УДК 681.324 : 621.3.049.77.001.2 : 519.2

ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ ОПТИМАЛЬНОГО ОБМЕНА ДАННЫМИ В МУЛЬТИПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

О.И. Богатов[1], Ю.С. Литвинов[1], Г.И. Стопченко[2], В.О. Богатов[2]
([1]Объединенный научно-исследовательский институт Вооруженных Сил,
[2]Харьковский Национальный университет радиоэлектроники)

Предложена методика организации оптимального по времени обмена, в соответствии с которой разработан алгоритм и создан комплект программ для нахождения оптимальных последовательностей обмена между процессорными элементами и моделирования их реализации в мультипроцессорной вычислительной системе (МВС).

мультипроцессорная вычислительная система, процессорный элемент, моделирование, оптимальный обмен данными, гиперкуб, квадратная перестановочная матрица

Постановка проблемы. При организации мультипроцессорных вычислительных систем (МВС) с большим количеством процессорных элементов (ПЭ) возникает одна проблема, когда практически невозможно обеспечить непосредственные связи между всеми ПЭ, а последовательное соединение ПЭ друг с другом в устройствах типа сдвиговых регистров приводит к недопустимо большому времени реализации. Поэтому становится актуальной задача организации регулярной сети обмена между ПЭ, сравнительно просто реализуемой и оптимальной по времени реализации.

Анализ литературы. Структура связей между ПЭ в МВС типа “n-мерный гиперкуб” считается одной из наиболее удобных для реализации параллельных алгоритмов, эффективные примеры конкретных задач приведены в [1]. Поэтому рассматривается вопрос организации процесса оптимального по времени обмена данными между ПЭ в МВС, взаимные соединения в которой организованы по схеме “n-мерный гиперкуб”.

Постановка задачи. Рассматривается вопрос организации процесса оптимального по времени обмена данными между ПЭ в МВС, взаимные соединения в которой организованы по схеме “n-мерный гиперкуб”. Критерий оптимальности, в соответствии с которым решается задача нахождения последовательности обменов между соответствующими ПЭ, определен следующим образом.

Пусть MZ – множество всех биективных отображений $f:\{0,1,\dots,N-1\}\to\{0,1,\dots,N-1\}$ для $N=2^n$. Поставим во взаимно однозначное соответствие каждой вершине некоторого n-мерного гиперкуба, соответствующей ПЭ в МВС, число $i\in\{0,1,\dots,N-1\}$ в качестве ее номера. Согласно определению гиперкуба обмен данными между двумя ПЭ возможен лишь тогда, когда номера соответствующих вершин отличаются только одним двоичным разрядом (в дальнейшем их будем именовать смежными). Предполагается, что соответствующие процессорные элементы имеют тот же номер.

Будем говорить, что две смежные вершины (соответственно ПЭ) с номерами i и j имеют k-й индекс смежности, если имеет место

$$|i-j|=2^k, \quad 0\le k\le n-1. \tag{1}$$

В настоящем статье рассматриваются процессы обмена с определенными ограничениями, сущность которых заключается в том, что одновременный обмен данными допускается лишь для ПЭ, имеющих одинаковый индекс смежности.

Для каждого отображения $f\in MZ$ реализацией $RZ(f)$ его на n-мерном гиперкубе будем называть некоторую последовательность взаимных обменов данными между парами его ПЭ (с учетом указанных выше ограничений), в результате которых в каждом ПЭ с номером i будут находиться данные, бывшие ранее в ПЭ с номером $f(i)$. Обозначим RZ множество всех таких реализаций, которые ставят в однозначное соответствие каждому $f\in MZ$ определенную реализацию $RZ(f)$ на n-мерном гиперкубе. В этом случае RZ можно рассматривать как некоторый алгоритм, согласно которому строится каждая реализация $RZ(f)$ для всех $f\in MZ$. Обозначим также через W множество всех таких алгоритмов RZ.

Пусть $T(RZ(f))$ – время, необходимое для реализации $RZ(f)$ отображения f на гиперкубе. В качестве критерия оптимальности по времени реализации примем

$$T_{опт} = \min_{RZ\in W}\ \max_{f\in MZ} T(RZ(f)), \tag{2}$$

согласно которому для каждого $f\in MZ$, при оптимальной реализации $T(RZ(f))\le T_{опт}$, существует такое f, для которого $T(RZ(f))=T_{опт}$, и для реализации которого дальнейшее уменьшение $T_{опт}$ уже невозможно. Примем время параллельного обмена между смежными ПЭ за условную единицу. Тогда можно показать, что существует алгоритм $RZ_{опт}$, обеспечивающий выполнение (2), причем $T_{опт}=2n-1$.

Математические основы организации обмена данными в МВС. При такой постановке задачи удобным математическим аппаратом для исследования и анализа процессов обмена оказывается теория групп [2], благодаря чему был обоснован алгоритм нахождения оптимальной реализации, положенный далее в основу соответствующего пакета программ.

В соответствии с этим для всех $1 \le i \le N/2$ и $1 \le j \le n$ обозначим

$$q_{ij} = \left(\left[\frac{i-1}{2^{j-1}} \right] \cdot 2^j + (i-1) \bmod 2^{j-1} + 1 \quad \left[\frac{i-1}{2^{j-1}} \right] \cdot 2^j + 2^{j-1} + (i-1) \bmod 2^{j-1} + 1 \right)$$

все транспозиции, соответствующие взаимному обмену данными между любыми двумя смежными вершинами гиперкуба, где [a] – целая часть числа a. (Отсюда следует, что $j-1$ здесь является индексом смежности для соответствующей пары ПЭ, а i – порядковым номером в множестве транспозиций с указанным выше индексом смежности).

Введем далее для всех $1 \le j \le n$ обозначение

$$Q_j(\alpha_1, \alpha_2, \ldots \alpha_{N/2}) = q_{1j}^{\alpha_1} \cdot q_{2j}^{\alpha_2} \cdots q_{N/2,j}^{\alpha_{T/2}}, \qquad (3)$$

где $\alpha_k \in \{0,1\}$ при $1 \le k \le N/2$, а $(ij)^1 = (ij), (ij)^0 = e$, где e обозначена единица симметрической группы. Естественно, что (3) определяет подстановки, соответствующие всем допустимым одновременным взаимным обменам данными на гиперкубе. Отсюда следует, что подстановки типа (3) могут быть приняты в качестве порождающих элементов симметрической группы S_N, каждый из которых соответствует одновременному взаимному обмену данными между ПЭ с одинаковым индексом смежности.

Основной результат настоящей работы, представленный в терминах подстановок, состоит в том, что любая подстановка $\sigma \in S_N$ (соответственно любой обмен данными между ПЭ) может быть представлена в виде

$$\sigma = Q_1(\beta_1) \cdot \ldots \cdot Q_{n-1}(\beta_{n-1}) \cdot Q_n(\alpha_n) \cdot Q_{n-1}(\alpha_{n-1}) \cdot \ldots \cdot Q_1(\alpha_1) \qquad (4)$$

для соответствующих векторов:

$\alpha_j = (\alpha_{1j}\, \alpha_{2j} \ldots \alpha_{N/2,j})$ для $1 \le j \le n$; $\beta_j = (\beta_{1j}\, \beta_{2j} \ldots \beta_{N/2,j})$ для $1 \le j \le n-1$,

где $\alpha_{ij}, \beta_{ij} \in \{0,1\}$, т.е.

$$Q_j(\alpha_j) = Q_j(\alpha_{1j}, \alpha_{2j}, \ldots, \alpha_{N/2,j}); \quad Q_j(\beta_j) = Q_j(\beta_{1j}, \beta_{2j}, \ldots, \beta_{N/2,j}).$$

Поскольку преобразования в группах с порождающими элементами типа подстановок $Q_j(\alpha_1, \alpha_2, \ldots, \alpha_{N/2})$ с неизвестными заранее значениями $\alpha_i \in \{0,1\}$ в теории симметрических групп не рассматривались, то целесообразно перейти к изоморфным им группам перестановочных матриц, для которых соответствующие преобразования можно заменить обычными операциями умножения матриц. Изоморфизм этот устанавливается заданием для каждой подстановки $Q_t(\alpha_1, \alpha_2, \dots, \alpha_{N/2})$ перестановочной N×N-матрицы $A(n,\alpha)$, где вектор $\alpha = (\alpha_1\ \alpha_2 \dots \alpha_{N/2})$. Каждый элемент a_{ij} этой матрицы определяется как

$$a_{ij} = \begin{cases} \overline{\alpha}_k, & i = j; \\ \alpha_k, & |i-j| = 2^{t-1}; \\ 0 & \text{в остальных случаях}, \end{cases}$$

где $k = \left[\frac{i-1}{2^t}\right] \cdot 2^{t-1} + (i-1) \bmod 2^{t-1} + 1$; $\overline{\gamma}$ – логическое отрицание (инверсия) двузначной переменной γ.

Тогда соответствующий (4) результат может быть представлен в виде перестановочной N×N-матрицы $\mathbf{Y}$:

$$Y = A_1(n,\beta_1) \times A_2(n,\beta_2) \times \ldots \times A_{n-1}(n,\beta_{n-1}) \times A_n(n,\alpha_n) \times$$
$$\times A_{n-1}(n,\alpha_{n-1}) \times \ldots \times A_2(n,\alpha_2) \times A_1(n,\alpha_1)\,. \qquad (5)$$

Алгоритм оптимального обмена данными в МВС. Возможность представления (5) основывается на том, что для любой перестановочной N×N-матрицы G имеет место

$$G = A_1(n,\beta) \times (U((n-1,1,0,F) + U(n-1,1,1,H)) \times A_1(n,\alpha)\,, \qquad (6)$$

где F и H – соответствующие перестановочные N/2×N/2-матрицы; α и β – некоторые N/2-мерные векторы с двоичными компонентами; а $U(n,k,s,M)$ для любой $2^n \times 2^n$-матрицы $M = \left\| m_{ij} \right\|$ определяется как $2^{n+k} \times 2^{n+k}$-матрица $\left\| u_{pq} \right\|$, каждый элемент которой равен

$$u_{pq} = \begin{cases} m_{ij}, & p = (i-1) \cdot 2^k + s + 1 \ \text{ и } \ q = (j-1) \cdot 2^k + s + 1; \\ 0 & \text{в остальных случаях}. \end{cases}$$

Нетрудно убедиться, что матрица, представленная в (6) выражением $(U((n-1,1,0,F) + U(n-1,1,1,H))$, позволяет однозначно выделить из нее матрицы F и H, имеющие размерность $N/2$. Поэтому разрешимость (6) при заданном значении матрицы G относительно матриц F и H, а также векторов α и β, позволит, кроме определения значения векторов, свести задачу размерности N к двум отдельным задачам такого же типа, но имеющим вдвое меньшую размерность. то дает возможность применить достаточно эффективный в вычислительной практике прием, известный под именем “разделяй и властвуй” [3], что в конечном итоге позволит свести задачу к $N/2$ задачам размерности 2, решение которых достаточно тривиально. По ходу такого решения будут также определяться и все векторы α и β из (5), что и является нашей основной задачей.

Рассмотрим далее некоторые особенности решения (6) относительно F, H, α и β. Естественно, должны быть принципиально решены две задачи: 1) обосновать, что уравнение (6) разрешимо для любых значений матрицы G; 2) найти алгоритм нахождения F, H, α и β для любого значения матрицы G, в конечном итоге обеспечивающий оптимальное решение согласно критерия (2). Поскольку как элементы матриц F и H, так и компоненты векторов α и β принимают значения из множества $\{0, 1\}$, то естественным математическим аппаратом для решения (6) является булева алгебра. Будем представлять матрицу G в виде

$$G = \begin{pmatrix} E_{11} & E_{12} & \cdots & E_{1,N/2} \\ E_{21} & E_{22} & \cdots & E_{2,N/2} \\ \cdots & \cdots & \cdots & \cdots \\ E_{N/2,1} & E_{N/2,2} & \cdots & E_{N/2,N/2} \end{pmatrix}, \tag{7}$$

где
$$E_{ij} = \begin{pmatrix} \overline{\alpha}_j\overline{\beta}_i f_{ij} \vee \alpha_j\beta_i h_{ij} & \alpha_j\overline{\beta}_i f_{ij} \vee \overline{\alpha}_j\beta_i h_{ij} \\ \overline{\alpha}_j\beta_i f_{ij} \vee \alpha_j\overline{\beta}_i h_{ij} & \overline{\alpha}_j\overline{\beta}_i f_{ij} \vee \alpha_j\beta_i h_{ij} \end{pmatrix}; \tag{8}$$

f_{ij} и h_{ij} – соответствующие элементы матриц F и H. (Следует заметить, что дизъюнкция в выражениях для каждого элемента матрицы E_{ij} в данном случае идентична обычному арифметическому сложению).

Тогда решение уравнения (6) относительно F, H, α и β будет сводиться к решению системы $4 \cdot \left(\frac{N}{2}\right)^2 = N^2$ булевых уравнений, полученных из (7) и (8). При традиционном подходе в случае выполнение конъюнкции в символьном виде из-за ее экспоненциальной сложности приводит к настолько громоздким выражениям, что практически их невозможно реализовать для нетривиальных случаев. Для эффективного решения такой задачи следует воспользоваться методом сопровождающих функций [4], в данном случае представленными функциями трехзначной логики, который, с одной стороны, позволяет показать, что (6) всегда разрешимо и, с другой стороны, дает возможность в общем виде найти некоторую последовательность определения всех ненулевых элементов матриц F, H и всех значений компонент векторов α и β соответственно.

Алгоритм сводится к последовательному просмотру единичных элементов матрицы G в следующем порядке. Пусть ранее на r-м шаге ($r \ge 1$) алгоритма рассматривался элемент $g_{ij} = 1$ (для определенности примем сначала, что $r - 1 \equiv 0\,(\mathrm{mod}\,2)$). Тогда индексы следующего единичного элемента g_{kl} , подлежащего просмотру на (r +1)-м шаге, определяются как

$$l = \left[\frac{j-1}{2}\right] \cdot 2 + \overline{(j-1)\bmod 2} + 1; \quad k = g^{-1}(l-1) + 1 .$$

(Предполагается, что для всех i, j $g_{ij} = 1$ эквивалентно $j - 1 = g(i-1)$). При этом индексы соответствующих единичных элементов f_{st} и h_{uv} матриц F и H соответственно будут равны:

$$s = \left[\frac{i-1}{2}\right] + 1; \quad t = \left[\frac{j-1}{2}\right] + 1; \quad u = \left[\frac{k-1}{2}\right] + 1; \quad v = \left[\frac{l-1}{2}\right] + 1 .$$

Соответственно определяются компоненты векторов:

$$\beta_s = \begin{cases} (i-1) \bmod 2, & r \equiv 1 (\mathrm{mod}\, 2); \\ \overline{(k-1) \bmod 2}, & r \equiv 0 (\mathrm{mod}\, 2); \end{cases} \qquad \alpha_t = \begin{cases} (j-1) \bmod 2, & r \equiv 1 (\mathrm{mod}\, 2); \\ \overline{(l-1) \bmod 2}, & r \equiv 0 (\mathrm{mod}\, 2). \end{cases}$$

Переход к очередному единичному значению $g_{pq} = 1$ на следующем (r + 2)-м шаге осуществляется для следующих значений индексов

$$p = \left[\frac{k-1}{2}\right] \cdot 2 + \overline{(k-1)\bmod 2} + 1; \quad q = g(p-1) + 1 ,$$

после чего процесс повторяется сначала до тех пор, пока не окажется $p = i$, где i – значение индекса первого единичного элемента матрицы G , с которого начинался данный процесс. Если при этом были перебраны не все единичные элементы матрицы G , то процесс повторяется с новыми значениями индекса i , значение которого выбирается как наименьшее из числа индексов всех невыбранных ранее единичных элементов. Построение алгоритма позволяет гарантировать, что такие явления, как зацикливание, при этом исключаются, и что алгоритм в любом случае приводит к решению (6).

Оптимальность по времени в соответствии с принятым критерием обосновывается следующим образом. Рассматривается конкретный пример обмена, которому соответствует некоторый элемент из симметрической группы S_N , для которого можно показать, что представление его в терминах порождающих элементов (3) не может быть выполнено менее чем за $2n - 1$ последовательных применений подобных подстановок. С другой стороны, из представления (5) следует, что реализация любого примера может быть выполнена за время, не большее, чем $2n-1$ шаг, откуда и следует оптимальность (5) в соответствии с принятым критерием (2). (Время, меньшее времени $2n-1$ шагов, получается в том случае, когда для каких-либо j имеет место $A_j(n,\alpha_j)=I_N$ или $A_j(n,\beta_j)=I_N$, где I_N – единичная матрица порядка N).

Моделирование оптимального обмена данными в пакете Hyper. Рассмотрим далее структуру пакета Hyper программ формирования и моделирования временных последовательностей оптимального по времени обмена данными между ПЭ. Кроме программ моделирования процесса обмена, в пакет включены также средств автоматического формирования поведенческой модели МВС во время обмена на языке VHDL (Very High Speed Integrated Circuit Hardware Description Language) [5], разработанного в США по заказу Министерства обороны и с 1987 года ставшего стандартом IEEE Std 1076, а в настоящее время ставшего по существу международным стандартом. (VHDL позволяет описывать цифровые схемы с любой степенью детализации и практически любой сложности, начиная с простейших вентилей и кончая самыми мощными современными вычислительными комплексами).

Исходными данными для пакета является информация о необходимом перераспределении данных между ПЭ, заключенных в файле Readr, пример которого приведен на рис. 1.

```
n = 3
PR0 <= PR2     PR4 <= PR1     PR1 <= PR3     PR5 <= PR7
PR2 <= PR5     PR7<=PR4       PR3 <= PR0
```

Рис. 1. Файл **Readr**

В первой строке содержится информация о величине n, определяющей размерность гиперкуба. В последующих строках находятся данные о необходимом перераспределении информации между ПЭ (обозначенных как PR i для $0 \le i \le 2^n - 1$, где i – номер ПЭ). В результате чего, данные из правой части строк всех ПЭ должны перейти в соответствующие ПЭ в левой части. Те ПЭ, в которых данные не изменяются, в файл не включаются. В исходные данные входит также файл Read_vhd, содержащий информацию, необходимую для генерирования модели на языке VHDL, которая будет рассмотрена несколько позже.

Результатом работы пакета является информация об организации последовательности обменов в системе “гиперкуб” (файлы Abps и Mod1) и соответствующая VHDL-модель.

Для моделирования процессов обмена должна быть выбрана соответствующая модель МВС хотя бы на поведенческом уровне. В настоящей статье принят следующий вариант.

МВС, организованная по типу "гиперкуб", содержит $N = 2^n$ ПЭ с номерами от 0 до $2^n - 1$. Каждый ПЭ с номером i (в дальнейшем именуемый как ПЭ i) может обмениваться данными с n другими смежными ПЭ j, для которых имеет место (1), при этом в процессе обмена ПЭ i в течение такта работы МВС получает данные из ПЭ j и наоборот, в ПЭ j оказываются данные, ранее находившиеся в ПЭ i.

После реконфигурирования МВС (в нашем случае настройки МВС на выполнение необходимого перераспределения данных между ПЭ) в каждом ПЭ i запоминается следующая информация, служащая для выработки сигналов управления процессом перераспределения информации в МВС: управляющий вектор $CPR\,i$, индивидуальный для каждого ПЭ, и два вектора REZ и VAC, общие для всех ПЭ. Все эти данные, необходимые для реконфигурирования МВС, содержатся в файле Abps (пример файла Abps для приведенного выше исходного файла Readr приведен на рис. 2).

	n = 3	
	REZ := 0 0 0 1	
CPR0 := 1 1 0 0	CPR4 := 1 1 0 1	CPR1 := 1 1 0 0
CPR5 := 1 1 0 0	CPR2 := 0 1 1 0	CPR6 := 0 1 1 1
CPR3 := 0 1 0 0	CPR7 := 0 1 0 0	

Рис. 2. Файл **Abps**

Каждый вектор $CPR\,i$ содержит некоторое количество бит (не более $2n - 1$), соответствующее числу тактов работы МВС, наличие "1" в некотором разряде которого свидетельствует о том, что в соответствующем такте происходит взаимный обмен данными ПЭ i с каким-либо смежным ПЭ j, и "0" – об отсутствии любых обменов с данным ПЭ i. Вектор REZ (признак результата) содержит такое же количество бит, как и $CPR\,i$, в том числе и единственный единичный бит в последнем разряде, свидетельствующий о том, что соответствующий ему такт является последним, а также, что в течении этого такта необходимое перераспределение данными между ПЭ будет произведено. И, наконец, вектор VAC для каждого такта содержит информацию о значении k из (1) в двоичном коде, т.е. косвенным образом задает информацию, какие именно ПЭ из числа отмеченных единичными битами во всех векторах $\mathrm{CPR}\,i$ должны обмениваться данными друг с другом в текущем такте. Поскольку из (1) следует, что максимальное значение k равно $n-1$, то на каждом такте требуется $[\log_2(n-1)]+1$ двоичных разрядов для представления k, т.е. при числе тактов, равной m, общая длина L вектора VAC составит

$$L = m\cdot([\log_2(n-1)]+1) \le (2n-1)\cdot([\log_2(n-1)]+1)$$

двоичных разрядов. Описанная выше поведенческая модель МВС совместно с файлом Abps является исходной информацией для моделирования ее работы в рамках пакета Hyper. Результаты моделирования заключены в файле $\mathrm{Mod1}$, пример которого в соответствии с рассмотренными здесь исходными данными, приведен на рис. 3.

```
MODELING OF TRANSFORMATION ON HYPERCUBE
(dimension N = 8 )
PR0 <= PR2 after DEL_OUT        PR4 <= PR1 after DEL_OUT
PR1 <= PR3 after DEL_OUT        PR5 <= PR7 after DEL_OUT
PR2 <= PR5 after DEL_OUT        PR7 <= PR4 after DEL_OUT
PR3 <= PR0 after DEL_OUT
Realization
STEP 1                          STEP 2
PR0 <= PR1 after DEL            PR0 <= PR2 after DEL
PR1 <= PR0 after DEL            PR1 <= PR3 after DEL
PR4 <= PR5 after DEL            PR2 <= PR0 after DEL
PR5 <= PR4 after DEL            PR3 <= PR1 after DEL
STEP 3                          PR4 <= PR6 after DEL
PR2 <= PR6 after DEL            PR5 <= PR7 after DEL
PR6 <= PR2 after DEL            PR6 <= PR4 after DEL
STEP 4                          PR7 <= PR5 after DEL
PR4 <= PR6 after DEL
PR6 <= PR4 after DEL
DEL_OUT = 4 * DEL
```

Рис. 3. Файл **Mod1**

Файл состоит из двух частей. Первая часть повторяет файл $\mathrm{R\ eadr}$, дополненный информацией о временной задержке $\mathrm{DEL\_OUT}$, спустя которую достигается необходимое перераспределение данных, и значение которой приводится в виде формулы в конце файла. Вторая часть состоит из ряда шагов $\mathrm{STEP1}, \mathrm{STEP2}, \ldots$, общим числом не более $2n-1$, в каждом из которых в течении одного такта работы МВС, определенного соответствующей задержкой DEL, происходит взаимный обмен данными между смежными ПЭ, указанными в текстовой части каждого шага.

Формирование VHDL-модели процесса обмена данными в пакете Hyper. Для генерации программы моделирования на языке VHDL модель МВС должна быть более детализирована. Так, конкретная длительность такта работы МВС задается в исходном файле Read_vhd величиной Tp, представленной в численной форме в общепринятых единицах времени. Непосредственно такт задается сигналами CLOCK, идущими с периодом Tp, каждый из полупериодов которого, может быть представлен соответственно нулевым и единичным сигналом. Началом работы МВС считается сигнал INSET предварительной установки, после которого можно считать, что во всех ПЭ находятся данные, подлежащие перераспределению. Предполагается, что в начале работы МВС сигнал INSET в течении первого полупериода находится в единичном состоянии, в то время как сигнал CLOCK будет находиться в нулевом. В процессе моделирования предполагается, что в это время происходит занесение исходных данных во все ПЭ. Для определенности здесь принято, в каждый ПЭ i для всех $0 \le i \le 2^n - 1$ заносится код, равный i соответственно. Соответствующий файл Read_vhd приведен на рис. 4.

Name Tst_Pr	Tp 100 ns

Рис. 4. Файл **Read_vhd**

Генерируемая моделирующая программа на языке VHDL представлена двумя файлами: Net n.vhd (в нашем случае файлом Net 3.vhd) и файлом, имя которого (без расширения) в соответствии с идентификатором Name указано в файле Read_vhd (т.е. в нашем случае файлом Tst_Pr.vhd). Файл Net n.vhd представляет собой программу на языке VHDL, описывающую структуру МВС с указанной выше степенью детализации. Естественно, что для всех МВС с одним и тем же значением n эти программы будут одинаковыми. Файл Tst_Pr.vhd представляет программу на языке VHDL, описывающую сам процесс перераспределения данных во времени в МВС, структура, которой описывается в файле Net n.vhd (в нашем случае файлом Net 3.vhd), в соответствии с исходными данными, содержащимися в файле Readr. В процессе моделирования в программе Tst_Pr.vhd в качестве исходной информации используются данные, содержащиеся в файле Abps. Однако здесь вместо векторов $CPRi$ для $0 \le i \le 2^n - 1$ используются 2^n-компонентные векторы CPA(j) того же назначения, где j представляет собой номер такта, что соответствует параллельной обработке данных для всех ПЭ в каждом текущем такте. Дополнительно ко всему пакет генерирует файл, имя которого также формируется на основе данных, хранящихся под идентификатором Name (в нашем случае – файл $Tst\_Pr\_tb\_run.do$). Этот файл используется при работе в оболочке Active-VHDL vers. 3.2 и содержит в себе все данные, необходимые для моделирования на языке VHDL и формирования временных диаграмм работы МВС.

Выводы. Для сетей обмена данными типа "гиперкуб" была предложена методика организации оптимального по времени обмена, в соответствии с которой был разработан алгоритм и создан комплект программ для нахождения оптимальных последовательностей обмена между процессорными элементами (ПЭ) и моделирования их реализации в мультипроцессорной вычислительной системе (МВС).

Для достоверности все результаты, как при непосредственном моделировании в пакете Hyper, так и при формировании соответствующих программ на языке VHDL контролируются путем обратного преобразования в исходные данные с последующим сравнением. Пакет разработан на языке $C++$. Для более тщательного контроля соединений ПЭ в МВС в пакете предусмотрена опция автоматического формирования исходных данных, основанная на генерации псевдослучайных чисел по заданию пользователя.

Пакет позволяет формировать и контролировать временные последовательности обменов в МВС, содержащей до 8192 ПЭ, а также формировать и контролировать соответствующие VHDL-модели в МВС, содержащей до 2048 ПЭ.

ЛИТЕРАТУРА

1. *Foster I. Designing and Building Parallel Programs. Addison-Wesley, 1995. – 126 p.*
2. *Курош А.Г. Теория групп. – М.: Наука, 1967. – 648 с.*
3. *Кун С. Матричные процессоры на СБИС. – М.: Мир, 1991. – 672 с.*
4. *Кондратьев В.Н., Кравченко В.Ф., Рвачев В.Л. О преобразовании сложных регулярных логических схем // ДАН СССР. – Т. 312, № 4. – 1990. – С. 808 – 814.*
5. *Армстронг Дж. Р. Моделирование цифровых систем на языке VHDL. – М: Мир, 1992. – 176 с.*

Поступила 19.03.2005

Рецензент: доктор технических наук профессор Е.И. Бобыр,
Харьковский гуманитарный университет «Народная украинская академия».