Параллелизм уровня команд

- Параллелизмом уровня команд называется наличие в коде **независимых** инструкций, которые могут выполняться параллельно.
- На параллелизме уровня команд основаны две основные техники повышения производитель: конвейеризация и суперскалярные исполнение.
- Зависимость между инструкциями задает естественный порядок их исполнения.

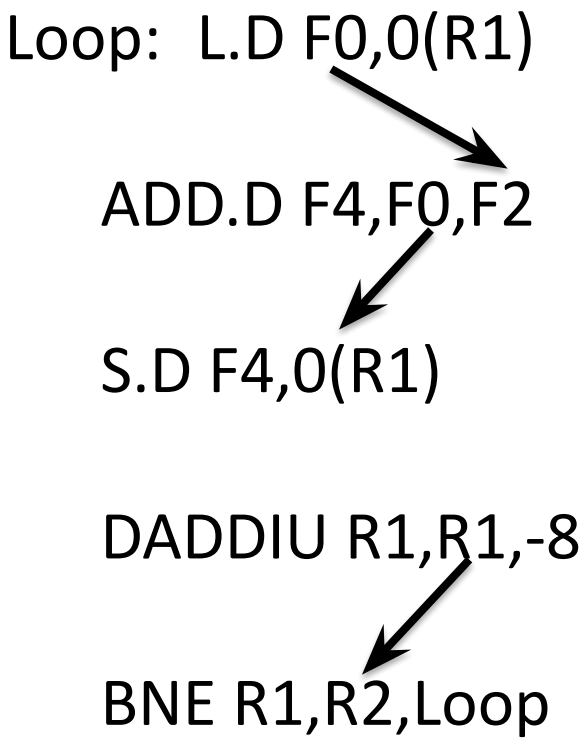
Зависимость между инструкциями

- зависимость по данным;
- зависимость по именам;
- зависимость по управлению.

Зависимость по данным.

- Инструкция J зависит по данным от инструкции I если результат работы инструкция I может быть использован инструкцией J.
- Отношение зависимости между инструкциями обладает свойством транзитивности, то есть если инструкция J зависит от инструкции K, а инструкция K зависит от инструкции I, то инструкция J зависит от инструкции I.

Зависимость по данным (пример).



Связь между инструкциями

- Через регистры
  - Просто выявить на этапе компиляции. Связь по имени регистра.
- Через память
  - Сложно, иногда невозможно выявить на этапе компиляции. Связь по значению регистра.

Зависимость по именам.

- Две инструкции зависят по именам, если использую одинаковые регистры или одинаковые ячейки в памяти, но при этом нет зависимости по данным и управлению.
- Являются устранимыми зависимостями, которые появляются в результате компиляции.

Тип зависимости по именам

- 1. Анти зависимость
  - между инструкциями I и J существует, когда инструкция J пишет в ячейку памяти или регистр, которую инструкция I читает.
- 2. Зависимость по выходу
  - существует , когда инструкция I и J пишут в одну ячейку памяти или регистр.

Пример зависимостей по именам.

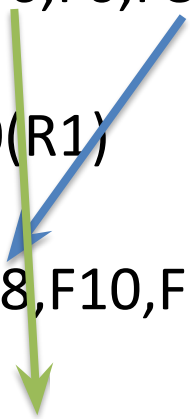
DIV.D F0,F2,F4

ADD.D F6,F0,F8

S.D F6,0(R1)

SUB.D F8,F10,F14

MULT.D F6,F10,F8



Анти зависимость

Зависимость по выходу

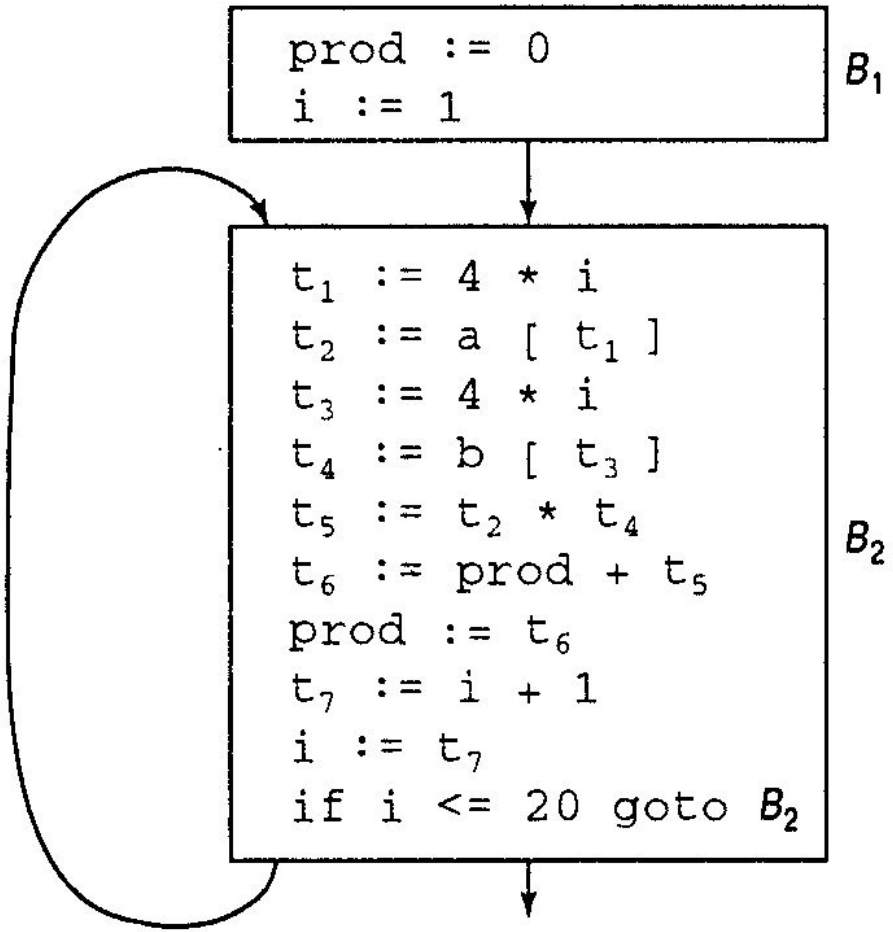
Базовый блок

- Это максимальная непрерывная последовательность инструкций, в которой не происходит ветвления, кроме последней инструкции.
- Первая инструкция базового блока
  - первая инструкция в программе
  - инструкция на которую ссылается инструкция перехода.
  - следующая инструкция за инструкцией перехода .
- Последняя инструкция базового блока
  - последняя инструкция программы
  - инструкция за которой следует, начало другого блока.

Граф потока управления (Control Flow Graph)

- Узлами графа являются базовые блоки.
- Направленная дуга идет от блока B1 к B2, если блок B2 может следовать за блоком B1 в некоторой последовательности исполнения.

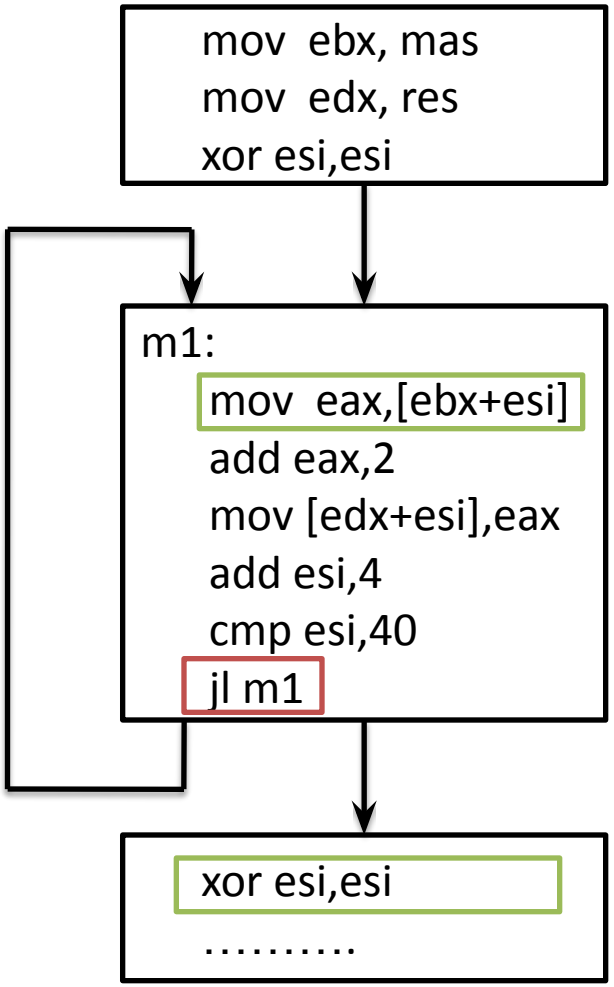
Пример граф потока управления



Зависимость по управлению

- Это зависимость всех инструкций базового блока от инструкции ветвления, которая ссылается на это блок.

Пример зависимости по управлению



Граф потока данных.

- Узлы графа – инструкции.
- Дуга связывает инструкцию производителя с инструкциями потребителями.

Критерии сохранения корректности программы

- 1. Сохранение потока данных
  - Поток значений передаваемых между инструкциями производителями и потребителями должен быть сохранен.
  - Инструкции ветвления делают поток данных динамическим.
- 2. Сохранение поведения исключений
  - Порядок, количество и тип генерируемых исключений должен сохраниться.

Когда применяются критерии?

- 1. При перестановки инструкции в программе.
- 2. При переходе от последовательного к параллельному исполнению.

Сохранение потока данных (пример).

```
    DADDU R1,R2,R3
    BEQZ R4,L
    DSUBU R1,R5,R6
L:
    OR R7,R1,R8
```

```
    DADDU R1,R2,R3
    BEQZ R12,skipnext
    DSUBU R4,R5,R6
    DADDU R5,R4,R9
skipnext:
    OR R7,R8,R9
```

Сохранение поведения исключений (пример)

```
DADDU R2,R3,R4
BEQZ R2,L1
LW R1,0(R2)
L1:
```

- Может быть сгенерировано исключение, если R2=0.