

УДК 629.7.05

ПРИНЦИПЫ ПОСТРОЕНИЯ КРЕЙТА БОРТОВОЙ МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ АВИОНИКИ ПЯТОГО ПОКОЛЕНИЯ

О.О. Жаринов, Б.В. Видин, Р.А. Шек-Иовсепянц

Рассматриваются принципы построения бортовой цифровой вычислительной системы в классе структур интегрированной модульной авионики. Определены значимые архитектурные признаки вычислительных систем. Рассмотрены модель ресурсов вычислительной системы и трехступенчатая иерархическая веерная модель. Рассмотрен метод декомпозиции функциональных задач бортовой вычислительной системы и их назначение на доступные вычислительные ресурсы.

Ключевые слова: принципы построения, бортовая вычислительная система, крейт.

Введение

Авиационные комплексы, находящиеся сегодня в эксплуатации, имеют системно-ориентированную детерминированную структурную организацию. Вычислительный ресурс этих комплексов регулярно распределен между информационными каналами связи посредством организации отдельных подсистем (перераспределение задач между подсистемами на системном уровне не предусматривается), что не обеспечивает достаточной и гибкой интеграции бортового оборудования. Реализуется лишь резервирование наиболее важных функций, решаемых средствами бортовой цифровой вычислительной системы (БЦВС). Кроме того, невозможна полномасштабная адекватная адаптация функциональных ресурсов БЦВС к различным ситуационным изменениям внешней обстановки. При этом возможность программного управления потоками информации (программная коммутация) в архитектурах современных БЦВС не реализуется вообще.

Для радикального повышения эффективности функционирования БЦВС разработчиками сегодня предпринимаются попытки ее реализации и проектирования в классе мультимикропроцессорных структур интегрированной модульной авионики (ИМА) с программируемой архитектурой, допускающей динамическое перераспределение вычислительной мощности аппаратуры в зависимости от приоритета решаемых задач. В связи с этим оказывается актуальной задача разработки такой вычислительной системы, организованной по типу интегрированной вычислительной среды, в которой изначально отсутствует регулярное распределение средств вычислительной техники по функциональным подсистемам и информационным каналам связи.

Иерархическая трехступенчатая веерная модель БЦВС

Открытая архитектура интегрированной модульной авионики строится по иерархическому принципу [1]:

- нижний уровень иерархии образуют унифицированные быстросъемные функциональные модули различного назначения, имеющие собственные вычислительные средства в компактном стандартном исполнении (европлаты с типоразмерами 3U, 6U, 9U);
- средний уровень иерархии образуют мультимикропроцессорные вычислительные системы, создаваемые из модулей нижнего уровня и конструктивно интегрированные в стандартный крейт;
- высший уровень иерархии представляет собой бортовую локальную вычислительную сеть, интегрирующую вычислительные средства боксов среднего уровня, на основе центрального сетевого интерфейса высокой пропускной способности.

В иерархической структуре ИМА заложен принцип присвоения более высокого ранга тем подсистемам управления и информационного обеспечения летательного аппарата (ЛА), которые в большей степени способствуют повышению эффективности использования ЛА в целом. Согласно принятой терминологии [2], схема иерархической организации БЦВС по концепции ИМА относится к классу трехступенчатых веерных структур, определяющих систему, в которой существует один привилегированный субъект (центр), который имеет возможность управлять остальными субъектами – посредниками П. Целевая функция центра Ц имеет вид $\Pi = \Pi(r_{x_1}, r_{x_2}, \dots, r_{x_k}; r_{y_1}, r_{y_2}, \dots, r_{y_k})$, где $\{r_{x_1}, r_{x_2}, \dots, r_{x_k}\}$ – ресурс управления, который находится в распоряжении центра Ц (воздействие центра на посредников $\Pi_1, \Pi_2, \dots, \Pi_k$). Целевые функции посредников Π_i имеют вид $\Pi_i = \Pi_i(r_{y_{i1}}, r_{y_{i2}}, \dots, r_{y_{ie}}; r_{z_{i1}}, r_{z_{i2}}, \dots, r_{z_{ie}})$, $i=1, 2, \dots, k$, где $\{r_{y_{ij}}\}$ – это управление Π_i , его воздействие на исполнителей I_{ij} (исполнителя номер j , подчиненного по уровню иерархии посреднику Π_i).

Таким образом, интересы посредников определяются ресурсом $\{r_{y_1}, r_{y_2}, \dots, r_{y_z}\}$, который находится в их собственном распоряжении, и ресурсом $\{r_{x_1}, r_{x_2}, \dots, r_{x_k}\}$, которым распоряжается центр.

Иерархия представляет собой свойство упорядоченного множества компонентов, между которыми установлено отношение приоритета. Компоненты комплекса, между которыми отсутствует предпочтительность, образуют один иерархический уровень. Приоритет субъектов проявляется в том, что центр назначает правила использования ресурса $\{r_{x_1}, r_{x_2}, \dots, r_{x_k}\}$, который зависит тем или иным образом от действий посредников, от их выбора $\{r_{y_1}, r_{y_2}, \dots, r_{y_z}\}$. Посредникам эти правила становятся известными в тот момент, когда они принимают решения об использовании $\{r_{y_1}, r_{y_2}, \dots, r_{y_z}\}$, тем самым центр в иерархической системе имеет возможность (правило первого хода) направлять в нужное русло усилия элементов низшего иерархического уровня, что создает предпосылки для организации реконфигурируемых вычислительных структур.

Целевые функции исполнителей $I_{ij} = I_{ij}(r_{y_j}; r_{z_j})$, $i = 1, 2, \dots, k$; $j = 1, 2, \dots, z_i$, где величины $\{r_{z_j}\}$ характеризуют действия исполнителей. Первый ход в трехступенчатой иерархической структуре делает центр – он сообщает посредникам правила назначения ресурсов $\{r_{x_1}, r_{x_2}, \dots, r_{x_k}\}$ в зависимости от их выбора. Следующий ход производит посредник, который сообщает правила выбора $\{r_{y_1}, r_{y_2}, \dots, r_{y_z}\}$.

Модель вычислительных ресурсов БЦВС

У каждого типа ресурса r_i существует набор функций $f = \{f_1^i, f_2^i, \dots, f_{j^i}^i\}$, которые он способен выполнять на множестве заданий $Z = \{Z_1, Z_2, \dots, Z_z\}$. Выполнение одного задания Z_i требует выполнения набора функций $\tilde{f}^i = \{\tilde{f}_1^i, \tilde{f}_2^i, \dots, \tilde{f}_{j^i}^i\}$. При этом [3]:

- ресурс $r_i \in R$ считается способным к работе над заданием $Z_i \in Z$, $i=1, 2, \dots, z$, если существует функция $f: f \in f^i$ и $f \in \tilde{f}^i$;
- ресурс $r_i \in R$ считается способным к работе над набором заданий $\bar{Z}_i \in Z$, $i=1, 2, \dots, 2^z$, если $\forall Z_{j_k} \in \bar{Z}_i$, где \bar{Z}_i – множество всех подмножеств заданий Z ;
- совокупность ресурсов $r_i \in R$ считается способной к работе над заданием $Z_i \in Z$, $i=1, 2, \dots, z$, если $\forall r_i \in \bar{R}$, где \bar{R} – множество всех подмножеств ресурсов R ;
- совокупность ресурсов $r_i \in R$ считается способной к работе над набором $\bar{Z}_i \in Z$, $i=1, 2, \dots, 2^z$, если $\forall Z_{j_k} \in \bar{Z}_i, \exists r_i \in \bar{R}_i$, способный к работе над Z_{j_k} .

Распределением заданий считается назначение совокупности $r_i \in R$ для каждого задания $Z_i \in Z$, пригодной к работе над ними, с условием, что никакие два $r_i \neq r_j$ не выполняют одну и ту же функцию f одного и того же задания Z .

Принципы построения аппаратных средств БЦВС

Детализация ИМА-структуры крейта вычислительной системы БЦВС показана на рис. 1 [4]. Приняты следующие обозначения: БР – буферный регистр, ЛОП – локальная оперативная память, ПДП – узел прямого доступа к памяти. Обмен данными между вычислителями MB_i , $i=1, 2, \dots, n$, осуществляется через системный интерфейс СИ и общий модуль памяти МП. Физическую основу реализации БЦВС ИМА-структуры (см. рис. 2) составляют модули-вычислители типа МВ80, модули памяти МП80, разработанные в ФГУП «СПб ОКБ „Электроавтоматика“ им. П. А. Ефимова» и выполненные по стандарту 6U на базе отечественного микропроцессорного комплекта «Мультиборт», который отвечает требованиям стандарта высокоскоростных коммуникаций системного интерфейса SpaceWire ECSS-E-50-12A, внедренного и поддерживаемого ESA (European Space Agency), NASA (США), JAXA (Япония), CSA (Канада) и Российским космическим агентством.

Идея реализации задач комплекса на вычислителях класса MB_1, MB_2, \dots, MB_n заключается в следующем. Каждый из MB_i , $i=1, 2, \dots, n$, функционирует по своей циклограмме, задаваемой препроцессором форматирования. Препроцессор форматирования ПФ $_i$, $i=1, 2, \dots, n$, распознает данные из общего

потока к абонентам или от абонентов, предназначенные для обработки на соответствующем вычислителе $МВ_i, i=1, 2, \dots, n$. В каждом из вычислителей $МВ_i, i=1, 2, \dots, n$, помимо препроцессора форматирования, имеются вычислительный узел ВУ, блок памяти БП и, в случае децентрализованного управления обменом с системным интерфейсом СИ, процессор обмена ПО. От соответствующего препроцессора форматирования в блок обмена БО поступают данные согласно принятому на борту ЛА протоколу взаимодействия. Согласно системе классификации Флинна, такая БЦВС относится к классу многопроцессорных систем с множественным потоком команд и множественным потоком данных (MIMD).

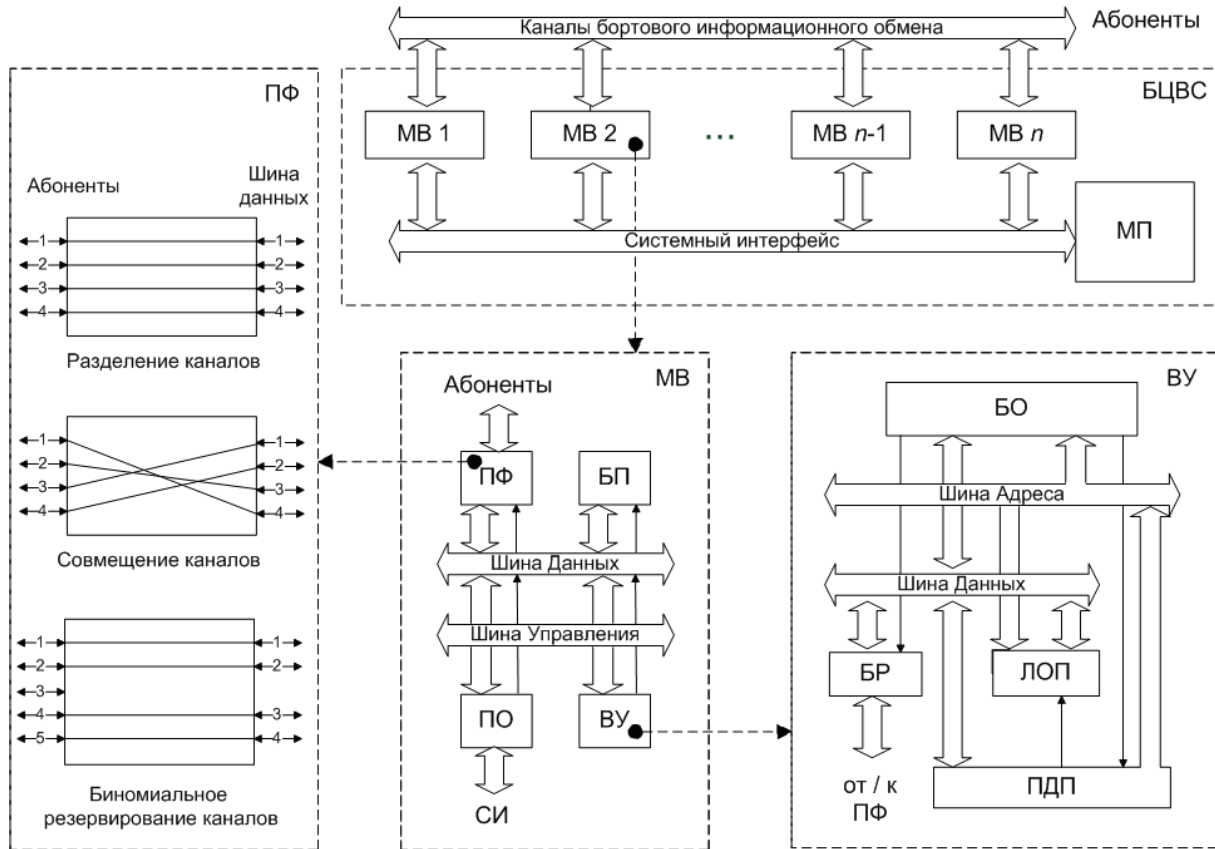


Рис.1. Обобщенная структура многопроцессорной вычислительной системы

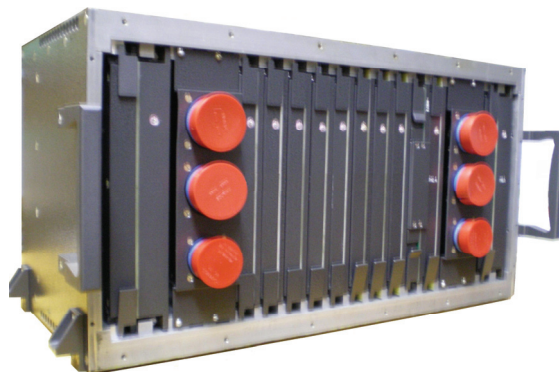


Рис. 2. Прототип бортовой цифровой вычислительной платформы „Крейт-6U“, разработанной в ФГУП «СПб ОКБ „Электроавтоматика“ им. П. А. Ефимова»

В архитектуре БЦВС структуры ИМА средствами модулей $МВ_i, i=1, 2, \dots, n$ предусматриваются три типа препроцессорной обработки [4]:

- форматирование с разделением каналов, когда ПФ устанавливает биективное соответствие между каналами связи с абонентами комплекса и каналами шины данных вычислителей $МВ_i, i=1, 2, \dots, n$;
- совмещение каналов, когда соответствия между каналами обменов и каналами шины данных вычислителей $МВ_i, i=1, 2, \dots, n$ являются произвольными (каждый с каждым, т.е. реализуется свойство отказоустойчивости – БЦВС остается работоспособной при наличии хотя бы одного исправного ресурса каждого типа);

– форматирование данных в условиях биномиального резервирования каналов обмена (при этом препроцессор форматирования устанавливает соответствие между каналами абонентов комплекса и каналами шины данных вычислителя $MB_i, i=1, 2, \dots, n$ с учетом введенной аппаратурной избыточности, обусловленной резервированием).

В соответствии с принятыми [4] способами препроцессорной обработки вычислители $MB_i, i=1, 2, \dots, n$ реализуют функции:

- препроцессорной обработки Θ средствами препроцессора форматирования ПФ;
- промежуточного хранения данных Ω (буферизацию), осуществляемого блоком памяти БП;
- обмена I с системным интерфейсом СИ с помощью процессора ПО или вычислительного узла ВУ;
- обработки P на вычислительном узле;
- управления U обменом. При централизованном управлении обменом функция U реализуется ВУ, а процессор обмена ПО отсутствует.

Соответствующие этим функциям архитектурные признаки БЦВС структуры ИМА являются наиболее значимыми и имеют вид $\Theta = \{\theta_1, \theta_2, \theta_3\}$, где θ_1 – разделение, θ_2 – совместное использование, θ_3 – биномиальное резервирование каналов обмена; $\Omega = \{\omega_1, \omega_2\}$, где ω_1 – наличие, ω_2 – отсутствие буферизации данных; $I = \{i_1, i_2\}$, где i_1 – пословный обмен, i_2 – обмен блоками слов в системном интерфейсе; $P = \{p_1, p_2, p_3, p_4\}$, где p_1 – обработка слов сразу после форматирования, p_2, p_3 – обработка перед буферизацией и после считывания данных из буфера обмена (блока памяти) соответственно, p_4 – обработка до и после буферизации; $U = \{u_1, u_2\}$, где u_1 и u_2 обозначают соответственно централизованное и децентрализованное управление.

Правила синтеза описаний $\Lambda = \{\Theta_{j_1}, \Omega_{j_2}, I_{j_3}, P_{j_4}, U_{j_5}\}$ формируются на основе известных знаний [4] о предметной области (назначения БЦВС) и исходя из допустимых принципов сочетаний вычислительных ресурсов в заданном наборе архитектурных признаков. Подмножество разрешенных частных описаний вычислителей $MB_i, i=1, 2, \dots, n$ и их композиции в составе БЦВС определяется набором допустимых сочетаний $\Lambda_1 = \{\theta_1, \omega_1, i_1, p_1, u_1\}$; $\Lambda_2 = \{\theta_j, \omega_1, i_1, p_2, u_1\}$, $j = 1, 2, 3$; $\Lambda_3 = \{\theta_j, \omega_1, i_1, p_3, u_1\}$, $j = 1, 2, 3$; $\Lambda_4 = \{\theta_j, \omega_1, i_1, p_4, u_1\}$, $j = 1, 2, 3$; $\Lambda_5 = \{\theta_j, \omega_1, i_2, p_1, u_1\}$, $j = 1, 2, 3$; $\Lambda_6 = \{\theta_j, \omega_1, i_2, p_3, u_2\}$, $j = 1, 2, 3$; $\Lambda_7 = \{\theta_j, \omega_1, i_2, p_4, u_2\}$, $j = 1, 2, 3$; $\Lambda_8 = \{\theta_j, \omega_2, i_1, p_1, u_1\}$, $j = 2, 3$. Сами по себе правила формирования описаний не формализуют конструктивного приема порождения вариантов (проектных альтернатив) архитектуры БЦВС. Эти правила лишь допускают либо запрещают определенные комбинации архитектурных признаков базиса модулей $MB_i, i=1, 2, \dots, n$. Сама же процедура синтеза связана с отображением операционной модели БЦВС на ее целевую структуру.

Отображение операционной модели БЦВС на доступные вычислительные ресурсы

Сущность метода отображения операционной модели на доступные вычислительные ресурсы заключается в следующем [5]. Пусть имеется n информационно связанных задач, которые необходимо решать с помощью крейта ИМА, и соответствующие 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 алгоритма ставится в соответствие вершина графа v_i , возле которой записывается ее вес – время выполнения данного функционального оператора в относительных единицах. Вершины v_i и v_j соединяются линией со стрелкой (дугой графа), направленной в v_i только в том случае, если результат, полученный после выполнения v_j , является одним из аргументов для v_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 крейта ИМА состоит в выборе числа вычислителей (модулей-вычислителей MB) и в определении электрических (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, начальные, а вершины, из которых выходят дуги результата решения задач или фрагментов вычисления, – конечными (так, на рис. 3 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 опера-

торов, не входящих в этот подграф, и, следовательно, каждый из них можно реализовать на отдельном МВ, обменивающимся информацией по «медленному» интерфейсу. Очевидно, число задействованных вычислителей МВ будет зависеть от числа независимых подграфов. Поскольку все вычислители будут работать параллельно, время выполнения всех n задач t_p при такой структуре будет минимальным, т.е. дальнейшее увеличение числа процессоров не уменьшит t_p .

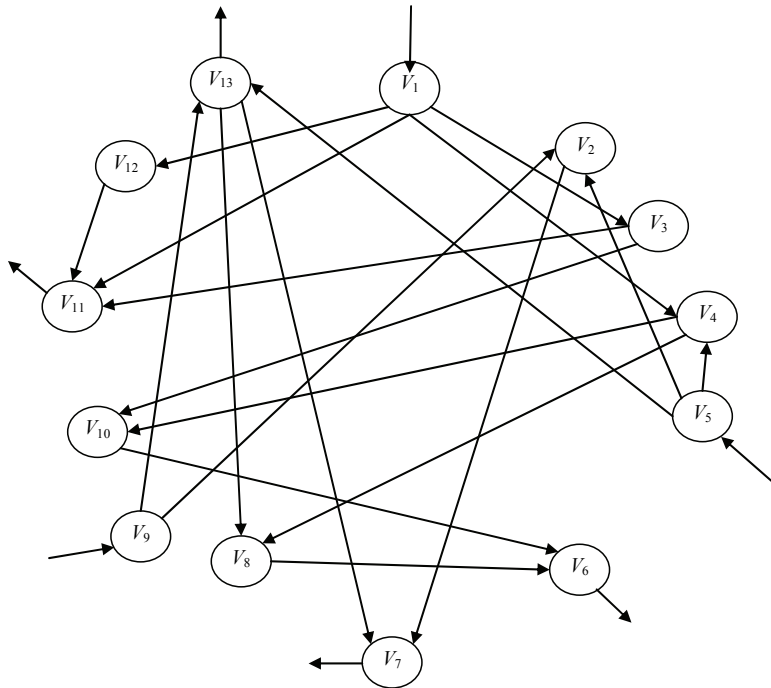


Рис. 3. Операционная модель G_{BC} вычислительной системы в виде многосвязного графа операций обмена и обработки информации

Рассмотрим основные положения, на которых строится алгоритм декомпозиции графа на независимые подграфы. Если существует множество вершин $v_{i1}, v_{i2}, \dots, v_{ik}$, 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 -го 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 E .

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 E .

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 E , 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 реализуется на отдельном МВ за время t_{pi} , то $t_p = \max\{t_{p1}, t_{p2}, \dots, t_{pN}\}$. Если имеются ограничения вида $\sum_{j=1}^l t_{pj} \leq t_p$, 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 многомашиной вычислительной системы дает оптимум по критерию минимума числа вычислителей МВ, задействованных в составе крейта БЦВС ИМА.

Применение алгоритма функционального разделения графа, показанного на рис. 3, позволяет получить три его независимых подграфа (см. рис. 4), каждый из которых соответствует своему алгоритму, исполняемому на отдельном МВ.

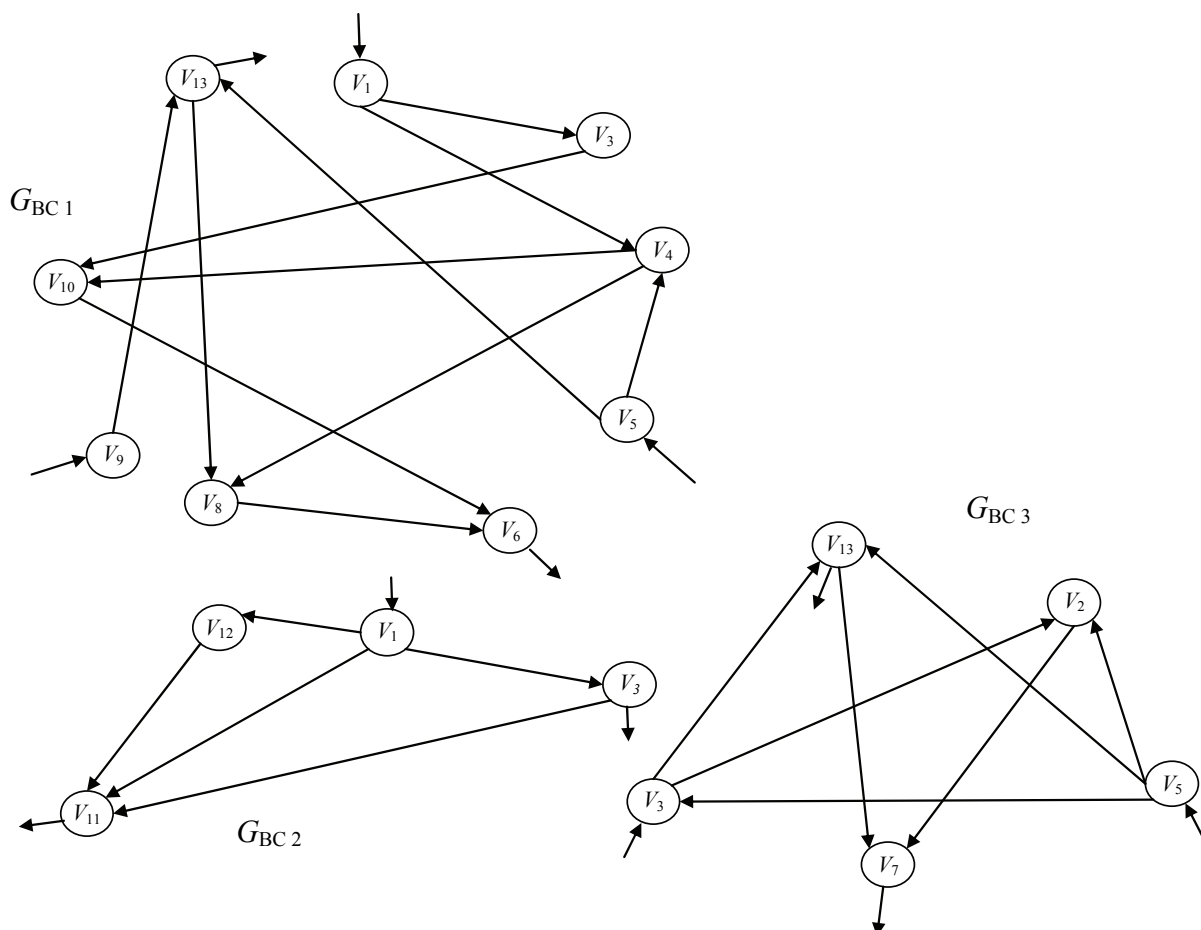


Рис. 4. Независимые подграфы $G_{BC 1}$, $G_{BC 2}$, $G_{BC 3}$ многосвязного графа G_{BC}

Заключение

В согласии с результатами работы [4] можно сделать вывод, что наиболее экономичным и предпочтительным по реализации мультипроцессорной БЦВС в условиях компромисса в пространстве состояний ($|HW|$, $|SW|$, T), где $|HW|$ – затраты на реализацию аппаратной платформы модулей МВ, $|SW|$ – сложность программного кода, T – ограничение на время выполнения бортовой задачи в вычислительной системе соответствующей архитектуры Λ_i , является вариант многопроцессорной системы «Крейт-6U» (см. рис. 2) с архитектурными признаками $\Lambda_3 = \{\theta_1, \omega_1, i_1, p_3, u_1\}$ – вычислители типа МВ80, модуль памяти МП80, разработанные в ФГУП «СПб ОКБ „Электроавтоматика“ им. П.А. Ефимова» в рамках реализации концепции ИМА бортовой авионики.

Рассмотренные в работе принципы построения БЦВС в соответствии с основными положениями концепции ИМА и методика оценки сложности вычислительного алгоритма БЦВС были применены также к анализу комплектов документации бортовых комплексов (систем) К-130, СОИ-У-25-1, СОИ-У-25-2, ССИ-80. Расчеты показывают, что внедрение структур ИМА на базе унифицированных конструктивно-функциональных модулей стандарта 6U с крейтово-модульным конструктивом БЦВС для решения бортовых задач указанных комплексов (систем) приводит к экономии материальных средств аппаратного обеспечения от 19% до 48%.

Литература

1. Павлов А.М. Принцип организации бортовых вычислительных систем перспективных летательных аппаратов // Мир компьютерной автоматизации. – 2001. – № 4 [Электронный ресурс]. – Режим доступа: www.mka.ru/?p=41177, открытый.
2. Моисеев Н.Н. Математические задачи системного анализа. – М.: Наука, Главная редакция физико-математической литературы, 1981. – 488 с.
3. Ивченко В. Д., Корнеев А. А. Анализ методов распределения заданий в задачах управления коллективом роботов // Мехатроника, Автоматизация, Управление. – 2009. – № 7. – С. 36–42.
4. Топорков В. В. Модели распределенных вычислений. – М.: ФизМатЛит, 2004. – 320 с.
5. Копорский Н. С., Видин Б. В., Жаринов И. О. Организация вычислительного процесса в многома-шинном бортовом вычислительном комплексе // Известия вузов. Приборостроение. – 2006. – Т. 49. – № 6. – С. 41–50.

Жаринов Олег Олегович

– Санкт-Петербургский государственный университет аэрокосмического приборостроения, кандидат технических наук, доцент, zharinov@hotmail.ru

Видин Борис Викторович

– ФГУП «СПб ОКБ “Электроавтоматика” имени П. А. Ефимова», кандидат технических наук, зам. главного конструктора, postmaster@elavt.spb.ru

Шек-Иовсепянц Рубен Ашотович

– ФГУП «СПб ОКБ “Электроавтоматика” имени П. А. Ефимова», доктор технических наук, профессор, главный конструктор, postmaster@elavt.spb.ru