

Модели вычислений

Модель вычислений -

представляет собой описание некоторой гипотетической машины, способной реализовать определенный алгоритмический процесс.

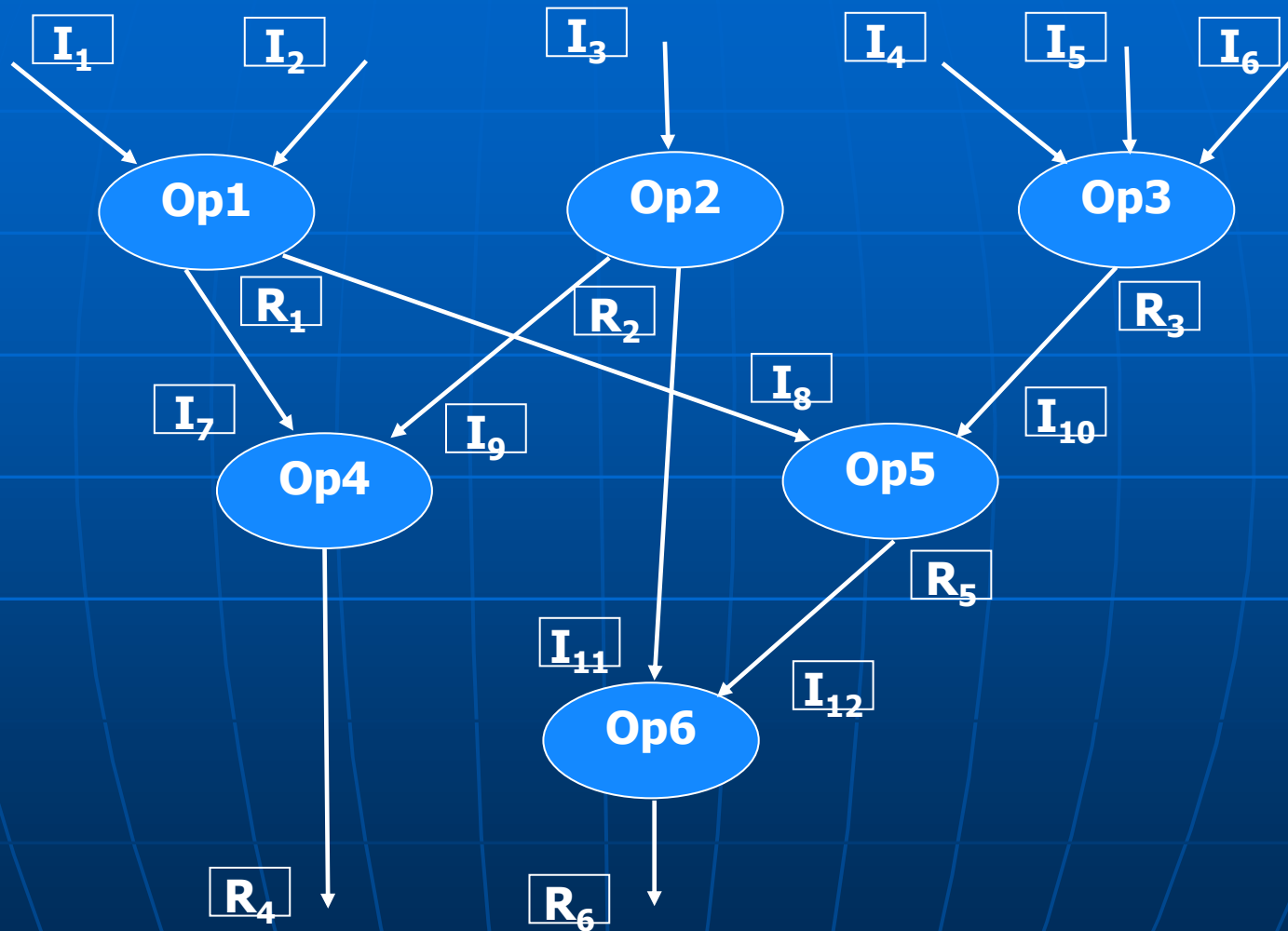
Такие модели могут создаваться с различными целями:

- для доказательства корректности алгоритмов,
- анализа семантики,
- определения эффективности программ,
- определения длительности выполнения программ

Графическое представление алгоритмов

- Информационный граф - ИГ
- Ярусно-параллельная форма - ЯПФ
- Сеть Петри

Информационный граф



«Плюсы» информационного графа

- + Позволяет выявить отношения «зависимости» между операторами – отношение **предшествования** и отношение **безразличия**.

Операторы **М** и **Н** находятся в отношении *безразличия*, если исходные операнды (**Г**) для каждого из них могут быть получены независимо от результатов выполнения другого оператора.

На графе отсутствует направленная дуга между ними.

«Минусы» информационного графа

- Отсутствует описание способов перехода от одного оператора к другому (порядок выполнения операторов).

Решение любой задачи на ЭВМ можно рассматривать как реализацию всей совокупности (необходимых) операторов информационного графа

Аппаратная поддержка реализации ИГ

- I. - В ВС предусмотрено одно место обработки (универсальное);
 - для хранения ИГ в виде некоторой программы, а также промежуточных результатов используется память;
 - совокупность операций устройства обработки представляет систему команд ВС.

- II. - Для каждого оператора ИГ предусмотрено своё устройство обработки;
 - все модули (блоки) связаны отдельными связями – сеть ВМ;
 - промежуточные результаты передаются между блоками;
 - программа представляет формализованное описание ИГ !;
 - настройка сети может выполняться статически или динамически (во время решения задачи).

Промежуточные решения – огромное число АСР вычислительных систем !

Последовательность выполнения операторов (модели вычислений)

I. Диррекативная

Управление потоком команд – следующим выполняется оператор, который явно определён в предыдущем (программистом !).

Указание на следующий оператор может быть явным или неявным.

Программа состоит из последовательности команд вида:
Op (I...), (R)

Все отношения предшествования учитываются
программистом на этапе составления программы.

Использование параллельных сред программирования может «улучшить» скорость выполнения программы – но выполнение недетерминированное !

II. Потокковая (data-flow)

Управление потоком данных – оператор может быть выполнен, если создались условия для его выполнения:

- на вход оператора поступили все операнды;
- «сработала» спусковая функция.

Результаты передаются по дугам другим операторам.

Программа представляет формальное описание ИГ:

$T = (Op, F, D..., An...) - \text{набор шаблонов}$
(вершин)

II. Потокковая (data-flow)

Готовность определяется аппаратно-программными средствами.

Все отношения предшествования учитываются автоматически !

Теоретически достижимо предельное распараллеливание !

- Проблемы связаны с неограниченным ростом числа параллельных процессов

III. Запросная (функциональная, редуccionная)

Управление потоком запросов– оператор может быть выполнен, если создались условия для его выполнения:

- на вход оператора поступили все операнды;
- поступил запрос на выполнение (результат) оператора;
- («сработала» спусковая функция.)

Если данные «не готовы» – формируется запрос к вышестоящей вершине. И так до «верхних» операторов.

Полученные результаты передаются по дугам нижестоящим операторам.

Готовность определяется аппаратно-программными средствами.

III. Запросная (функциональная, редуccionная)

Каждый выполненный оператор (подграф!) заменяется его значением и при следующих запросах не требуется его повторного выполнения!

Методы управления получили название методы редуccionии, а машины на его основе – редуccionионные.

Языки – функциональные.

- Проблемы связаны с двухпроходным режимом реализации ИГ: запросы – данные.

Механизмы передачи операндов

(механизмы вызова данных-операндов)

В общем случае операнд может быть представлен:
литералом, значением, ссылкой.

В виде литерала — значение становится известно в момент компиляции — в каждую команду помещается «копия» операнда. В процессе выполнения программы такие операнды не передаются между командами.

Строго говоря, механизмов вызова операндов два:

- вызов по значению;
- вызов по ссылке.

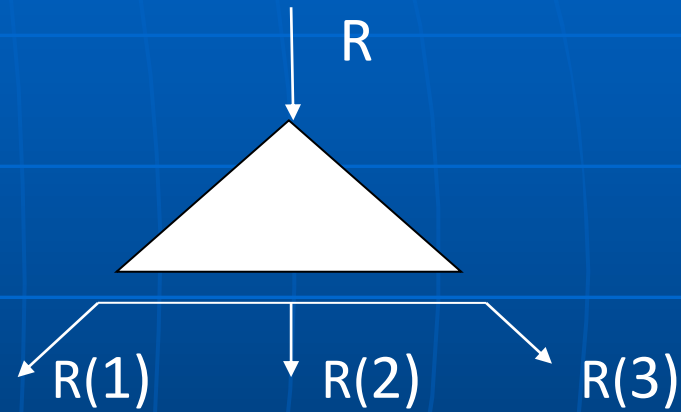
Вызов операнда по значению

Предполагает непосредственную передачу копий результата во все команды

Обычно вводят специальную операцию копирования:

Рассылка в виде сообщений, которые имеют:

- адрес процесса-получателя;
- собственно результат;
- указатель конца сообщения.



ММожно использовать для синхронизации параллельных процессов, что приводит к уменьшению числа ошибок при программировании.

ППример – транспьютерные системы (нужны links)

Вызов операнда по ссылке

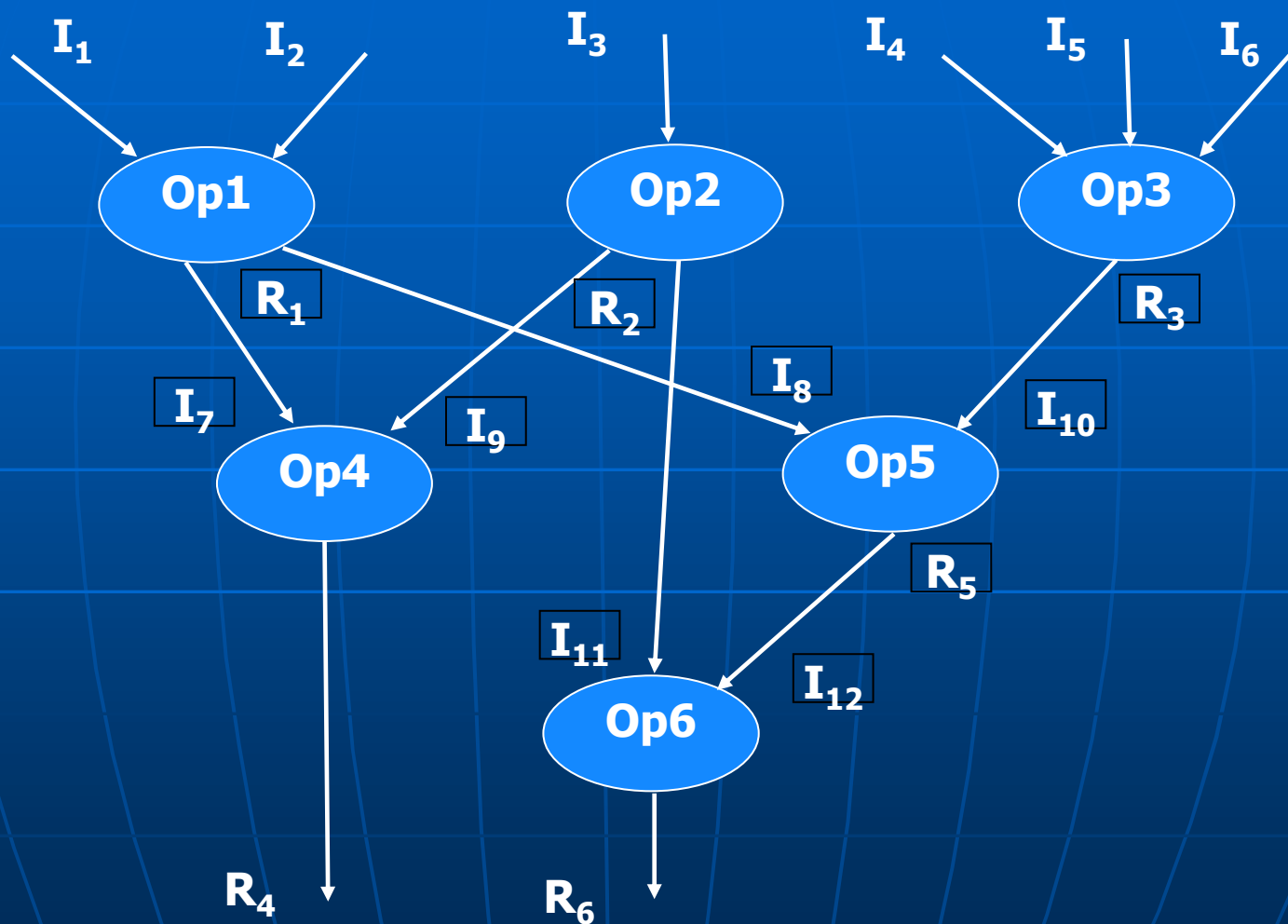
Используется выделенная область памяти (ячейка A), куда команда, порождающая R, помещает результат.

Командам, нуждающимся в R, необходимо сделать запрос $?(A)$ и получить операнд.

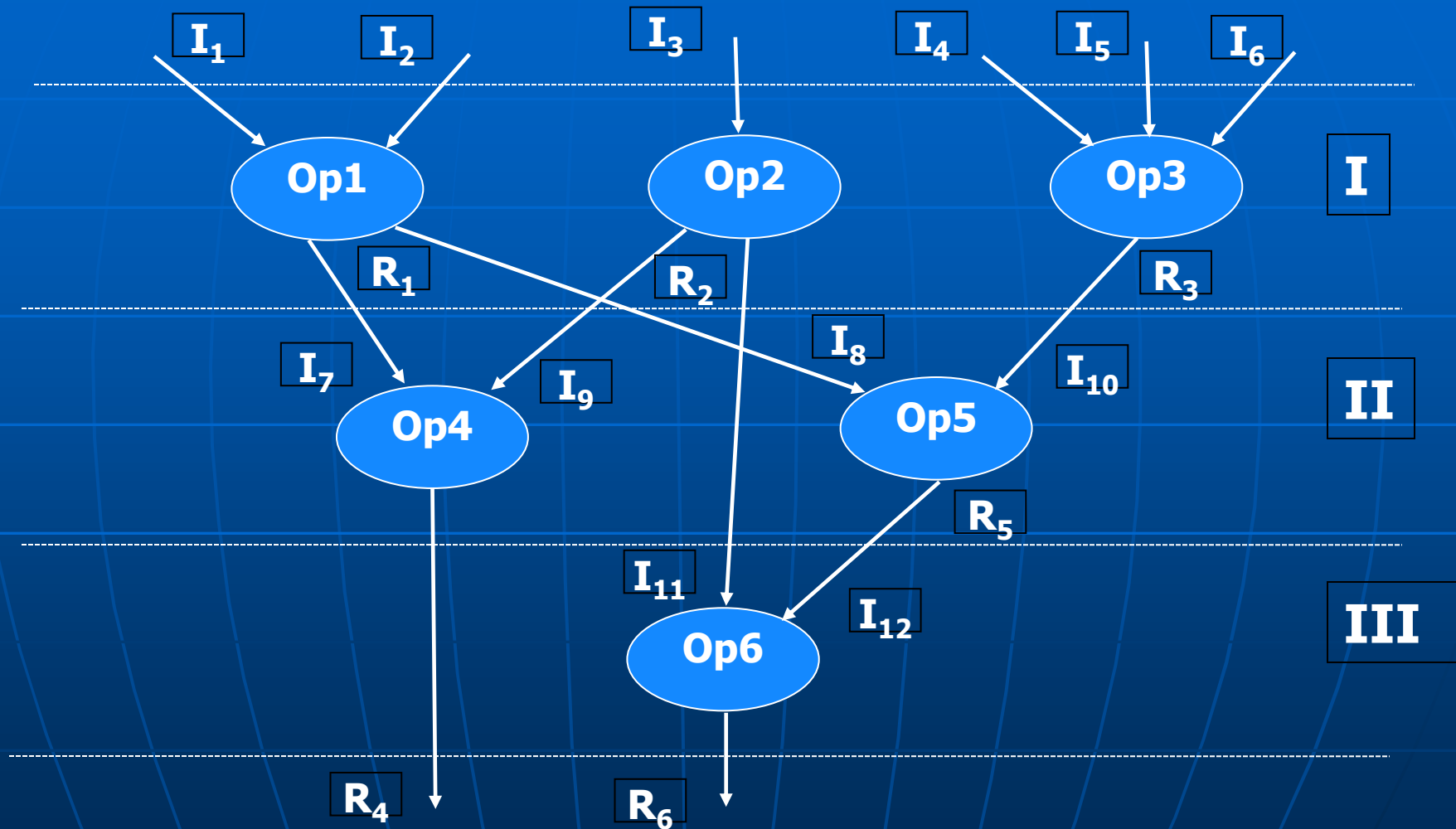
Расстановка ссылок происходит на этапе компиляции.

Механизм прост, но является источником ошибок и ограничивает параллелизма (общая память !)

Информационный граф

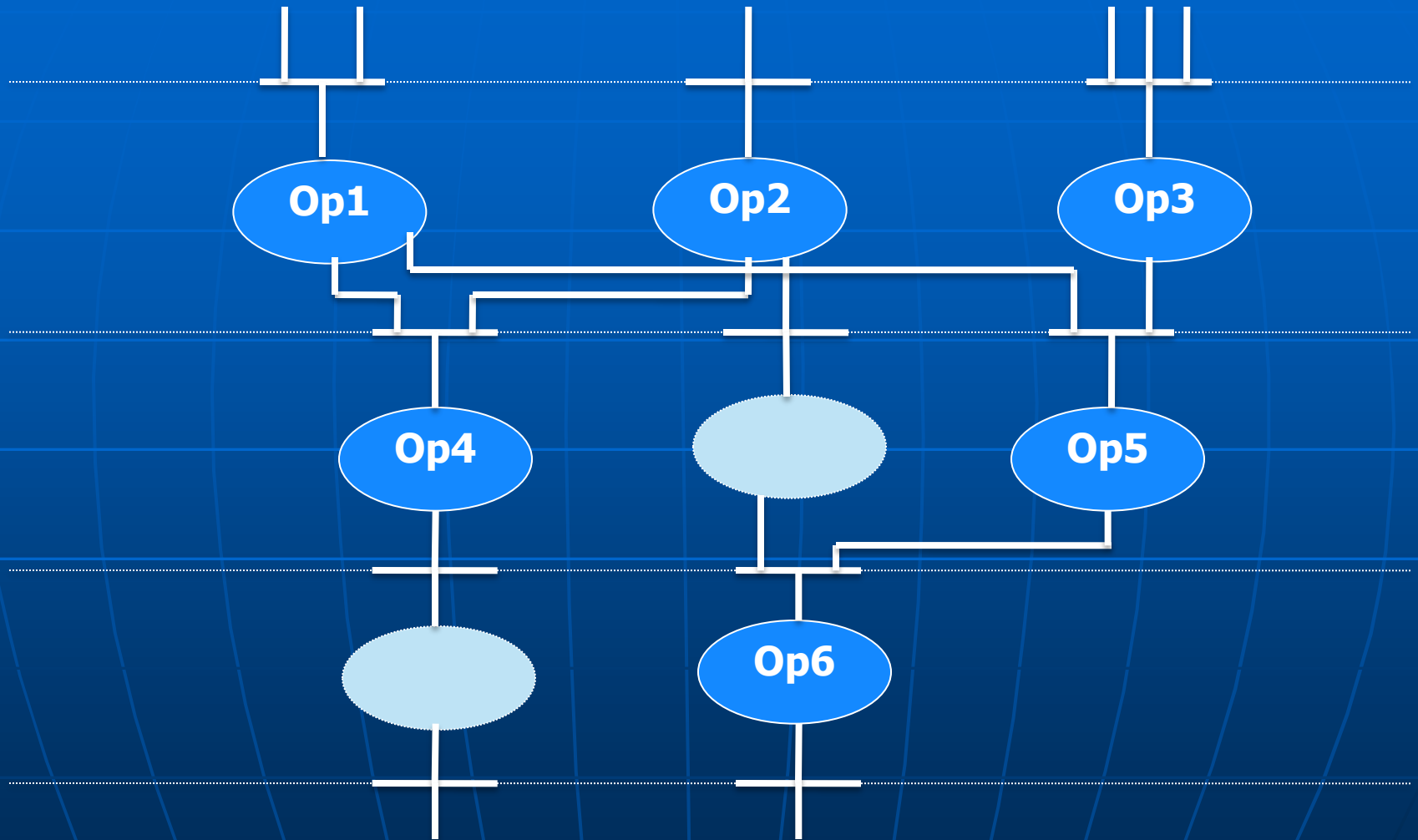


Ярусно-параллельная форма



Качественная оценка однородности, регулярности связности и параллельности

Сеть Петри



Срабатывание переходов – синхронно или асинхронно