

可编程保护测控功能通用性设计与实现^①

余群兵, 陈宏君, 金震, 丁力, 徐舒

(南京南瑞继保电气有限公司, 南京 211102)

摘要: 介绍了保护测控装置中可编程保护测控功能的概念、应用场景及实现要求, 提出适用于不同软硬件平台的可编程功能通用化设计方案。在分析可编程功能与保护测控装置标准或通用功能之间的数据接口及其描述方法的基础上阐述了编程语言的设计以及用户程序转换为中间代码和目标代码的实现方法, 讨论了装置虚拟机通过构造逻辑堆栈高效地执行用户程序的过程, 并提出了配套图形化编程工具软件的实现思路。该方案为保护测控装置提供了灵活的通用化二次可编程接口。

关键词: 可编程保护测控功能; 中间代码; 虚拟机; 逆波兰符号法; 控制方程

Generalized Design and Implementation of the Programmable Function for Protection and Monitoring

YU Qun-Bing, CHEN Hong-Jun, JIN Zhen, DING Li, XU Shu

(NR Electric Co. Ltd., Nanjing 210012, China)

Abstract: This paper introduces the concept, application scenarios and implementation requirements of the programmable function in protection and monitoring device and proposes the design scheme of programmable function applicable to different hardware and software platform. Based on the analysis of data interface and definition between programmable function and standard or universal function, the design of programming language and the realization of compiling user programs to intermediate code and target code are elaborated. Finally, this paper demonstrates the method of executing user programs efficiently by constructing a logical stack in the virtual machine and proposes an approach of supporting software of graphical programming. The design scheme can provide flexible application programming interface for secondary development of protection and monitoring device.

Key words: programmable function for protection and monitoring; intermediate code; virtual machine; reverse polish notation; governing equation

随着微机保护测控装置的广泛使用, 应用场合的复杂多样化对装置的软件灵活性提出了更高的要求, 这就要求新型的保护测控软件能够提供灵活的二次可编程接口, 即可编程保护测控功能, 方便工程人员和用户实现现场的保护测控功能扩展、联闭锁逻辑以及其他特殊需求。国外多数微机保护厂家都实现了一定规模的可编程逻辑^[1], 例如 SEL 的 Slogan、GE 的 FlexLogic、AREVA 的 PSL、ABB 的 FUPLA 等都实现了简单的可编程数字逻辑功能; 近几年, 部分国内保

护测控产品中也已具备较复杂的可编程功能^[2], 但大多实现原理并未将适应现场应用的二次可编程与用于提高产品开发效率、依赖于丰富的硬件资源和强大的系统软件、工具软件的研发级可编程严格区分开来^[3-5]。

PLC 在工业现场应用中, 由于其组态灵活、编程方法易掌握、操作方便, 受到广大技术人员的欢迎和重视。本文在分析和借鉴 PLC 的基本工作原理和编程思想的基础上, 提出了适用于不同软硬件平台的可编程保护测控功能的通用化设计方案, 着重论述其具体

^① 收稿时间:2015-05-22;收到修改稿时间:2015-06-27

实现方法, 并举例介绍应用编程的方法和步骤.

1 可编程保护测控的概念

可编程保护测控功能是指面向用户和工程、可在线编程、在标准或通用的保护测控功能基础之上、基于有限资源和有限访问权限的二次可编程接口, 方便工程人员和用户实现工程现场特殊的保护和测控逻辑, 是对传统保护测控功能的有效补充. 可编程保护测控功能增强了保护测控软件的灵活性, 有利于产品在工业企业、电气化铁路、地铁、新能源等领域及海外市场的应用推广; 另一方面, 电力公司等运行单位从管理角度考虑对保护测控装置功能的标准化、版本严控要求越来越高, 而工程应用时又不可避免地存在差异化需求, 该项功能则能很好地解决这一矛盾.

早期的可编程功能又称作可编程逻辑, 通常仅支持简单的数字逻辑, 如与、或、非、上升/下降沿检测、定时器、计数器等, 为了更灵活地满足应用需求, 应扩展模拟量等信息类型, 支持比较、算术运算、以及较复杂的保护和测控等功能.

应用场景的复杂多变对可编程功能的要求越来越高, 因此可编程功能与标准或通用的保护测控功能之间必须要有非常明确的接口界面, 并且允许使用的资源和访问权限也应有限且可控可见, 确保可编程功能与装置原有标准保护测控功能之间解耦.

2 可编程功能的设计原理

对比保护测控程序和 PLC 循环扫描、I/O 映像、顺序化运算的结构特点, 如图 1 所示, 可以看出保护测控程序的周期处理机制与 PLC 循环扫描的工作原理基本相同^[6,7]. 因此, 可借鉴 PLC 的基本工作原理和编程思想实现保护测控装置中的可编程保护测控功能.

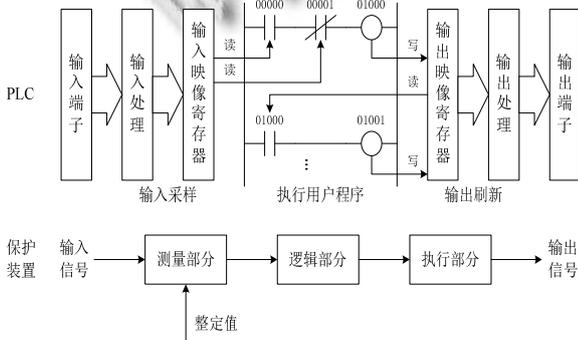


图 1 PLC 及保护装置扫描工作方式

本文设计的可编程保护测控功能主要由上位机编程工具软件、装置虚拟机两部分实现. 其中编程工具软件编辑器提供支持功能块图(Function Block Diagram, FBD)编程语言的用户编程环境, 编程工具软件编译器将用户程序编译生成中间代码, 经调试通讯口可将中间代码文件下载到装置的用户程序存储器中.

装置收到下载的中间代码文件后重启, 由虚拟机初始化程序负责申请动态内存, 并注册实例化功能块, 并设置相关的参数和变量的初始值, 最后将中间代码转换为目标代码存入随机存储器. 装置运行时, 虚拟机解释程序按扫描周期逐行解释执行目标代码, 然后把所得结果保存到输出映像区或暂存区, 直至所有目标代码执行完毕. 最后由装置 I/O 驱动程序将输出映像区的结果输出到指定通道, 完成用户所需的逻辑运算和控制功能^[8]. 可编程系统总体结构如图 2 所示.

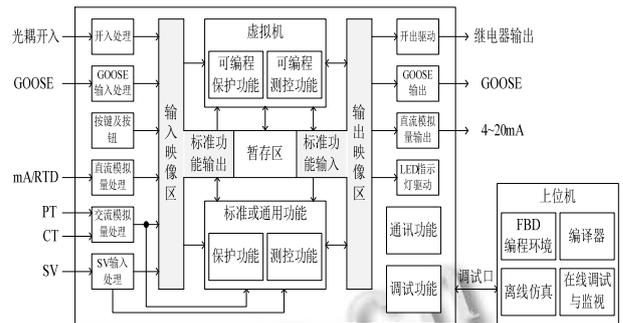


图 2 可编程系统结构图

3 可编程功能的设计与实现

可编程保护测控功能的设计与实现主要包括以中间代码为核心的编程语言设计、目标代码的生成与执行、配套图形化编程工具软件设计等部分内容.

3.1 中间代码语言元素

中间代码语言元素主要包括功能和功能块, 以及反映信号流的操作数^[9].

3.1.1 功能和功能块

显然, 元件化是可在编程功能在保护测控装置中使用的前提和基础^[10]. 关于元件化分类原则和封装方法已有大量研究, 这里不再赘述. 参照 IEC61131-3 和 IEC-61850 标准对常用功能进行元件化封装, 包括标准功能(Function, FUN)和标准功能块(Function Block, FB)以及扩展功能和功能块^[11-13], 在装置中均采用 C 语

言编程实现. 可编程系统主要的功能和功能块如表 1 所示.

表 1 功能和功能块举例

类型	名称	运算符
功能	位逻辑	AND, OR, XOR, NOT
	逻辑运算	ANDB, ORB, XORB
	比较	EQ, NE, GT, LT, GE, LE
	数学运算	ADD, SUB, MUL, DIV
	选择	SEL, MAX, MIN
功能块	触发器	SR, RS
	边沿检测	P, N, PN
	定时器	TON, TOF, TP, TDW
	计数器	CTU, CTD, CTUD
扩展功能块	保护	PTOC, PTOV, PTUV, PTRC 等
	保护相关	RDIR, RVCE, RBRF, RREC 等
	测量	MFREQ, MPHASE 等
	控制	XCBR, SWI, CTRL 等

3.1.2 I/O 映像

保护测控装置以定时采集的方式获取现场输入信号, 在每一次扫描周期的开始, 读取所有输入量的状态或值并存放于随机存储器中的输入映像区, 供执行用户程序时取用. 同样, 用户程序执行的各项结果也并非立即向外输出, 而是暂存于随机存储器中的输出映像区, 在输出刷新阶段根据输出映像区中所有信息更新输出端口的状态. 输入、输出映像区的内容反映了与装置硬件和标准或通用功能相关的各种输入、输出接口资源的配置.

此外, 为了方便存取程序中间变量还增设了暂存区, 与输入、输出映像区统称为 I/O 映像区. 建立 I/O 映像区并统一编址, 使用户程序运行时只与内存单元建立信号的存取关系. 可编程系统 I/O 映像区主要的存储器标识符如表 2 所示.

表 2 I/O 映像区存储器标识符举例

类型	名称	标识符	备注
开关量输入	开入	BI, POS, GOIN	遥信开入、位置开入
	压板	EBI, VBI	硬压板、软压板
	其他	KBI	按键或按钮输入
开关量输出	开出	BO, GOOUT	继电器输出、GOOSE 输出
	指示灯	LED	面板指示灯输出
模拟量输入	电压	U3P, UIP	三相电压、单相电压
	电流	I3P, IIP	三相电流、单相电流
	频率	FRQ	
标准功能输出	保护	OP, ST	保护动作、保护启动
	告警	ALM	装置自检、运行告警

标准功能输入	使能	ENA	可编程功能的输出
	闭锁	BLK	
公用定值	数字型	PA	
	控制字	PB	
暂存区	整型	I, O, VIO	支持位域寻址
	变体	VAR	数据类型自适应

3.1.3 资源描述

结合以上两类语言元素的设计, 可通过符号库编辑工具配置描述装置可编程资源和接口的装置库文件, 供图形化编程使用. 装置库文件主要包括以下内容:

元件库: 功能和功能块的图形化符号描述, 对外接口一般由输入、输出和参数三部分组成, 辅助信息包括执行时间、占用内存大小等.

I/O 库: I/O 映像区存储器标识符定义, 每类存储器的大小和下标范围, 每个存储单元的辅助信息包括数据类型、数据长度、刷新周期、读写权限、描述标签等.

3.2 中间代码语法规则

用户程序编译生成由参数定值列表和类似于指令表(Instruction List, IL)的控制方程(Governing Equation)组成的中间代码.

控制方程是由一条或多条语句组成用来完成一项或多项保护测控功能的语句组, 每条语句由功能或功能块运算符、赋值符和操作数按照一定的运算规则组合而成. 控制方程采用逆波兰符号法(Reverse Polish Notation, RPN)编写, 与指令表相比主要改进内容如下:

1)指令表程序的 LD、OLD、ALD 等指令都对堆栈进行了操作, 逆波兰式控制方程使用了指令表的编程规则但省略了堆栈处理指令, 即内部自动处理逻辑堆栈.

2)由于逻辑操作的操作数个数可能较多, 因此多操作数指令需要指定操作数的个数, 例如, 二输入“位与”写作“AND(2)”.

3)功能块的实例化命名为便于辨识实例号加中括号, 例如第二个实例化的定时器写作“TON[1]”.

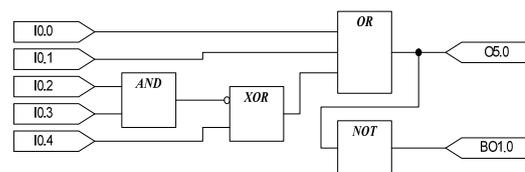


图 3 功能块图编程举例

逆波兰式控制方程书写、阅读非常方便, 例如图3所示的功能块图可写成如下表达式语句

```
O5.0 = IO.0|IO.1|(!IO.2&IO.3)^IO.4
BO1.0 = ! O5.0
```

采用逆波兰符号法表示为

```
O5.0 IO.0 IO.1 IO.2 IO.3 & ! IO.4 ^ || =
BO1.0 O5.0 ! =
```

对应的逆波兰式控制方程如图4所示.

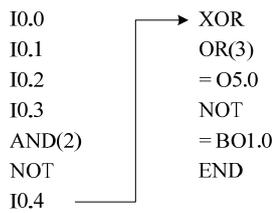


图4 逆波兰式控制方程举例

从上例中可以看出, 逆波兰式控制方程实际上只是将逆波兰符号法表示形式由从左至右改为从上至下, 而仅在赋值符的处理上作了改进, 控制方程的形式像是简化了的指令表程序. 同时, 该改进处理使控制方程更具连贯性, 不必因赋值符的限制强行把完整的功能用分散的控制方程实现.

逆波兰式控制方程编写的主要语法规则如下:

- 1)控制方程除赋值符外均采用逆波兰符号法编写, 作为输入量的操作数必须紧邻运算符且在运算符之前.
- 2)运算符仅支持一个输出缓存, 如果一个运算符的输出用作多个输入, 则运算符输出必须写入输出映像区, 若无指定专用区域则使用暂存区.
- 3)运算符操作的结果写入输出映像区和暂存区必须通过赋值符完成.
- 4)功能块被实例化以后仅能被分配使用一次, 如TON[1]在用作某个功能的延时定时器后不能再作他用.
- 5)对于未指定且允许使用缺省值的输入, 控制方程中用“DFLT”伪操作数标识; 功能运算符输入个数、功能块实例化编号必须在声明范围之内.

3.3 目标代码生成与执行

3.3.1 目标代码生成

目标代码生成由虚拟机初始化程序完成, 装置重启时初始化程序首先分析下载到装置用户存储器中的

中间代码, 检查控制方程每条语句的合法性、操作数地址的有效性以及操作数存取访问权限等; 中间代码检查无误后申请动态内存、注册实例化标准功能块和扩展功能块, 并设置相关的参数定值; 最后将每条语句中的功能或功能块运算符、赋值符和操作数一一转换为解释程序可识别的二进制编码, 以方便装置运行时解释程序高效执行.

3.3.2 目标代码执行

虚拟机解释程序是可编程功能的执行引擎. 解释程序通过构造一个逻辑堆栈存取操作数来实现对控制方程的分析, 为了简化压栈、出栈处理, 逻辑堆栈数据宽度按最大的操作数数据结构长度设计. 由于解释程序在其自己的数据结构中直接维护环境, 因而它的任务就很简单^[14].

逆波兰式控制方程中的各个运算是按顺序执行的, 需从上至下按照控制方程语句的次序依次执行每一条指令. 每遇到一个操作数就将其压栈至栈顶暂存, 每遇到一个功能或功能块运算符则取出栈顶的一个或多个操作数(个数由运算符指定或默认固定)进行运算, 并将运算结果压至栈顶暂存, 然后再继续扫描下一条语句直至结束. 如果运算符为赋值符, 则将栈顶的值赋给指定的操作数, 但不做出栈操作, 直到执行 END指令清堆栈回到控制方程执行前状态.

由于目标代码与中间代码的控制方程语句存在一一对应关系, 因此为了方便分析仍以图4中的中间代码文本形式的控制方程为例, 逻辑堆栈操作过程以及每条语句执行前后栈底(bottom of stack, bos)和栈顶(top of stack, tos)的变化情况如图5所示.

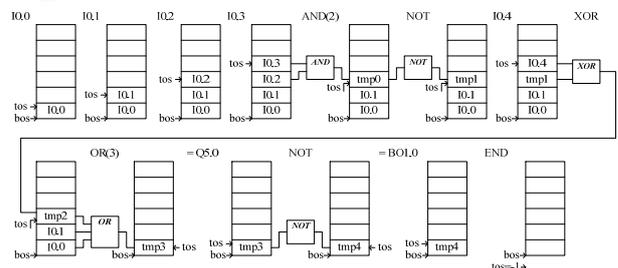


图5 逻辑堆栈操作示例

实际上, 根据应用需求可以设置多个任务等级和执行周期的解释程序任务, 例如可编程保护程序按高优先级(0.833 ms 周期)执行, 可编程测控程序按低优先级(20 ms 周期)执行.

3.4 图形化编程工具

上位机图形化编程工具软件以用户程序和装置软硬件资源接口一体化管理的设计理念,采用模块化可重用的元件库搭建应用功能,实现保护测控功能的灵活配置。

3.4.1 图形化编程工具软件功能和架构

图形化编程工具主要功能: ①采用功能块图图形化编程语言,以图形化模块化的方式搭建应用功能. ②从实例化形成的变量库中挑选信号实现应用功能接口信息配置. ③用户程序编译和管理,按照中间代码语法规则将图形化用户程序编译生成中间代码. ④在线调试功能,包括实时变量访问、事件查看、波形分析、定值整定等。

图形化编程工具软件采用由数据层、逻辑层、表示层 3 层体系构成的分层可重构的组件化架构,提供跨平台、高效率的图形编辑器^[15],如图 6 所示.数据层由层次数据模型、编程符号库、实时变量库组成,是逻辑处理的基础,实现数据信息的建模和管理功能.逻辑处理层是应用功能的实现层,由驱动包管理、配置建模文件形成、脚本解析处理、权限管理、报文处理等模块组成.表示层包括工具的主框架界面、图形系统界面、装置资源配置界面、事件和录波配置界面、定值整定界面、在线调试浏览界面等。



图 6 装置配套工具软件架构

3.4.2 图形化编程配置实现

图形化编程以导入的装置库文件中描述的功能和功能块、I/O 映像区内单元为语言元素,采用“所见即所得”功能块图图形化编程语言,根据现场应用功能需求在可视化页面中挑选并实例化合适的功能和功能块元件,然后进行输入输出信号连线,以模块化图形化的方式快速便捷地搭建应用功能.图 7 显示了从装置库文件的元件库中挑选各功能模块编程配置应用

功能的过程。

完成图形化编程配置后,图形化编程工具软件按照用户程序中的全部功能和功能块连接线的先后关系以及从左到右、从上到下的原则确定功能和功能块的执行顺序,对于交叉点若无指定专用存储区域则使用暂存区且数据类型自适应,并整理相关的参数定值列表,最后编译生成中间代码。

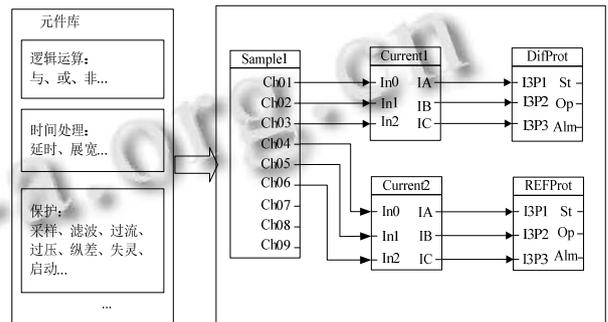


图 7 基于模块化的图形化编程示例

4 编程方法及应用举例

编写功能块图用户程序步骤如下:首先,创建用户程序工程文件,导入装置库文件,创建用户程序页面;然后,根据应用功能的要求从 I/O 库中拖入需要的输入、输出信号,从模块库中拖入需要的功能和功能块,并在输入、输出信号和功能、功能块的输入、输出之间连线,如此反复,直到功能块图实现的保护测控功能满足应用要求;完成功能块图图形编辑后,设置公用定值和功能块参数缺省值,并标记开放用户整定的参数定值,更改其名称描述;最后,编译用户程序生成中间代码文件。

以 10kV 母线增加简易母线保护为例,保护逻辑如下:如果母线上所连接的主变元件过流,而该母线上所连接的其他元件无过流闭锁信号,则快速切除主变低压侧开关.考虑采取在标准主变低压侧后备保护中增加简易母线保护功能、在标准馈线保护中增加闭锁简易母线保护功能的实现方案,后备保护中简易母线保护功能块图图形程序如图 8 所示。

馈线保护中闭锁简易母线保护功能要求如下:速断过流保护、任一段过流保护启动后,瞬时发闭锁信号,用于闭锁简易母线保护;保护动作故障切除后或 200 ms 故障未消失,闭锁信号瞬时返回.功能块图图形程序如图 9 所示。

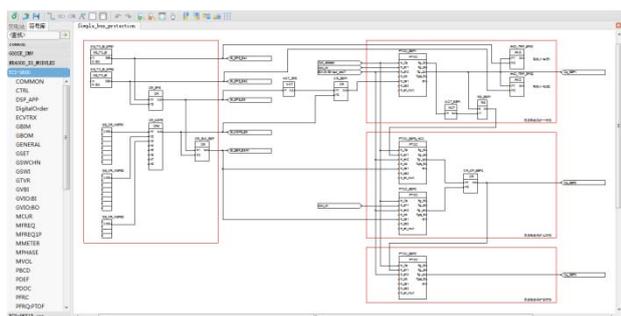


图8 简易母线保护功能逻辑图

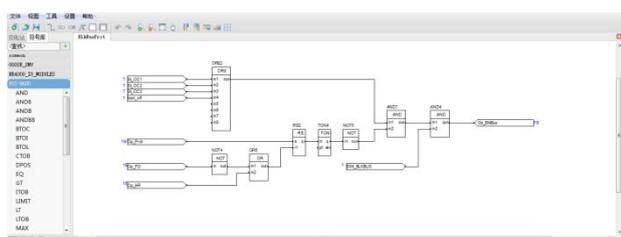


图9 闭锁简易母线保护功能逻辑图

5 结语

本文所提出的可编程保护测控功能设计与实现方案,采用中间代码文件使用户程序与装置标准软、硬件之间解耦,增加了可编程功能的通用性;用户程序编译时转换为逆波兰式控制方程,使程序在运行时按顺序逐条被解释执行,避免了顺序调整和跳转带来的额外开销,大大地提高了程序的执行效率。由于代码转换规则简单,技术起点低,比较容易实现,因此具有很高的实用价值。随着变电站自动化技术的发展,保护测控装置的通讯功能及需求越来越多,实现也越来越复杂,本文下一步的研究方向是编程通讯功能的实现。

参考文献

1 谢志迅,徐礼葆.微机保护装置中 PLC 功能的实现和应用. 电力自动化设备,2007,27(2):121-123.

2 卢娟,李兆成,陈刚杰,等.保护与测控领域通用可视化平台的设计与实现.电力系统自动化,2005,29(4):58-61.

3 李健,来轶,李燕,等.基于模块元件的图形化保护平台设计. 电工电气,2010,7:23-25.

4 魏巍,余群兵,陈刚.图形化控制算法组态在微机保护中的应用.江苏电机工程,2005,24(3):28-30.

5 郭玮,田录林,张永良,等.基于嵌入式 PLC 软核的通用保护平台设计与实现.电力系统保护与控制,2014,42(16):122-126.

6 常斗南,李全利.可编程序控制器原理及应用.北京:电子工业出版社,2006:8-12.

7 贺家李,李永丽,董新洲,等.电力系统继电保护原理.北京:中国电力出版社,2010:3-4.

8 张爱民,蒋刚,张连原,等.软 PLC 的设计思想在继电保护装置中的应用.高压电器,2007,43(6):444-447.

9 GB/T 15969.3-2005 可编程序控制器第 3 部分:编程语言.2005.

10 黄海悦,缪欣,权宪军,等.基于元件化和可编程逻辑构建的继电保护平台.继电器,2006,34(14):11-14.

11 IEC 61131-3 Programmable Controllers, Part 3: Programming Languages.2003.

12 Communication Networks and Systems in Substations - Part 5: Communication Requirements for Functions and Device Models, 2003.

13 Communication Networks and Systems in Substations - Part 7-4: Basic Communication Structure for Substation and Feeder Equipment - Compatible Logical Node Classes and Data Classes, 2003.

14 Kenneth C.Louden,冯博琴.编译原理及实践.北京:机械工业出版社,2004.

15 陈宏君,刘克金,冯亚东,等.新一代保护测控装置配套工具软件设计与应用.电力系统自动化,2013,37(20):92-96.