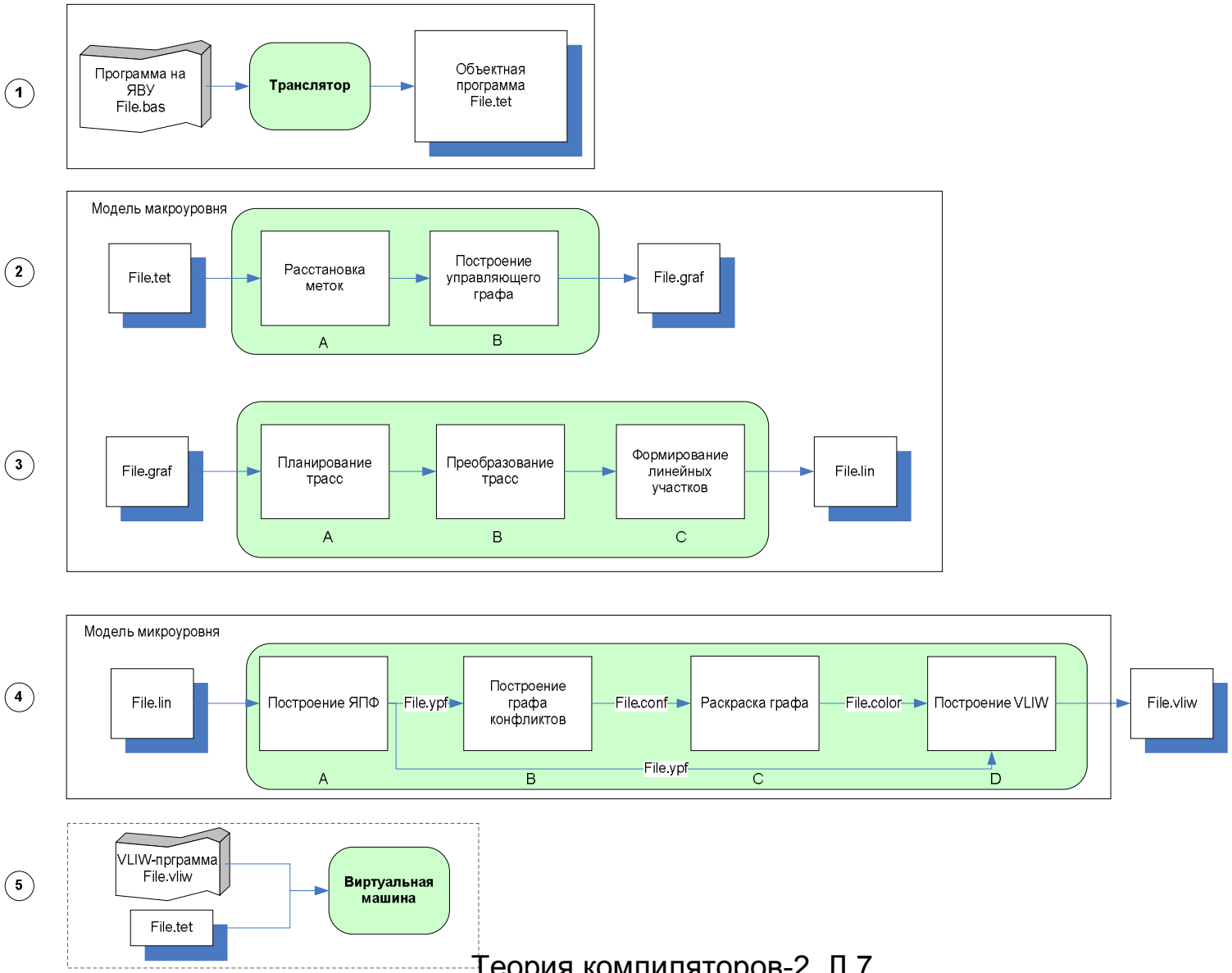
Теория компиляторов

Часть II

Лекция 7. Курсовой проект

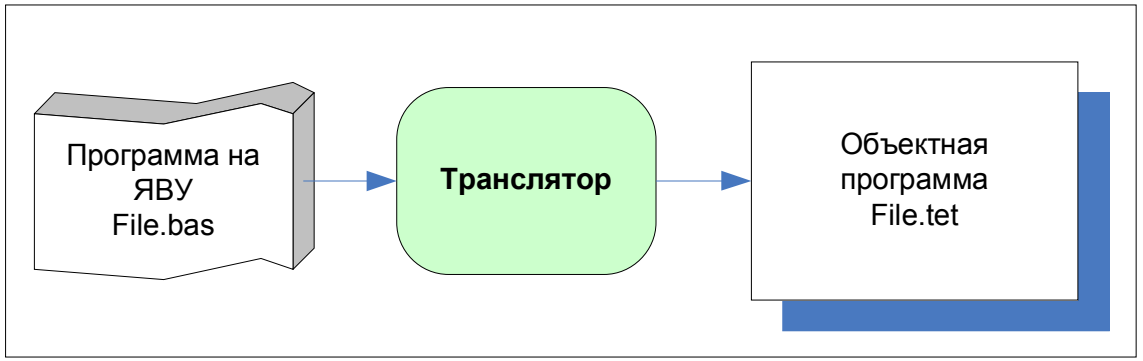
Общая схема ПО



1. Создание объектного файла

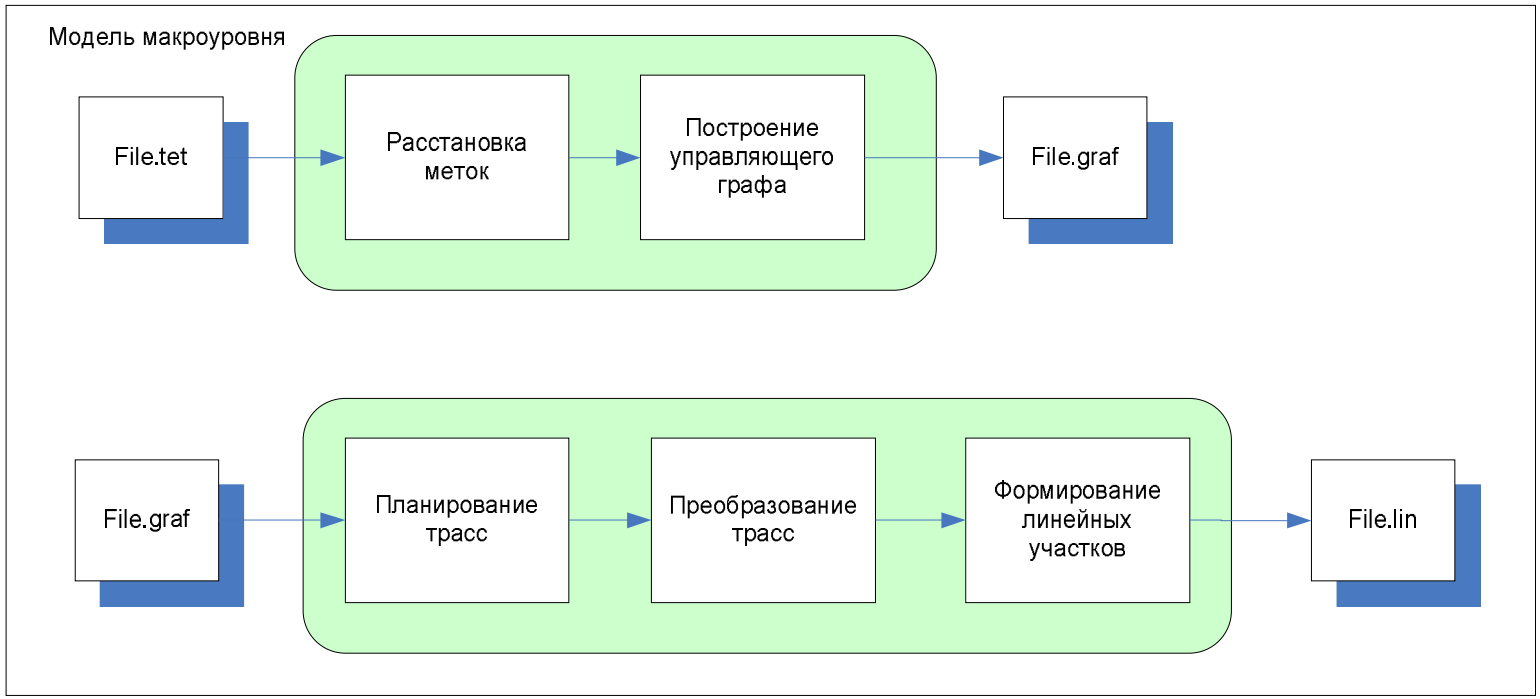
Пример объектного файла

```
<HEAD>
  DATA_LEN = 6
  TEXT_LEN = 4
  STACK_LEN = 20
  ; Описание архитектуры ВМ
  FU_NUM = 3   ; Количество ФУ в ВМ
  RF_SIZE = 10 ; Количество регистров в РФ
</HEAD>
<DATA>
  (1,0,50)          ; 000: переменная, число, 50
  (1,0,3.14)        ; 001: переменная, число, 3.14
  (0,1,”qwerty”)    ; 002: константа, строка
  (1,0,1)           ; 003: переменная, 1
  (1,1,”any string”) ; 004: переменная, строка
  (0,0,0)           ; 005: константа, число, 0
</DATA>
<TEXT>
  ((+,0,1,6))
  ((+,1,6,6);(*,3,5,7);(:=,1,5,))
  ((out,4,,);(+,7,3,1))
  ((out,1,,))
</TEXT>
```



Заголовок <HEAD> DATA_LEN TEXT_LEN STACK_LEN Прочая информация </HEAD>
Сегмент данных <DATA> </DATA>
Сегмент текста <TEXT> </TEXT>

2. Модель макроуровня



- **Алгоритм расстановки меток**
- **Алгоритм построения линейных участков и УГ**
- **Алгоритм построения трасс**

Алгоритм расстановки меток

```
-- Расстановка меток для тетрад
Цикл ДляКаждого в СписокТетрад: ТекущаяТетрада
  Если ТекущаяТетрада.Оператор = (OUT | IN | POP | PUSH | POPR | PUSHR) ТО
    ТекущаяТетрада.Метка := ПлохаяИнструкция
  Иначе Если ТекущаяТетрада.Оператор = (BREQ | BRNEQ | ... | BRNE | BRPL) ТО
    Если ТекущаяТетрада.Аргумент1 является числом ТО
      ТекущаяТетрада.Метка := Развилка
    Иначе
      ТекущаяТетрада.Метка := ПлохаяИнструкция
    Кесли
  Иначе Если ТекущаяТетрада.Оператор = JMP ТО
    Если ТекущаяТетрада.Аргумент1 является числом ТО
      ТекущаяТетрада.Метка := БезусловныйПереход
    Иначе
      ТекущаяТетрада.Метка := ПлохаяИнструкция
    Кесли
  КЕсли
КЦикла
```

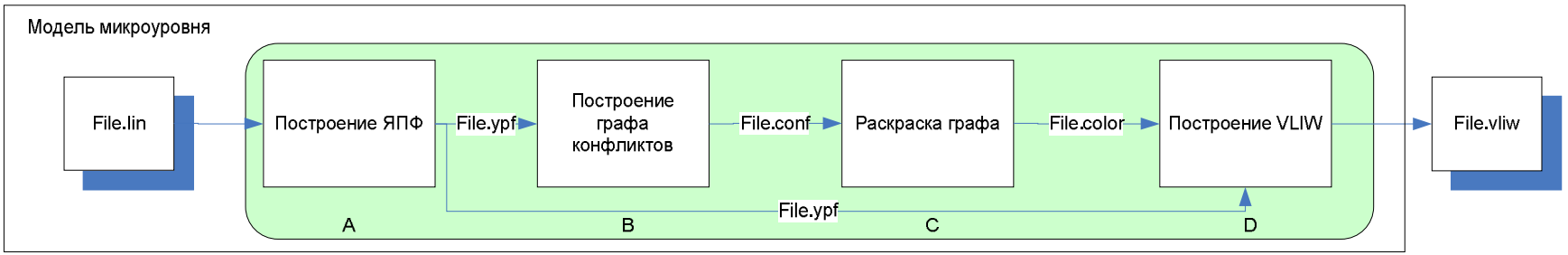
Алгоритм построения линейных участков и УГ

```
-- Формирование управляющего графа
Цикл ДляКаждого в СписокТетрад: ТекущаяТетрада
  Если ТекущаяТетрада является стоком И ТекущаяТетрада.Метка = ХорошаяИнстркция ТО
    УправляющийГраф.ДобавитьВершину
    УправляющийГраф.ПоследняяВершина.Дуга1 := номер следующей тетрады
  Кесли
  -- Граница линейного участка
  Начало блока последней вершины УГ := номер текущей тетрады
  ТекущаяТетрада.НомерЛинейногоУчастка := номер последней вершины УГ
  Если ТекущаяТетрада.Метка = ПлохаяИнстркция ТО
    УправляющийГраф.ДобавитьВершину
    УправляющийГраф.ПоследняяВершина.Дуга1 := номер следующей тетрады
  Иначе Если ТекущаяТетрада.Метка = Развилка ТО
    УправляющийГраф.ДобавитьВершину
    УправляющийГраф.ПоследняяВершина.Дуга1 := номер следующей тетрады
  Иначе Если ТекущаяТетрада.Метка = БезусловныйПереход ТО
    УправляющийГраф.ДобавитьВершину
    УправляющийГраф.ПоследняяВершина.Дуга1 := -1
  КЕсли
КЦикла
-- Расстановка дуг управляющего графа
Цикл ДляКаждого в СписокТетрад: ТекущаяТетрада
  Если ТекущаяТетрада.Метка = (Развилка | БезусловныйПереход) ТО
    Сдвиг := ТекущаяТетрада.Аргумент1
    ТекущаяТетрада.Дуга2 := ТекущаяТетрада.НомерЛинУчастка + Сдвиг
  КЕсли
КЦикла
```

Алгоритм построения трасс

```
Цикл ДляКаждого в УправляющийГраф: ТекущаяВершина
  Если ТекущаяВершина есть в трассе ТО
    СписокТрасс.ДобавитьТрассу
    Вернуться по списку возврата
  КЕсли
  ТекущаяТрасса.ДобавитьТекущуюВершину
  ТекущаяВершина = есть в трассе
  Если ТекущаяВершина.Дуга2 <> -1 ТО // т.е. переход
    Если ТекущаяВершина.Дуга1 <> -1 ТО // т.е. условный переход
      Если ТекущаяВершина.ПерваяТетрада.Оператор = (BREQ | BRNEQ)
        СчетчикЦикла := ТекущаяВершина.Дуга2
      КЕсли
      Если текущий вершины нет в списке возврата ТО
        Добавить в список возврата
      КЕсли
    Иначе // т.е. безусловный переход
      СчетчикЦикла := ТекущаяВершина.Дуга2
  Иначе Если ТекущаяВершина.Дуга1 = -1 ТО // конец трассы
    СписокТрасс.ДобавитьТрассу
  КЕсли
КЦикла
```

3. Модель микроуровня



Преобразование множества линейных участков в множество VLIW

- Вход: множество линейных участков (file.lin)
- Выход: множество VLIW (file.vliw)

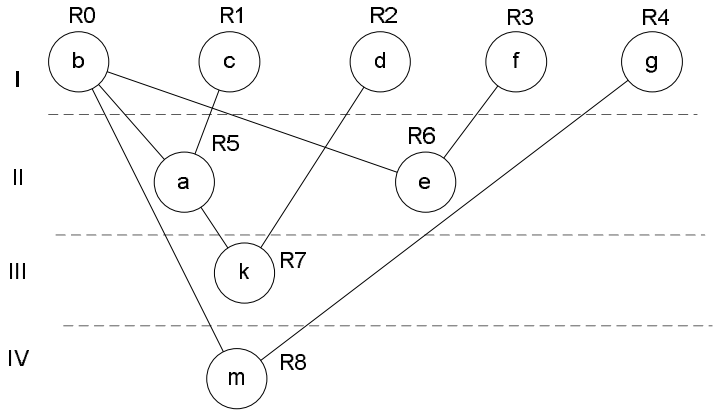
1. Построение ЯПФ.
   Вход: множество линейных участков (file.lin)
   Выход: ЯПФ (file.ypf);
2. Формирование графа конфликтов.
   Вход: file.ypf,
   Выход: граф конфликтов (file.conf)
3. Раскраска графа конфликтов.
   Вход: file.conf
   Выход: ЯПФ с раскраской графа по регистрам (file.color)
4. Формирование множества VLIW.
   Вход: file.color
   Выход: file.vliw

1. Файл множества линейных участков file.lin

Формат файла:
(OP,A1,A2,R)
…
#
Пример файла file.lin:
(+,,b,c)
(+,a,b,c)
(+,a,b,b)
(+,d,e,f)
(:=,b, ,a)
(:=,b,,f)
(:=,h, ,g)
#
(+,,b,c)
(+,a,b,c)
(+,a,b,b)
(+,d,e,f)
(:=,b, ,a)
(:=,b,,f)
(:=,h, ,g)
#

2. Файл ЯПФ file.ypf

Формат файла:
(NAME,OP,LEFT,RIGHT,RANK,ID,)
…
#
Пример файла file.ypf:
(b,,NULL,NULL,0,1,)
(a,,NULL,NULL,0,3,)
(d,,NULL,NULL,0,6,)
(e,,NULL,NULL,0,7,)
(h,,NULL,NULL,0,11,)
(c,+,NULL,1,1,2,)
(f,+,6,7,1,8,)
(g,:=,11,NULL,1,12,)
(c,+,3,1,2,4,)
(b,+,3,1,2,5,)
(a,:=,5,NULL,3,9,)
(f,:=,5,NULL,3,10,)
#
(b,,NULL,NULL,0,1,)
(a,,NULL,NULL,0,3,)
(d,,NULL,NULL,0,6,)
(e,,NULL,NULL,0,7,)
(h,,NULL,NULL,0,11,)
(c,+,NULL,1,1,2,)
(f,+,6,7,1,8,)
(g,:=,11,NULL,1,12,)
(c,+,3,1,2,4,)
(b,+,3,1,2,5,)
(a,:=,5,NULL,3,9,)
(f,:=,5,NULL,3,10,)
#

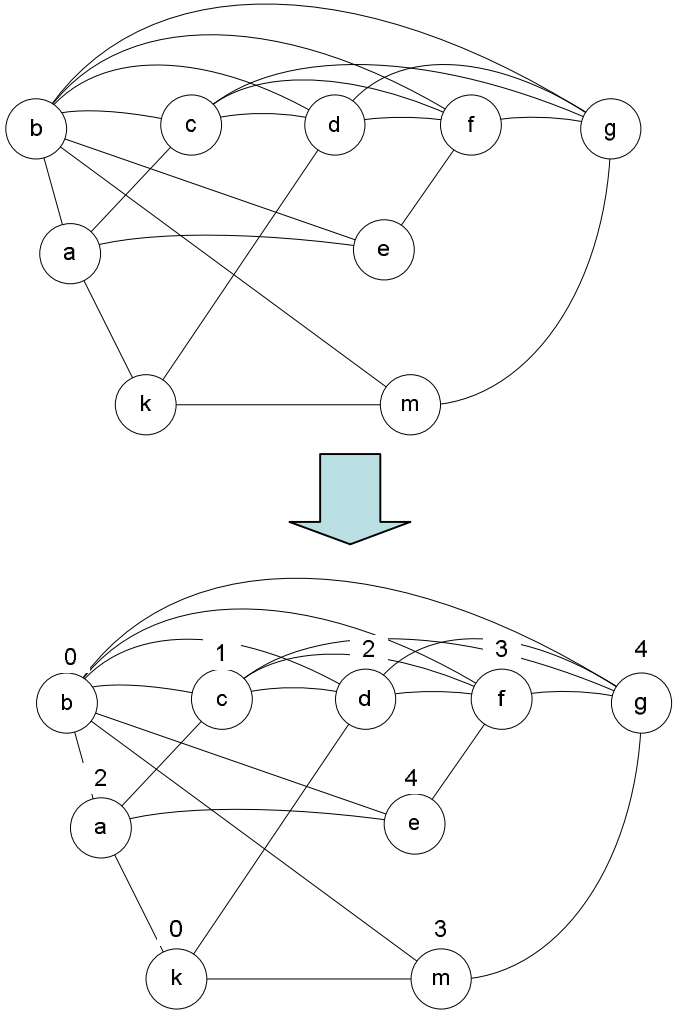


3. Файл ЯПФ с раскраской графа по регистрам file.color

Формат файла:
(NAME,OP,LEFT,RIGHT,RANK,ID,COLOR)
…
#
Пример файла file.color:
(b,,NULL,NULL,0,1,0)
(a,,NULL,NULL,0,3,1)
(d,,NULL,NULL,0,6,2)
(e,,NULL,NULL,0,7,3)
(h,,NULL,NULL,0,11,4)
(c,+,NULL,1,1,2,1)
(f,+,6,7,1,8,0)
(g,:=,11,NULL,1,12,2)
(c,+,3,1,2,4,2)
(b,+,3,1,2,5,3)
(a,:=,5,NULL,3,9,0)
(f,:=,5,NULL,3,10,1)
#
(b,,NULL,NULL,0,1,0)
(a,,NULL,NULL,0,3,1)
(d,,NULL,NULL,0,6,2)
(e,,NULL,NULL,0,7,3)
(h,,NULL,NULL,0,11,4)
(c,+,NULL,1,1,2,1)
(f,+,6,7,1,8,0)
(g,:=,11,NULL,1,12,2)
(c,+,3,1,2,4,2)
(b,+,3,1,2,5,3)
(a,:=,5,NULL,3,9,0)
(f,:=,5,NULL,3,10,1)
#

4. Граф конфликтов file.conf

Формат файла:
Матрица смежности графа
Число вершин
Матрица смежности графа
#
Пример файла file.conf:
12
0 1 1 1 1 1 1 0 0 0 1 0
1 0 0 0 0 0 0 1 0 0 0 1
1 0 0 1 1 1 1 0 0 0 1 0
1 0 1 0 1 0 0 0 0 0 0 0
1 0 1 1 0 0 0 0 1 1 0 0
1 0 1 0 0 0 1 1 0 0 1 0
1 0 1 0 0 1 0 1 0 0 1 0
0 1 0 0 0 1 1 0 0 0 0 1
0 0 0 0 1 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0 1 0 0 0
1 0 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 0 0 1 0
#
12
0 1 1 1 1 1 1 0 0 0 1 0
1 0 0 0 0 0 0 1 0 0 0 1
1 0 0 1 1 1 1 0 0 0 1 0
1 0 1 0 1 0 0 0 0 0 0 0
1 0 1 1 0 0 0 0 1 1 0 0
1 0 1 0 0 0 1 1 0 0 1 0
1 0 1 0 0 1 0 1 0 0 1 0
0 1 0 0 0 1 1 0 0 0 0 1
0 0 0 0 1 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0 1 0 0 0
1 0 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 0 0 1 0
#



5. Файл множества VLIW file.vliw

Формат файла:
[(OP,A1,A2,R) (OP,A1,A2,R)… (OP,A1,A2,R)]
…
#
Пример файла file.vliw:
[(load,b,,R0)(load,a,,R1)(load,d,,R2)(load,e,,R3)(load,h,,R4)]
[(+,,R0,R1)(+,R2,R3,R0)(:=,R4,,R2)]
[(store,R1,,c)(+,R1,R0,R2)(+,R1,R0,R3)(store,R0,,f)(store,R2,,g)]
[(store,R2,,c)(:=,R3,,R0)(:=,R3,,R1)]
[(store,R0,,a)(store,R1,,f)]
#
[(load,b,,R0)(load,a,,R1)(load,d,,R2)(load,e,,R3)(load,h,,R4)]
[(+,,R0,R1)(+,R2,R3,R0)(:=,R4,,R2)]
[(store,R1,,c)(+,R1,R0,R2)(+,R1,R0,R3)(store,R0,,f)(store,R2,,g)]
[(store,R2,,c)(:=,R3,,R0)(:=,R3,,R1)]
[(store,R0,,a)(store,R1,,f)]
#

4. Выполнение

- ВМ должна уметь выполнять как «обычный» поток тетрад, так и поток VLIW