

Архитектура параллельных ЭВМ

1. Особенности архитектуры мультипроцессорных ЭВМ
2. Классификация ЭВМ по числу потоков команд и данных
3. Механизмы управления и архитектура ЭВМ

Знать: понятие архитектуры ЭВМ и влияние на нее сложности модуля обработки, числа потоков команд и данных, механизма управления; классификацию ЭВМ по Флинну; особенности архитектур ЭВМ с директивным, потоковым и запросным управлением.

Помнить: об "извлечении" ЭВМ с потоковым управлением параллелизма из выполняемого алгоритма

Литература: Страбыкин Д.А. Организация машин параллельного логического вывода: Учебное пособие для вузов. – Киров, изд-во ВятГТУ, 1999 с.116-128.

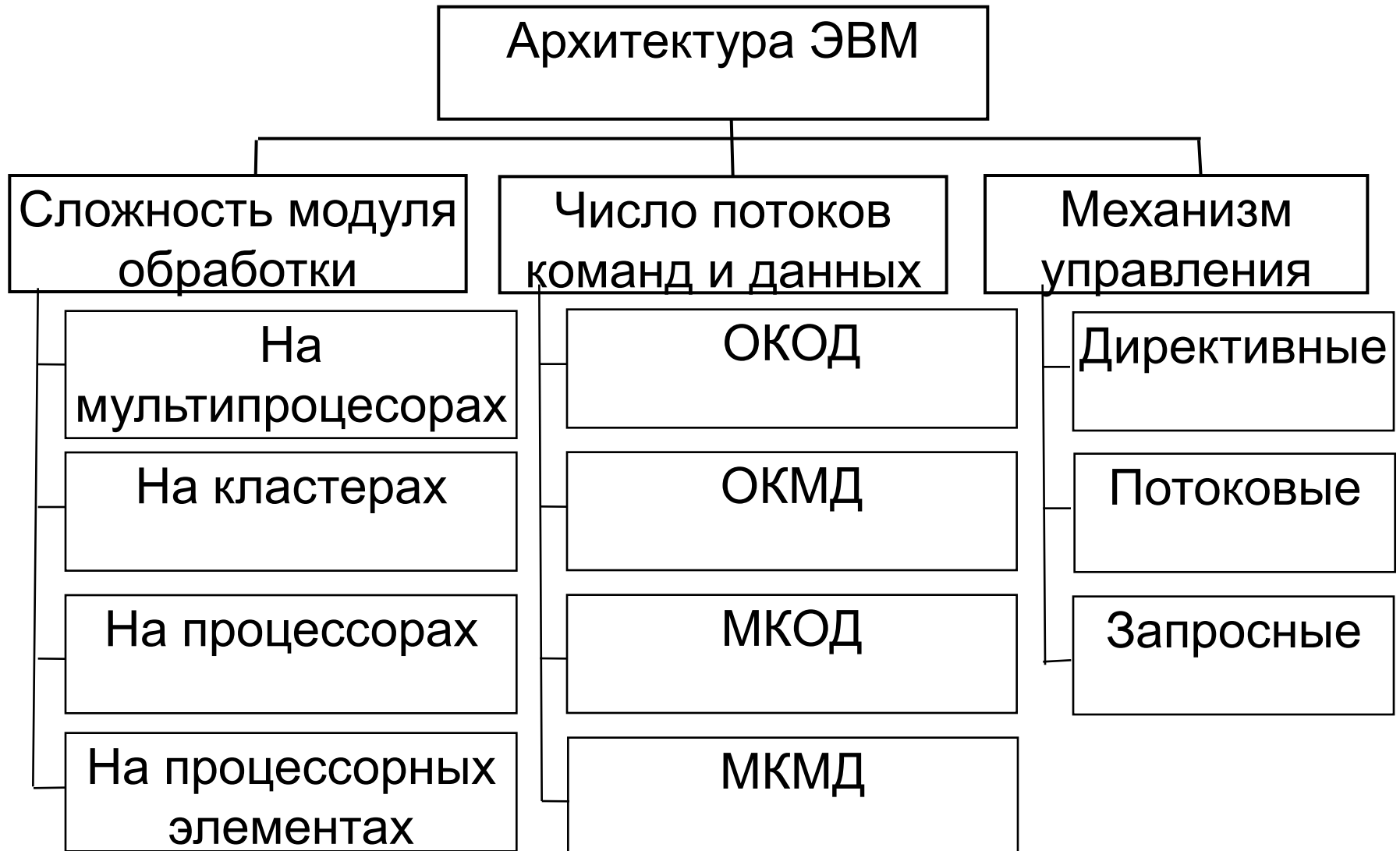
1. Особенности архитектуры мультипроцессорных ЭВМ

- Архитектуру машины можно рассматривать как отображение модели вычислений на аппаратно-программную среду, т.е. в виде абстрактного представления физической системы (аппаратных и микропрограммных средств машины) с точки зрения системного программиста, разрабатывающего системное программное обеспечение.
- Архитектура определяет принципы организации ЭВМ и функции компонентов, но не отражает внутренние механизмы их функционирования, например управление и передачу данных внутри компонента, его конструктивные и технологические особенности.

1.1. Многоуровневое представление

- Многопроцессорная машина может быть представлена в виде системы с многоуровневой организацией, понятие архитектуры, первоначально сформулированное для однопроцессорных машин, может быть распространено и на многопроцессорные машины (рисунок 1).
- На каждом уровне иерархии многопроцессорной машины можно выделить совокупность обрабатывающих модулей процессоров и модулей памяти, связанных между собой коммутационной средой. Для описания наиболее сложных из реально существующих многопроцессорных машин оказывается достаточно 3-4 уровней иерархии.

1.2. Архитектура параллельных ЭВМ



1.3. Типовая структура машины

Типовая структура машины в этом случае может быть описана следующим образом:

- множество (специализированных) процессоров и модулей памяти нулевого уровня образуют алгоритмически универсальный модуль первого уровня;
- множество процессоров и модулей памяти первого уровня образуют кластер (или процессор второго уровня);
- множество кластеров вместе с модулями памяти второго уровня образуют мультикластерную конфигурацию – мультипроцессор второго уровня.

1.4. Описание архитектуры многопроцессорной машины

- Определение архитектуры нижнего уровня:
 - формы представления программы и правил ее интерпретации,
 - программистские структуры процессоров,
 - системы команд, способы адресации;
 - типы, форматы и формы представления данных.
- Определение механизмов и правил управления последовательностью выполнения команд, взаимодействия процессоров между собой и с устройствами ввода-вывода.
- Дополнение описания архитектуры нижнего уровня распределением функций и правилами взаимодействия между уровнями.

2. Классификация ЭВМ по числу потоков команд и данных

- Одна команда в машине определяет операцию над одним элементом данных или над целой структурой.
- В случае операции над одним элементом формируется лишь один результирующий элемент данных, который затем может подвергаться последующей обработке следующей командой. Возникает “поток” команд для обработки одного “потока” данных.
- Если одна команда определяет операцию над несколькими элементами данных, то имеет место один поток команд и несколько потоков данных.
- В том случае, когда несколько команд, предназначенных каждая для обработки одного элемента данных, выполняются одновременно, существует несколько потоков команд и несколько потоков данных.

2.1. Критерий Флинна

Согласно этому критерию все ЭВМ делятся на четыре группы.

- **ОКОД – одиночный поток команд, одиночный поток данных.** В каждый момент времени обработке может подвергаться один объект данных (одно слово) по одной команде. В таких машинах, тем не менее, может существовать параллелизм “низкого уровня” – конвейеризация команд, сопроцессирование и др.
- **ОКМД – одиночный поток команд, множественный поток данных.** Эти машины используют параллелизм объектов, т.е. в них существует общее управление (единый поток команд), которое обеспечивает одновременную и синхронную обработку в разных процессорных элементах нескольких элементов данных. Все одновременно обрабатываемые элементы данных подвергаются одинаковому преобразованию.

2.2. Машины типа МКОД и МКМД

- ***МКОД – множественный поток команд, одиночный поток данных.*** К таким машинам следовало бы отнести гипотетические ЭВМ, в которых имеется множество процессоров, обрабатывающих один и тот же элемент данных, но под управлением различных команд. На практике трудно представить подобный алгоритм, поэтому многие исследователи считают этот класс машин вакантным.
- ***МКМД – множественный поток команд, множественный поток данных.*** В нескольких автономно работающих процессорах производится обработка различных объектов данных под управлением различных команд. К этому классу относят машины, которые служат для непосредственной реализации моделей параллельных вычислений, т.е. классические мультипроцессорные машины, машины с управлением потоком данных, редукционные машины.

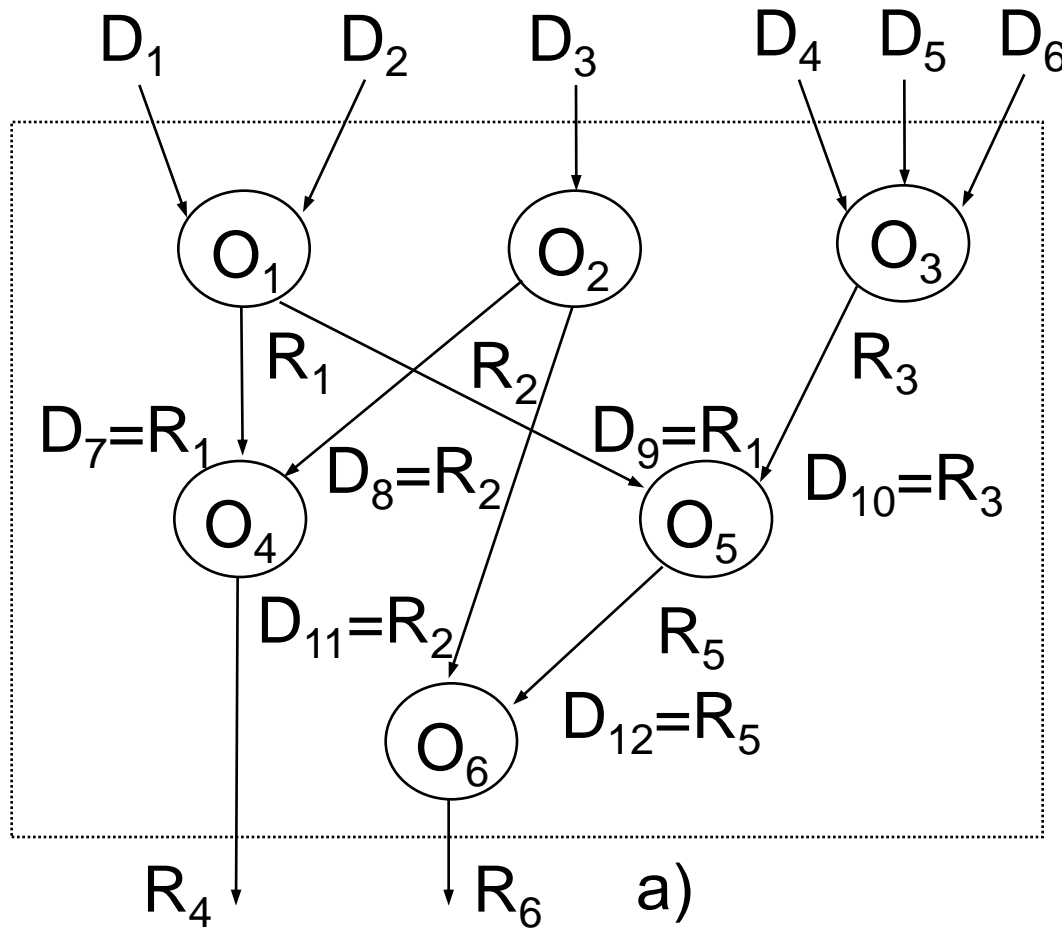
3. Механизмы управления и архитектура ЭВМ

- Механизм управления, используемый в модели вычислений, на основе которой строится ЭВМ, оказывает существенное влияние на архитектуру машины.
- В зависимости от применяемого механизма управления можно выделить три типа машин:
 - директивные,
 - потоковые,
 - запросные

3.1. Архитектура машин с директивным управлением

- Машины с директивным (принудительным) управлением имеют традиционную фон-неймановскую архитектуру. Каждая команда помимо кода операции и операндов содержит явное (указывается адрес перехода) или неявное (к адресу текущей команды добавляется константа) указание, к выполнению какой следующей команды должна приступить ЭВМ.
- Программа представляет собой списочную форму описания информационного графа и состоит из последовательности команд k вида:
 $O(D), (R),$
где O определяет примитивную операцию, соответствующую машинной операции, макрооперации, подпрограмме и т.п., (D) – множество входных операндов, (R) – множество выходных результатов.

Информационный граф (а) и списочная форма последовательной программы (б)

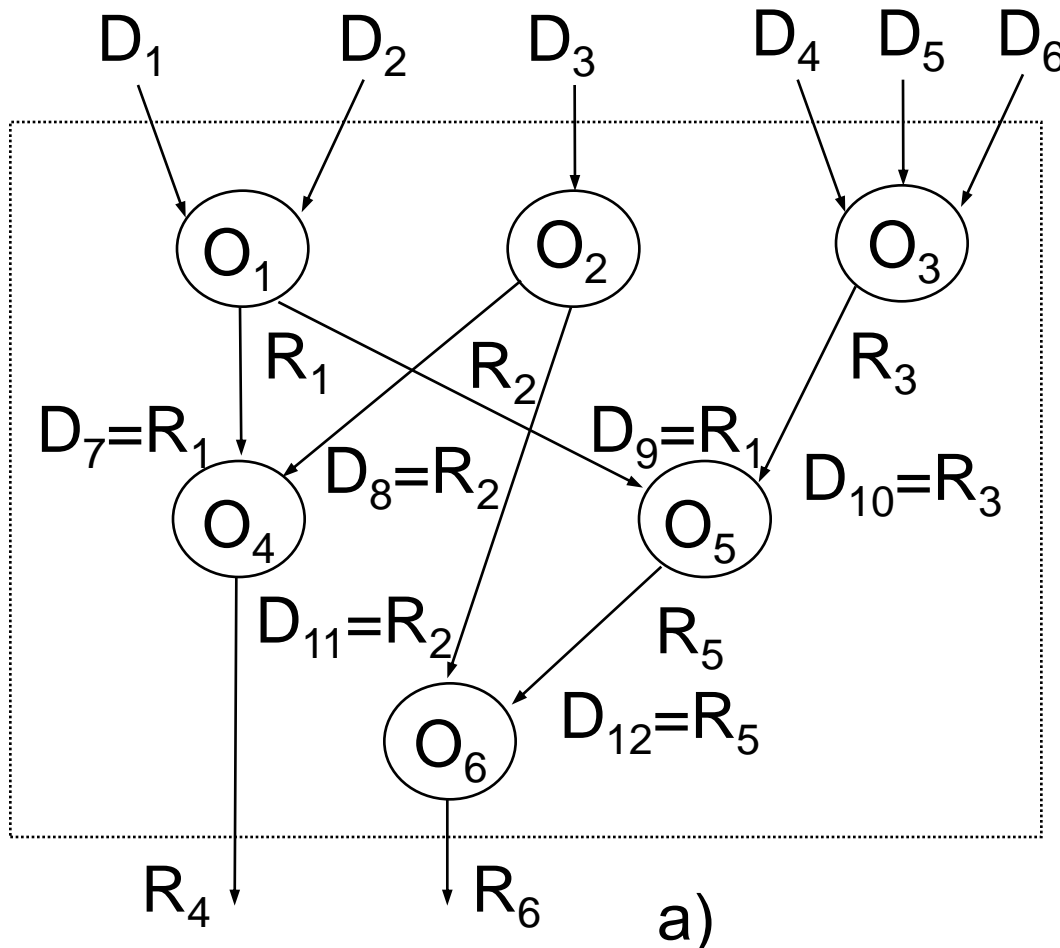


k1: $O_1 (D_1, D_2), (R_1)$;
 k2: $O_2 (D_3), (R_2)$;
 k3: $O_3 (D_4, D_5, D_6), (R_3)$;
 k4: $O_4 (R_1, R_2), (R_4)$;
 k5: $O_5 (R_1, R_3), (R_5)$;
 k6: $O_6 (R_2, R_5), (R_6)$.

б)

По информационному графу можно построить несколько последовательностей команд, приводящих к получению одинаковых результатов; но программа определяет лишь одну детерминированную последовательность.

Информационный граф (а) и списочная форма параллельной программы (б)



Если алгоритм допускает параллельное выполнение нескольких операций, то они могут быть объединены единым описанием – “макрокомандой” параллельного исполнения. Аппаратные средства ЭВМ должны выявлять параллельные конструкции **par** k и организовывать их выполнение.

par k1: ($O_1(D_1, D_2), O_2(D_3), O_3(D_4, D_5, D_6), (R_1, R_2, R_3)$);
par k2: ($O_4(R_1, R_2), O_5(R_1, R_3), (R_4, R_5)$);
 k3: $O_6 (R_2, R_5), (R_6)$. б)

3.2. Архитектура машин с потоковым управлением

- Потокковые машины переходят к выполнению оператора (операции), как только для него создаются условия, характеризующие возможность его выполнения.
- Такой способ управления называют также управлением потоком данных.
- Оператор может быть выполнен, если для этого создались все необходимые условия, т.е. на соответствующую вершину информационного графа поступили все необходимые операнды по входным дугам. Говорят, что такая вершина “сработала” и после этого ее результат по направленным дугам передается на другие вершины.
- Результат решения задачи будет получен после срабатывания последней вершины в информационном графе.

Списочная форма потоковой ЭВМ

- Аналогично машинам с директивным управлением информационный граф также может быть представлен списочной формой. Однако такая форма отражает не единственную последовательность операторов, а общую структуру алгоритма.
- Элемент такого списка, часто называемый шаблоном, может быть представлен в следующем виде:

$O, F, (d), (r),$

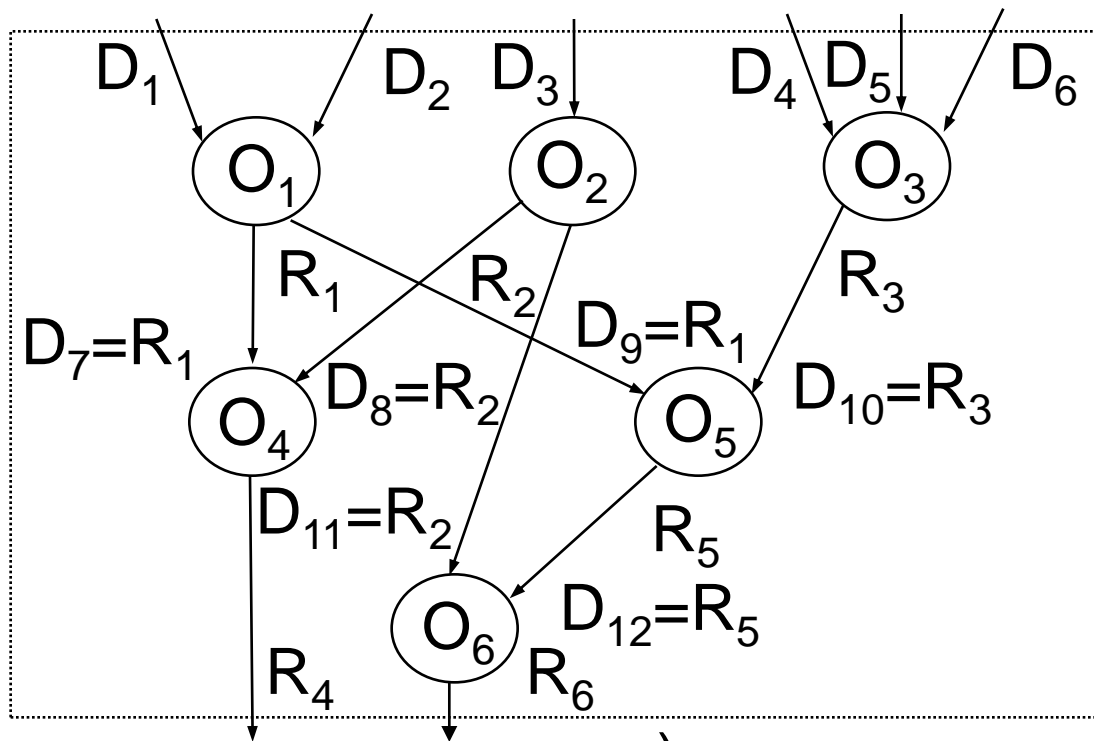
где O определяет примитивную операцию, соответствующую машинной операции, макрооперации, подпрограмме и т.п.,

F – условие срабатывания вершины данной операции в информационном графе (спусковая функция);

d – множество полей операндов, заполняемых по мере выполнения предшествующих операторов (срабатывания предшествующих вершин в графе); эти операнды могут использоваться как для выполнения операции O , так и для вычисления спусковой функции F ;

r – множество полей назначения, указывающих, каким шаблонам должны быть переданы результаты выполнения операции O .

Информационный граф (а) и списочная форма потоковой машины (б)



а)

$k1: t(x,x,1) O1 (d1,d2), (k4/1,k5/1);$
 $k2: t(x,1,1) O2 (d3), (k4/2,k6/1);$
 $k3: t(x,x,x) O3 (d4,d5,d6), (k5/2);$
 $k4: t(x,x,1) O4 (r1,r2), (r4);$
 $k5: t(x,x,1) O5 (r1,r3), (k6/2);$
 $k6: t(x,x,1) O6 (r2,r5), (r6).$

б)

$k1, \dots, k6$ –
 идентификаторы
 шаблонов,
 $t(x,x,1)$ – тег (Т)
 готовности операндов,
 операнды получены,
 если $T=t(1,1,1);$
 $O1, \dots, O6$ –
 примитивные операции;
 $d1, \dots, d6$ и $r1, \dots, r5$ – поля
 операндов;
 kJ/n – поля назначения
 (вход n вершины kJ в
 информационном
 графе).

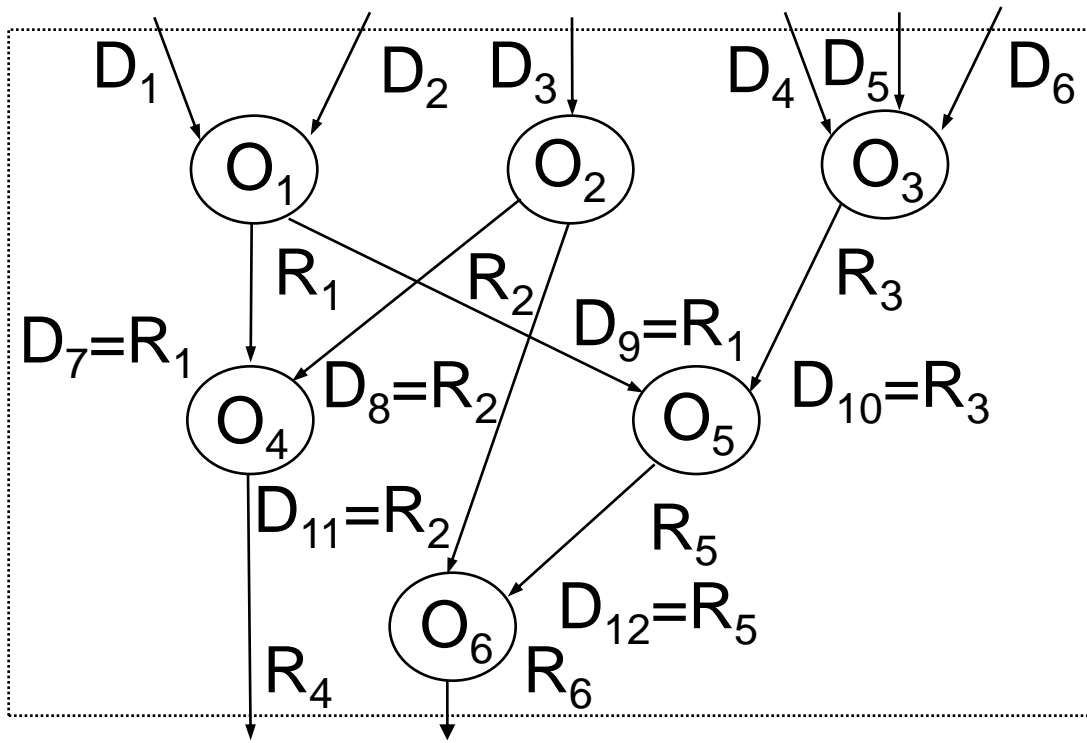
Рабочий цикл потоковой машины

- На каждом шаге работы потоковой машины анализируется поле тегов всех шаблонов и выполняются операции тех шаблонов, у которых $T=t(1,1,1)$.
- После выполнения операции полученные результаты в соответствии с заданными полями назначений поступают во входные поля операндов других шаблонов. Например, результаты выполнения операции $O1$ подаются в качестве первых операндов (на первые входы вершин информационного графа) шаблонов $k4$ и $k5$.
- При поступлении операнда в шаблон устанавливается соответствующий ему бит тега. В данном случае $T4=t(1,x,1)$ и $T5=t(1,x,1)$. После выполнения оператора соответствующий ему шаблон из списка шаблонов удаляется.
- Выполнение программы завершается, как только обнаружится, что список шаблонов пуст.

Работа потоковой машины по шагам

K1	t(1,1,1)	O1 (D1,D2)	(k4/1,k5/1)	Шаг 1. Выполнение и удаление готовых шаблонов
k2	t(1,1,1)	O2 (D3)	(k4/2,k6/1)	
k3	t(1,1,1)	O3 (D4,D5,D6)	(k5/2);	
k4	t(x,x,1)	O4 (r1,r2)	(r4)	
k5	t(x,x,1)	O5 (r1,r3)	(k6/2)	
k6	t(x,x,1)	O6 (r2,r5)	(r6)	
k4	t(1,1,1)	O4 (R1,R2)	(r4)	Шаг 2. Выполнение и удаление готовых шаблонов
k5	t(1,1,1)	O5 (R1,R3)	(k6/2)	
k6	t(1,x,1)	O6 (R2,r5)	(r6)	
k6	t(1,1,1)	O6 (R2,R5)	(r6)	Шаг 3. Выполнение и удаление готовых шаблонов
				Шаблонов нет

3.3. Архитектура машин с управлением по запросам



В таких машинах выполнение операций осуществляется в последовательности, определяемой запросами операндов.

Предположим, что возникла необходимость в получении результата R_6 , т.е. сформирован “запрос” R_6 . Этот запрос используется для вызова операции O_6 , однако последний не может быть выполнен из-за отсутствия операндов $D_{11} = R_2$ и $D_{12} = R_5$. В результате такого предварительного анализа операции O_6 порождаются два запроса: “запрос R_2 ” и “запрос R_5 ”.

Принцип работы запросной машины

- Процесс порождения запросов будет продолжаться до тех пор, пока не будет встречена операция, готовая к выполнению. В рассматриваемом примере такими операциями являются O1, O2 и O3.
- Процесс порождения запросов сопровождается ростом числа параллельных процессов, однако этот рост легко контролировать – можно задержать формирование запроса, а следовательно, и рост числа порождаемых процессов.
- Каждая выполненная операция заменяется соответствующим результатом, который передается по обратному пути, обеспечивая возможность реализации всех предыдущих запросов более высокого уровня.
- При этом можно сохранить результаты выполнения каждой операции, заменив ими спецификацию операции; тогда при последующих обращениях-запросах к этим операциям не потребуется повторного выполнения.

Управление потоком запросов и редукционные машины

- Рассмотренный способ управления вычислениями также называют управлением потоком запросов, он хорошо отражает особенности параллельной обработки и позволяет при этом контролировать число порождаемых параллельных процессов.
- Замена операций (операторов) и подграфов вычисленными значениями приводит к сокращению информационного графа (или соответствующей программы), в силу чего такие методы вычислений называются также методами редукции, а машины, реализующие различные их варианты, – редукционными машинами.