

# 一个小型机上的通用电路分析程序——GCAPN\*

洪先龙 孙家广 吴启明 柳西玲 王泽毅  
(清 华 大 学)

## 提 要

电路分析技术是电路设计工作者有力的助手。在小型机上实现电路分析程序将更有利于它的普及和推广，可为更多的电路设计工作者使用。本文介绍的 GCAPN 是配置在配有磁盘操作系统 RDOS 的 DJS-130 机或 NOVA 机上的通用电路分析程序，该机具有 64 K 字节的内存。

GCAPN 具有一个直观易学的输入语言，采用“改进节点法”列方程，用稀疏矩阵技术和隐式变步长的积分法来求解方程，在数据结构和软件技术上采用了适于小型机特点的方法。GCAPN 中还备有 MOS 和双极型晶体管的模型。用 GCAPN 可进行直流、交流和瞬态分析。

计算机辅助电路分析程序是电路设计的有力工具，它对于研制一般电子线路设备及大规模集成电路所起的作用是人所共知的。目前，越来越多的电子线路的设计师和大规模集成电路设计者重视和使用这个技术，并且在实际使用中取得了越来越显著的效果。

国外计算机辅助电路分析技术发展很快，在很多大型计算机上配备了通用的电路分析程序。如美国有 ASTAP、SPICE、ECAP-II、SCEPTRE 等。目前除了进一步研究如何使电路分析技术适应大规模网络和提高分析能力这一方面以外，也有不少人研究在小型机上实现电路分析程序，使它更普及。例如美国近几年来发表的 BIAS-D、小型机上的 ECAP-II、min MSINC 等都是这方面的例子。国内也有不少单位在研制电路分析程序，但从国内现有条件看，小型机较为普遍，要分析的电路有不少是二、三十个节点的电路，即使是节点数较多的大型电路，也可以分解成若干块来分析，或者其中关键部分电路的规模不大。因此，研制小型机上的通用电路分析程序，能为更多的电路设计师使用这一先进技术。

GCAPN 是配置在有 RDOS 支持下的 DJS-130 机或 NOVA 机上的通用电路分析程序，如作少量修改可搬到同样规模的小型机（如 PDP-11）或微处理机上。程序是用 FORTRAN 语言编写的，它采用了适于小型机的数值方法、数据结构和软件技术。用它可进行线性和非线性电路的直流、交流和瞬态分析。

\* 1980 年 10 月 6 日收到。

## 一、GCAPN 输入语言的梗概和特点

GCAPN 输入语言是面向电路分析的专用语言, 它为不熟悉程序设计或不熟悉电路分析的使用者提供使用 GCAPN 的工具。由于它直观、简单, 因此易学。

一个电路分析程序, 若没有一个直观方便的输入语言, 即使程序功能很强, 但对用户来说是不受欢迎的。因为在每一个电路中, 元件种类和参数都很多, 各种元件的表示方式也不相同, 电路连接关系也很复杂, 如没有一个直观方便的语言去描述, 用户很容易弄错了。因此, 向用户提供一个好的语言对提高分析效率和便于普及推广都是十分重要的。

GCAPN 提供了十一种语句供用户描述要分析的电路的结构、元件参数、分析类型和输出要求等。其中元件语句和模型语句是语言中的核心部分。元件语句允许用户描述十八类元件, 包括各种无源元件、互感和控制源、各类晶体管以及各种电源。通过元件语句和模型语句的结合, 提供了各种非线性元件的描述方法。程序中已提供了若干种常用非线性元件(例如二极管、MOS 晶体管、双极型晶体管等)的数学模型, 并且还可以根据需要加以扩充。如用户需要, 也可以按照一定的规格自行编制 FORTRAN 子程序嵌入到系统中, 然后按照用户自行定义的模型计算。

GCAPN 语言是一种自由格式语言, 每个语句行的格式类似于 FORTRAN 程序行, 书写自由, 没有过多的限制。各语句的先后顺序除了元件语句不能紧跟在模型语句之后以外, 可以自由安排。每类语句以第一个字符作关键字。为了便于记忆和阅读, 采用英语词汇作语句标识符, 以常用电路元件符号作元件类型的关键字。如 BEGIN 为开始语句, END 为终了语句, PRINT 为打印语句, MODEL 为模型语句等。又如用 R、L、C 分别表示电阻、电感、电容, PNP 和 NPN 表示三极管类型等。在语言中还增设了以 C 为开头的注解语句, 其作用和 FORTRAN 中注解行类似, 便于阅读和交流电路源程序。

通常, 在使用 GCAPN 前, 用户应将要分析的电路的每个元件命名, 并将电路中各节点编上号, 设置参考节点(即为接地点), 然后再根据 GCAPN 语言把电路写成源程序。例如

```
R12, 1—8, 1 K
C3, 2—0, 100
L345, 1—4, 50
```

就表示一个名叫  $R_{12}$  的电阻, 连接在 1 与 8 号节点之间, 其值为 1 千欧姆; ……。

有关 GCAPN 输入语言使用方法的更详细情况可参见 [1]。

## 二、GCAPN 中的“改进节点法”

GCAPN 采用了一种适于小型机特点的“改进节点法”来建立电路方程, 它具有方程阶数低、方法直观简单、且保持了导纳矩阵的稀疏性等优点。

对于仅含有电导和独立电流源的电路, 节点法很容易建立如下的方程组:

$$YV = I \quad (1)$$

其中  $Y$  是节点导纳矩阵,  $V$  是未知节点电压矢量,  $I$  是已知电流矢量。节点法具有方法直观、程序简单、方程阶数低等优点。但它不能处理独立电压源和与支路电流相关的元件, 支路电流也不能作为未知量。1975年美国的 Chung-Wen Ho 等提出的改进节点法 (Modified Nodal Approach 简称作 MNA)<sup>[2]</sup> 克服了上述矛盾, 应用到美国目前的一些电路分析程序中。改进节点法引入了支路电流作为未知量。这样, 对与电流相关的元件如电感, 则可很方便地建立起以支路电流作为未知量的电路方程。对独立电压源, 在利用节点法写出与独立电压源相连的节点方程式时, 由于独立电压源的内阻为零, 无法写出此支路上的电流表达式。改进节点法在独立电压源支路中引入支路电流作为未知量。这样, 一个独立电压源就可以方便地建立起节点电压和支路电流两个方程。对于如电阻、电容这样一些元件, 如要求这些元件上的电流时, 除了对这些元件列一个节点电压方程外, 还要列一个支路电流方程。改进节点法列出的方程组如下所示:

$$\begin{bmatrix} Y_R & B \\ C & D \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} = \begin{bmatrix} J \\ F \end{bmatrix} \quad (2)$$

其中  $Y_R$  是导纳矩阵,  $B$  是由电流未知量支路中产生的, 其中只有 +1 或 -1,  $C$  和  $D$  是表示电流未知量支路关系的,  $C$  的非零结构与  $B$  的转置相同。 $V$  是要求的节点电压矢量,  $I$  是要求的支路电流矢量,  $J$  和  $F$  是已知的电流、电压矢量。此方法一共要求解  $(Nd + 2E_0 + Id)$  个独立方程, 其中  $Nd$  是除与独立电压源相接的节点外的其它未知电压节点,  $E_0$  是独立电压源数,  $Id$  是支路电流未知量数。在[2]中给出了一个改进节点法列方程的实例, 其电路图如图 1 所示, 列出的方程是:

$$\begin{bmatrix} G_1 & -G_1 & 0 & 0 & -1 \\ -G_1 & G_1 + G_2 + G_3 & -G_3 & 0 & 0 \\ 0 & -G_3 & G_3 & 1 & 0 \\ 0 & 0 & 1 & -L/\Delta t & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_L \\ I_E \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{L}{\Delta t} I_{LP} \\ E \end{bmatrix} \quad (3)$$

其中  $\Delta t$  是积分步长,  $I_{LP}$  是前一时刻电流。

但是, [2]中的改进节点法存在着三个问题。其一是对一端接地的独立电压源列方程时, 由于该电压源非接地端往往连接有很多个支路。例如 MOS 1024 位动态 RAM 的读放电路中, 有六个这样的电压源, 其中三个非地端连接的支路达 20 条以上。这样, 导纳矩阵相应的有 20 多个非零元。这就破坏了导纳矩阵的稀疏性; 其二是对每个一端接地的独立

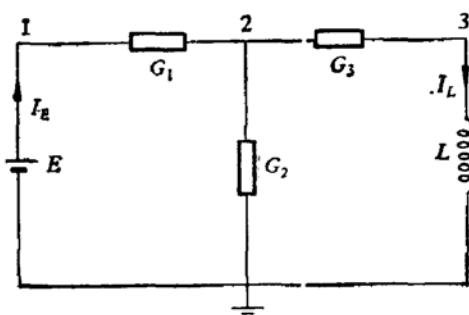


图 1

电压源要各建立一个节点电压方程和支路电流方程，因此一个这样的电压源就要增加两个方程；其三是对每类电路元件既可列节点电压方程，又可列支路电流方程；这样势必要求对这些元件准备两种建立方程的程序模块，从而增加了程序量和编译、列方程时的工作量。

GCAPN 对上述三个问题作了改进，使之更适合于小型机的特点（对大型机也是有益的）。GCAPN 对一端接地的电压源不列方程，而将此类电源非地端连接的各支路上的电导乘以相应电压源值移到右端自由项中。这样不仅对每个这样的电压源可减少两个方程，而且减少了导纳矩阵中的非零元，保持了矩阵的稀疏性。对于要输出支路电流时，考虑到此种情况一般较少，采取了引进零值电压源作为输出支路电流的元件。虽然这样做要增加一个节点，因而要增列一个节点电压方程（如稍加处理也可不列），但却简化了编译和送导纳值的工作，减少了程序量。

如用 GCAPN 的改进节点法，图 1 的电路所列的方程为：

$$\begin{bmatrix} G_1 + G_2 + G_3 & -G_3 & 0 \\ -G_3 & G_3 & 1 \\ 0 & 1 & -L/\Delta t \end{bmatrix} \begin{bmatrix} V_2 \\ V_3 \\ I_L \end{bmatrix} = \begin{bmatrix} G_1 * E \\ 0 \\ -\frac{L}{\Delta t} I_{LP} \end{bmatrix} \quad (4)$$

(4) 比 [2] 中列出的方程组 (3) 少 2 个方程。

利用 GCAPN 列出的方程一般形式为：

$$\begin{bmatrix} Y_{out} & Y_N & Y_{IN} \\ Y_N^T & Y_{ND} & 0 \\ Y_{IN}^T & 0 & Y_{ID} \end{bmatrix} \begin{bmatrix} V_{out} \\ V_{IN} \\ I \end{bmatrix} = \begin{bmatrix} J \\ F \end{bmatrix} \quad (5)$$

其中  $V_{out}$  是对应未知节点电压矢量， $V_{IN}$  是晶体管模型等效电路内部的极电阻所形成的节点（我们称之为内节点）所对应的节点电压矢量， $I$  是未知支路电流矢量。其方程个数为  $N_d + I_d$  个，比 MNA 少  $2E_0$  个（当独立电压源都是一端接地时）。

### 三、数据结构和软件技术

小型机具有价格低、使用简便、易于普及推广等优点。但它内存小、字长短、速度慢。GCAPN 采用了适于小型机的数据结构和软件技术，以克服小型机的困难，提高软件效率。

#### 1. 分类存放元件的数据结构

电路分析程序中主要的原始数据是各类元件的名字、节点号、值或参数。如何在有限的内存中尽可能多地存放元件和以较少时间查找这些元件是数据结构要解决的问题。这关系到能分析的电路规模和解题速度。

GCAPN 中允许 18 种元件，如按每种元件分别存放，则需 18 种存元件名字、节点号、值或参数的数组。而 GCAPN 是用 FORTRAN 写的，公共数组的大小只能预先定义。这样，对一个具体电路，上述数组中有不少（甚至是大部分）是空着不用，造成存储上

的浪费。但此种存贮方法查找速度较快。还有一种存贮方法是将 18 种元件统统集中存放，然后每个(或每种)元件用指针数组指示存放位置。这种方法数组利用效率很高，节省存贮。但由于各种元件所需要的节点数和参数个数都不相同，查找时比较麻烦和费时间。而解题时，特别是瞬态分析时，每迭代一次都需要查找一遍所有元件(可以只查找非线性元件)来建立方程组。一般瞬态的非线性电路分析总要几百次甚至上千次迭代，这样解题时间要大大增加。GCAPN 采用了上述两种方法的折衷：分类元件存放方法。

GCAPN 设置四类数组，按无源元件类(R、C、L 和二极管)、电源元件类、晶体管元件类以及互感和控制源元件类分别存放十八种元件。由于各类元件所要求的节点数、参数个数基本相同，可规定相同维数的数组而不必指示长度。这样不但提高了数组的利用效率，而且查找速度也比第二种快得多。存贮和查找时按图 2 所示树结构关系进行。

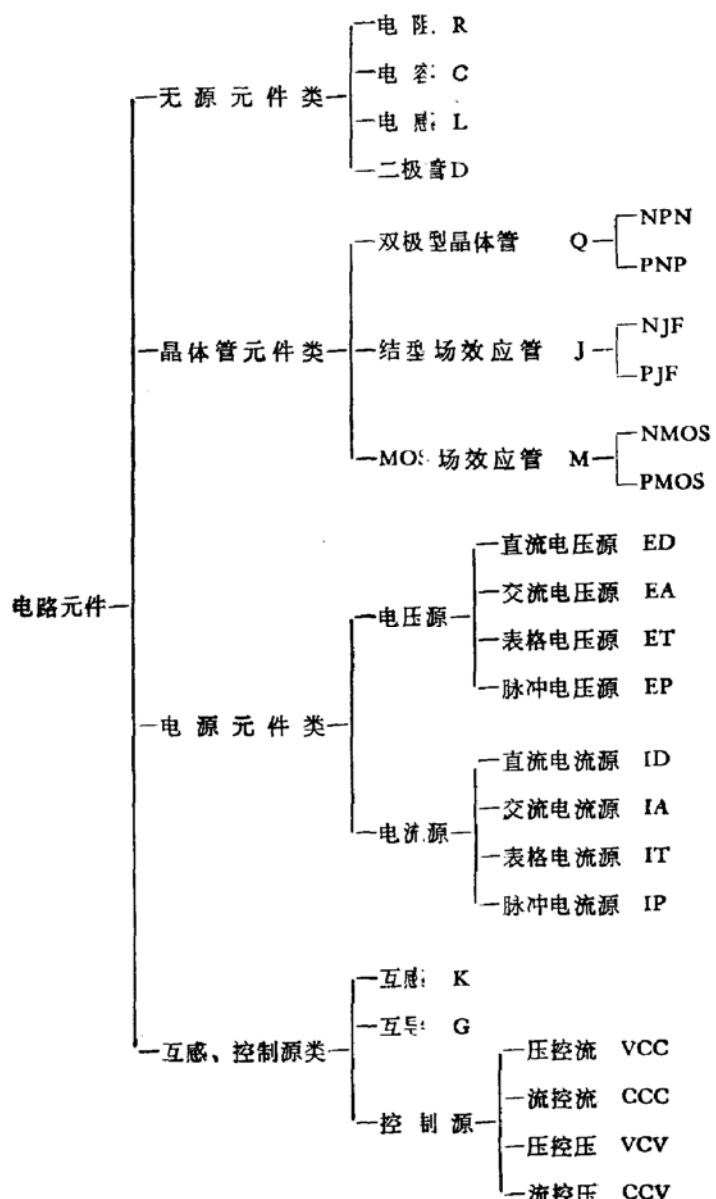


图 2

## 2. 模块结构和多种装配方案

一个通用电路分析程序，它要针对各种元件的电路类型和各种分析要求。因此它的程序量很大，这与小型机内存小是一个尖锐的矛盾。DJS-130 机除了操作系统 RDOS 常驻部分占用外，可供用户使用的内存只有 20 K 字左右。为了解决这个矛盾，我们采用了模块结构和多种装配方案，并在程序中使用了交换链接技术。

考虑到一个具体用户要分析的电路类型大都是单一的，因此我们组装成四个程序方案供不同要求的用户使用。

- 1) GCAPN 1——MOS 电路直流、瞬态分析程序。
- 2) GCAPN 2——双极型电路直流、瞬态分析程序。
- 3) GCAPN 3——双极型电路交流分析程序。
- 4) GCAPN 4——混合类型电路分析程序。

还可以根据需要组装成其它电路类型的分析程序。虽然在磁盘操作系统支持下，程序可以放在磁盘上，然后通过“覆盖”、“交换”、“链接”等手段按要求送到内存运行。但是，过多的与磁盘交换将会大大增加计算时间。电路类型单一的程序存贮量小，与磁盘交换次数少，可以提高计算速度。

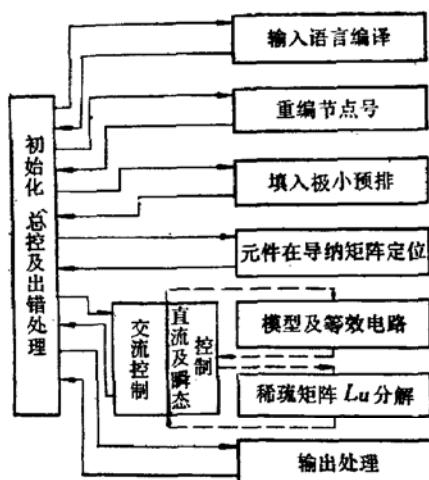


图 3

组装成上述四种程序方案的程序模块有下述十种：

- 1) 输入语言编译。
- 2) 重编节点号。
- 3) 稀疏矩阵填入极小预排列。
- 4) 确定元件在导纳矩阵中位置。
- 5) 模型与等效电路。
- 6) 稀疏矩阵 Lu 分解法。
- 7) 直流及瞬态分析控制。
- 8) 交流分析控制。
- 9) 输出处理。
- 10) 初始化、总控及出错处理。

各模块间的联系如图 3 所示。

图 3 中各框一般作为一个独立程序，实线箭头的连接表示用“程序交换”处理的，虚线箭头表示用子程序调用处理的。

只有 5、6、7、8 四个模块在计算时需反复迭代，其它模块只需进行一次。为了减少与磁盘的交换次数，提高计算速度，最好能将 5、6 和 7 或 5、6 和 8 这几个模块都能放在内存中。对于 20 K 字内存，考虑将上述模块能一起放在内存，所剩的部分可容纳 50 个晶体管元件、30 个无源元件、8 个电源、40 个计算节点这样规模的电路数据。

对用户来说，无论哪种电路类型、其电路源程序都同样按 GCAPN 的输入语言书写，只是在上机时根据需要调用相应程序执行。

## 四、结 束 语

已用 GCAPN 计算了上百个不同类型的电路例子，结果良好，有些已用于实际设计中。平均每只晶体管迭代一次需 60 毫秒。下面表 1 中列出了四个电路的计算情况。

表 1

电 路 名	CMOS 二级反相器	地址缓冲寄存器	环形振荡器	1024 MOS RAM 读放电路
总节点数	9	13	20	28
总元件数	14	22	38	60
晶体管数	8	15	34	48
求解节点数	4	7	17	22
产生填入数	0	2	0	18
迭代次数	165	294	272	520
积分步数	106	71	50	190
分析时间	3(分)	18(分)	34(分)	45(分)

软件研制过程中得到北京工业大学谈根林、本校电工教研室孙树勤等同志的帮助，邮电部 508 厂为我们使用计算机提供了方便，仅向他们表示感谢。

## 参 考 文 献

- [1] 清华大学计算机系 CAD 小组，“GCAPN 输入语言及其编译思想”(工作报告)。
- [2] Ho Chung-Wen, et al., “The Modified Nodal Approach to Network Analysis”, *IEEE Trans. on Circuits and Systems, CAS 22*, 504 (1975).

## A General Purpose Circuit Analysis Program for Minicomputer—GCAPN

Hong Xianlong, Sun Jiaguang, Wu Qiming,

Liu Xiling and Wan Zeyi

(*Qinghua University*)

### Abstract

It is well known that the computer aided circuit analysis technique is a powerful tool for the circuit designers. The circuit analysis program realized on minicomputer is beneficial to its wide spread use. The GCAPN described in this paper is a general purpose circuit analysis program, which is run on a DJS-130 computer (or NOVA computer) with realtime disk operating system RDOS. There is a 64 K bytes memory in this computer.

The GCAPN has a intuitive and obvious input language. The modified nodal approach to establishment of the equation system is adopted in it, and the sparse matrix method and the implicit variable step size integral method for solving the equation system are used. The special data structure and the software technique are adopted for fitting the minicomputer. Besides, there are models of both MOS transistor and bipolar transistor in the GCAPN. The DC, AC and transient analysis can be realized with GCAPN.