоненциальному закону и отказ одного из элементов ведет к нарушению работы всего вычислительного устройства. Матрица надежности системы, состоящей из нескольких процессоров, будет иметь следующий вид:

$$\lambda = \begin{vmatrix} \lambda_{11} & & \dots & & \lambda_{1n} \\ & \lambda_{22} & & & \\ \cdot & & & & \cdot \\ \cdot & & \lambda_{ij} & & \cdot \\ \cdot & & & & \cdot \\ \lambda_{n1} & & \dots & & \lambda_{nn} \end{vmatrix}$$

где λ_{ij} – показатели надежности собственно процессоров; $\lambda_{ij} (i \neq j)$ – показатели надежности элементов взаимосвязи i и j -ого процессоров. Суммарный показатель надежности λ_{Σ} , соответствующий заданному числу процессоров БЦВС и их взаимосвязей, определяется согласно матрице соединения $\lambda_{\Sigma} = \sum_{j=1}^n \sum_{i=1}^n \lambda_{ij} C_{ij}$, где C_{ij} - элементы матрицы соединения.

Модель массы и габаритных размеров БЦВС. Математическая модель массы и габаритных размеров строится аналогично модели надежности. Матрица массы (габаритов) имеет следующий вид

$$P = \begin{vmatrix} P_{11} & \cdot & \cdot & \cdot & \cdot & P_{1n} \\ \cdot & P_{22} & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & P_{ij} & & & \cdot \\ \cdot & & & & & \cdot \\ P_{n1} & \cdot & \cdot & \cdot & \cdot & P_{nn} \end{vmatrix}$$

где P_{ij} – масса (габаритные размеры) собственно процессоров; $P_{ij} (i \neq j)$ – масса (габаритные размеры) элементов взаимосвязи i и j -ого процессоров. Оценка суммарной массы (габаритных размеров) определяется по формуле $P_{\Sigma} = \sum_{j=1}^n \sum_{i=1}^n P_{ij} C_{ij}$.

Алгоритм выбора оптимальной структуры БЦВС. Для определения оптимальной (в смысле определенного ряда показателей) структуры бортовой вычислительной (многопроцессорной) системы целесообразно использовать алгоритм, блок-схема которого показана на рис.6. Алгоритм построен следующим образом.

В блок 01 вводятся исходные данные, которые включают наборы матриц соединения для каждой рассматриваемой задачи; зависимость времени решения от вида графа решения; производительность процессоров; количество операций при выполнении операторов задач; данные по массам, надежности и габаритным размерам процессоров и элементов их взаимосвязи.

В блоке 02 вычисляется суммарная матрица соединения рассматриваемых вариантов задач цикла. Показатели систем вычисляются в блоке 03 по математической модели в соответствии с суммарной матрицей соединения.

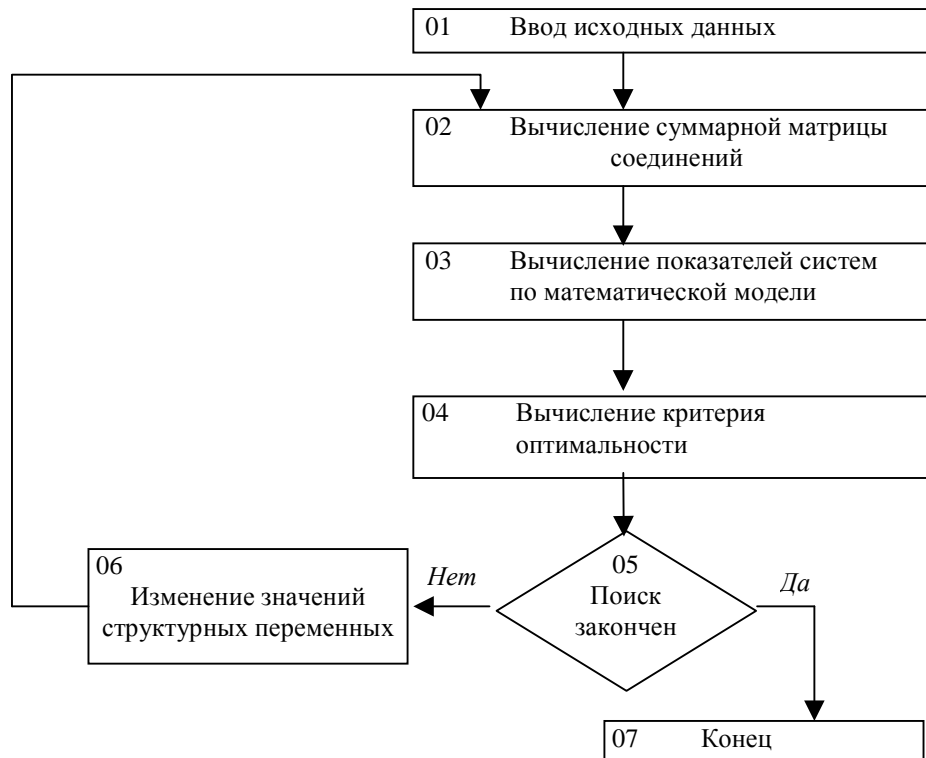


Рис.6. Блок-схема алгоритма структурной оптимизации БЦВС.

В блоке 04 вычисляется критерий оптимальности, в качестве которого целесообразно использовать линейную комбинацию показателей $F = F_1 + \sum_{i=1}^k C_i F_i$, где

$$C_i = \begin{cases} 0 & \text{при } F_i \leq F_i^* \\ C_i & \text{при } F_i > F_i^* \end{cases}, F_1 - \text{минимизируемый показатель системы; } F_i - \text{показатели, которые}$$

не должны быть меньше заданных F_i^* .

В блоке 04 также собирается информация о статистических свойствах функции распределения критерия и ее параметрах, необходимых для выбора наиболее эффективных методов поиска, причем наиболее эффективный метод выбирается с помощью предварительного анализа условий задачи оптимизации: числа переменных, количества принимаемых ими значений и параметров функции распределения значений критерия.

В блоке 05 проверяются условия завершения оптимизации, а в блоке 06 осуществляется изменение значений структурных переменных в соответствии с выбранным методом поиска.

Полученные в ходе расчетов результаты включают номера вариантов решения рассматриваемых задач, структуру БЦВС (электрическую и логическую схемы связи вычислительных устройств на уровне связности системы внутри БЦВМ и всей БЦВС), а также значения оцениваемых показателей математической модели БЦВС.

Литература

1. Федосеев Е. П. Сквозная технология синтеза систем информационного обмена информационно-управляющих комплексов — реальный путь к снижению стоимости разработки бортового радиоэлектронного оборудования. // World wide web recourse: http://www.inr.troitsk.ru/~elics/icsnet/koi/section5_koi.html.