

## 仿真语言自动排序器\*

杜铁塔 胡守仁

(电子计算机系)

**摘要** 连续系统高级仿真语言是一种非过程的程序设计语言,程序的书写顺序与执行顺序无关。为了将仿真语言程序编译成能在计算机上正确执行的程序,必须对源程序进行排序。本文在讨论分析了高级仿真语言动力学段特性的基础上,给出了并行仿真软件 PAR-SIM 中高级仿真语言 PARCSSL 编译器的自动排序器的实现算法。

**关键词** 仿真语言, 自动排序, 编译器

**分类号** TP391. 9

连续系统高级仿真语言(如 CSSL-IV, ACSL, ADSIM 等)编制的仿真程序常包含三大部分:初始化段、动力学段以及运行说明段。动力学段是仿真程序的核心,用于描述系统的数学模型。为了使动力学段程序更接近数学模型的代表形式,仿真语言文本规定,该段程序中语句的书写顺序与其执行顺序无关。用户可简单地将数学模型转换成仿真程序,而无须象 FORTRAN 等过程式程序设计语言那样要考虑如何按计算的先后次序来编制仿真程序,因此仿真语言是一种非过程的程序设计语言。这种非过程特性无疑方便了用户,但增加了仿真语言编译器的负担,它必须将一个以任意顺序书写的程序转换成具有正确执行顺序的仿真程序。这一过程称之为“自动排序”,完成这项工作的程序称为“自动排序器”。本文将介绍我们研制的并行仿真软件 PARSIM<sup>[1]</sup>中高级仿真语言 PARCSSL 编译器的自动排序器。

### 1 关于动力学段的几点说明

由于动力学段中的程序代码需作排序,故对这一段代码有如下要求。

**变量需定义** 每一个在表达式中引用的变量必须以下述三种方式之一确定其值:或出现在动力学段内某个赋值语句的左端;或曾在动力学段外被赋值;或为某一状态变量,运算时取其初值或上一帧的值。

**不含代数环** 各变量(状态变量除外)不能直接地或间接地引用自身。如不能出现如下形式的语句:

$A = A + 1$  或

\* 1991年11月21日收稿

```
A=B
B=C
C=A
```

出现这种形式的代码在仿真语言中称为包含了“代数环”。

**单一赋值** 所有变量只能被赋值一次。

**无控制语句** 由于动力学段的非过程特性，故不允许出现如转换之类的控制语句。

**过程代码用过程块引入** 当过程式代码必须时，如循环、分枝，可用过程代码块(procedural block)引入。过程块的结构为

```
procedural (outlist=inlist)
  ... (过程式代码)
end procedural
```

outlist 和 inlist 分别列出了在过程块中欲引用的变元和将被赋值的变元，以便正确排序。

动力学段包含两类语句，即赋值语句和过程块。赋值语句又可分为单输出赋值语句和多输出赋值语句。前者如同 FORTRAN 语言的赋值语句，后者形式为：var1, var2, ... = fun(in1, in2, ...)。赋值语句右边为一函数引用。就排序而言，过程块的处理原理上与多输出赋值语句一样。为简便计，下面仅讨论如何对动力学段中赋值语句进行排序。

## 2 自动排序器

### 2.1 排序表

为了描述赋值语句中输出变元(赋值号“=”左边出现的变元)和输入变元(赋值号右边表达式中出现的变元)之间的关系，我们引入关系 R，

$$R = \langle V_{out}, V_{def}, F \rangle$$

$$V_{out} = F(V_{in})$$

$$V_{in} = V_{out} \cup V_{def}$$

其中：  $V_{out} = \{v | v \text{ 是动力学段中被赋值量(输出变元)}\}$

$V_{def} = \{v | v \text{ 是动力学段之外被赋值的量或状态变量}\}$

$F = \{f | f \text{ 是某个表达式运算规则}\}$ 。

经编译器第一遍扫描后，即可获得上述关系。

为了表征关系 R，我们定义一种称为“排序表”的数据结构。排序表由若干纪录组成，每个赋值语句对应一个记录。每个记录定义如下：

输出变元名	已赋值标志	已排序标志	尚未被求值的 输入变元计数	尚未被求值的 输入变元名表	语句号
-------	-------	-------	------------------	------------------	-----

输入变元名表由赋值号右端表达式中互异的变量名组成。随着排序的不断进行，变元表中变元个数不断减少。输入变元计数等于零时，意味着对应语句成为可计算的了。这时可根据“语句号”产生相应的赋值语句，该语句具有正确的执行顺序。每个记录中还用到两个标志域，一个是“已赋值”标志，用于指示该输出变元是否可引用；另一个是“已排序”标志，用于表征该输出变元是否已被排序。例如有两个赋值语句：

$$X = \text{INTEG}(X\text{DOT}, X\text{IC})$$

$$X\text{DOT} = X + \text{SIN}(Y) * 0.2 + A * B$$

这里 X 是状态变量。对应的记录为

X	ON	OFF	2	XDOT		XIC	
XDOT	OFF	OFF	4	X	Y	A	B

“已排序”与“已赋值”两个标志不等同。如上面第一个语句中, X 是“已赋值”的, 因为在动力学段执行时使用的是 X 上一帧的值, X 为已知。但 X 所在的赋值语句可能尚未被排序。另一方面, “已排序”者一定是“已赋值”的了。

## 2.2 排序算法

构造好排序表后, 即可遵循下面的算法进行排序。

设立一工作集合  $V_{out}$ 。

$$(1) V_{out} \leftarrow \Phi;$$

$$(2) \text{求 } V'_{out} = \{v \mid v \in V_{out} \text{ 且 } v = f_v(V_s)\}$$

其中,  $f_v$  为变量  $v$  的求值规则,  $V_s \subseteq V_{def}$ ;

(3) 输出  $V'_{out}$  所对应的赋值语句集,

$$V_{def} \leftarrow V_{def} \cup V'_{out},$$

$$V_{out} \leftarrow V_{out} \setminus V'_{out};$$

(4) 重复①②③, 直到下述条件之一被满足为止

i)  $V_{out} = \Phi$  (排序正常结束),

ii)  $V'_{out} = \Phi$ , 但  $V_{out} \neq \Phi$  (出现代数环)。

## 2.3 实现时的几点考虑

(1) 对于单输出赋值语句, 每个语句对应排序表中一个记录。对于多输出赋值语句, 情形稍有不同。虽然我们在排序表中为每一个输出变元构造一个记录, 但只有第一个输出变元对应的记录中, 输入变元名表由右端表达式中各输入变元组成, 从第二个输出变元开始, 它们的输入变元名表不同于第一个输出变元, 仅由第一个输出变元组成。之所以如此处理, 是因为只要对第一输出变元完成了排序, 随后各变元便随之排好了。

(2) 集合  $V_{def}$  在实现时以线性表形式表示。集合  $V'_{out}$  中的元素并入  $V_{def}$  时, 我们将其依次排在  $V_{def}$  的尾部, 而不作任何分类排列。从排序算法可以看出,  $V_{def}$  在排序过程中不断增大。当判断一尚未排序的记录的输入变元是否属于  $V_{def}$  时, 我们自  $V_{def}$  的线性表的末尾向前作线性搜索, 其平均搜索次数远小于依“从头到尾”顺序搜索  $V_{def}$  线性表的平均搜索次数, 其道理是显而易见的。

(3) 如有一些输入变元没有在动力学段之外被计算过, 又没有在动力学段中被赋值, 那么这些变元很可能是用户写程序时写错了, 或忘了赋值。排序器将检测出这些输入变元, 并报告给用户以示警告。这样为用户调试程序提供了方便。

(4) 通过排序, 还可以发现一些赋值语句(如果有的话), 它们所引用的量(包括状态变量)都是动力学段以外赋值的。这意味着它们在一个采样周期或更长一个时间间隔内其

值不变。而一个采样周期内动力学段一般将被循环执行很多次，因此有必要将其提到动力学段之外执行，以减少动力学段的执行时间，从而缩短了仿真求解时间。

### 参 考 文 献

- 1 杜铁塔. 同构型并行仿真机系统结构及其软件支撑环境的研究. 国防科技大学博士论文, 1990

## Auto-sorter of Simulation Language

Du Tieta Hu Shouren  
(Department of Computer Science)

### Abstract

Continuous system simulation language is a non-procedural programming language. The written order of the statements bears no relation to the execution order. The statements must be sorted (or re-arranged) automatically by the compiler so as to generate the object codes with correct executing order. Based on analysing the characteristics of the dynamic section of simulation programs, the algorithms of auto-sorter implemented in the simulation language PARCSSL compiler of the software for parallel simulation PARSIM are given in this paper.

**Key words** simulatin language, compiler, auto-sorting

---

(上接第 21 页)

## The Study of 1.25cm Gunn Effect Oscillator

Chen Guoqiang He Chongjun  
(Department of Electronic Technology)

### Abstract

In this paper, a kind of waveguide Gunn effect oscillator at 1.25cm is discussed. Its main cavity uses the structure of adjustable coaxial height reduced waveguide. This makes the circuit match the diode well. The paper puts stress on discussing the influence of the size of each part on the characteristic of the oscillator. High Q  $H_{011}$  mode cylindrical cavity is used as the steady-frequency cavity and metal materials with different line-expand coefficients for temperature compensation is also used. These improve the frequency stead degree. In 23GHz-25GHz band, using WT55 Gunn effect diode, the output power of the oscillator is larger than 30mw, the frequency stead degree is  $5 \times 10^{-5}$ .

**Key words** microwave, waveguide, oscillator, 1.25cm Gunn effect diode