туру функций и архитектуру системы. Помимо этого, получаемые на основе модели оценки эффективности и классификации элементов являются средством сравнения вариантов архитектуры и определения содержания необходимой корректировки каждого варианта. Предлагаемый метод анализа эффективности проектных решений может быть использован для управления процессом проектирования архитектуры широкого класса АСНИ.

Литература

1. Ванд Л.Э. Оперативный анализ, оценка и выбор решений при многих критериях и ограниченной информации. Методические рекомендации. - М.: ЦНИПИАСС, 1978. - 30 с.

2. Гермейер Ю.Б. Введение в теорию исследования операций. - М.: Наука, 1971, с. 36-44.

3. Беллман Р., Заде Л. Принятие решений в расплывчатых условиях. - В сб.: Вопросы анализа и процедуры принятия решений. -М.: Мир, 1976, с. 172-216.

4. Кендалл М.Дж., Стьюарт А. Статистические выводы и связи. - М.: Наука, 1973, с. 58-109.

5. Смоляк С.А., Титаренко Б.П. Устойчивые методы оценивания. -М.: Статистика, 1980. - 158 с.

УДК 681.3.06

В.К.Погребной, И.Н.Кошовкин, Т.Г.Балова

ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ АСНИ, ПРЕДСТАВЛЕННЫХ НА ЯЗЫКЕ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ

(г. Томск)

Применение микропроцессоров (МП) и микроЭВМ при разработке АСНИ существенно изменило представление об алгоритме функционирования программного обеспечения: произошел переход от централизованного сбора информации и управления к децентрализованному, что привело к созданию распределенных систем на базе автономных контроллеров, выполняющих локальные задачи эксперимента и управление им. Построение распределенной системы предполагает разбиение функциональных алгоритмов

АСНИ на множество законченных функций путем декомпозиции алгоритмов и сопоставления им вычислительных процессов, для которых задается два типа отношений, определяющих порядок выполнения, по информационным связям (ограничения заданы алгоритмическими преобразованиями) и по внешним событиям (ограничения, накладываемые реальным временем).

Для этого в рамках системы автоматизации модульного проектирования комплексов программ (САМПР) решаются следующие задачи [I]: разработана методика анализа алгоритмов, выделения алгоритмических модулей, отображения их в виде графовых моделей и моделирования их работы; разработаны методы декомпозиции графовых моделей для выделения вычислительных процессов и задания между ними отношений порядка выполнения; разработаны методы настройки графовых моделей на конкретное применение, их оптимизации и синтеза объектных программных модулей АСНИ.

Отображение алгоритмов АСНИ графовыми моделями

Для отображения алгоритмов АСНИ и разработки формализованных методов решения перечисленных задач разработан язык ЭФ-М. Программа на языке ЭФ-М представляется графовой моделью алгоритма (МГА), вершинами которой являются базовые программные модули - элементарные функции алгоритмов (ЭФА) и модули более высоких рангов, между которыми устанавливаются управляющие и информационные связи. В качестве модулей могут выступать универсальные модули, реализованные на языке ЭФ-М, и подпрограммы, полученные на других языках, а также фиктивные модули, которые имитируют работу соответствующих алгоритмов, но программно не реализованы.

Отображение алгоритмов графовыми моделями в САМПР осуществляется в два этапа: анализ исходных алгоритмов АСНИ и выявление функционально законченных фрагментов алгоритмов (А-модулей); программная реализация выделенных А-модулей на языке ЭФ-М и моделирование их работы. Предполагаем, что некоторый А-модуль m_i реализует функцию f_i, определенную на множестве параметров R_i, и вычисляет значения множества параметров $R_{m_i} = f_i(R_i)$ - результатов А-модуля. Выделение модулей предполагает установление между двумя модулями m_i и m_j информационных связей по следующим двум правилам: во-первых, по совместному использованию одних и тех же параметров, когда $R_i \cap R_j \neq \phi$; во-вторых, результат одного А-модуля является исходным параметром другого $R_{m_i} \subseteq R_j$ и $R_{m_j} \subseteq R_i$. Применение данных правил позволяет выполнить децентрализацию функций алгоритма и упорядочить выделенные

А-модули по информационным связям. Программная реализация каждого А-модуля на языке ЭФ-М начинается с упорядочения исходных данных, определения их элементами, массивами, матрицами и таблицами, которые выделяются в автономные вершины МГА. После графического изображения вершин исходных параметров А-модуля осуществляется построение МГА для функциональной части А-модуля. Моделирование работы МГА осуществляется в режиме полной интерпретации, в результате чего устанавливается правильность функциональных преобразований МГА и информационная упорядоченность ее составных частей.

Декомпозиция вычислительного процесса, описываемого графовыми моделями

Машинное представление МГА отобразим в памяти двумя компонентами: U - операционной (задаваемой программой) и $\mathcal{J}$ - информационной (исходные данные и промежуточные результаты). Тогда можно говорить о вычислительном процессе в $S = U \times \mathcal{J}$, определяемом через конечное множество $P = \{p\}$ процессов, между которыми заданы отношения порядка с учетом вышеупомянутых ограничений по информационной упорядоченности и синхронизации событий. Каждому процессу заданы характеристики: $t(p)$ - время выполнения процесса, $t_0(p)$ - время окончания процесса. Множество процессов P может быть реализовано, если задана вычислительная система $T = V \times W \times K$, где $V = \{v\}$ - множество процессоров, $W = \{w\}$ - множество независимых компонент памяти, $K = \{k\}$ - множество каналов типа "общая шина". В таком случае для реализации вычислений, задаваемых операционной компонентой U над информационной компонентой $\mathcal{J}$, необходимо выделить в S множество процессов $P = \{p\}$ и определить для них отображение $S \rightarrow T$ так, чтобы минимизировать стоимость вычислительной системы:

$$C(T) \rightarrow \text{мин} \qquad (1)$$

при следующих условиях:

$$t'(p) \leqslant t(p), \quad p \in P, \qquad (2)$$

$$t'_0(p) \leqslant t_0(p), \quad p \in P, \qquad (3)$$

где $t'(p)$, $t'_0(p)$ - реальное время выполнения и окончания процесса.

Решение задачи (1) - (3) предполагает создание методов для принятия оптимальных структурных и функциональных решений, устанавливающих синхронизацию программных процессов с учетом их протекания в реальном

времени. Формирование множества процессов $P=\{p\}$ осуществляется путем разбиения графа МГА на подграфы, каждый из которых сопоставляется цепочкой последовательных процессов, для чего задается отображение множества вершин МГА $y\in Y$ во множество состояний S^* пространства $S=U\times J$. Для решения задачи построим два двудольных графа $G'=(Y,X,\Gamma)$ и $G^2=(Y,Z,\Omega)$, где $Y=\{y\}$, $X=\{x\}$, $Z=\{z\}$ - множества вершин МГА, соответствующих базовым программным модулям (ПМ), информационным модулям (ИМ) и операторам управления (ОУ). Под ИМ понимаются вершины исходных данных и результаты ЭФА, ОУ - условно введенный оператор переключения процессов. Множество дуг $\Gamma=\{\gamma\}$ задано на Y и X и отражает информационные связи между ПМ и ИМ, а $\Omega=\{\omega\}$ задается на Y и Z и определяет управляющие связи между ПМ и ОУ. Сформируем граф $G=G'\cup G^2=(L,Q)$. Необходимо найти разбиение графа G на подграфы $G_v(v=1,\dots,r)$ через разбиение множества Y на подмножества Y_v так, чтобы

$$\forall y_i\in Y_v\,(Y_v\cap Y_s=\emptyset)\;\exists z_i\in z_v,$$

где $z_v=\{z\,|\,\rho(z,y_i,\omega)\}$, $|z_v|=|Y_v|$.
Кроме этого, Y_v соответствует $X_v=\{x\,|\,\rho(x,y_i,\gamma)\}$, где $y_i\in Y_v$, $\rho(a,b,\alpha)$ - инцендентор: между вершинами α и b существует дуга α.

Процедура формирования множества процессов $P=\{p\}$ включает два этапа [2]: доопределение связей между ПМ и ОУ (построение параллельных ветвей); разрезание графа МГА с учетом дополнительных ограничений.

Затем осуществляется отображение $S\to T$, когда T задано с ограничением $|V|=|W|=|K|=1$, заключающееся в составлении расписания последовательного выполнения процессов P на одном процессоре. После этого выделяется множество процессов $P^\circ\subseteq P$, не удовлетворяющих ограничениям (2) и (3) в полученном на предыдущем этапе расписании, и формируется окончательное отображение $S\to T$ посредством следующих процедур:

для процессов $p\in P^\circ$, не удовлетворяющих условию (2), выполняется разбиение p на более простые $P(p)=\{p^1,\dots,p^n\}$, которые могут выполняться параллельно, и при этом

$$\max\, t'(p^i)\leqslant t(p),\; p^i\in P(p); \qquad (4)$$

для процессов $p\in P^\circ$, не удовлетворяющих условию (3), распараллеливаются процессы, предшествующие p (обозначим их через $P^{-r}(p)$)

так, чтобы

$$max\ t_0^i(p^i) + t^i(p) \leq t_0(p),\ p^i \in P^{-1}(p); \qquad (5)$$

для процессов $p \in P^0$, не удовлетворяющих условию (2), и для которых невозможно его разрешить с использованием первой процедуры, отображение задается выбором специализированных процессов;

для множества взаимосвязанных процессов P^* , не удовлетворяющих ограничению

$$\sum_{p_i \in P^*} t^i(p) \leq \sum_{p \in P^*} t(p), \qquad (6)$$

с учетом того, что существуют $p \in P^-$, не удовлетворяющие условию (2), и для которых разрешение противоречий невозможно действиями первой и третьей процедур, назначаются приоритеты.

Синтез объектных программных модулей в кодах микропроцессора

Синтез объектных программ с языка ЭФ-М в коды МП осуществляется в четыре этапа: преобразование графовой модели для упрощения ЭФА; формирование генерирующих последовательностей; выполнение оптимизационных процедур; генерация объектной программы.

Преобразование МГА заключается в упрощении сложных ЭФА в примитивные элементарные функции. Выделение множества примитивных ЭФА связано с системой команд МП. В большинстве случаев МП имеет один сумматор и набор команд $S = \{S_1, S_2, S_3\}$,

где S_1 - команда $LDA\ M$ (содержимое ячейки М помещается в сумматор), S_2 - это $STA\ M$ (содержимое сумматора помещается в ячейку M); S_3 - это $\Theta A, R$ (бинарная операция Θ над содержимым сумматора A и регистра R). К примеру, для МП К580 R может принадлежать множеству регистров $\{B, C, D, E, H, L\}$ или множеству спаренных регистров $\{BC, DE, HL\}$. Примитивные ЭФА позволяют использовать для выполнения действия команду S_3 с загрузкой сумматора по командам S_1 и S_2 . Упрощение сложных ЭФА и генерация примитивных осуществляется по алгоритмам, использующим методы декомпозиции графов, представленных в виде дерева.

Формирование генерирующих последовательностей достигается созданием таблицы признаков, число строк которой равно общему числу ЭФА в МГА, а столбцы определяются признаками, характеризующими заполнение

параметров ЭФА: указывают на тип и разрядность поступающих по информационным входам аргументов ЭФА, порядок загрузки регистров МП, определяют необходимость генерации меток и команд переходов и в целом задают всю исходную информацию для генерации объектной микропрограммы.

Графовое представление алгоритмов позволило разработать ряд специальных оптимизационных процедур, решающих задачи машинно-независимой оптимизации на МГА: распределения памяти; загрузки регистров; упорядочения информационных и управляющих связей ЭФА; устранения лишних и избыточных вычислений; чистки циклов; замены сложных вычислений на простые; оптимизации арифметических выражений.

Данная методика разработана в рамках САМПР как инструментальный пакет на ЕС ЭВМ с синтезом объектных программных модулей для компонент АСНИ на базе МП К580.

Литература

1. Погребной В.К. Автоматизация проектирования систем управления. - Томск: ТПИ, 1980. - 95 с.

2. Кошовкин И.Н., Балова Т.Г. Об одной задаче оптимизации программного обеспечения иерархических систем управления, работающих в реальном времени. - В кн.: Методы и программы решения оптимизационных задач на графах и сетях. Ч. I. Алгоритмы, программы, применения. Новосибирск, 1982, с. 93-96.

УДК 62.501.72

К.Д.Колесников

АЛГОРИТМИЗАЦИЯ ИДЕНТИФИКАЦИИ ЛИНЕЙНЫХ ОДНОМЕРНЫХ ОБЪЕКТОВ ПО ПЕРЕХОДНЫМ ФУНКЦИЯМ С ПРИМЕНЕНИЕМ ЭКСПОНЕНЦИАЛЬНО-ГАРМОНИЧЕСКОЙ АППРОКСИМАЦИИ

(г. Куйбышев)

В качестве одной из задач решения автоматического управления многими авторами выдвигается задача идентификации объектов управления [2 - 7]. В.В.Солодовниковым [1] была предложена постановка задачи авто-