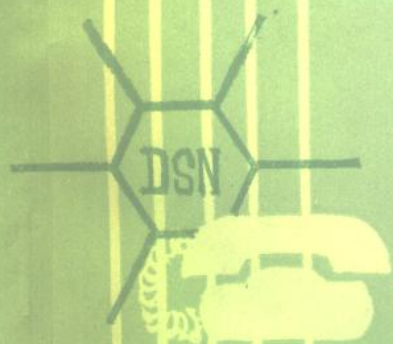


# CHILL 语言

## ——程控交换程序 设计语言

王钟馨 著  
盛友招



12  
8/1

北京邮电学院出版社

TP312  
WZX/1

# CHILL 语 言

——程控交换程序设计语言——

王钟馨 盛友招 编

北京邮电学院出版社

1987年

**CHILL语言**  
——程控交换程序设计语言  
王仲馨 盛友招 编

北京邮电学院出版社出版发行 各地新华书店经售  
北京邮电学院印刷厂印刷

850×1168毫米 1/32 印张 8 字数 212千字  
1987年11月第一版 1987年11月第一次印刷  
印数 1—2500册

统一书号：15523·001 ISBN 7—5635—0000—6/TP  
定价：1.60元

**011302**

## 内 容 简 介

本书是一本主要用于程控交换的工具书。

CHILL 语言是由 CCITT 推荐、主要用于程控交换系统的一种程序设计语言，也是我国在大量引进程控交换机形势下急需掌握的一种高级语言。该语言具有丰富完备的数据模式、清晰明了的通用语句和模块结构，书写格式自由、运行效率高、应用灵活、简明易学等优点，除此之外，它还能使程序设计中大部分差错在编译时就被检测出来。这一特点，保证了程序设计的高效和高可靠性，深受软件研制者的欢迎，目前它已在国际上得到越来越广泛的应用。

本书综合了国外多种文献资料和培训实用教材，并在比利时贝尔电话公司 (BTMC) 大量上机和在国内多次课堂教学实践等基础上编写而成的。书中有较多的例题和习题，内容通俗易懂，深入浅出，可作有关大专院校的教科书，也可供高等院校有关专业师生，以及其它有关部门的科研、工程技术人员学习参考。

## 序 言

《CHILL 语言》是 CCITT 推荐的一种适用于程控交换系统，并能应用于计算机网络通信等系统的高级系统程序设计语言。它具有高效、可靠、灵活、功能强，易于模块化、结构化，简明易学等优点，在国际上已得到了广泛的应用。当前，在一些较先进的数字程控交换系统中，庞大而复杂的程序，百分之八十以上是用 CHILL 语言编写的。

近二、三年来，我国很多省、市大量引进了、并正在继续大量引进数字程控交换机，在此形势下，CHILL 语言已成为我国程控交换工作者必须掌握的一种工具。但由于该语言是近七、八年才发展起来的，我国又起步较晚，至今还未见到一本通俗易懂、深入浅出适合于教学和自学的教材。为了适应当前这种形势的需要，作者编写了这本书。

本书的两位作者都是较早从事程控交换教学和科研的教师，前几年，他们在国外学习程控交换软件，收集和整理了有关 CHILL 语言的多种文献资料。在认真学习和大量上机实践的基础上，写出了初稿。尔后，又通过多次课堂讲授等教学实践，对初稿进行了修改。作为审阅人，我仔细通读全稿，提出了具体的修改意见。本书是在综合资料、上机和教学实践的基础上，经过几个人审阅和多次修改后编写而成的。教材内容由浅入深，概念清楚，层次分明，通俗易懂，大量的程序举例将帮助读者更好地理解 CHILL 语言的程序结构，文中所采用的语法图将帮助读者更形象地、正确地掌握 CHILL 语言的语法规则，这些对初学者来说都是有益的。在教材中，还与 PASCAL 语言作了适量的对比，它有助于熟悉传统语言的读者，更快地掌握 CHILL 语言和进一步了解 CHILL 语言的特点和优点。

我希望本书不仅成为一本适用的大学教科书，并能成为一本面向广大通信科技工作者的自学读物。

沈树雍

1987年3月

# 前 言

CHILL 是 CCITT HIGH LEVEL LANGUAGE 的缩写，即国际电报电话咨询委员会高级语言。它是一种主要用于存贮程序控制交换系统中的程序设计高级语言。存贮程序控制 (STORED PROGRAM CONTROL—SPC) 简称程控。

随着程控交换技术的不断发展和程控交换机数量的日益增多，软件设计在程控交换系统中占着越来越显著的地位。我们知道，程控交换机中的软件系统是十分庞大和十分复杂的，“它是软件建立以来最庞大和最复杂的一种软件，而且将来还会发展”（见参考文献 11），它不仅是决定程控交换系统工作性能的关键要素，而且对数字程控交换（包括多种通信业务）的发展还会起很大的作用。因此，软件系统的质量是评定程控交换系统优劣的决定性依据。

为了提高软件系统的水平，自 1967 年开始，英国、法国、美国、日本、瑞典等国根据 SPC 程序设计经验都研制过适合于本国程控交换系统的高级语言，由于各自承袭自己发展的道路，因此无法统一，这对各系统间的互相联系和软件系统的发展都带来极大的困难。

CCITT 针对此问题着手拟订了一套完整统一的、适用于程控交换系统的高级语言，并于 1980 年正式提出 CHILL 语言建议书——《CHILL 语言定义》，十一月该建议书得到了 CCITT 全体会议的批准，在 1984 年 4 月 CCITT 对它又加以补充和修改。

CHILL 语言吸收了 ALGOL、PASCAL、及 PL/1 等高级语言的优点，并弥补了它们应用于程控交换系统的不足。CHILL 语言虽然主要用于程控电话交换，但对于其它方面也作了足够的考虑，如考虑到应用于报文交换 (MESSAGE SWITCHING) 和分组交换 (PACKET SWITCHING) 等。它具有丰富完备的数据模式，清晰明了的通用语句和模块结构，书写格式自由，运行效率高，功能强，应用

灵活, 简明易学等优点。此外, 它还能使程序设计中的大部分差错在编译阶段就被检测出来, 这一特点保证了程序设计的高效和高可靠性, 这些都深受程控交换软件研制者的欢迎。目前一些较先进的数字程控交换机如: 美国、欧洲 ITT 公司的 SYSTEM-12、日本冲电气、日立、日电、富士通公司的 D70、荷兰 PHILIPS 公司的 PRXD、西德西门子公司的 EWSD 和 HICOM 等, 包括一些较先进的数字程控用户小交换机都采用 CHILL 语言。CHILL 语言自从在 CCITT 全体会议上被确定为程控交换等通信系统的程序设计标准语言以来, 已得到进一步修改和完善, 相信它将会在通信系统中得到越来越广泛的应用。

在我国大量引进程控交换机的形势下, CHILL 语言已成为我国从事程控交换生产、维护、科研工作者急需掌握的一种高级语言。但目前我国还缺少一本通俗易懂、深入浅出、适合教学和自学的这方面的书籍, 本书正是为了适应这种形势和要求而编写的。我们综合了 ITT、BTM 公司的有关教材、资料和手册, 参考了 CCITT 的文献和国内外其它文献资料, 并结合在比利时贝尔电话制造公司的上机实践经验编写了这本书。

编写本书时, 我们把重点放在应用于程控交换的内容上, 并从我国实际情况出发, 根据大多数初学者的特点, 在书中首先简明地介绍了 CHILL 语言的程序结构、数据模式和基本语句, 先给读者一些概貌性的知识, 然后逐步展开, 进行比较详细的讨论。在讨论中, 较多地采用“从具体例题入手, 逐步引出概念, 层次分明地加以阐述, 最后再作归纳”的方法。对一些数据模式、用户自定义模式、各种动作语句、过程和函数的引出、定义、特点和在程序中的应用, 结合举例作了比较确切的说明, 并和 PASCAL 语言作了适当的对比。根据多次上机实践及课堂讲授等教学实践经验, 对一些容易混淆和出错的地方也作了对比说明。书中有较多的例题和习题, 这些都有利于加强和巩固对基本概念的理解。

在本书中, 各种数据模式和各种语句是基础, 过程和函数、模

块以及它们的标识符（名字）的作用域（或称可见性），同义模式定义、新模式定义应该作为重点学习和掌握，而结构模式和引用模式在程控交换软件中用得极为广泛，必须深入理解和掌握。

对于已掌握 PASCAL 语言或其它高级语言的读者来说，若能注意比较 CHILL 语言与这些高级语言的差别，将能较快地掌握 CHILL 语言和进一步了解 CHILL 语言的特点和优点。CHILL 语言与一般高级语言明显的不同点大致有：

1. 它的 CASE 语句可以具有多个选择表达式，从而可以获得多个选择条件下的更多的组合，这大大增加了程序的灵活性。

2. 它提供了只读模式，可用以保护程序中某些特定单元的内容。

3. 它还提供了两种供用户自定义模式的手段——同义模式定义语句和新模式定义语句。前者给名字赋予与定义模式同义的模式，后者给名字赋予与定义模式不同义的模式。合理使用模式定义可大大简化一致性检查，改善程序的可读性，并使程序易写、易修改、少出错。

4. 为了满足程控交换系统的实时性要求，CHILL 语言提供了并发机构。

5. 它还提供了多赋值语句，一条多赋值语句可以对多个单元（对应多个变量）赋予相同的值。

本书共八章，其中第一章、第二章、第六章、第七章由王钟馨编写，第三章至第五章及第八章由盛友招编写。

在编写本书的过程中，得到了沈树雍教授和陈崇昕副教授的热心指导和帮助，他们认真仔细地审阅了全书，提出了很多宝贵的意见和建议；胡文经、陈家珏高级工程师对本书的内容作了仔细的审查，提出不少宝贵的修改意见；我们还得到邮电高校“计算机通信与应用”教材编审委员会文瑜主任的热情鼓励和帮助；此外，在资料等方面，我们得到姚应俊、安远亮、龚双璩、王立言诸位同志的大力支持，在此，我们一并表示衷心的感谢。



由于我们水平有限，又缺乏经验，本书一定有不少缺点和错误，敬请有关专家和广大读者批评指正。

编者

# 目 录

## 前 言

### 第一章 基础知识和数据模式

- §1.1 基础知识..... ( 1 )
- §1.2 数据模式分类..... ( 9 )
- §1.3 基本语句..... (16)
- §1.4 基本运算符..... (20)
- §1.5 集合模式..... (30)
- §1.6 范围模式..... (37)
- §1.7 串模式..... (42)
- §1.8 数组模式..... (56)

### 第二章 控制语句

- §2.1 控制语句的简介..... (67)
- §2.2 IF 语句..... (67)
- §2.3 CASE 语句..... (74)
- §2.4 DO 语句..... (81)
- §2.5 出口语句 (EXIT 语句)..... (99)
- §2.6 GOTO 语句..... (101)

### 第三章 同义词、同义模式、新模式

- §3.1 同义词定义..... (102)
- §3.2 INIT 设施 (初始化模式)..... (103)
- §3.3 READ 设施 (只读模式)..... (103)
- §3.4 用户模式定义..... (104)
- §3.5 明显的模式变换..... (111)

### 第四章 过程和函数

§4.1	概述 .....	(113)
§4.2	过程特性 .....	(113)
§4.3	无参数过程 .....	(113)
§4.4	数值参数 .....	(118)
§4.5	存贮单元参数 .....	(122)
§4.6	函数 .....	(128)
§4.7	标识符的作用域 .....	(136)
§4.8	过程属性 .....	(140)
§4.9	前向引用 .....	(142)
<b>第五章 模块</b>		
§5.1	概述 .....	(143)
§5.2	模块结构 .....	(143)
§5.3	举例 .....	(144)
§5.4	准许语句 .....	(145)
§5.5	占有语句 .....	(146)
<b>第六章 结构模式</b>		
§6.1	结构模式定义及固定结构模式 .....	(151)
§6.2	DO WITH 语句及其在结构模式中的应用 .....	(157)
§6.3	嵌套 .....	(158)
§6.4	程序模块举例 .....	(165)
§6.5	变化结构 变化场 标号场 .....	(166)
§6.6	举例 .....	(169)
§6.7	变化场、无标号场 .....	(173)
§6.8	用同义模式或新模式定义的优点 .....	(174)
<b>第七章 引用模式</b>		
§7.1	引用模式概述 .....	(177)
§7.2	受限引用模式 .....	(177)
§7.3	自由引用模式 .....	(180)
§7.4	程序举例 .....	(181)

§7.5	空指针 .....	(182)
§7.6	反引用 .....	(183)
§7.7	运算符 (包括引用和反引用运算符等) 的优先级 .....	(188)
§7.8	程序举例 .....	(190)
§7.9	引用模式的用途 .....	(194)
<b>第八章 并发执行</b>		
§8.1	CHILL 语言 进程的定义 .....	(195)
§8.2	进程的状态 .....	(197)
§8.3	区域 .....	(199)
§8.4	进程的同步和通信 .....	(203)
<b>附录</b>		
附录 1:	CCITT No.5 字母表/ASCII 字符集 .....	(212)
附录 2:	保留字 .....	(213)
附录 3:	CHILL 语法图.....	(218)
参考文献	.....	(242)

# 第一章 基础知识和数据模式

这一章是为本书提供一些基础知识，并介绍 CHILL 语言的数据模式。其中包括程序设计基础知识、CHILL 语言的程序格式、语法图、基本语句、基本运算符和数据模式。

## §1.1 基础知识

### 1.1.1 程序设计基础知识

程序设计是伴随着电子计算机的出现而产生的一门学科，简单地说，就是人们把需要计算机做的工作写成一种计算机能直接或间接接受的程序。随着程序设计的方式和水平的不断改善和提高，程序所用的语言也从简单到复杂，从低级到高级。在计算机问世初期，用二进制代码表示计算机指令系统，并用它们来编写程序，这就是“机器语言”。由于机器语言使用很不方便，为此人们用一些简单而又形象的符号来代替每一条具体的指令，而这些指令又对应于具体机器的二进制指令码，这就形成了“符号语言”。在此基础上，把一些子程序、存贮器地址等也用符号来表示，这就形成了“汇编语言”。从汇编语言转换成计算机所能理解和执行的机器语言，中间要经过翻译，这个翻译任务由所谓“汇编程序”完成。机器语言和汇编语言都和具体所用的计算机有关，它们是为特定的机器服务的，所以被称为面向机器的语言。

由于不同类型的计算机具有不同的指令系统，从而它们的机器语言和汇编语言也不同，因而这种面向机器的语言，通用性差，使用仍然不方便，为此人们在汇编语言的基础上设计出一种不依赖于任何特定机器的、通用性强的语言，这种语言用一些符号来表示程

序员解题的意图，而又能够通过各类机器的翻译程序即可变换成能在各类机器上运行的机器语言。这种语言就是高级语言。高级语言可以在不同类型的计算机上通用，只是它们的翻译程序各不相同，这就为用户带来很大的方便。目前国内外比较通用的计算机高级语言有几十种，其中最普遍的有 BASIC，FORTRAN，ALGOL，COBOL，PL/1 以及 PASCAL 等。在多数系统中，BASIC 语言的翻译程序是解释程序，BASIC 语言的源程序是通过它的解释程序边解释边执行的。而其它几种语言的翻译程序称为编译程序，这些语言源程序要经过各自的编译程序编译之后才能执行。因为计算机本身是不懂得这些高级语言的，只有通过翻译程序才能把用一种高级语言编写成的源程序转换成能为计算机理解和执行的机器语言。

请注意，虽然不同类型的计算机可以使用同一种高级语言，但由于它们的指令系统不同，因而翻译程序也不同。

随着计算机应用的迅速发展，为了解决各种具体应用问题，人们又研制设计了各种专用高级语言。CHILL 语言，SDL 语言，MML 语言就是 CCITT 为了解决程控交换系统中程序设计、操作维护等问题而研制的专用高级语言系列。其中，SDL (SPECIFICATION AND DESCRIPTION LANGUAGE) 是说明和描述语言，它主要用来说明和描述程控交换系统的功能和规范。MML (MAN-MACHINE LANGUAGE) 是人机语言，是维护人员和输入/输出终端之间通信用的语言。CHILL (CCITT HIGH LEVEL LANGUAGE) 是国际电报电话咨询委员会高级语言，它主要用于程控交换系统中的程序设计。CHILL 语言不仅继承了现有各种高级语言的主要优点，并弥补了现有各种高级语言应用于程控交换系统的不足。它的编译系统较一般高级语言的编译程序要庞大复杂得多，这种编译系统具有较强的差错校验功能，能使程序设计中的大部分差错在编译阶段就能被检测出来。

对于学习 CHILL 语言的大多数读者来说，程序设计基础主要

是指掌握 CHILL 语言的基本概念，会阅读有关程序，会编制有关的应用程序。至于 CHILL 语言的编译系统和管理软、硬件资源的操作系统在本书就不作介绍了。

一般所说的程序设计，大致要经历以下几个步骤：

- (1) 根据任务提出问题；
- (2) 确定方法——构造数学模型，选择运算方法；
- (3) 编制程序——画出框图，编写程序，反复调试；

(4) 在实践中检查程序的结果，如果正确无误，则设计基本完成。否则，要反复修改，直到满足要求为止。对于用 CHILL 语言所编制的程序来说，由于它的编译系统有较强的差错校验功能，故第 4 个步骤的工作量一般较少。

程序员的职责，就是编制高质量的程序。所谓高质量的程序，其基本要求是正确可靠和简单清晰。同时在满足程序设计模块化要求的前提下，尽量做到程序短，占据内存单元少，程序运行时间省，还应做到程序便于修改、调试，通用性好，便于移植等。

### 1.1.2 程序的格式和举例

一个程序语言的基本功能是对数据进行描述和操作。一个程序，从本质上讲，它是对给定数据处理过程的描述。

CHILL 语言是如何描述数据？又是如何对数据进行操作的呢？用 CHILL 语言编写的程序，其一般格式如何？这是用户首先关心的问题。下面先看一个简单的例题，题意是对两个整数变量赋值，并计算两整数之和。程序如下：

行号

```
1  SLMPIE; MODULE;           (程序头)
2  DCL  A,B,K INT;          (数据描述)
```

```

3 MAIN: PROC( );
4   A: = 103;
5   B: = 98;
6   K: = A + B;
7 END MAIN;
8 END SIMPLE;

```

(运算的描述)

(程序结束)

程序设计语言有语法要求，这和英语中每个句子都必须以一个大写字母开始是相似的。

CHILL 语言编写的一段程序由三个基本部分所组成：程序头、程序结束；数据描述；运算描述。下面我们就对这些内容以及其他一些规定结合例题介绍如下：

(1) 程序头和程序结束：

第一行 SIMPLE:MODULE; 是程序头(PROGRAM HEADING)，其中 MODULE (模块) 作为程序模块的标志符，而 SIMPLE 是该程序模块的名字。模块名由用户自己定义，名字可以形象化一些，以增加可读性，也可以随便取。本例用 SIMPLE，表示该程序模块是一个简单的源程序。语法规定，在模块名后总是跟有一个冒号(:)和 MODULE 这个字，MODULE 后的分号(;)是任选的，一般可不写。最后一行 END SIMPLE; 是程序的结束(PROGRAM END)标志，语法规定，以“END”和“模块名”结束程序。

(2) 数据的描述(DESCRIPTION OF DATA):

它是程序的说明部分，用来描述程序运行中所需用到的数据。例中第二行 DCL A,B,K INT; 是一条数据说明语句，它说明变量 A,B,K 是整数模式。DCL 是 DECLARATION (说明) 的缩写。INT 是 INTEGER (整数) 的缩写。整数模式是数据模式的一种模式。语法规要求程序中要有数据描述。除了用数据说明语句描述数据外，还可以用数据定义语句，这在以后再作介绍。

(3) 运算的描述 (DESCRIPTION OF ACTIONS):



它是程序的执行部分，由一系列语句所组成。每一条语句执行一定的动作，完成一定的任务。

第四行 A := 103;

第五行 B := 98;

} 为两条赋值语句

第六行 K := A + B; 为求和执行语句，实际上也是赋值语句。

在程序的运算描述部分，有所谓“过程头”和“过程结束”，第三行 MAIN:PROC( )；是过程头。其中 PROC 是 PROCEDURE（过程，又称子程序）的缩写，它是过程的标志符。MAIN 是过程名。（ ）内无任何文字和数，表示无参数（详见第四章）。

在一个程序模块中，可以有几个过程，各个过程，有各自不同的过程名。

第七行 END MAIN；是过程结束语句。语法要求：

在每一个过程结束时，都要有一个过程结束语句“END 过程名；”。


另外，每条语句末尾都有一个分号“；”。以表示本语句结束。所以分号“；”被叫做语句结束符（STATEMENT TERMINATOR）。

CHILL 格式自由（FREE FORMAT），非常灵活，允许一行写几个语句，也允许一个语句写成几行。（这与 PASCAL 相同）所以这比 BASIC 和 FORTRAN 语言要方便些。

### 1.1.3 语法图(SYNTAX DIAGRAM)

语法图是用来规定程序语法的一种明确而有效的方法。所有的语法图都有一个输入和一个输出，它们被集中编写在附录 3 中。

假如在程序中有一个差错，而你又不能容易地去识别时，就可参考相关的语法图，来加以比较、鉴别。在语法图中，

矩形方框  表示要参照另一个语法图，