

УДК 004.4 (043.2)

3943.208-OK.2

Гамаюн В. П. д-р техн. наук

**ПОТОКОВАЯ ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА**

Институт кибернетики им. В. М. Глушкова НАН Украины

*Предложена организация вычислительного процесса с прямой адресацией данных. Адресное поле команды определяется частью операнда, остальная часть операнда адресуется по этому же значению.*

**Введение**

Повышение производительности компьютерных средств и систем предусматривает, прежде всего, достижение высокой скорости исполнения программ. Классическая схема исполнения элемента программы – команды, заключается в реализации нескольких необходимых фаз командного цикла:

- выбор команды из памяти;
- дешифрация и интерпретация кода операции;
- определение адресов операндов;
- чтение операндов из запоминающей среды;
- исполнение преобразования операндов в результат;
- запись результата в память.

Одним из факторов такого исполнения программы является фон Неймановская структура ЭВМ и соответствующая организация вычислительного процесса. В компьютерных средствах с другой структурой и другой организацией вычислительного процесса возможны отличительные решения, в частности не только по следованию фаз реализации команды, но и по структуре самих команд.

Развитие принципов самоорганизации, повышение уровня машинного (компьютерного) интеллекта является основой высокопроизводительной обработки.

**Постановка задачи**

Принципы потоковой организации вычислительного процесса позволяют реализовать параллелизм обработки на

основе потоков признаков, потоков данных. Общая концепция заключается в применении потоков компонент структур данных или самих структур данных для инициализации обработки в операционных слотах архитектуры [1, 2].

Поток данных является также иницирующим компонентом на операционном уровне вычислений. Примером могут служить методы блочно-групповой обработки, методы с применением вычисляемых таблиц [1].

Такие методы применяются для задач, в которых поток данных может быть использован как адресно-управляющая информация. Для вычисления оператора суммы пар произведений  $Y = \sum_{i=1}^2 A_i \times B_i$  представим каждый операнд  $B_i$  в позиционной системе счисления с основанием  $p = 2$ :

$$Y = \sum_{i=1}^2 A_i \times B_i = \sum_{i=1}^2 A_i \sum_{k=1}^n b_{i,k} p^k,$$

где  $n$  – разрядность данных,  $k$  – номер разряда,  $b_{i,k}$  – значение  $k$ -го разряда,  $p$  – основание системы счисления.

Изменяя порядок суммирования, получаем

$$Y = \sum_{i=1}^2 A_i \times B_i = \sum_{k=1}^n p^k \sum_{i=1}^2 A_i b_{i,k}.$$

Сумма при фиксированном  $k$  будет содержать только те значения  $A_i$ , для которых  $b_{i,k}$  отлично от нуля. Поэтому при определенных сочетаниях  $b_{i,k}$  в общий результат необходимо добавлять соответствующие значения сумм  $A_i$  или значения

$A_i$ . Область значений  $A_i$  (все варианты сумм и значений) помещаются в определенную область памяти, адресная часть которой управляется  $b_{i,k}$ . При этом формируется следующая процедура вычисления  $Y$ :

- заполняется таблица вспомогательных сумм и значений из  $A_i$ ;
- по значениям  $b_{i,k}$  считываются соответствующие величины и накапливается результат.

Для  $i=2$  вспомогательная таблица имеет вид:

Табл.1 Значения вспомогательных сумм

Значения $A_i$	Значения $b_{1,k} b_{2,k}$
0	00
$A_1$	01
$A_2$	10
$A_1 + A_2$	11

При вычислении  $Y$  операнды  $B_i$  поразрядно подаются на адресные шины, и считываемые значения суммируются со сдвигом с предыдущими значениями. Структура команды для операционного устройства при этом может быть следующая

КОП СложСД	$b_{1,k} b_{2,k}$	адрес РР
------------	-------------------	----------

где: адрес РР – адрес регистра-аккумулятора;

КОП СложСД – код операции сложения сдвига;

$b_{1,k} b_{2,k}$  – адреса обращения к вспомогательной таблице.

Адреса обращения заполняются при непосредственной загрузке  $B_i$  или могут аппаратно соединены с адресными шинами в операционном устройстве.

### Методика решения

Использование потока данных в качестве адресно-управляющей информации может быть распространено на общую организацию вычислительного процесса в компьютерных средствах. Концепция предлагаемой организации обра-

ботки заключается в использовании потока данных в качестве адресно-операндной информации. Адресные поля в структуре команды заменяются непосредственными значениями операндов или частями операндов – полями старших, младших разрядов или комбинированными группами разрядов.

Если в адресном поле команды записывается часть операнда, то остальная информация о данном записывается в памяти компьютера. При этом возникает задача уникальности, неповторяемости кодов, размещаемых в таких полях.

При записи полного операнда в адресном поле команды исключается фаза обращения к памяти и таким образом исполнение команды реализуется за меньшее время.

Механизм реализации команды с потоковой адресацией по двум вариантам представлен следующим рисунком:



Рис.1. Механизмы потоковой адресации

Вариант адресно-операндной организации предусматривает поэтапное выполнение команды:

- выбор данных по адресной части с параллельным исполнением команды над операндной частью;
- выполнение и корректировка вычислений по данным из запоминающей среды (формирование результата).

Рассмотрим пример реализации вычисления полинома 4-й степени с применением потоковой организации

$$Y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4.$$

При следующем числовом значении переменных выделим старшие значения

разрядных полей операндов в качестве адресно-операндных форматов команды:

$a_0 = 001011$	000001	011
$a_1 = 011101$	000011	101
$a_2 = 100001$	000100	001
$a_3 = 001111$	001111	
$a_4 = 101111$	000101	111
$x = 000111$	000111	

Адресно-операндная часть      Часть данных, хранимая в ОЗУ

Программа с адресно-операндным механизмом исполнения команд следующая:

Номер команды	Тип операции	Поле 1	Поле 2	Действия
1	КОП 1	000111	000111	$PP = X^2$
2	КОП 2	000110(001)		запись PP
3	КОП 1	000011	000111	$PP = a_1 \times X$
4	КОП 2	001011(1101)		запись PP
5	КОП 1	000100	000110	$PP = a_2 \times X^2$
6	КОП 2	001100(1010001)		запись PP
7	КОП 1	000000	000110	$PP = X^3$
8	КОП 2	1010(10111)		запись PP
9	КОП 1	000110	000110	$PP = X^4$
10	КОП 2	010101(111001)		запись PP
11	КОП 1	001010	001111	$PP = a_3 \times X^3$
12	КОП 2	010100(00011001)		запись PP
13	КОП 1	010101	00101	$PP = a_4 \times X^4$
14	КОП 2	101001(100001101)		запись PP
15	КОП 3	000001	001011	$PP = a_0 + a_1 \times X$
16	КОП 3	001100		$PP = PP + a_2 \times X^2$
17	КОП 3	010100		$PP = PP + a_3 \times X^3$
18	КОП 3	101001		$PP = PP + a_4 \times X^4$

В рассмотренном примере коды операций сложения, умножения и записи обозначены соответственно как КОП 3, КОП 1, КОП 2. В командах записи в адресно-операндной части в скобках указаны части операндов, записываемых в память.

Команды, в которых поля адресной информации (Поле1 и Поле2) заполнены значениями операндов, исполняются без фаз выборки данных из оперативной памяти. Например, команда № 1:

КОП 1 000111 000111

реализует вычисление квадрата переменной  $X$  способом умножения одинаковых сомножителей.

Так как разрядности в адресно-операндных достаточно для полного представления операндов, то значения таких операндов указаны в Поле1 и Поле2 – 000111. Указанные значения поступают в операционные средства компьютер (процессор) непосредственно после чтения команды.

В других командах, например №11 – КОП 1 001010 001111 – Поле 1 содержит

старшие разряды первого сомножителя 1010 \*\*\*\*\*, младшие разряды считываются из памяти по значению 1010 и равны \*\*\*\*10111. В Поле 2 указано полное значение второго операнда – 001111.

Результат перемножения равен 1010000011001, причем сразу было вычислено произведение  $101000000 \times 001111$ , что составило около 70% значения результата.

В командах записи – №14 – КОП 2 101001(100001101) – в Поле 1 указан адрес записи содержимого регистра результата, которые одновременно являются и старшими разрядами результата. В скобках указано то значение, которое записывается по данному адресу. Значение в скобках не приводится при реальном заполнении полей команды. Вторая часть адресно-операндного поля не заполнена.

Возможны использование вариантов построения команды при потоковой организации вычислений. Например, в командах записи Поле 2 может содержать информацию о значащих разрядах операнда, которые размещаются в памяти. Для реализации такого варианта необходимо иметь достаточное количество разрядов в формате команды.

Анализируя команды умножения и сложения по объёму данных, которые непосредственно указаны в форматах полей команд, можно сделать вывод, что 72%

времени на выполнение фаз считывания данных исключается и соответственно общее время вычисления полинома уменьшается на 43%.

Следует указать, что данная программа представляет в основном пример реализации потоковой организации вычислений и не является оптимальной относительно заполнения адресно-операндных полей. Решение последней указанной задачи определяет еще большее сокращение времени вычисления полинома.

### **Выводы**

Рассмотренный материал является развитием метода прямой адресации, развиваемый для построения высокопроизводительных средств вычислительной техники, в частности специализированных арифметических расширителей, проблемно-ориентированных операционных полей.

### **Список литературы**

1. Гамаюн В. П., Богданова Е. В. Параллельно-последовательный алгоритм вычисления суммы парных произведений // Средства микропроцессорной техники. – Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР. 1987. – С.21-24.
2. СуперЭВМ. Аппаратная и программная организация / Под. ред. С. Фернбаха – М.: Радио и связь, 1991. – 320 с.