

На правах рукопису

Шахновський Юрій Сергійович

АЛГОРИТМИ ОПТИМІЗАЦІЇ ГОРИЗОНТАЛЬНИХ МІКРОПРОГРАМ

05.13.11 - математичне і програмне забезпечення  
обчислюваних машин та систем

Автореферат дисертації на здобуття вченого ступеня  
кандидата технічних наук

Харків - 1994

Праця виконана у Харківському політехнічному університеті

Наукові керівники:

- доктор технічних наук,  
професор Костенко Д. Т.
- кандидат технічних наук,  
доцент Малих О. М.

Офіційні опоненти:

- доктор фіз.-мат. наук,  
професор Яковлев С. В.
- кандидат технічних наук,  
доцент Мілов О. В.

Ведуче підприємство:

- Інститут кібернетики  
ім. В. М. Глушкова  
АН України

Захист відбудеться " 17 " березня 1994 р. о "     " годині  
на засіданні спеціалізованої ради К 068.37.03 у Харківському  
технічному університеті радіоелектроніки (310059, м. Харків,  
пр. Леніна, 14).

З дисертацією можна ознайомитися у бібліотеці Харківського  
технічного університету радіоелектроніки.

Автореферат розіслано " 28 " січня 1994 р.

Вчений секретар  
спеціалізованої ради  
канд. техн. наук, доцент

*Д. Сукесов*

Сукесов Е. А.

ЛННБ України ім. В. Стефаника



00753683 (W)

ЛННБ ім. В. Стефаника  
АН України

## ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність проблеми. Мікропрограмне управління мають центральні процесори класу 8086, співпроцесори арифметики / вводу-виводу типу 8087/8089 та трансп'ютери. Крім того мікропрограмне управління це основний спосіб реалізації обчислень в спеціалізованих багатопроцесорних обчислювачах. Такий варіант управління спрощує схемотехнічну реалізацію мікропроцесорів та обчислювачів, тому що кожна команда реалізується не апаратно, а шляхом виконання інтерпретувачої мікропрограми (МП), котра розміщується в управляючій пам'яті.

Найбільш складну організацію має горизонтальне мікропрограмне управління, при якій в горизонтальних мікрокомандах (МК), складаних МП, можуть одночасово розмішуватися і, як наслідок, виконуватися декілька операцій.

Складність процесу заповнення, тобто упаковки, МК операціями з врахуванням усього спектру інформаційних та ресурсних обмежень обумовила розробку систем автоматизації мікропрограмування, елементом яких являється підсистема упаковки мікрокоду.

Відрізняють локальну та глобальну упаковку. При локальній упаковці операції одного лінійного участку мікропрограми (ЛУМ) розміщуються в МК без врахування операцій інших ЛУМ. Під час глобальної упаковки одночасно розміщуються операції різних ЛУМ.

Вибір задачі локальної упаковки як основного об'єкту досліджень зумовлений тим, що локальна упаковка являється основою глобальної і для задачі локальної упаковки відсутні ефективні методи рішень. Останнє пояснюється тим, що дана задача відноситься до розряду комбінаторних багатоекстремальних задач з важко знайденим допустимим рішенням. Цільова функція задачі визначена на множині перестановок операцій, складаних ЛУМ.

Традиційною моделлю задачі локальної упаковки являється граф залежності з даними (ГЗД), відбиваючий умови зв'язку операцій з даних. Ці умови встановлюють порядок використання спільних для операцій ресурсів з пам'яттю (типу регістр) обчислювача. Порядок використання таких ресурсів визначається початковим представленням ЛУМ. Джерелом варіантності розв'язання при фіксованому ГЗД являються форматні та функціональні обмеження, що визначають правила заповнення горизонтальних МК та порядок завантаження функці-

ональних ресурсів (комбінаційних схем та спільних ліній зв'язку). Серед умов взаємодії операцій з даних особливу роль відіграють ті з них, котрі характеризують обмін даними між операціями, тобто визначають потоки даних в початковому ЛУМ. Останні умови взаємодії операцій на статичних ресурсах встановлюють порядок їх використання різноманітними потоками даних ЛУМ.

Суть нового підходу до вирішення задачі локальної упаковки полягає в тому, що як початкові розглядаються тільки ті умови зв'язку між операціями, які визначають обмін даними, а останні умови передумання на статичних ресурсах формуються в процесі розв'язування задачі. Інакше кажучи, ГЗД скорочується до графу обмінів, створеного заданими потоками даних.

При такому підході предметом пошуку являється не тільки порядок використання функціональних ресурсів, але і послідовність загрузки ресурсів з пам'яттю різноманітними потоками даних, складаючими ЛУМ. Це забезпечує вихід на якісно новий простір рішень, котрі раніш не враховувались при реалізації даної задачі.

Метод роботи являється:

розробка методів та алгоритмів одержання допустимих рішень задачі локальної упаковки з нової області пошуку;

розробка алгоритмів, орієнтованих на кращу якість рішень, які формуються;

розповсюдження розроблених методів розв'язування задачі локальної упаковки на глобальну упаковку мікрокоду.

Методи досліджень будуються на теорії графів, теорії розпису, теоретичному програмуванні та теорії складності алгоритмів.

Наукова новизна роботи: 1) зроблена постановка задачі локальної упаковки, яка дозволила установити новий простір рішень; 2) доведені умови існування допустимого рішення в задачі локальної упаковки; 3) розроблені два методи та їх алгоритмічна реалізація для одержання допустимого рішення локальної упаковки; 4) розроблені 8 алгоритмів для покращення якості допустимого рішення локальної упаковки; 5) розроблено алгоритм, дозволяючий використовувати пріоритети теорії розпису в задачі локальної упаковки; 6) доведені коректність та кінцевість, досліджена складність розроблених алгоритмів; 7) розроблені 3 алгоритми глобальної упаковки, основані на запропонованих методах локальної

упаковки; 8) проведено числений експеримент, який дозволив встановити обчислювану ефективність розроблених алгоритмів.

Практична цінність роботи. Розроблений комплекс алгоритмів та програм використовувався при створенні систем автоматизації мікропрограмування, які були випробувані та впроваджені у НВО точних приладів (м. Москва) та у НДІ приладобудування (м. Москва). Загальний економічний ефект від впровадження розробок склали 60 тисяч карбованців (1990 р.).

Апробація результатів праці Результати роботи доповідались на семінарі "Теория автоматов и ее применение" в інституті кібернетики академії наук України (Київ, 1990) та на I Всесоюзній науково-технічній конференції "Координирующее управление в технических и "природных системах" (Крим, 1991).

Публікації. Результати роботи відбиті в дев'яти друкованих роботах.

Обсяг і структура роботи. Дисертація складається з вступу, п'яти розділів, заключення, викладених на 130 сторінках; списку використаної літератури з 30 назв та трьох додатків; містить 19 малюнків.

Перший розділ присвячено змістовній постановці задачі локальної і глобальної упаковки мікрокоду та аналізу відомих методів їх розв'язку, на базі якого сформульовано перелік задач досліджень та розробок; другий розділ - формування та аналізу математичної моделі задачі локальної упаковки мікрокоду, методам будування допустимих рішень та способам покращення їх якості при врахуванні обмежень на ресурсах з пам'яттю; третій розділ - способам покращення якості мікропрограми при врахуванні форматних та функціональних обмежень; четвертий розділ - методам глобальної упаковки; п'ятий розділ - оцінці обчислюваної ефективності розроблених алгоритмів.

Перший додаток містить приклади, ілюструючі роботу системи мікропрограмування та розроблених алгоритмів упаковки; другий додаток - результати експериментального дослідження алгоритмів; третій додаток - інформація про впровадження результатів роботи.

## ЗМІСТ РОБОТИ

Задача локальної упаковки горизонтального мікрокоду вважається

ся заданою, якщо визначені:

1. Початкова послідовність з  $n$  операцій  $Q = (O_i | i = \overline{1, n})$ , складаючих лінійний учасок вертикальної МП.

2. Для кожної операції  $O_i$  перерахунок використаних ресурсів, що складає множину ресурсів з пам'яттю  $P_i$  (тобто статичних ресурсів) та множину функціональних ресурсів (тобто без пам'яті)  $\Phi_i$ . Множина  $P_i$  складається з двох підмножин  $Z_i$  та  $Ч_i$ , ресурси яких використовуються відповідно в режимах запису та читання.

3. Моменти початку  $T_{i\Gamma}$  і завершення  $TE_{i\Gamma}$  інтервалу використання кожного ресурсу  $\Gamma \in P_i \cup \Phi_i$  відносно початку, в загальному випадку, багатофазної  $O_i$ .

4. Формати мікрокоманд обчислювача як множина складаючих їх полів  $F_1 = (P_{1j} | j = \overline{1, J_1})$ ,  $1 = \overline{1, L}$ .

5. Множина полів  $\langle P_{1j} \rangle$  форматів мікрокоманд для розміщення кожної  $O_i$ .

Нехай  $O_i < O_j$  позначає умову передування операцій в початковій послідовності.

Операція  $O_i$  взаємодіє з  $O_j$ , якщо існує такий ресурс з пам'яттю  $\Gamma$ , що

$$\begin{cases} \Gamma \in Z_i \cap P_j, \\ \Gamma \in P_i \cap Z_j. \end{cases}$$

Взаємодіючі операції  $O_i$ ,  $O_j$  обмінюються інформацією по ресурсу з пам'яттю  $\Gamma$  та  $O_i$  являється джерелом, а  $O_j$  - приймачем, якщо:

$$\begin{cases} \Gamma \in Z_i \cap \Phi_j, \\ O_i < O_j, \\ \forall k \text{ такого, що } \Gamma \in Z_k, \text{ виконується } \begin{cases} O_k < O_i, \\ O_j < O_k. \end{cases} \end{cases}$$

Належність будь-якої операції  $O_i$  до МК горизонтальної МП визначається моментом початку її виконання  $T_i$ . Якщо інтервал виборки МК, тобто машинний цикл обчислювача, являється однофазним, одну МК складають операції з однаковим моментом початку. У випадку багатофазного машинного циклу одну МК складають усі операції, моменти початку яких належать до одного циклу.

В допустимій горизонтальній МП моменти початку операцій повинні відповідати слідуючим обмеженням:

1. Якщо  $O_i$  обмінюється з  $O_j$  по  $\Gamma$ , то

$$\left\{ \begin{array}{l} T_j \geq T_i + TE_{ig} - TB_{jg}, \\ \forall O_k \text{ такої, що } g \in Z_k, \text{ виконується } \left[ \begin{array}{l} T_k \leq T_i + TB_{ig} - TE_{kg}, \\ T_k \geq T_j + TE_{jg} - TB_{kg}. \end{array} \right. \end{array} \right.$$

2. Якщо операція  $O_i$  взаємодіє з  $O_j$  на ресурсі  $g$ , то

$$\left\{ \begin{array}{l} T_j \leq T_i + TB_{ig} - TE_{jg}, \\ T_j \geq T_i + TE_{ig} - TB_{jg}. \end{array} \right.$$

3. Якщо операція  $O_i$  має спільний функціональний ресурс  $g$  с  $O_j$ , то

$$\left\{ \begin{array}{l} T_j \leq T_i + TB_{ig} - TE_{jg}, \\ T_j \geq T_i + TE_{ig} - TB_{jg}. \end{array} \right.$$

4. Якщо  $T_i, T_j$  належить одному машинному циклу, то існує формат  $F_1$  такий, що

$$\left\{ \begin{array}{l} \Pi_{1i}, \Pi_{1j} \in F_1, \\ \Pi_{1i} \neq \Pi_{1j}. \end{array} \right.$$

Необхідно сформуувати горизонтальну МП, тобто упакувати складарчі її МК операціями так, щоб мінімізувати заданий критерій якості  $\max T_i$ .

Математична модель локальної упаковки. Нехай  $PB$  - множина пар, взаємодірчих операцій, а  $PO$  - множина пар, взаємодірчих операцій яких обмінюються даними.

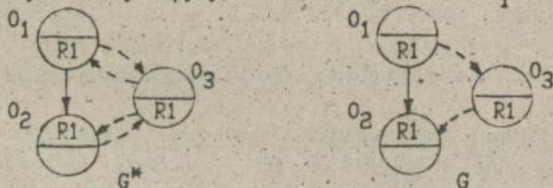
Графовор моделлю обмінів початкової послідовності операцій  $O$  являється безконтурний оргграф обмінів  $GO=(O, DO)$ , де множина дуг обмінор  $DO=\{(i, j) | (O_i, O_j) \in PO \text{ и } O_i < O_j\}$ .

Будемо вважати, що в множині пар  $PB \setminus PO$  порядок використання операціями спільних статичних ресурсів не визначається початковою послідовністю  $O$ . Інакше кажучи, між операціями кожної пари з  $PB \setminus PO$  відсутня умова передування, не дивлячись на її існування в  $O$ . Але поскільки для кожної пари операцій повинен бути розв'язаний конфлікт на спільному ресурсі, тобто встановлений порядок його використання, можливі варіанти розв'язання конфліктів зображуються шляхом доповнення  $GO$  множиною диз'юнктивних дуг взаємодії  $DDB=\{(i, j), (j, i) | (O_i, O_j) \in PB \setminus PO\}$ . У результаті отримуємо диз'юнктивний граф взаємодії  $DGV=(O, DO \cup DDB)$ .

Поскільки множина  $DDB$  зображає сукупність потенційних умов передування операцій, конкретизація останніх здійснюється вибором одного з двох можливих напрямків для кожної диз'юнктивної дуги. У

результаті множина ДДВ перетворюється в множину дуг взаємодії  $ДВ = \langle (i, j) \mid \langle O_i, O_j \rangle \in ПВ \setminus ПО \rangle$ , які визначають оргграф взаємодії  $ГВ = (O, ДО \cup ДВ)$ .

Проте не у всякому довільно знайденому ГВ відповідає виконувану МП. Одним з умов недопустимості формуваного ГВ являється наявність у ньому заборонених підграфів, появлення яких у процесі орієнтації диз'юнктивних дуг приводить до порушення умов обміну. Нехай  $G^*$  - деякий підграф графу ДГВ, в якому ресурси читання  $Ч_1$  кожної операції  $O_i$  зображуються над діаметром, а ресурси запису  $З_1$  - під діаметром у відповідній вершині. У цьому підграфі  $(1,2)$  являється дугою обміну, а вершини пар  $(O_1, O_3)$  і  $(O_2, O_3)$  з'єднані диз'юнктивними дугами. У результаті розв'язання конфліктів на диз'юнктивних дугах одержимо заперечний підграф  $G$ , умови зв'язку в якому порушують наданий обмін між  $O_1$  і  $O_2$ .



Твердження. ГВ являється допустимим тоді і тільки тоді, коли у ньому не існує заперечних підграфів та контурів.

При наявності в початковій послідовності  $O$  множини пар операцій ПФ з спільними функціональними ресурсами варіантність порядків їх використання зображується множиною диз'юнктивних дуг  $ДДФ = \langle (i, j), (j, i) \mid \langle O_i, O_j \rangle \in ПФ \rangle$ . При доповненні ДДФ до ГВ одержуємо диз'юнктивний граф  $ДГФ = (O, ДО \cup ДВ \cup ДДФ)$ , відбиваючий варіантність використання функціональних ресурсів. Вибір одного з двох можливих напрямків для кожної диз'юнктивної дуги в ДГФ приведе до оргграфу порядку використання спільних ресурсів мікропрограми  $ГМП = (O, ДО \cup ДВ \cup ДФ)$ , де  $ДФ = \langle (i, j) \mid \langle O_i, O_j \rangle \in ПФ \rangle$ .

ГМП являється допустимим, тобто з нього можливо отримати виконувану МП, якщо він побудований на основі допустимого ГВ та не має контурів.

Форматні обмеження (обмеження 4 в постановці задачі) не мають графової інтерпретації. Їх врахування здійснюється одним з двох способів. Згідно до першого з них будується перестановка операцій,

у якій дотримується упорядкування, задане ГМП. Вона може не співпадати з початковою послідовністю. На основі цієї перестановки здійснюється будівництво горизонтальної МП шляхом послідовного включення операцій в МК. Другий спосіб передбачає побудову сполучень з множин операцій, предки яких в ГМП вже включені в складені МК. Потім з сполучень вибирається те, яке, як передбачається, приведе до мінімальної по подовженню МП. Операції цього сполучення включаються в МК одночасово.

Якщо відсутні форматні обмеження, побудова виконуваної МП зводиться до обчислення сітки, сформованої з ГМП. Довжина дуги  $(i, j)$  цієї сітки визначається як  $\max_{r \in R_{ij}} (TE_{ir} - TB_{jr})$ , де  $R_{ij}$  — множина, яка складається з спільних статичних ресурсів взаємодії та функціональних ресурсів операцій  $O_i, O_j$ .

При формуванні МП зменшення значення  $\max T_i$  досягається вибором орієнтації диз'янктивних дуг при побудові ГВ, вибором орієнтації диз'янктивних дуг при побудові ГМП та вибором перестановки чи сполучень при урахуванні форматних обмежень. Оскільки процес оптимізації складається з трьох етапів, можливе допущення, що зменшення  $\max T_i$ , розрахованого на наступному етапі, приведе до зменшення  $\max T_i$  для слідуючого етапу.

Алгоритм побудови допустимого ГВ. Побудова ГВ починається з ГО. Потім в ГО по одній добавляються дуги між парами операцій, маючими спільну диз'янктивну дугу в ДТВ. Побудова ГВ закінчується при визначенні напрямку дуг для усіх таких пар. Орграфі, виникаючі при переході від ГО до ГВ, будемо називати графами стримуваних дуг (ГСД). Якщо при побудові ГВ в ГСД створився контур чи заперечний підграф, то він залишиться у результативному ГВ. Тому необхідно запобігти створенню контурів чи заперечних підграфів ще в ГСД. Аналіз структури заперечного підграфу показує, що він складається з трьох вершин, які зв'язані однією дугою обміну та двома дугами взаємодії, введеними при визначенні напрямку диз'янктивних дуг. Так як ці дуги взаємодії вводяться послідовно, то в процесі розв'язання існує ГСД, в якому одна з них вже введена, а друга ще ні. Підграф такого ГСД, створений з вершин, приймаючих участь в заперечному підграфі, називається небезпечним. Небезпечний підграф необхідно знайти при його виникненні та ввести в нього дугу, протилежну тій, яка приведе до заперечного підграфу. Така дуга

називається блокувчою. Аналогічний прийом використовується для запобігання контурів. У контурі містяться три дуги взаємодії. Якщо знаходити підграфи, в яких є дві з цих дуг, то введення дуги протилежної до тої, яка приводить до контуру, запобігає його створенню.

Будемо називати безальтернативною дугу  $(O_i, O_j)$ , якщо введення в ГСД протилежно направленої дуги  $(O_j, O_i)$  приводить до створення ГСД, з якого не може бути одержано допустимий ГВ. Доведено, що дуги, блокувчі небезпечні підграфи, та дуги, запобігачі створенню контуру, являються безальтернативними.

Наступний алгоритм буде з ГО допустимий ГВ.

Вхід: ГО, список ресурсів з пам'яттю та тип їх використання для кожної операції.

Вихід: допустимий ГВ.

1. Перетворити ГО в початковий ГСД.
2. Знайти в ГСД усі безальтернативні дуги і ввести їх до графу.

3. Якщо в ГСД усі пари взаємодіючих операцій зв'язані дугами, то зупинитися.

4. Вибрати у ЛГВ пару операцій, з'єднаних диз'юнктивною дугою; до ГСД ввести між цими операціями дугу в одному з двох напрямків; з ЛГВ виключити виділену диз'юнктивну дугу.

5. Перейти до п. 2.

У цьому алгоритмі після введення дуги на кроці 4, може слідувати введення значної кількості безальтернативних дуг на кроці 2. Можливо вважати, що введення цих безальтернативних дуг зумовлено введенням дуги на кроці 4. У протилежність дугам, що вводяться на кроці 2, дуги, які вводяться на кроці 4, будемо називати альтернативними.

Для різноманітних варіантів вибору пари операцій та напрямку дуги між ними на кроці 4 приведенний алгоритм може будувати будь-який допустимий ГВ.

У приведенному алгоритмі суворо не визначені такі дії:

1. На кроці 1 алгоритм побудови початкового ГСД з ГО.
2. На кроці 2 правила пошуку безальтернативних дуг.
3. На кроці 4 правила вибору пари операцій та напрямку дуги між ними.

Пошук безальтернативних дуг. Одним із типів безальтернативних дуг являються дуги, попереджувачі контур. Дуга, попереджувача контур, може бути знайдена, як дуга транзитивного замикання двох інших дуг підграфу, що приводить до контуру. Транзитивне замикання графу можна побудувати за допомогою алгоритма Кліні<sup>1</sup>, який будує транзитивне замикання будь-якого графу за час  $O(n^3)$ . При побудові ГВ треба багатократно будувати транзитивне замикання ГВ. Тому для збільшення швидкості розроблено алгоритм, будує транзитивне замикання графу, який був транзитивно замкнений, але втратив цю якість у результаті введення до нього нової дуги.

Алгоритм відновлення транзитивного замикання графу.

Вхід: транзитивно замкнений граф;  $(i, j)$ - дуга, що вводиться до нього.

Вихід: транзитивне замикання нового орграфу.

1. Нехай  $k_1=1$ . 2. Якщо у графі немає дуги  $(k_1, i)$  та при цьому  $i \neq k_1$  чи у графі є дуга  $(k_1, j)$ , то перейти до п.7. 3. Ввести до графу дугу  $(k_1, j)$ . 4. Покласти  $k_2=1$ . 5. Якщо є дуга  $(j, k_2)$  та немає дуги  $(k_1, k_2)$ , то ввести до графу дугу  $(k_1, k_2)$ . 6. Покласти  $k_2=k_2+1$ ; якщо  $k_2 \leq n$ , то перейти до п.5. 7. Покласти  $k_1=k_1+1$ ; якщо  $k_1 \leq n$ , то перейти до п.2. 8. Зупинитися.

Приведений алгоритм вводить до графу також дугу, яка викликала його запуск. Його часова складність  $O(n \times l)$ , де  $l$  - кількість дуг, введених алгоритмом до графу. Оскільки до орграфу, з якого дуги не виключались, не може бути введено більш ніж  $O(n^2)$  дуг, то сумарна складність використання цього алгоритму до одного графу не перебільшує  $O(n^3)$ .

Інший приклад безальтернативних дуг - дуги, блокуєчі небезпечні підграфи. Небезпечний підграф містить одну дугу з Г0 та одну дугу взаємодії. Побудова цього графу зв'язана з введенням дуги взаємодії. Тому небезпечні підграфи можливо знайти, перевіряючи для кожної дуги, що вводиться до ГВ, чи не створє вона небезпечного підграфу. Перевірку появи небезпечного підграфу необхідно виконувати також для дуг, що вводяться алгоритмом транзитивного замикання. Після введення блокуєчої дуги необхідно відновити транзитивну замкненість орграфу.

Г) Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М: Мир, 1979, 536с.

Алгоритм введення до графу дуг, блокувчих небезпечні підграфи, та дуг транзитивного замикання.

Вхід: ГО, транзитивно замкнений ГСД без небезпечних підграфів,  $(i, j)$  - альтернативна дуга, що вводиться до графу.

Вихід: транзитивно замкнений ГСД без небезпечних підграфів.

1. Вмістити до стеку дугу  $(i, j)$ . 2. Вибрати з вершини стеку дугу. Нехай ця дуга  $(i_1, j_1)$ . 3. Для усіх операцій  $O_k$ , нащадків  $O_{i_1}$  у ГО, виконувати п.4. 4. Нехай  $g$  - ресурс обміну між  $i_1$  і  $k$ . Якщо  $g \in Z_{j_1}$  та  $(k, j_1) \in$  ГСД, то виконати алгоритм відновлення транзитивного замикання для дуги  $(k, j_1)$ , зробивши занесення даних про кожну дугу, введenu цим алгоритмом до стеку. 5. Для всіх  $O_k$ , предків  $O_{i_1}$  у ГО, виконувати п.б. 6. Нехай  $g$  - ресурс обміну між  $k$  і  $j_1$ . Якщо  $g \in Z_{i_1}$  та дуга  $(i_1, k) \in$  ГСД, то виконати алгоритм відновлення транзитивного замикання для дуги  $(i_1, k)$ , зробивши занесення даних про кожну дугу, введenu цим алгоритмом до стеку.

Приведений алгоритм має часову складність  $O(n^2)$ , де  $l$  - кількість дуг, введених до ГСД цим алгоритмом. Сумарна складність його використання до графу, з якого не виключалися дуги  $O(n^3)$ .

Відсутність заперечних підграфів та контурів достатня для допустимості ГВ. Для ГСД ця умова необхідна, але недостатня. Тому безальтернативні дуги не вичерпуються дугами транзитивного замикання та дугами, блокувчими небезпечні підграфи. Алгоритми пошуку інших безальтернативних дуг називаються алгоритмами провокацій та визначаються рекурсивно. Як алгоритм провокацій нульового порядку використовується алгоритм, який знаходить блокувчі дуги та дуги транзитивного замикання. Алгоритм провокацій порядку  $k$  визначається через алгоритм провокацій порядку  $(k-1)$  таков процедурою.

Вхід: ГО, ГСД.

Вихід: ГСД, що можливо містить додаткові безальтернативні дуги.

1. Нехай  $i=1$ . 2. Покласти  $j=1$ . 3. Якщо  $i=j$  чи в ГСД є дуга  $(i, j)$  чи  $(j, i)$ , то перейти до п.7. 4. Ввести до ГСД дугу з  $i$  в  $j$  за допомогою алгоритму пошуку безальтернативних дуг порядку  $(k-1)$ . 5. Якщо у ГСД створився цикл, то виключити з нього дуги, введені при останньому виконанні кроку 4, ввести за допомогою алгоритма порядку  $(k-1)$  дугу  $(j, i)$  і перейти до п.1. 6. Виключити з ГСД

дуги, введені в нього при останньому виконанні кроку 4. 7. Покласти  $j=j+1$ ; якщо  $j \leq n$ , то перейти до п.3. 8. Покласти  $i=i+1$ ; якщо  $i \leq n$ , то перейти до п.2. 9. Зупинитися.

В практичних цілях рекомендується використовувати таку каскадну схему вибору порядку використовуваних провокацій.

1. Нехай  $k=0$ . 2. Зробити спробу побудови ГВ за допомогою алгоритма порядку  $k$ . Якщо в процесі побудови ГВ не створено циклу, то вважати одержаний ГВ - шуканим та зупинитися. 3. Покласти  $k=k+1$ ; якщо  $k \leq k_{\max}$ , то перейти до п.2. Інакше як шуканий узяти ГВ початкової послідовності.

Наведені аргументи на користь NP повноти задачі пошуку усіх безальтернативних дуг.

Побудова початкового ГСД з ГО. Граф обмінів у загальному випадку являється мультиграфом, оскільки з одної його вершини до іншої може іти декілька дуг, що характеризуються різними ресурсами обмінів. Початковий ГСД з ГО можна побудувати заміною усіх дуг зв'язуючих пару вершин однією дугою. Проте такий граф буде містити не введені безальтернативні дуги. Тому найбільш ефективним є наступний алгоритм побудови початкового ГСД.

Вхід: ГО.

Вихід: початковий ГСД, що не містить безальтернативних дуг виявляемих алгоритмом порядку  $k$ .

1. Для всіх дуг, належачих до ГО, виконувати алгоритм провокацій порядку  $k$ . 2. Зупинитися.

Пошук кращого ГВ. Якість ГВ визначається вибором пари операцій та напрямку дуги між ними на кроці 4 алгоритму побудови ГВ. ГВ з найкращим значенням  $\max T_i$  можна одержати за допомогою описаного в роботі алгоритму, що використовує метод вітвей та границь. Але оскільки це потребує великих витрат обчислюваного часу, то окрім нього запропоновано 8 евристичних правил, які вибирають пару операцій та напрямку дуги між ними, виходячи з сіткових характеристик поточного ГСД. Якщо для усіх операцій поточного ГСД розрахувати ранні строки початку операцій  $T_{pi}$  та пізні строки  $T_{pi}$ , то можливо побудувати такі 4 характеристики для кожної упорядкованої пари операцій  $(i, j)$ :

$$ПР1 = T_{pi} - T_{pj}; \quad ПР2 = T_{pi} - T_{pj}; \quad ПР3 = T_{pi} - T_{pj}; \quad ПР4 = T_{pi} - T_{pj}.$$

У якості вводимі альтернативної дуги вибирається та, яка надає

найбільше значення вибраного критерія. Якщо для кожної з можливих альтернативних дуг побудувати  $L_{ij}$  - множину, яка складається з дуги  $(i, j)$  та безальтернативних дуг, що викликані вводом цієї дуги, то можливо одержати такі 4 пріоритети:

$$\begin{aligned} \text{ПР5} &= \max_{(a,b) \in L_{ij}} T_{pa} - T_{pb}; & \text{ПР6} &= \max_{(a,b) \in L_{ij}} T_{ra} - T_{rb}; \\ \text{ПР7} &= \max_{(a,b) \in L_{ij}} T_{pa} - T_{pb}; & \text{ПР8} &= \max_{(a,b) \in L_{ij}} T_{ra} - T_{rb}; \end{aligned}$$

Розроблено алгоритм, дозволяючий використовувати для побудови ГВ з кращою якістю пріоритети теорії розпису.

- Вхід: ГО, список ресурсів з пам'яттю та тип їх використання для кожної операції.

Вихід: перестановка операцій, відмінна від початкової послідовності, ГВ якої може бути використано для побудови горизонтальної МП.

1. Побудувати початковий ГСД з ГО.
2. Вважати усі операції невідмічені.
3. Серед невідмічених операцій виділити підмножину в яку не заходять дуги з інших невідмічених операцій. Ця підмножина називається множиною операцій доступних з даних.
4. Користувачись пріоритетами загальної задачі теорії розпису вибрати з множини операцій доступних з даних одну. Нехай це буде  $O_1$ .
5. Покласти  $j=1$ .
6. Якщо  $O_j$  вже відмічена, то перейти до п. 9.
7. Якщо  $O_1$  та  $O_j$  не взаємодіють чи вже з'єднані дугою, то перейти до п. 9.
8. Ввести в ГСД дугу  $(i, j)$  за допомогою одного з алгоритмів пошуку безальтернативних дуг. Якщо в ГСД з'явилася дуга, яка виходить з невідміченої операції в операції  $i$  чи контур, то перейти до п. 12.
9. Покласти  $j=j+1$ ; якщо  $j \leq n$ , то перейти до п. 6.
10. Відмітити  $O_1$  та додати її в кінець формульчої послідовності. Якщо не всі операції відмічені, то перейти до п. 3.
11. Зупинитися.
12. Виключити  $O_1$  з множини доступних з даних; з ГСД виключити дуги, які введені в нього під час останнього виконання циклу в пп. 5-9; перейти до п. 4.

#### Врахування функціональних ресурсів та форматних обмежень

зручно виконувати одночасово. Це досягається шляхом побудови перестановки операцій, яка включає в собі упорядкування, задане ГВ. Для визначення порядку використання спільних функціональних ресурсів використовується порядок операцій в перестановці. Побудова перестановки визначає також порядок включення операцій до

мікрокоманди з врахуванням форматних обмежень.

Побудова перестановки по ГВ здійснюється шляхом присвоєння операціям пріоритетів. Операція з більшим пріоритетом буде мати менший номер. Традиційні засоби визначення пріоритетів розраховані на одноциклові операції. У цій роботі пропонується використовувати як пріоритети операції, пізні строки робіт сітки, розрахованої по ГВ, що дозволяє розрахувати пріоритети також для багаточислових операцій.

Якщо відсутні функціональні обмеження між операціями різних МК можна використовувати алгоритми, які будувть горизонтальну МП без побудови перестановки, шляхом вибору сполучень операцій з множини, предки яких у ГВ вже включені до МК. Для багаточислових операцій кращу якість розв'язку з цієї групи дає алгоритм описаний в <sup>2)</sup>. Його недолік у тому, що в визначених умовах він коректує вже виконану частину розв'язки на невизначену глибину. Така корекція може потребувати більших витрат обчислюваного часу. У дисертаційній роботі запропонована модифікація цього алгоритму, яка позбавлена такого недоліку.

При відсутності форматних обмежень врахування порядку використання ресурсів з пам'яттю та функціональних можна виконувати в єдиній техніці, шляхом одержання з ГО безпосередньо ГМП. Для цього необхідно замінити усі функціональні ресурси, які використовуються операціями на ресурси з пам'яттю, і вважати, що вони використовуються в режимі запису. Тепер застосування алгоритму побудови ГВ надає відразу порядок використання ресурсів з пам'яттю та функціональних ресурсів.

Глобальна упаковка. Запропоновано метод, зберігачий обміни між операціями різних ЛУМ при зміні порядку використання ресурсів з пам'яттю всередині ЛУМ. Це досягається введенням до початкового ГСД додаткових дуг.

Кожний метод глобальної упаковки складається з двох структурних частин. Це метод локальної упаковки і принцип виділення суміжних ЛУМ, які будуть разом упаковуватися. Згідно з J. Toso M., Tamura E. et al. An Approach to microprogram optimization considering resource occupancy and instruction formats. //Proceedings of 10<sup>th</sup> Annual Workshop on Microprogramming. - 1977. - pp. 92-108.

правилами виділення таких груп ЛУМ методи локальної упаковки поділяються на шляхо-орієнтовані, в яких для кожного застосування алгоритму локальної упаковки виділяються ЛУМ, які лежать на одному шляху, тобто ті, між якими можлива безпосередня передача управління, та блок-орієнтовані, в яких операції можуть пересуватися тільки між двома суміжними ЛУМ. Для усіх знайдених в літературі шляхо-орієнтованих методів можливо використання алгоритмів, змінюючи порядок застосування спільних ресурсів з пам'яттю, шляхом простої заміни ними традиційних алгоритмів локальної упаковки.

Усі блок-орієнтовані методи зберігають відображення однієї операції вертикальної мікропрограми в одну операцію горизонтальної мікропрограми. У роботі знайдена максимальна область, в якій виконується ця відповідність - нижнє дерево. Розроблено алгоритм, використовувачий зміну порядку застосування спільних ресурсів з пам'яттю, та працюючий на нижньому дереві.

Запропонована нова оцінка якості горизонтальних мікропрограм, не потребує априорної інформації про ймовірності виконання ЛУМ, що базується на припущенні, що усі нащадки конкретного ЛУМ одержуть від нього управління рівноімовірно. Ця оцінка якості мікропрограми названа інтегральною.

Експериментальне дослідження алгоритмів. Перша група мікропрограм генерувалась з застосуванням датчика випадкових чисел. Експерименти дозволили виділити краще з пріоритетних правил, використовувється при виборі альтернативної дуги. Цим правилом стало правило ПР2. Було підтверджено речення, що зменшення  $\max T_i$ , розрахованого по ГВ, приводить до зменшення довжини одержуваної з нього МП. При роботі алгоритма з 400 згенерованих МП, тільки в одній з них необхідно було застосування провокацій першого порядку. У всіх інших виявилось достатнім алгоритма нульового порядку. Це підтверджує обчислювану ефективність каскадної схеми при побудові ГВ.

У другій частині експерименту досліджувались 23 МП з математичного забезпечення реального обчислювача. Досліджувалась якість МП в залежності від застосування глобальної упаковки та зміни порядку використання ресурсів з пам'яттю. Незалежно застосування глобальної упаковки та способу зміни порядку

використання ресурсів з пам'яттю дають зрівняні результати, а ефект їх сумісної реалізації наближається до подвійного ефекту роздільного застосування.

## ВИСНОВКИ

1. Сформована та досліджена математична модель задачі локальної упаковки мікрокоду.

2. Використовувана модель дозволила знайти новий клас рішень, які не використовувались традиційними методами упаковки.

3. Досліджені умови допустимості графу взаємодії, застосовуваного при формуванні допустимого рішення задачі локальної упаковки.

4. Розроблено метод одержання допустимого графу взаємодії по графу обмінів.

5. У рамках методу одержання графу взаємодії розроблені алгоритми пошуку безальтернативних дуг, як дуг транзитивного замикання та дуг, блокуючих створення заперечних підграфів з небезпечних.

6. Розроблено алгоритм відновлення транзитивного замикання графу, який був транзитивно замкнений, але втратив цю властивість внаслідок вводу до нього нової дуги. Це дозволило підвищити швидкість роботи алгоритмів у  $N^2$  разів, де  $N$  - число операцій мікропрограми.

7. Одержана група алгоритмів, яка дозволяє знаходити безальтернативні дуги, що не знаходяться як при транзитивному замиканні, так і при пошуку небезпечних підграфів.

8. Розроблена каскадна схема пошуку безальтернативних дуг, дозволяюча суттєво зменшити час упаковки.

9. Запропоновано 8 нових пріоритетних правил та метод, дозволяючий застосовувати пріоритети теорії розпису для побудови графу взаємодії, що гарантує мікропрограми кращої якості.

10. Запропоновано пріоритетне правило, дозволяюче у єдиній техніці враховувати форматні та функціональні обмеження для багатоциклових операцій.

11. Усунена необхідність повернень при використанні алгоритму, будувачого горизонтальну мікропрограму вибором сполучень.

12. Запропоновано метод, враховуючий обмеження на функціональних ресурсах та ресурсах з пам'яттю в єдиній техніці.

13. Для усіх розроблених алгоритмів проведені дослідження їх часової складності.

14. Одержані методи локальної упаковки мікрокоду поширені на глобальну упаковку.

15. Проведені обчислювані експерименти, що доводять ефективність розроблених схем рішень та реалізуючих їх алгоритмів.

#### ПУБЛІКАЦІЇ ПО ТЕМІ ДИСЕРТАЦІЇ

1. Малых О.Н., Ардельян А.В., Боблак М.Ю., Наумов Л.Н., Шахновский Ю.С. Алгоритмы упаковки многоцикловых микроопераций. //Вестн. Харьк. политехн. ин-та. 1987. - №240: Техническая кибернетика и ее прил. - Вып. 7. - С. 59-62.

2. Малых О.Н., Ардельян А.В., Кожин Ю.Н., Шахновский Ю.С. Моделирование микропрограмм сетями Петри. //Вестн. Харьк. политехн. ин-та. - 1988. - №252: Техническая кибернетика и ее прил. - выпуск 8. - С. 41-45.

3. Малых О.Н., Ардельян А.В., Кожин Ю.Н., Шахновский Ю.С. Этапы анализа структуры микропрограммируемого SIMD-вычислителя. //Вестн. Харьк. политехн. ин-та. - 1989. - №263: Техническая кибернетика и ее прил. - выпуск 9. - С. 59-61.

4. Шахновский Ю.С., Малых О.Н., Ардельян А.В., Кожин Ю.Н. Об одном подходе к решению задачи расписания с динамическими ограничениями. //Вестн. Харьк. политехн. ин-та. - 1990. - №277: Техническая кибернетика и ее прил. - выпуск 10. - С. 71-75.

5. Шахновский Ю.С. Формирование микрокода с помощью правила, определяющего порядок включения операций. //Вестн. Харьк. политехн. ин-та. - 1992. - №2: Техническая кибернетика и ее прил. - выпуск II. - С. 36-38.

6. Кожин Ю.Н., Малых О.Н., Прокопенков В.Ф., Шахновский Ю.С. Настраиваемая система автоматизации микропрограммирования матричного вычислителя. //Вестн. Харьк. политехн. ин-та. - 1992. - №2: Техническая кибернетика и ее прил. - выпуск II. - С. 87-91.

7. Акопов В.И., Губарев Н.В., Евдокимов А.В., Малых О.Н., Кожин Ю.Н., Прокопенков В.Ф., Шахновский Ю.С. Мультипроцессорный вычислитель для параллельных вычислений: структура и система

автоматизации микропрограммирования. - Научно-технический сборник "Вопросы авиационной науки и техники", серия "Бортовые приборы навигации, контроля и управления", выпуск 7, 1992, Московский институт электромеханики и автоматики. С. 23-31.

8. Малых О. Н., Шахновский Ю. С. Новый подход к локальной упаковке микрокода. //Вестн. Харьк. политехн. ин-та. - 1993. - Резание и инструмент. - выпуск 47. - С. 161.

9. Кожин Ю. Н., Малых О. Н., Прокопенков В. Ф., Шахновский Ю. С. Описание перенастраиваемой системы генерации микропрограмм. //Вестн. Харьк. политехн. ин-та. - 1993. - Резание и инструмент, выпуск 47. - С. 162.

*Шах*

AB 29.065

**AB 29.065**