

УДК 004.4 (043.2) ,

Гамаюн В. П., д-р техн. наук

## ПОТОКОВАЯ РЕКОНФИГУРАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР

Институт кибернетики им. В. М. Глушкова НАН Украины

*Рассмотрена реконфигурация вычислительной структуры на основе потока данных. Уменьшение глубины алгоритма (реализации) достигается за счет реальных значений данных.*

### Введение

Адаптация вычислительной структуры, компьютерной операционной среды признана как направление в организации высокопроизводительных средств или систем. Такое направление поддерживается развитием элементно-технологической базы, в частности применением программируемых логических интегральных схем (ПЛИС). Настройка компьютерной среды на заданный алгоритм производится либо установкой необходимых связей между соответствующими слотами структуры, либо загрузкой всей структуры в случае использования ПЛИС. При такой реконфигурации операционной структуры не учитываются возможности сокращения, уменьшения глубины реализуемых алгоритмов за счет анализа, структуры и "семантики" реальных данных, потока данных, что может значительно изменить уровень адаптации. Для систем для обработки массивов предложены подходы по реконфигурации компьютерной среды на основе анализа потока преобразуемых данных.

Организация вычислительного процесса для многоагентных, параллельных вычислительных структур на основе потока данных исследуется несколько десятилетий (начиная с 60-х годов). Получены значительные теоретические результаты по ряду проблем реализации потокового управления – статического и динамического способов инициализации обработки [1]. Преимущество статического подхода заключается в простоте управлением операций – команды реализуются при поступлении операндов. При втором подходе для выполнения команды собираются все

операнды, обозначенные одной меткой, что значительно усложняет реализацию данного подхода. Для статического подхода применяется структура с фиксированными связями, а для динамического – универсальная, например матричная структура. Примерами реализации потоковой концепции в организации вычислительного процесса являются потоковая машина Массачусетского технологического института, Манчестерская потоковая машина, "U-интерпретатор" и другие [1]. Однако применение потока данных для реконфигурации операционной среды не реализовывалось ни в одном из проектов.

### Постановка задачи

Рассмотрим общие положения адаптации вычислительной структуры к структуре алгоритма. Граф алгоритма для потоковой машины со статическим способом инициализации содержит операции и скалярные переменные в неструктурированной форме. Такое упрощение принято для иллюстрации подхода потоковой реконфигурации. Использование структурированных данных требует расширения концепции потоковой обработки и определения операций над такими данными.

### Методика исследования

Пусть задана параллельная структура общего типа ( $A$ ) с множеством операционных вершин  $\{n_i\}$  и множеством дуг  $\{d_i\}$ .

На рис. 1 представлена параллельная сильносвязанная структура на  $N=12$  операций.

Допустим, что граф алгоритма отображается полностью на вычислительную структуру, другими словами операторной

вершине соответствует процессорный элемент (ПЭ), а дуге графа некоторая коммутативная линия.

Множество априорно заданных переменных обозначим как  $\{a_i\}$ , множество промежуточных переменных и результатов как  $\{F_i\}$ . Причем индекс исходных данных присваивается произвольно, а номер промежуточных переменных соответствует номеру операторной вершины (процессорного элемента), в котором это данное получено.

Реконфигурация вычислительной структуры заключается в следующем.

На основе многомерного анализа возможно установить изменение структуры вычислительной, операционной среды с учетом реальных данных, преобразуемых по такому алгоритму. Преобразование прежде всего определяется такими значениями операндов, которые определяют выполняемую в некоторой вершине операцию как не изменяющую входные данные или, как приводящую к нулевому результату.

Пусть  $a_i 0$  означает нулевой операнд, а  $a_i 1$  – единичный операнд.

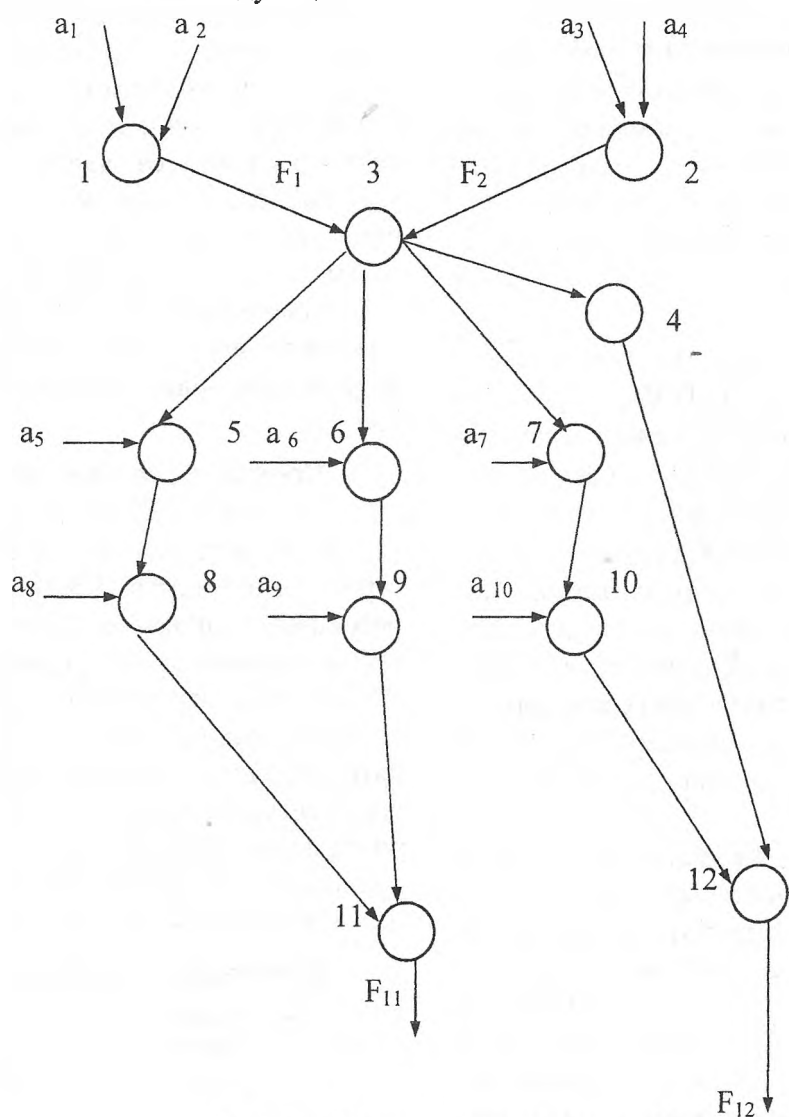


Рис. 1. Параллельная структура общего вида

При следующих значениях исходных данных  $a_{11}, a_{40}, a_{71}, a_{91}$  и типах операций, выполняемых в процессорных элементах ПЭ<sub>1</sub> – сложение, ПЭ<sub>2</sub> – сложение, ПЭ<sub>7</sub> – умножение, ПЭ<sub>9</sub> – умноже-

ние получаем, что ПЭ<sub>1</sub> выполняет заданную функцию (команду), ПЭ<sub>2</sub> выполняет команду передачи операнда с входа на выход, ПЭ<sub>7</sub> и ПЭ<sub>9</sub> выполняют также передачу с входа на выход.

Таким образом, если считать функции (команды) по передаче данных через ПЭ как "прозрачную" команду, то вычислительная структура приобретает следующую конфигурацию, показанную на рис. 2.

Рассмотренное распределение по реальным данным  $a_{11}$ ,  $a_{40}$ ,  $a_{71}$ ,  $a_{91}$  не

приводит к уменьшению глубины графа, а следовательно и к сокращению временных затрат при использовании редуцированной структуры.

При следующем варианте распределения данных реконфигурация вычислительной структуры может быть следующей.

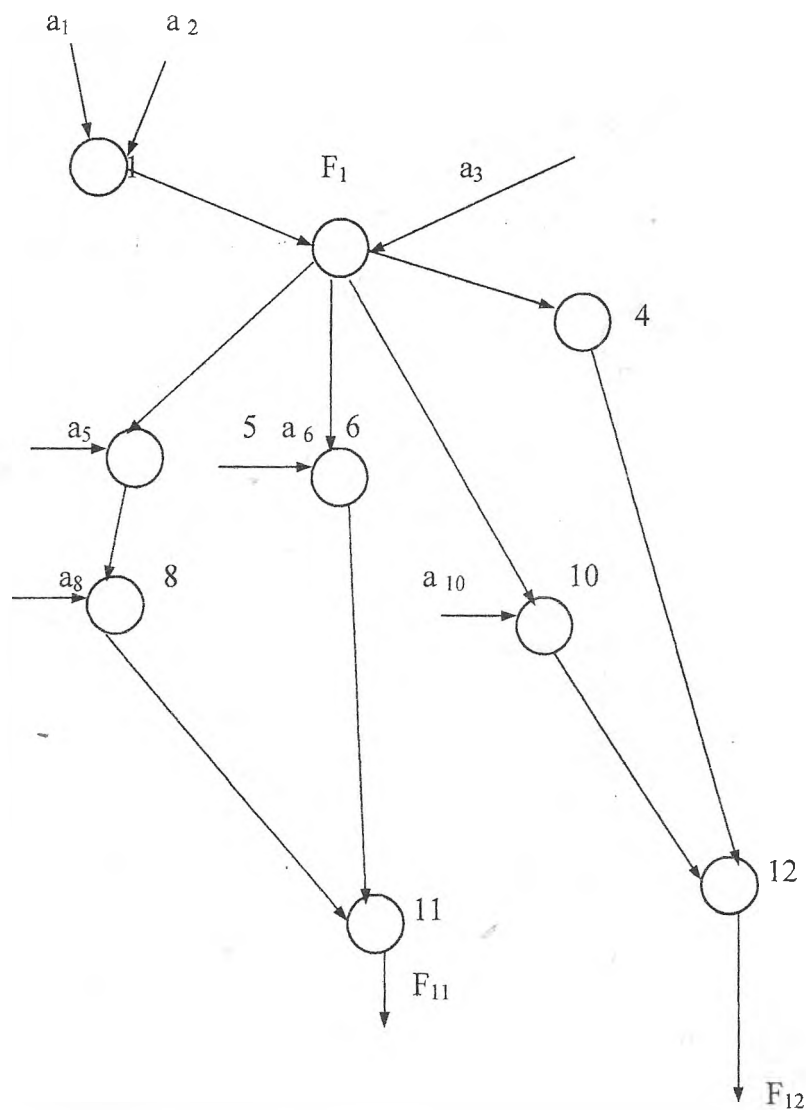


Рис. 2. Первый вариант реконфигурации структуры

Для второго варианта распределения данных  $a_{11}$ ,  $a_{40}$ ,  $a_{71}$ ,  $a_{91}$ ,  $a_{80}$  и команд, выполняемых в ПЭ<sub>1</sub> – сложение, ПЭ<sub>2</sub> – умножение, ПЭ<sub>7</sub>, ПЭ<sub>8</sub>, ПЭ<sub>9</sub> умножение – получаем реконфигурацию структуры, представленную на рис. 3.

Глубина алгоритма уменьшилась и, следовательно, сократилось время решения задачи в целом. Рассмотренные варианты реконфигурации определяют апри-

орную реконфигурацию структуры на основе анализа исходных данных, априорных данных.

Кроме данных анализируется командно-операционный набор, применяемый для реализации решения.

Таким образом реализуются этапы адаптации вычислительной структуры:

- настройка структуры на заданный алгоритм;

- анализ потока исходных данных;
- анализ командно-операционного набора алгоритма;
- определение "прозрачных" вершин в алгоритме или коммутационных каналов в структуре;

- реконфигурация вычислительной структуры с учетом семантики потока данных и соотношений следования в алгоритме.

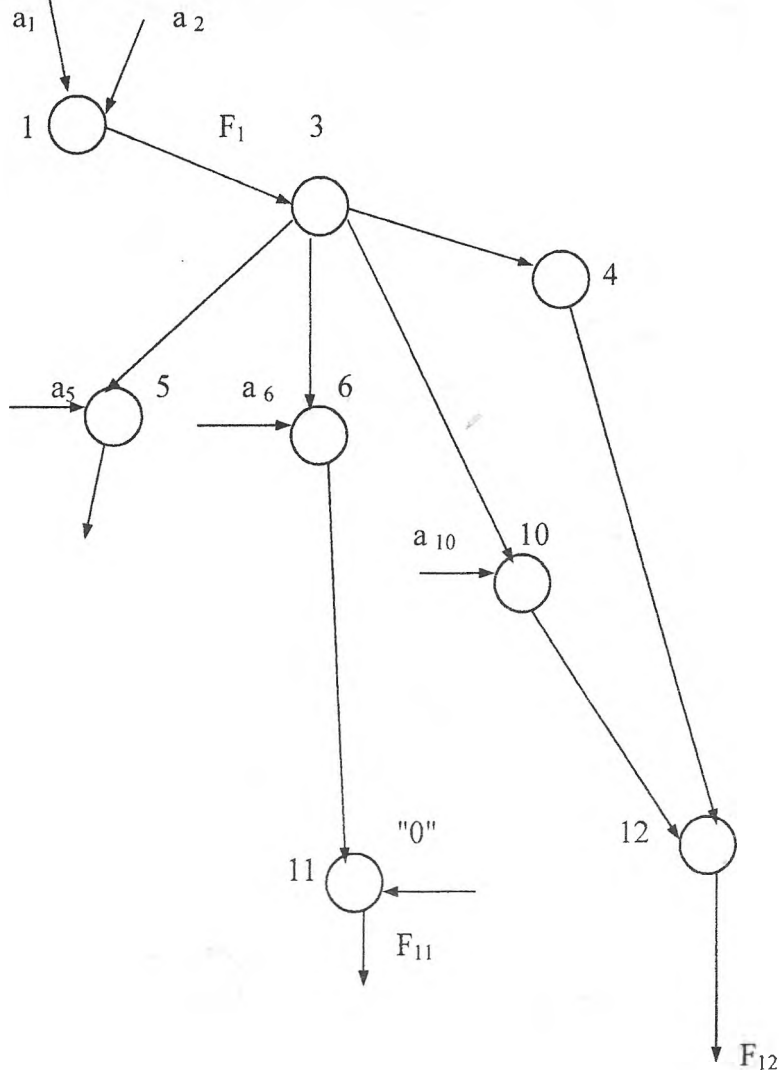


Рис. 3. Второй вариант реконфигурации структуры

### Выводы

Предложенные подходы базируются на вычислительной структуре, элементами которой должны быть слоты, подобные процессорным элементам, но со следующими обязательными функциями: записи-чтения, распознавания, транзита-передачи и традиционным набором команд по обработке. Выигрыш при адаптации возможен при реализации операций

транзита-передачи за меньшее время, чем операции обработки. Примером такой архитектуры может быть система, подобная транспьютерной или с быстрыми мультиплексорно-матричными коммутаторами.

### Список литературы

1. *СуперЭВМ*. Аппаратная и программная организация / Под. ред. С. Фернбаха – М.: Радио и связь, 1991. – 320 с.