А 05.13.16
А-379

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК РЕСПУБЛИКИ АРМЕНИЯ

На правах рукописи

УДК 681.3.06

АЙМАН АЛИ АШУР

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СПЕЦИАЛЬНЫХ ПЛАТ РАСШИРЕНИЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

Специальность 05.13.16 - применение вычислительной техники, математического моделирования и математических методов в научных исследованиях

05.13.11 - математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей

А в т о р е ф е р а т
диссертации на соискание ученой степени
кандидата технических наук

Ереван-1993

© National Library of Armenia OCR by Portmind

Работа выполнена в Ереванском государственном университете

Научные руководители: - доктор физико-математических наук, профессор ШУКУРЯН С.К.

- кандидат физико-математических наук, доцент ЕГИАЗАРЯН В.С.

Официальные оппоненты: - академик НАН РА, доктор технических наук, профессор АТОЯН Р.В.

- кандидат физико-математических наук, с.н.с. АСЛАНЯН Л.А.

Ведущая организация - Ереванский Государственный Инженерный Университет, кафедра вычислительных машин, комплексов, систем и сетей (г. Ереван)

Защита состоится "3" декабря 1993г. в 11.00 час. на заседании специализированного совета К 005.21.01 Института проблем информатики и автоматизации НАН РА по адресу: 375044, г. Ереван, ул. П.Севака, 1.

С диссертацией можно ознакомиться в научно-технической библиотеке института.

Автореферат разослан "3" ноября 1993г.

2617-94

Ученый секретарь специализированного совета, к.э.н ,с.н.с. МЕЛКОНЯН А.Е.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. Существующие в настоящее время платы расширения (ПР) и сопроцессоры для персональных компьютеров, расширяющие функциональные возможности и границы их применения, обеспечивают достаточно широкий диапазон потребностей от эмуляции архитектуры вычислительных машин (ВМ) до быстрого выполнения специальных функций. Стоимость ПР и сопроцессоров колеблется в пределах от 300 до 3000 долларов, что сравнимо с стоимостью самого персонального компьютера. В последнее время в США даже в оборонных исследованиях в связи с сокращением ассигнований стало уделяться большое внимание адаптерным платам, связывающим недорогие персональные компьютеры с шинами, сответствующими военным стандартам. Подобные платы эмулируют и имитируют работ дорогостоящих компьютеров или периферийных устройств научного и технического назначения стоимостью в несколько сотен тысяч долларов при разработке прикладных программ, а в определенных случаях вместе с персональным компьютером могут их заменять. Особое место занимают платы расширения, ориентированные на автоматизацию научного эксперимента, одной из определяющих характеристик которого является время. Неуклонное расширение сферы применения компьютеров, быстрый рост их количества, постоянное изменение запросов рынка делают важной и актуальной проблемой разработку информационных технологий проектирования, полностью охватывающих процесс проектирования ПР и сопроцессоров и позволяющих с минимальными затратами и быстро откликаться на любые запросы.

Идеальное решение для проблем, связанных с проектированием ПР, заключается в автоматической генерации полного комплекта документации для ее изготовления по ее исходному описанию на некотором формальном языке. Однако, в силу больших трудностей принципиального характера, сложности универсальных алгоритмов генерации, жестких требований к времени проектирования ПР и, соответственно, инструментарию для ее проектирования, оно в настоящее время недостижимо.

© National Library of Armenia

В то же время активно разрабатываются различные приближения к этому решению. Настоящая работа является составной частью исследований, проводимых в последние годы по разработке адаптивной информационной технологии проектирования специальных семейств ПР, имеющих общий базовый микропроцессор и отличающихся друг от друга содержимым управляющей памяти микропрограмм этого микропроцессора. Эффективность подхода во многом определяется возможностями настройки базового микропроцессора и программ связи с центральным микропроцессором персонального компьютера на конкретную задачу.

Целью диссертационной работы является обоснование и разработка алгоритмов и программ инструментальной оболочки для автоматизированной генерации интерфейсного и инструментального программного обеспечения по описанию ПР на языке высокого уровня.

Научная новизна. Предложены:

- многоуровневая модель компьютера и метод синтеза реализаций программы на этой модели;
- структура базы знаний для метода синтеза;
- алгоритмы генерации реализаций программ для различных уровней модели;
- алгоритм проверки корректности для класса системных программ ПР;
- алгоритмы построения частных реализаций команд и программ.

Практическая ценность работы.

Разработанное алгоритмическое и программное обеспечение включено в состав системы автоматического проектирования семейств специальных плат расширения персональных компьютеров, совместимых с IBM PC, разработка которой ведется в рамках гранта Национального фонда развития науки и инноваций Республики Армения, а также использовано при разработке конкретных плат расширения персональных компьютеров в различных

предприятиях и организациях.

Апробация работы. Конкретные результаты работы обсуждались на семинарах в Ереванском государственном университете и Ереванском НИИ математических машин.

Публикации. Основные результаты работы отражены в 2 публикациях, список которых приведен в конце автореферата.

Структура и объем работы. Диссертация состоит из введения, трех глав, заключения, списка литературы и приложения. Объем работы 103 стр., включая 5 стр. рисунков, I стр. таблиц, 10 стр. списка цитируемой литературы, насчитывающего 90 наименований, и I стр. приложения.

СОДЕРЖАНИЕ РАБОТЫ

Во введениии обосновываются актуальность, научная новизна и практическая ценность диссертационной работы и формулируется цель работы. Рассмативается краткое содержание работы по главам.

В главе I проведен анализ тенденций в развитии ПР для персональных компьютеров, указана возможность унификации семейств ПР по функциональным возможностям и быстродействию, интерфейсу с персональным компьютером и.т.д. и приведен подход к проектированию семейств специализированных ПР для персональных компьютеров, заключающийся в следующем:

I. В качестве аппаратной основы - центрального процессора плат семейства выбирается некоторый микропрограммируемый микропроцессор, удовлетворяющий унифицированным требованиям для всех плат семейства.

2. Отдельные ПР семейства отличаются друг от друга содержимым постоянного запоминающего устройства (ПЗУ) микропрограмм. В ПЗУ содержится микрокод реализации той конкретной функции (совокупности функций), на эффективное выполнение которой ориентирована данная ПР.

3. Настройка на данный микропроцессор и генерация микрокода для ПЗУ выполняются автоматически по входному

© National Library of Armenia

описанию ПР.

4. Обращение и использование ПР проводится с помощью специальных системных программ, также генерируемых автоматическки.

Преимущества рассматриваемого подхода:

- возможность уменьшения времени выполнения для произвольной функции (совокупности функций), выполняемых на персональном компьютере;

- единая технология изготовления всех плат семейства;

- минимальный объем затрат при переходе от одного базового микропроцессора к другому.

Детально рассмотрено применение предложенного подхода для разработки семейства ПР, ориентированного на конкретный микропроцессор. Архитектура этого микропроцессора соответствует стандарту США, обеспечивает достаточно сложный набор машинных команд и достаточно широко используется в авиационной электронике как для гражданских, так и для военных целей. В качестве входного описания функции ПР рассматриваются фрагменты программ из машинных команд микропроцессора.

Приведена и обоснована типовая структура системных программ для проектирования и использования рассматриваемых ПР. Выделены функции и компоненты, необходимые при проектировании специальных ПР, проанализированы необходимые связи между компонентами и трудности при разработке компонентов.

В §I.2 на основе проведенных рассмотрений приведена общая структура системы автоматического проектирования семейств ПР, обоснованы и выделены ее основные подсистемы.

Сформулированы требования к технологическому окружению проектирования системных программ, настраеваемых на конкретную ПР, и основные задачи автоматической генерации

Глава 2 посвящена математическому и алгоритмическому обоснованию и решению ряда задач проектирования технологической среды генерации программ, настраеваемых на конкретную ПР.

Вычислительную машину можно представить в виде иерархии

уровней, каждому из которых соответствует некоторая виртуальная или реальная машина со своим машинным языком. Одной из наиболее актуальных и важных проблем при таком представлении является отображение решаемой задачи на эту многоуровневую структуру, позволяющее оптимально использовать возможности различных уровней при ее решении.

Отображение можно проводить путем адаптации отдельных уровней в иерархии к решаемой задаче. При этом диапазон используемых средств может варироваться от мягких программных и, в том числе, микропрограммных до жестких аппаратных средств, включая и комбинированные программно-аппаратные методы. Например, в базовых компиляторах для вычислительных машин с архитектурой очень длинных командных слов VLIW используются чисто программные методы, а проблема адаптации вычислительной машины к задачам, требующим быстрого доступа к часто используемым операндам, уменьшения времен выборки и декодирования команд, оптимизации организации и хранения промежуточных результатов, уменьшения времени запоминания и восстановления совместно используемых ресурсов при выполнении переходов с возвратом как чисто аппаратными методами с помощью процессоров архитектуры RISC так и микропрограммными и аппаратно-микропрограммными средствами.

В §2.I предложен метод синтеза реализаций программы для произвольного уровня вычислительной машины, являющийся обобщением метода, выполняющего адаптацию микропрограммного уровня вычислительной машины к решаемым задачам*).

Метод состоит из двух шагов:

I. Анализ исходной программы уроня i и его преобразование в промежуточное представление (в программу на промежуточном языке).

2. Генерация реализации программы на уровне l (l>i) по полученному промежуточному представлению.

---

*) Шукурян С.К. Подход к решению задач синтеза микропрограммных поддержек. // Кибернетика._ I989. :4, с. 26-32.

© National Library of Armenia

Метод основан на следующих трех компонентах базы знаний для каждого оператора исходного и промежуточного языка:

- множестве описаний соотношений "вход-выход";
- множестве правил преобразования;
- множестве схем частных реализаций на уровне микрокода.

Каждая схема может быть превращена в конкретную реализацию на уровне 1 после задания значений неопределенных параметров.

Шаг I включает детальный анализ управляющего графа преобразуемой программы и поиск соотношений между переменными программы с помощью указанных выше множеств соотношений "вход-выход". Найденные соотношения и результаты анализа определяют множество применимых правил преобразования и порядок их применения.

Каждое правило добавляет к преобразуемым операторам новые параметры, которые позволяют оптимизировать реализацию на уровне j за счет удаления избыточных действий, выделения общих частей и совмещения независимых действий.

Шаг 2 включает выбор подходящей схемы реализации для каждого оператора промежуточного языка, ее преобразование в конкретную реализацию с помощью значений соответствующих параметров и объединение полученных реализаций в единую реализацию.

Приведем соответствующие формальные определения.

Пусть $O^{(i)}$ некоторое множество, называемое множеством опера ций уровня i $(i=0,1,\dots,2d)$*), и пусть задано разбиение множества $O^{(i)}$ на непересекающиеся классы $O^{(i)}=\bigcup_{j=1}^{n_i} O_j^{(i)}$, $h\neq l \Rightarrow$

*) Введение четного количества уровней позволяет разделить задачи оптимизации при реализации программы на более низком уровне и оптимизации программы на данном уровне. На четных уровнях можно рассматривать вопросы реализации, на нечетных уровнях - совмещения команд.

$O_h^{(i)} \cap O_l^{(i)} = \emptyset$, $n_0=1$, и если $i=2k$, то $n_{i+1}=n_{i+2}$, $n_i<n_{i+2}$, причем в каждом классе $O_j^{(i)}$ существует специальная пустая операция $e_j^{(i)}$. Будем считать, что с $O^{(i)}$ связано множество переменных $R^{(i)}$. Множество значений переменных из $R^{(i)}$ обозначим через $D^{(i)}$, множество состояний переменных $(D^{(i)})^{R^{(i)}}$ - через $S^{(i)}$, при этом если $i<l$, то $R^{(i)}\subseteq R^{(l)}$, $D^{(i)}\subseteq D^{(l)}$, $R^{(2k)}=R^{(2k+1)}$, $D^{(2k)}=D^{(2k+1)}$. С каждой операцией $o\in O^{(i)}$ свяжем множество переменных $R[o]$ и функцию $f[o]$, выполняемую операцией o, $f[o]:S^{(i)} \Rightarrow S^{(i)}$. В множестве $R[o]$ выделим два подмножества $RW[o]$ и $RR[o]$, определяемые следующим образом:

$$r \in RW[o] \Leftrightarrow \exists\, s \in S^{(i)} \quad s(r) \neq f[o](s)(r);$$

$$r \in RR[o] \Leftrightarrow \exists\, r' \in RW[o],\ s_1, s_2 \in S^{(i)},\ \forall r'' \in R[o]\setminus\{r'\}$$

$$s_1(r'') = s_2(r''),\ s_1(r') \neq s_2(r) \Rightarrow f[o](s_1)(r') \neq f[o](s_2)(r').$$

Множество переменных уровня i $\bigcup_{\forall o\in O_j^{(i)}} R[o]$ обозначим через $R_j^{(i)}$, множество их значений- через $D_j^{(i)}$, множество состояний переменных $(D_j^{(i)})^{R_j^{(i)}}$ - через $S_j^{(i)}$, а множество переменных уровня i $\bigcup_{\forall o\in O_j^{(i)}} RW[o]$ через $RW_j^{(i)}$. При этом для $\forall i\ j \neq h \Rightarrow RW_j^{(i)} \cap RW_h^{(i)} = \emptyset$. Состояние $PR_h(s)$ из множества $S^{(h)}$ назовем проекцией состояния s из множества $S^{(i)}$ на уровне h, если для $\forall r \in R^{(i)}$ $PR_h(s)(r) = s(r)$.

На каждом уровне выделим специальное упорядоченное подмножество переменных $CC^{(i)}\subseteq R^{(i)}$, $\forall l\in L^{(i)}$ $s(l)\in\{0,1\}$, $i<h \Rightarrow CC^{(i)}\subseteq CC^{(h)}$. Вектор переменнных $CC^{(i)}$, состоящий из всех элементов множество $CC^{(i)}$, назовем вектором условий перехода уровня i, вектор его значений в состоянии $s\in S^{(i)}$- кодом условий в состоянии s.

© National Library of Armenia

Вектор $c = (|o^{(i)}_{c_1}, o^{(i)}_{c_2}, \ldots, o^{(i)}_{c_{n_i}}|)$ назовем командой уровня i, где $o^{(i)}_{c_j} \in O^{(i)}_j$ $(j=1,2,\ldots,n_i)$, а $o^{(i)}_{c_j}$-компонентом $comp(j,c)$ команды c. Множество команд уровня i, которое можно отождествлять с множеством $O^{(i)}_1; O^{(i)}_2; \ldots; O^{(i)}_{n_i}$, обозначим через $C^{(i)}$.

Для команды уровня i введем следующие обозначения:

$R[c] = \bigcup_{j=1}^{n_i} R[o_{c_j}]$, $RW[c] = \bigcup_{j=1}^{n_i} RW[o_{c_j}]$, $RR[c] = \bigcup_{j=1}^{n_i} RR[o_{c_j}]$,

$f[c](s) = (f[o_{c_1}](s), f[o_{c_2}](s), \ldots, f[o_{c_{n_i}}](s))$, где $s \in S^{(i)}$.

Реализацией $re_l(o,s)$ операции $o \in O^{(i)}_j$ на уровне l в состоянии $s \in S^{(i)}$ назовем последовательность команд $c_1, c_2, \ldots, c_{w(o)}$ уровня l такую, что $f[o](s)(r) = f[c_{w(o)}](\ldots f[c_2](f[c_1](PR_l(s)))\ldots))(r)$ для $\forall r \in RW[o]$.

Предположим, что множества $O^{(i)}$ построены таким образом, что для $\forall o \in O^{(i)}$ и $\forall s \in S^{(i)}$ $\forall l > i$ существует хотя бы одна реализация $re_l(o,s)$, причем $i=2k$, а $l=2k+1$, то $w(j)=1$, и для $\forall r_1, r_2 \in RW[o]$, $\exists$ уровень l такой, что выражение $(r_1 \in RW[o']) \vee (r_2 \in RW[o'])$ истинно для $\forall c_v$ из $re_l(o,s)$ и $\forall o' = comp(j, c_l)$.

Реализацией $re_l(b,s)$ команды b уровня i на уровне l в состоянии $s \in S^{(i)}$ назовем последовательность команд уровня l $c_1, c_2, \ldots, c_{w(b)}$ такую, что для $\forall r \in RW[o]$ $f[b](s)(r) = f[c_{w(b)}](\ldots f[c_2](f[c_1](PR_l(s)))\ldots))(r)$.

Разделяемой реализацией $re_l(b_1, b_2, s)$ команд b_1 и b_2 уровня i на уровне l в состоянии s назовем последовательность команд $c_1, c_2, \ldots, c_{w(b_1,b_2)}$ уровня l такую, что для $\forall r \in RW[o]$ $f[c_{(w(b_1,b_2)}](\ldots f[c_2](f[c_1](PR_l(s)))\ldots))(r) = f[b_2](f[b_1](s)(r))$.

Многоуровневая модель процессора A- это иерархическая структураб, каждый уровень i $(i=0,1,\ldots,2d)$ которой представляет собой автомат $A^{(i)}$, являющийся произведением автоматов $A^{(i)}_j$

$(j=1,2,\ldots,n_i)$, где n_i- число непересекающихся классов множества $O^{(i)}$. Каждый автомат $A^{(i)}_j$- это автомат, множеством состояний которого является $S^{(i)}_j$, начальным состоянием- $s^{(i)}_0$, множеством входных воз- действий - $O^{(i)}_j$, а функция переходов $\sigma^{(i)}_j$ и функция выходов $\psi^{(i)}_j$ такие, что $\forall o \in O^{(i)}_j$, $s \in S^{(i)}_j$, $\forall r \in R^{(i)}$ $\sigma^{(i)}_j(s,o)(r) = f[o](s)(r)$, $\psi^{(i)}_j(s,o) = f[c](s)(cc^{(i)})$. Будем говорить, что модель процессора A задает микропрограммный процессор машинных команд, если команда уровня 0 соответствует машинной команде, команда уровня $2d$- микрокоманде, а операция уровня $2d$- микрооперации, причем операции с различными операндами воспринимаются как различные воздействия.

Для каждой команды c уровня i и для произвольного уровня $l \geq i$ существует хотя бы один граф реализации на уровне l $G^{(l)}(c)$, представляющий собой ориентированный помеченный граф, вершины которого соответствуют командам из $C^{(l)}$, а дуги помечены кодами условий уровня l. Одна дуга может быть помечена несколькими кодами условий, множество всех отметок дуг, исходящих из одной вершины, совпадает с множеством всех кодов условий уровня l, а две дуги, исходящие из одной вершины, не могут быть помечены одним и тем же кодом условий. В $G^{(l)}(c)$ выделены входная и выходная вершины.

Путем выполнения в графе $G^{(l)}(c)$ в состоянии s автомата $A^{(l)}$ назовем путь в графе, состоящий из вершин, помеченных командами $c_1, c_2, \ldots, c_{p(s)}$, где c_1 соответствует входной, $c_{p(s)}$- выходной вершине, а дуга, связывающая вершину c_{v-1} с вершиной c_v $(v=2,3,\ldots,p(s))$, помечена кодом условий, совпадающим со значением функции выходов автомата $A^{(l)}$ в состоянии $\sigma_l(\ldots\sigma_l(\sigma_l(s,c_1),c_2),\ldots),c_{v-1})$. Последовательность команд $c_1, c_2, \ldots, c_{p(s)}$, соответствующих вершинам пути выполнения, обозначим через $p(s, G^{(l)}(c))$.

Граф реализации назовем графом частной реализации команды c уровня i на уровне $l \geq i (l=2k)$ по множеству $S' \subset S^{(i)}$, и обозначим через $GP^{(l)}(c,S')$, если для $\forall s \in S'$ $p(s, G^{(l)}(c)) \in RE_l(c,s)$. и графом универсальной реализации

© National Library of Armenia

$GN^{(l)}(c)$, если $S'=S^{(i)}$. Множество графов частной реализации всех команд уровня i на уровне $l\geq i(l=2k)$ по множеству $S'\subset S^{(i)}$ обозначим через $QP^{(l)}(S')$.

С каждым графом частной реализации $GP^{(l)}(c,S')$ свяжем $rc(rc\geq 1)$ целочисленных положительных характеристик $\alpha_1,\alpha_2,\ldots,\alpha_{rc}$, определяющих эффективность реализации по различным критериям (объем, время выполнения, точность вычисления и.т.д.). Будем считать, что граф частной реализации $GP_1^{(l)}(c,S')$ не хуже графа частной реализации $GP_2^{(l)}(c,S')$ по характеристике α_q, если $\alpha_q(GP_1^{(l)}(c,S'))\leq\alpha_q(GP_2^{(l)}(c,S'))$. Будем считать граф частной реализации $GP^{(l)}(c,S')$ оптимальным графом частной реализации по множеству S' и характеристике α_q, если он не хуже по характеристики α_q любого другого графа частной реализации по множеству S'. Графы частной реализации соответствуют частным случаям выполнения команд, что и обуславливает их рассмотрение только на четных уровнях модели.

Введем также понятие графа пустой реализации, соответствующего удалению команды при ее реализации.

С произвольной командой c уровня i связывается некоторая переменная - параметр частичной реализации $par^{(l)}[c]$, характеризирующий ее реализацию на уровне $l\geq i(l=2k)$ и принимающий значение в множестве $\{S'|GP^{(l)}(c,S')\in QP^{(l)}(c)\}$ и будем считать, что каждому значению параметра соответствует некоторый граф частной реализации команды c (В частности, в качестве графа частной реализации можно рассматривать граф универсальной реализации или граф пустой реализации).

Значение параметра, соответствующее универсальному графу реализации обозначим через $uparv^{(l)}[c]$, а значение, соответствующее графу пустой реализации - через $eparv^{(l)}[c]$.

Пусть c- команда уровня i, $G(c)$- граф, состоящий из одной вершины, помеченной командой c. Преобразованием $TR^{(l)}(c,S)$ команды c на уровне $l\geq i)$, задаваемым значением S параметра частной реализации $par^{(l)}[c]$, называется замена графа $G(c)$ графом частной реализации $GP^{(l)}(c,S')$. При этом преобразование $TR^{(l)}(c,S)$ и значение S параметра частной реализации $par^{(l)}[c]$

называются оптимизирующими по характеристике α_q при переходе от уровня i к уровню $l>i$, если $\alpha_q(GP^{(l)}(c,S'))<\alpha_q(GP^{(i)}(c,S'))$, а преобразование $TR_1^{(l)}(c,S_1)$ считается лучше преобразования $TR_2^{(l)}(c,S_2)$ по характеристике α_q при переходе от уровня i к уровню $l>i$, если $\alpha_q(GP^{(l)}(c,S_1))<\alpha_q(GP^{(l)}(c,S_2))$. Преобразование, задаваемое значением $uparv^{(l)}[c]$ обозначается через e, а преобразование, задаваемое значением $eparv^{(l)}[c]$ - через del.

Для нечетных уровней $l=2k+1$ модели аналогично графу частной реализации команды определим граф разделяемой реализации $GU^{(l)}(c,b)$ для команд c и b уровня i на ~~уровне l~~, путь выполнения и последовательность команд $p(s,GU^{(l)}(c,b))$ в графе $GU^{(l)}(c,b)$. Граф реализации назовем графом разделяемой реализации по множеству Z, определяющему информационные связи между переменными команд c и b уровня i, и обозначим через $GU^{(l)}(c,b,Z)$, если для $\forall z\in Z$, $\forall s\in S^{(i)}$ $p(s,GU^{(l)}(c,b,Z))\in RE_1(c,b,s)$.

С парой команд c и b уровня i свяжем некоторую переменную - параметр разделяемой реализации $par^{(l)}[c,b]$, характеризирующий ее реализацию на уровне $l\geq i$ $(l=2k+1)$ и принимающий значение в множестве $\{Z'|GU^{(l)}(c,b,Z')\in QP^{(l)}(c,b)\}$. Установим соответствие между значениями параметра разделяемой реализации и графами разделяемой реализации, а также определим преобразование и оптимизирующее преобразование, задаваемые значениями параметра разделяемой реализации.

Поскольку параметры частной и разделяемой реализаций действуют на различных уровнях, то с каждой командой можно одновременно связать все комбинации параметров: параметр частной реализации, параметр разделяемой реализации, оба параметра вместе.

При этом в случае двух параметров уровень одного из параметров должен совпадать с уровнем команды, а преобразование команды с двумя параметрами состоит в последовательном применении преобразований, соответствующих отдельным параметрам, начиная с преобразования по параметру, уровень

© National Library of Armenia

которого равен уровню команды.

Программой $P^{(i)}$ уровня i называется ориентированный помеченный граф, вершины которого соответствуют командам из $C^{(i)}$, а дуги помечены кодами условий уровня i, причем требования на отметки дуг аналогичны требованиям на отметки дуг в графе реализации команды. Число вершин программы $P^{(i)}$ обозначим через N_P. Будем считать, что в программе $P^{(i)}$ существует одна входная вершина c_N и одна выходная вершина c_X. Аналогично пути выполнения и последовательности команд $p(s,G(c))$ в графе реализации определим путь выполнения и последовательность команд $p(s,P^{(i)})$ для программы $P^{(i)}$ на автомате $A^{(i)}$ при входном состоянии $s\in S^{(i)}$. Если последовательность $p(s,P^{(i)})$ состоит из команд $c_{s_1},c_{s_2},\dots,c_{s_P}$, то состояние $\sigma_i(\dots\sigma_i(\sigma_i(s,c_{s_1}),c_{s_2}\dots c_{s_P}\dots)$ автомата $A^{(i)}$ назовем результатом работы программы $P^{(i)}$ при входном состоянии $s\in S^{(i)}$ и обозначим через $se(s,P^{(i)})$, а через $se(s,c,P^{(i)})$ обозначим состояние, в котором оказывается автомат $A^{(i)}$ после выполнения команды c при входном состоянии s.

Пусть p- вершина программы $P^{(i)}$, помеченная командой c. Множество всех состояний автомата $A^{(i)}$ перед выполнением команды c программы $P^{(i)}$ обозначим через $EN(c,P^{(i)})$ и назовем множеством входных состояний команды c в программе $P^{(i)}$, $EN(c,P^{(i)})\subseteq S^{(i)}$. Множество $EN(c_N,P^{(i)})$ назовем входным множеством программы $P^{(i)}$. Кроме того, будем считать, что с каждой командой c программы $P^{(i)}$ связано некоторое множество характеристик $Z(c,P^{(i)})$, определяющих информационные связи между командами. Граф $G(c,S')$ частной реализации команды c программы $P^{(i)}$ назовем допустимым для программы $P^{(i)}$, если $EN(c,P^{(i)})\subseteq S'$, а граф $G(c,b,Z')$ разделяемой реализации команд c и b программы $P^{(i)}$- допустимым для программы, если $Z(c,P^{(i)})\cap Z(b,P^{(i)})\subseteq Z'$. Соответственно, значение параметра $par^{(l)}[c]$ команды c уровня i является допустимым для программы, если соответствующий ему граф реализации является допустимым для программы. Значение S' допустимого параметра частичной реализации команды c уровня $i(i=2k)$ будем считать оптимальным на данном уровне по характеристике α_q, если для любого другого

допустимого значения S'' этого же параметра $\alpha_q(GP^{(i)}(c,S'))\le\alpha_q(GP^{(i)}(c,S''))$.

Программы $P_1^{(i)}$ и $P_2^{(i)}$ назовем эквивалентными, если для $\forall s\in EN(c_N,P_1^{(i)})\cup EN(c_N,P_2^{(i)})$, $\forall r\in R^{(i)}$ $se(s,P_1^{(i)})(r)=se(s,P_2^{(i+1)})$.

Программы $P_1^{(i)}$ и $P_2^{(l)}(l>i)$ назовем эквивалентными на уровне i, если для $\forall s\in EN(c_N,P_1^{(i)})$, $\forall r\in R^{(i)}$ $se(s,P_1^{(i)})(r)=se(PR_1(s),P_2^{(i+1)})$.

Пусть $P^{(i)}$-программа уровня i, состоящая из N_P вершин, пронумерованных от 1 до N_P, вершина с номером v помечена командой c_v и с каждой командой c_v связаны $v[p]$ $(v[p]\in\{0,1,2\})$ параметров реализации. Вектор, состоящий из значений параметров команд программы, обозначим через $\overline{par}^{(l)}$, где l- максимальный уровень параметров программы. Реализацией $RE(P^{(i)},\overline{par}^{(l)})$ программы $P^{(i)}$ уровня i на уровне l по вектору значений $\overline{par}^{(l)}$ назовем программу, полученную в результате преобразований команд программы $P^{(i)}$, задаваемых компонентами вектора значений $\overline{par}^{(l)}$. Если значения всех компонентов вектора значений $\overline{par}^{(l)}$ являются допустимыми для программы $P^{(i)}$, то вектор $\overline{par}^{(l)}$ считается корректным. Имеет место следующая теорема об эквивалентности программы $P^{(i)}$ и ее реализации $RE(P^{(i)},\overline{par}^{(l)})$ по корректному вектору значений параметров $\overline{par}^{(l)}$.

Утверждение. Пусть $P^{(i)}$ программа уровня i, а $\overline{par}^{(l)}$ корректный вектор значений параметров. Тогда $P^{(i)}$ эквивалентна программе $RE(P^{(i)},\overline{par}^{(l)})$ на уровне i.

Вектор $\overline{par}^{(l)}$ называется оптимизирующим для программы $P^{(i)}$ при переходе от уровня i к уровню l, если он содержит значение какого-либо параметра, задающего оптимизирующее преобразование при переходе от уровня i к уровню l.

Пусть $\overline{par}_1^{(l)}$ и $\overline{par}_2^{(h)}$ корректные вектора значений параметров для программы $P^{(i)}$. Будем считать, что реализация $RE(P^{(i)},\overline{par}_1^{(l)})$ лучше реализации $RE(P^{(i)},\overline{par}_2^{(h)})$ по характеристике α_q, если

© National Library of Armenia

$$\sum_{j=1}^{N_P} \alpha_q(G_1^{(l)}(c_j)) < \sum_{j=1}^{N_P} \alpha_q(G_2^{(h)}(c_j)),$$

где $G_1^{(l)}(c_j))$ и $G_2^{(h)}(c_j))$- графы реализаций, определяемые векторами $\overline{par}_1^{(l)}$ и $\overline{par}_2^{(h)}$ соответственно. Корректный вектор $\overline{par}_1^{(l)}$ называется оптимальным вектором программы $P^{(i)}$ уровня l по характеристике α_q, если не существует корректного вектора $\overline{par}_2^{(l)}$ такого, что реализация $RE(P^{(i)}, \overline{par}_2^{(l)})$ лучше реализации $RE(P^{(i)}, \overline{par}_1^{(l)})$ по характеристике α_q, и просто оптимальной по характеристике α_q, если не существует уровня $h>l$ и корректного вектора $\overline{par}_2^{(h)})$ уровня h для программы $RE(P^{(i)}, \overline{par}_1^{(l)})$ таких, что $RE(RE(P^{(i)}, \overline{par}_1^{(l)}), \overline{par}_2^{(h)})$ лучше $RE(P^{(i)}, \overline{par}_1^{(l)})$ по характеристике α_q.

Из проведенных рассмотрений вытекает метод синтеза эффективной реализации программы уровня i на произвольном уровне l, заключающийся в последовательном решении трех следующих задач:

I. Построение множеств частных и разделяемых реализаций для команд уровня i.

2. Построение оптимального вектора корректных параметров для исходной программы, соответсвующих полученным множествам реализаций.

3. Генерация реализации программы по оптимальному вектору параметров.

В работе указана возможность использования ранее полученных результатов для решения задачи 2 и детально рассмотрено решение задач I и 3.

Разработан алгоритм получения частных реализаций команд, имеющих линейную внутреннюю структуру и доказана его правильность.

В §2.2 введено понятие схемы реализации, характеризующей структуру реализации, приведен алгоритм генерации реализации и обоснована возможность использования в качестве основы

генератора реализаций готового генератора расширений макроконструкций некоторого языка программирования. Предложен алгоритм перевода схемы реализации в макроопределение.

В §2.3 описан алгоритм проверки корректности класса линейных программ, обрабатывающих булевские массивы и использующих групповые операторы присваивания. Алгоритм имеет полиномиальную сложность и может быть применен для проверки корректности линейных программ обработки булевских массивов, в которых выражения в правых частях операторов присваивания, входящих в групповой оператор, либо константы, либо не используют общих переменных.

В главе 3 рассмотрены основные характеристики языкового и программного обеспечения для автоматической генерации системных программ ПР.

В §3.I описан процесс построения реализации по исходной программе в системе автоматического проектирования плат расширения, приведены характеристики и основные параметры промежуточного языка.

Построение реализации по исходной программе состоит из двух шагов.

На первом шаге выполняется перевод исходной программы в программу на промежуточном языке с помощью последовательного применения системы преобразований. Преобразовния этой системы выполняют либо удаление оператора или группы операторов, либо замену одной последовательности операторов другой, либо уточнение параметров оператора. Применяемость преобразования определяется наличием характеристик несущественности выполнения оператора или группы операторов перехода или наличием соотношений между переменными программы,к которой применяются преобразования. Условия применяемости преобразования проявляются на различных уровнях: для отдельных операторов, этапов выполнения данного оператора, элементарных действий при выполнении этапа и.т.д.

По результирующей программе на промежуточном языке с помощью схем реализации (заготовок) операторов промежуточного языка генерируется реализация программ.

При этом множество параметров операторов промежуточного языка разбито на 3 группы:

© National Library of Armenia

I. Параметры, уточняющие алгоритмы выполнения данного оператора или последовательности операторов.

2. Параметры, уточняющие алгоритмы выполнения отдельных этапов данного оператора.

3. Параметры, уточняющие алгоритмы выполнения отдельных действий во время выполнения данного этапа.

Параметры группы $k(k=2,3)$ могут присутствовать операторе промежуточного языка только в том случае, если в нем присутствуют параметры группы $k-1$.

В соответствии с используемыми группами параметров можно считать, что есть четыре уровня промежуточного языка L:

I. Уровень L_0 - это операторы промежуточного языка без рассмотренных выше параметров и, в частности, операторы уровня i.

2. Уровень L_i - это операторы уровня $k-1$, дополненные параметрами группы $k(k=I,2,3)$.

Преобразования операторов промежуточного языка всех уровней задаются для каждого отдельного оператора языка и имеют единое представление:

ЕСЛИ <схема соотношения> ТО <схема выполняемого действия>

Схема соотношения - это выражение вида $t_1 \alpha t_2$, где t_1, t_2 - термы, составленные из символов переменных исходной программы, констант и функциональных символов, соответствующих действиям, выполняемым операторами исходной программы, α - символ некоторого бинарного отношения, характеризующего зависимости между значениями и/или адресами переменных перед выполнением операторов.

Схема выполняемого преобразования имеет вид
<тип преобразования> <список изменяемых параметров промежуточного языка>

Могут быть три типа преобразований: добавление - ADD удаление - DEL изменение - CHNG.

Список изменяемых параметров - это совокупность изменяемых параметров, отделенных друг от друга запятой. Список параметров заключается в скобки. Каждый изменяемый параметр в свою очередь

представляется следующим образом:

<имя параметра> = t, где t - терм.

Применяемость преобразования к данному оператору OP определяется следующим образом.

Проверяется инвариантность соотношения, задаваемого схемой, т.е. истинность соотношения для всех возможных текущих значений адресов и операндов перед выполнением оператора OP.

Если соотношение инвариантно, то в зависимости от типа преобразования происходит добавление, удаление или изменение параметров оператора. Новые значения параметров определяются значениями термов в схеме выполнения преобразования.

Если инвариантность соотношения невозможно проверить, или если соотношение не является инвариантным, то преобразование не применяется к данному оператору.

Поскольку количество параметров операторов конечно, а термы в схеме выполнения преобразований для различных параметров не содержат общих переменных, то из результата §2.3 следует, что при наличии полной системы преобразований для отдельных операторов языка можно построить полную систему преобразований для произвольных последовательностей операторов и линейных участков в программе. Таким образом, для программ линейной структуры в данном случае можно достаточно эффективно получать правильные и оптимальные реализации. Например, драйверы для рассматриваемых ПР в среде MS-DOS имеют линейную структуру (заголовок, стратегия устройства, программа обработки прерываний и, следовательно, можно построить эффективные алгоритмы их автоматической генерации.

В §3.2 на основе результата §2.I предложен алгоритм построения частных реализаций произвольных команд и программ. Описан макрогенератор, используемый в качестве инструментального средства в системе автоматического проектирования ПР ПК, совместимых с IBM PC, и показано использование макрогенератора при генерации системных программ в системе автоматического проектирования ПР.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В диссертационной работе получены следующие результаты:

© National Library of Armenia

I. Обоснован подход к решению задачи автоматического проектирования унифицированных семейств плат расширения персональных компьютеров.

2. Выделены основные этапы процесса получения реализации по исходному описанию алгоритма и предложена структура соответствующей базы знаний.

3. Построена многоуровневая вычислительная модель и разработан метод синтеза реализаций программы на введенной модели.

4. Разработаны алгоритмы генерации реализации программы по оптимальному вектору корректных параметров.

5. Разработаны алгоритмы получения частных реализаций команд и программ.

6. Описан алгоритм проверки корректности для класса линейных системных программ для плат расширения.

7. Показано применение разработанных алгоритмов при автоматической генерации системных программ для плат расширения персональных компьютеров, совместимых с IBM PC.

По теме диссертации опубликованы следующие работы:

I.Ашур А.А., Шукурян С.К. Генерация реализации программы для многоуровневой вычислительной модели с использованием макропроцессора.//Рукопись, депонированная в АрмНИИНТИ I4.07.93, : I8-Ар93.

2.Арутюнян Л.Л., Ашур А.А., Шукурян С.К. Метод синтеза реализаций программы для многоуровневой вычислительной модели. //Рукопись, депонированная в АрмНИИНТИ OI.II.93, : 35 -Ар93.

أيمن عاشور

Р Е З Ю М Е

Аймен Али Ашур "Автоматическая генерация программного обеспечения специальных плат расширения персональных компьютеров"

Диссертационная работа Аймана А.А. посвящена обоснованию и разработке алгоритмов и программ инструментальной оболочки для автоматизированной генерации системного программного обеспечения по описанию платы расширения на языке высокого уровня.

В работе получены следующие основные научные результаты:

1. Построена многоуровневая модель процессора и в рамках этой модели предложен метод синтеза реализаций программ для произвольного уровня модели.
2. Разработаны алгоритмы для этапа генерации реализации программ в методе синтеза.
3. Показано приложение математических методов к построению полных инвариантов программ для проверки корректности системных программ специальных плат расширения персональных компьютеров.
4. Разработаны основные конструкции языкового обеспечения и утилиты программного обеспечения для автоматизированной генерации системных программ.

Результаты диссертации могут быть использованы при решении различных задач автоматизации процессов разработки системных программ для специальных плат расширения персональных компьютеров в различных отраслях экономики.

© National Library of Armenia

15.05.2014

ԱՐՁԱՆԱԳՐՈՒԹՅՈՒՆ

«Ազգային գրադարանի գրքերի ինվենտարացման գործընթացի կազմակերպման ու անցկացման մասին» ՀՀ մշակույթի նախարարի հրաման

Ղեկավարվելով «Գրադարանային գործի մասին» ՀՀ օրենքի 4-ի 1-ին մասի 1-ին կետի պահանջներով, հրամայում եմ.

1. Սահմանել, որ ՀՀ գրադարանային գործի կազմակերպիչ մարմինը՝ ՀՀ մշակույթի նախարարությունը, ապահովում է գրքերի ինվենտարացման գործընթացի կազմակերպումը ու անցկացումը:
2. Սինթեմատ մեթոդով ինվենտարացման գործընթացում մշակել ՀՀ գրադարանների գրքերի ինվենտարացման գործընթացի կանոնակարգ, որում ներառել հետևյալ մեթոդները:
3. Ներդրել և տարածքային գրադարանների գրքերի ինվենտարացման գործընթացում կիրառել գրքերի ինվենտարացման գործընթացի կանոնակարգ՝ ՀՀ գրադարանների գրքերի ինվենտարացման գործընթացի կանոնակարգ:
4. Համակարգչային գրադարանի գրքերի ինվենտարացման գործընթացում կիրառել գրքերի ինվենտարացման գործընթացի կանոնակարգ՝ ՀՀ գրադարանների գրքերի ինվենտարացման գործընթացի կանոնակարգ:

Սույն հրամանն ուժի մեջ է մտնում հրապարակմանը հաջորդող տասնհինգերորդ օրվանից:

Նախարար՝ Ս. Ավետիսյան

© National Library of Armenia

© National Library of Armenia