АКАДЕМИЯ НАУК СССР
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 1 1988

УДК 681.142.2

М. А. БЕРЕЗОВСКИЙ, А. Л. МИНКИН
(Москва)

РАЗВИТИЕ КООРДИНАТНОГО МЕТОДА РАСПАРАЛЛЕЛИВАНИЯ DO-ЦИКЛОВ

В последнее десятилетие все большее распространение получают параллельные ЭВМ. Использование их наиболее эффективно, если программы прямо учитывают особенности архитектуры компьютера. Основная тяжесть работы по соответствующему преобразованию пользовательских программ при современном подходе к конструированию математического обеспечения должна быть возложена на специализированные препроцессоры, осуществляющие автоматизированную конвейеризацию, векторизацию и т. п. Для программ на языках высокого уровня одна из основных проблем при разработке препроцессоров — распараллеливание гнезда вложенных DO-циклов. Данная работа посвящена развитию и применению к конвейерным векторным ЭВМ так называемого «координатного» метода [1]. Указанный алгоритм выявляет подмножество циклов из исходного гнезда, по которым вычисления можно вести параллельно в синхронном режиме.

При перестройке тела цикла может быть изменен порядок следования операторов, а также, что существенно, введены дополнительные операторы присваивания. Приведем кратко основные положения и терминологию из [1], которые будут использоваться далее.

Переменные, стоящие в левой части арифметического оператора, называются генерируемыми, а в правой — используемыми. Области изменения индексов n вложенных циклов образуют n-мерный параллелепипед S_n — пространство итераций. Каждый вектор $\mathbf{P}$ из S_n определяет одно выполнение тела цикла. Для сравнения векторов из S_n используется лексикографический порядок.

Переход от пространства S_n к пространству S_k, $k<n$ (результат распараллеливания), задается отображением π: $S_n \to S_k$; $S_{n-k} = S_n \backslash S_k$ — пространство итераций распараллеленных циклов.

Вводится функция индексных выражений $T_{A_i}(\mathbf{P})$ — $T_{A_i} : S_n \to Z_r$, где A_i — i-е вхождение в тело цикла элемента массива A; r — размерность массива A; l-я компонента $T_{A_i}(\mathbf{P})$ есть значение индекса в l-м измерении индексного выражения при A_i на итерации цикла, задаваемой вектором $\mathbf{P} \in S_n$. Определяется отношение предшествования: $A_m \to B_l$ (A_m предшествует B_l), если вхождение A_m происходит раньше вхождения B_l в теле цикла, т. е. либо A_m находится в операторе, предшествующем оператору с B_l, либо, если они в одном операторе, A_m — используемая переменная, а B_l — генерируемая.

Вводится множество n-мерных векторов направлений:

$$\langle A_m, A_l \rangle = \{\mathbf{X} : T_{A_m}(\mathbf{P}) = T_{A_l}(\mathbf{P} + \mathbf{X}),\ \mathbf{P} \in S_n,\ \mathbf{X} \in Z_n\}.$$

Распараллеливание возможно, если отображение π удовлетворяет следующим требованиям [1]:

Правило $S1$. Для любых двух вхождений A_m и A_l элементов одного массива, хотя бы одно из которых является генерацией, если $\mathbf{X} \in \langle A_m, A_l \rangle$ и $\mathbf{X} > \mathbf{0}$, то для π должно выполняться:

1) $\pi(\mathbf{X}) \geqslant \mathbf{0}$; 2) если $\pi(\mathbf{X}) = \mathbf{0}$, то $A_m \to A_l$.

Правило $S2$. Для любых двух вхождений A_m и A_l элементов одного массива. хотя бы одно из которых является генерацией, если $\mathbf{0} \in$ ности, т. е. если $A_m \to B_l$ и $B_l \to C_f$, то $A_m \to C_f$. Если в результате появились отношения предшествования вида $A_m \to A_m$, то по выбранному подмножеству циклов распараллелиться нельзя и требуется задать другое π. Если же все условия выполнены, то далее строится такая последовательность операторов тела цикла, при которой выполняются все отношения предшествования между переменными. Причем если используемая переменная A_m должна стоять в i-м операторе, а в j-м операторе ($j < i$) уже стоит переменная B_l такая, что $A_m \to B_l$, то вводится дополнительный k-й оператор ($k < j$) вида $\mathrm{TEMP} = A_m$ (TEMP — новая переменная) и в i-м операторе вместо A_m используется TEMP (рис. 1, *а*, *б*).

Построение оптимальной последовательности операторов тела распараллеленного цикла. Максимальное быстродействие параллельных ЭВМ достигается при полной и постоянной загрузке параллельно работающих процессорных элементов. Это возможно лишь в том случае, если устройства памяти успевают обеспечивать все необходимые для этого операции записи и считывания данных.

В общем случае это не так, и увеличение числа операций с памятью при оптимизации циклов может существенно сказаться на эффективности работы компьютера [2].

Построение нового тела цикла с помощью «координатного» метода может потребовать введения дополнительных операторов, требующих пересылки данных из одной области памяти в другую. Естественно, что такие добавления увеличивают общее время работы оптимизируемого цикла и требуют резервирования памяти для хранения пересылаемых переменных. Поэтому возникает задача построения тела цикла с минимальным числом дополнительных операторов.

Их число определяется структурой последовательности, в которой размещаются операторы нового тела цикла. Между генерируемыми переменными могут существовать отношения предшествования. При этом соответствующие операторы должны располагаться в определенной последовательности. Так, пример, приведенный в [1] (рис. 1,

а
```
     DO 4 I=2,M
     DO 4 J=2,N
1    A(I,J)=B(I,J)+C(I)
      A1     B1     C1
2    C(I)=B(I-1,J)
      C2   B2
3    B(I,J)=A(I+1,J)**2
      B3     A2
4    CONTINUE
```
б
```
     DO 4 J=1,N
     DO 4 SIM FOR ALL (2 <=I <=M)
     TEMP(I)=A(I+1,J)
               A2
1    A(I,J)=B(I,J)+C(I)
      A1     B1     C1
3    B(I,J)=TEMP(I)**2
      B3
2    C(I)=B(I-1,J)
      C2   B2
4    CONTINUE
```
в
```
     DO 4 J=1,N
     DO 4 SIM FOR ALL (2 <=I <=M)
     TEMP1(I)=B(I,J)
               B1
3    B(I,J)=A(I-1,J)**2
      B3     A2
     TEMP2(I)=C(I)
               C1
2    C(I)=B(I-1,J)
      C2   B2
1    A(I,J)=TEMP1(I)+TEMP2(I)
      A1
4    CONTINUE
```

Рис. 1. Исходный цикл (*а*), распараллеленный цикл с оптимальной (*б*) и неоптимальной (*в*) последовательностями операторов *а, б*), содержит генерируемые переменные, связанные отношением $B_3 \to C_2$; естественно, что оператор, содержащий B_3, должен стоять в новом теле цикла выше оператора, содержащего C_2. Однако в цикле могут быть операторы, генерируемые переменные которых не связаны отношением такие переменные. Занумеруем все переменные, встречающиеся в теле цикла так, что первые r номеров будут соответствовать генерируемым переменным G_i (r — число операторов присваивания), а оставшиеся — используемым U_j. Введем матрицу предшествования M размерности $t \times t$, где t — полное число переменных:

$$M_{ij} = \begin{cases} 1, & \text{если } i \to j; \\ 0, & \text{если } i \not\to j. \end{cases}$$

Соответствующий матрице M граф информационных зависимостей с множеством вершин V естественным образом может быть представлен в виде r групп вершин $V = \bigcup\limits_{i=1,r} V_i$ (рис. 2) по правилу: каждому оператору по группе. При этом самая левая вершина в группе соответствует генерируемой переменной: $V_i = G_i \cup \{U_k \xrightarrow{S3} G_i\}$ (запись $\xrightarrow{S3}$ означает предшествование по правилу $S3$). Граф является ориентированным: из вершины i в вершину j направлена дуга, если $i \to j$. Упорядочение (нумерация) групп графа отвечает некоторому размещению операторов тела цикла. Группу, из которой выходит дуга, будем называть исходной. Дуги, входящие в группы с номерами, большими, чем номера исходных, соответствуют выполненным отношениям предшествования. Дуги, входящие в группы с меньшими номерами, чем у исходных, порождают дополнительные операторы. При упорядочении все дуги, соединяющие самые левые вершины в группах (отношения $G_i \to G_j$), должны входить в группы с большими номерами, чем у исходных, так как разрешение таких отношений путем введения дополнительных операторов невозможно.

Из графа на рис. 2 видно, что k-й и j-й операторы нельзя поменять местами, так как есть отношение $G_j \to G_k$, а i-й и j-й операторы целесообразно поменять местами, поскольку при этом вверх будут направлены не две, а одна дуга графа и потребуется соответственно один дополнительный оператор.

Вообще, новое тело цикла оптимально, если в соответствующем ему графе информационных зависимостей минимально число дуг, входящих в группы с меньшими номерами, чем у исходных. Фактически для оптимального выбора последовательности операторов нужно переупорядочить лишь вершины, соответствующие генерируемым переменным. Поэтому целесообразно перейти от исходного графа к ориентированному взвешен-

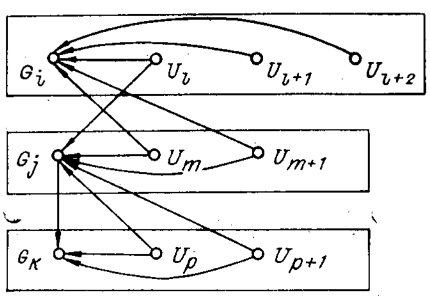

Рис. 2. Граф информационных зависимостей

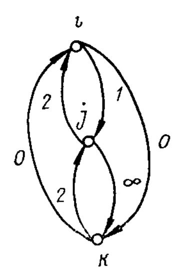

Рис. 3. Ориентированный взвешенный граф

ному графу с r вершинами, отвечающими генерируемым переменным. Вес дуги из i-й вершины в j-ю будет равен числу отношений предшествования, наложенных на переменные из j-го оператора переменными i-го оператора (их необходимо разрешить, если j-й оператор будет предшествовать i-му). Введем матрицу весов W, элемент W_{ij} которой равен весу дуги из i-й вершины в j-ю. Матрица W вычисляется с помощью матрицы M, которую можно разбить на четыре подматрицы. Подматрица $M\,[1:r,\ 1:r]$ содержит отношения вида $G_i \to G_j$, которые изначально должны быть выполнены, поэтому соответствующие $W_{ij} = \infty$. Таким образом, имеем

$$\forall i,\ j = 1 - r;\ W_{ij} = \begin{cases} \infty, & \text{если } M_{ij} = 1; \\ 0, & \text{если } M_{ij} = 0. \end{cases}$$

Подматрица $M\,[r+1:\ t,\ 1:r]$ включает отношения вида $U_i \to G_j$. Элементы W_{ij} определяются так: $\forall i, j = 1, \ldots, r$, $\forall k = r+1, \ldots, t : U_k \overset{S3}{\to} G_i$; $W_{ij} = W_{ij} + M_{kj}$. Подматрицу $M\,[1:r,\ r+1:t]$ составляют отношения вида $G_i \to U_j$. Однако $\forall U_j \exists G_k : U_j \overset{S3}{\to} G_k$ и по правилу транзитивности должно выполняться $G_i \to G_k$. Но такое отношение уже учтено в подматрице $M\,[1:r,\ 1:r]$ и переходит в матрицу W в виде $W_{ik} = \infty$, следовательно, рассматриваемую подматрицу не следует учитывать. Элементы подматрицы $M\,[r+1:t,\ r+1:t]$ — это отношения вида $U_i \to U_j$. Однако отношения такого вида появляются лишь по правилу транзитивности с участием хотя бы одной генерируемой переменной, т. е. $\exists G_k$: $U_i \to G_k \wedge G_k \to U_j$, и уже учтены в рассмотренных ранее подматрицах. Таким образом, матрица W формируется на основе подматрицы $M\,[1:t,\ 1:r]$. На рис. 3 изображен взвешенный граф для приведенного ранее графа информационных зависимостей (см. рис. 2).

Теперь решаемая задача свелась к поиску гамильтонова пути наименьшего веса на построенном ориентированном взвешенном графе. Вес пройденного пути при этом вычисляется по формуле $S = \sum_{j=1}^{n} \sum_{k=1}^{j-1} W_{jk}$.

Полученная постановка известна среди оптимизационных задач как наилучшая приближенная триангуляция квадратной матрицы, т. е. приведение матрицы к виду, в том или ином смысле минимально отличающемуся от верхнетреугольной. В работе [3] изложен алгоритм решения этой задачи.

Ускорение работы координатного метода. При выборе из n вложенных циклов подмножества, для которого проверяются условия распараллеливания, возможно $2^n - 1$ комбинаций. Однако можно сократить число возможных вариантов без проверки условий, заранее установив невозможность распараллеливания по ним.

Индексные выражения при элементах массивов могут не содержать все индексы циклов. Такие индексы в [1] называются отсутствующими. Если индекс l-го цикла (l-го измерения S_n) отсутствует у генерируемого элемента A_i, то $\langle A_i,\ A_i \rangle$ имеет следующий вид:

$$\langle A_i,\ A_i \rangle = \{(x_1,\ \ldots,\ x_n),\ (\forall j \neq l,\ x_j = 0),\ x_l \in Z\}. \tag{1}$$

Утверждение 1. Если какой-либо индекс цикла отсутствующий, то по соответствующему циклу распараллеливание произвести невозможно.

Доказательство. Пусть у генерируемой переменной A_i l-й индекс цикла отсутствующий. Тогда для нее выполняется соотношение (1), т. е.

$$(\forall \mathbf{X} > \mathbf{0} : \mathbf{X} \in \langle A_i,\ A_i \rangle) \Rightarrow ((\forall j \neq l,\ x_j = 0) \wedge (x_l > 0)). \tag{2}$$

Пусть выбрано отображение π такое, что можно произвести распараллеливание и по l-му циклу, т. е. $\pi = (i_1,\ \ldots,\ i_k)$: A_j, $i_j \neq l$. Отсюда следует, что для любого $\mathbf{X}$, удовлетворяющего (2), $\pi(\mathbf{X}) = \mathbf{0}$. В этом случае по правилу $S1$: $A_i \to A_i$, что свидетельствует о невозможности распараллеливания при выбранном π. Утверждение доказано.

Таким образом, если из n вложенных циклов m имеют отсутствующие индексы, то нужно проверить лишь $2^{n-m}-1$ комбинаций циклов, которые потенциально могут быть распараллелены. Надо отметить, что наличие отсутствующих индексов довольно характерное явление в вычислительных программах.

Векторы $\langle A_i, A_j\rangle$, рассматриваемые в правиле $S1$, удовлетворяют условию $\langle A_i, A_j\rangle > \mathbf{0}$. Из этого условия следует, что самая левая из отличных от нуля компонент $\langle A_i, A_j\rangle$ положительная, однако более правые ненулевые компоненты могут быть и отрицательными. Сформулируем теперь

Утверждение 2. Если среди векторов направлений есть $\langle A_i, A_j\rangle > \mathbf{0}$, m-я компонента которого отрицательная, то m-й цикл не может быть самым внешним из нераспараллеливаемых циклов.

Доказательство. Пусть отображение $\pi=(i_1, \ldots, i_k)$ таково, что $i_1=m$. Пусть $\mathbf{X}\in\langle A_i, A_j\rangle$, тогда по условию $\mathbf{X}>\mathbf{0}$ и $x_m<0$. Но при данном выборе π имеем $\pi(\mathbf{X})<\mathbf{0}$, т. е. правило $S1$ нарушено. Следовательно, распараллеливание невозможно. Утверждение доказано. Отсюда следует вывод: если m-я компонента $\langle A_i, A_j\rangle > \mathbf{0}$ отрицательна, то нужно рассматривать только $2^n-2^{n-m}-1$ комбинаций для π.

Из двух приведенных утверждений вытекает

Следствие. Если m-й цикл является самым внешним, содержащим отсутствующий индекс, и m-я компонента $\langle A_i, A_j\rangle > \mathbf{0}$ отрицательна, то среди нераспараллеленных обязательно должен быть цикл, уровень вложенности которого меньше m. В этом случае число рассматриваемых для π комбинаций сокращается до $2^{n-1}-2^{n-m}-1$.

Применение координатного метода к ЭВМ конвейерной архитектуры. Изначально рассматриваемый метод был разработан для ЭВМ типа ОКМД (одна команда — множество действий) [2]. В этом разделе приведено обоснование его применимости к ЭВМ конвейерного типа [2], а также некоторое расширение метода с учетом специфики таких ЭВМ.

Существенное отличие работы ЭВМ типа ОКМД от конвейерной состоит в разном построении последовательности вычислений арифметических операторов и записи их результатов в память.

В ЭВМ первого типа вначале одновременно вычисляются значения правой части арифметического оператора для всего набора значений, принимаемых индексами распараллеленных циклов, а затем уже происходит одновременное присвоение результатов левой части оператора (запись в память). В конвейерной ЭВМ результаты вычисления оператора для некоторой итерации распараллеленных циклов могут быть записаны в память (присвоены левой части) до того, как вычислены значения того же оператора для других итераций таких циклов. Указанные отличия могут влиять на сохранение информационных зависимостей.

В результате распараллеливания происходит переход от стандартных скалярных арифметических операторов к векторным, которые осуществляют поэлементную обработку векторов. Длина вектора l определяется по числу итераций распараллеленных циклов. Введем вектор итераций $\mathbf{P}=\{(\mathbf{P}_1, \ldots, \mathbf{P}_l)\colon \forall i=1-l,\ \mathbf{P}_i\in S_{n-k}\}$, задающий однозначное соответствие между элементами векторных операторов и исходными итерациями распараллеленных циклов.

Утверждение 3. Последовательность действий, заданных гнездом циклов при распараллеливании «координатным» методом, выполняется на ЭВМ конвейерного типа без нарушения информационных зависимостей, если $\mathbf{P}$ удовлетворяет условию: $\forall i,\ j=1-l$, если $i>j$, то $\mathbf{P}_i>\mathbf{P}_j$.

Доказательство. При сравнении двух упомянутых ЭВМ речь шла о различии в обработке каждого отдельного арифметического оператора, поэтому информационные зависимости могут быть нарушены лишь в том случае, если генерируемая переменная и некоторые из используемых являются элементами одного и того же массива. Пусть A_i — генерируемая переменная оператора, а A_j — одна из используемых. По правилу $S3$ существует отношение $A_j\to A_i$. Если выбрано π, удовлетворяющее всем необходимым правилам, то возможны следующие варианты:

$$\text{если } \langle A_j,\ A_i\rangle > \mathbf{0}, \text{ то } \pi(\langle A_j,\ A_i\rangle) > \mathbf{0} \text{ или } (\pi(\langle A_j,\ A_i\rangle) = \mathbf{0} \text{ и } A_j \to A_i);$$

$$\text{если } \langle A_i,\ A_j\rangle > \mathbf{0}, \text{ то } \pi(\langle A_i,\ A_j\rangle) > \mathbf{0}. \qquad (3)$$

Вариант, когда $\langle A_i,\ A_j\rangle > \mathbf{0}$ и $\pi(\langle A_i,\ A_j\rangle) = \mathbf{0}$, невозможен, так как при этом по правилу $S1$ возникает отношение $A_i \to A_j$, противоречащее отношению $A_j \to A_i$. Если в соотношениях (3) для π выполняется неравенство, то вычисление элементов A_i и A_j происходит в разных последовательных итерациях нераспараллеленных циклов, и, следовательно, [illegible] на более ранней итерации, чем для A_j. Следовательно, при параллельной конвейерной обработке нарушения информационных зависимостей не произойдет, если вычисления для A_i будут проведены раньше, чем для A_j. Это верно для вектора итераций $\mathbf{P}$, удовлетворяющего условию $\forall i, j = 1, \ldots, l$, если $i > j$, то $\mathbf{P}_i > \mathbf{P}_j$. Утверждение доказано.

Возможность при конвейерной обработке записать результаты вычислений в память до завершения их для всех итераций распараллеленных циклов позволяет ослабить правило $S1$.

Утверждение 4. Пусть в теле n вложенных циклов есть оператор вида $A_i = F(A_j)$, где F — некоторая арифметическая функция, вычисляемая на конвейерной ЭВМ за τ шагов, и $\langle A_i,\ A_j\rangle = (x_1, \ldots, x_n) > \mathbf{0}$ (имеет место рекуррентное соотношение). Пусть отображение π таково, что $\pi(\langle A_i,\ A_j\rangle) = \mathbf{0}$. Тогда выбранное π будет допустимо, если либо $\exists x_k \geqslant l_k$, либо $\tau \leqslant x_r \prod\limits_{\substack{r < m \leqslant n \\ m \in \Omega}} l_m$, где l_m — количество итераций, задаваемых m-м циклом; Ω — множество номеров распараллеленных циклов; x_r — самая правая положительная компонента вектора $\langle A_i,\ A_j\rangle$ (она всегда существует, так как $\langle A_i,\ A_j\rangle > \mathbf{0}$).

Доказательство. Отметим, что исходный координатный метод данное π определит как недопустимое. Этот факт уже рассматривался в доказательстве предыдущего утверждения.

В операторе $A_i = F(A_j)$ элементы массива A являются как входными, так и выходными данными. Из условия $\langle A_i,\ A_j\rangle > \mathbf{0}$ следует, что результат вычисления этого оператора на некоторой итерации может в дальнейшем использоваться этим же оператором на более поздней итерации, т. е. может иметь место рекуррентное соотношение. Если $\exists x_k \geqslant l_k$, то множества генерируемых и используемых переменных рассматриваемого оператора не пересекаются, и, следовательно, при таком π нарушения информационных зависимостей произойти не может. Иначе обстоит дело, если $\forall i,\ x_i < l_i$. Так как по условию $\pi(\langle A_i,\ A_j\rangle) = \mathbf{0}$, то как используемые, так и генерируемые на некоторой итерации элементы массива A попадают в один вектор A^*, обрабатываемый параллельно в конвейерном режиме. Элементы вектора A^* формируются из элементов A на основе утверждения (3). Так как оператор F вычисляется на ЭВМ с длиной конвейера τ, то с момента начала вычисления некоторой k-й компоненты вектора A^* до его завершения на обработку будет подано еще $\tau - 1$ компонент. Тогда нарушения информационных зависимостей при параллельном выполнении не произойдет в том случае, если среди этих компонент отсутствуют такие, для которых необходимо использовать результат вычисления k-й компоненты (т. е. если компоненты, связанные рекуррентным соотношением, разделяет не менее τ элементов). Значение x_r — самого правого положительного элемента вектора $\langle A_i,\ A_j\rangle$ — определяет количество компонент вектора A^*, разделяющих A_i и A_j, метода распараллеливания DO-циклов. Предложен подход к построению нового тела цикла с минимальным числом дополнительных операторов присвоения, что позволяет получать в результате распараллеливания более эффективный цикл.

Показана применимость координатного метода к конвейерной ЭВМ. Кроме того, расширены возможности его использования на таких компьютерах с учетом обработки рекуррентных соотношений.

Предложены способы ускорения работы алгоритма, позволяющие избежать полного перебора возможных комбинаций распараллеливаемых циклов.

Перечисленные возможности по развитию метода были использованы и апробированы в программной разработке по автоматическому распараллеливанию циклов для процессора А-12* [4]. Опыт обработки реальных циклов для этого процессора показывает, что отсутствующие индексы в пользовательских программах — довольно частое явление. Построение тела цикла, как правило, ведет к появлению не более трех дополнительных операторов.

ЛИТЕРАТУРА

1. **Lamport L.** The parallel execution of DO loops // Comm. of ACM.— 1974.— V. 17, N 2.
2. **Хокни Р., Джессхоуп К.** Параллельные ЭВМ.— М.: Радио и связь, 1986.
3. **Белкин А. Р.** Приближенная триангуляция матриц в задачах ранжирования и обработки межотраслевого баланса // Техн. кибернетика.— 1981.— № 1.
4. **Бродский И. И. и др.** Высокопроизводительный периферийный векторный процессор А-12 // Вопр. кибернетики.— 1985.— № 104.

Поступила в редакцию 21 апреля 1987 г.

УДК 681.3.06

Р. ВЕРБОВА, А. НЯГОЛОВ, П. ТАНЕВА, А. ТРЕНЕВ

(София, Болгария)

ПОДХОДЫ К УПРАВЛЕНИЮ СПЕЦИАЛИЗИРОВАННЫМИ АРИФМЕТИЧЕСКИМИ ПРОЦЕССОРАМИ

Введение. Микрокомпьютерные системы автоматизации занимают в последние годы центральное место в области автоматизации научных исследований, управления непрерывными производственными процессами, САПР и т. д. Разнообразие и непрерывное совершенствование элементной базы приводят к созданию разных типов систем автоматизации в соответствии с конкретными требованиями применений. Использование современных магистрально-модульных интерфейсов обеспечивает их гибкую структуру. Основным требованием к таким системам, работающим, как правило, в реальном масштабе времени, является быстродействие. Функции управления чаще всего связаны с выполнением большого числа вычислительных операций в широком диапазоне входных данных и выходных реакций. Вычислительные возможности микропроцессоров, однако, весьма ограниченны. Все это приводит к необходимости включения

---

* Публикация по разработке планируется во втором номере журнала «Автометрия» за 1988 г.