

多 CPU 内部通信测试技术研究及应用

韩悦, 张 文, 马全霞, 李娟娟, 信亚磊

(许继电气股份有限公司, 河南 许昌 461000)

摘要: 在电力系统保护与控制领域, 多 CPU 协同工作已经是常见的应用方式, 为了保证智能变电站继电保护与控制装置运行的可靠性, 需要对多 CPU 间内部通信管理、数据异常处理等进行充分验证, 以保证电力系统保护装置在各种工况下不会出现误动作; 由于继电保护装置运行环境复杂, 所有内部通信数据均是通过以太网总线直连, 现阶段缺少对内部通信数据处理有效性的验证手段; 采用在各 CPU 应用程序中设计一个专用检测模块, 可以在不改变原有接线布局的情况下, 对 CPU 之间的应用数据进行控制, 从而验证了各种运行环境下多 CPU 板卡间内部通信的正确性, 对智能变电站继电保护与控制装置的各种正常数据和异常数据处理能力进行充分的测试和验证, 保证了保护与控制装置的运行可靠性。

关键词: 多 CPU 通信; 测试技术; 数据处理; 可靠性

Research and Application of Multi-CPU Internal Communication Testing Technology

HAN Yue, ZHANG Wen, MA Quanyia, LI Juanjuan, XIN Yalei

(XJ Electric Co., Ltd., Xuchang 461000, China)

Abstract: In the field of power system protection and control, more CPU work has is a common way of application, in order to guarantee the reliability of the intelligent substation relay protection and control device to run, the need for more internal communication between the CPU data management, exception handling etc. To fully verify, to ensure the power system to the maloperation of the protection device under various conditions will not; Due to the complex operation environment of relay protection device, all the internal communication data processing. Uses in each CPU application design a special detecting module, can under the conditions of without changing the original wiring layout, to control the application of data between the CPU and various operating condition is verified the correctness of the internal communication between multiple CPU boards, the intelligents substation relay protection and control device of all kinds of normal data and abnormal data processing ability adequately testing and verification ensure the reliability of the protection and control device.

Keywords: Multi-CPU communication; Testing techniques; Data processing; Reliability

0 引言

在电力系统变电站自动化技术领域, 多 CPU 协同控制技术已经得到了广泛的应用, 为了保证数据通信的实时性和可靠性, 各 CPU 之间主要通过以太网进行通信^[1-2]。同时以太网通讯技术广泛应用于现代社会生产生活的各个环节, 通过专用网络设备配合交换机^[3-4]、路由器等能够对外部通信接口的各种正常和异常数据进行全面的测试和验证。继电保护装置调试口和应用网口分离, 硬件回路、协议栈在设计上实现分离, 提供给应用的有站控层接口和过程层接口, 满足站控层网络和过程层网络的冗余要求。其中装置调试口既是管理 CPU 的调试口, 也可以通过该调试口实现对各个 CPU 板卡的访问需求, 板卡之间的网卡点对点连接。在各种工程应用中, 由于运行环境复杂、内部通信数据通过总线直连, 对内部数据验收不充分, 实际运行中就会出现各类小概率事件, 无法有效的进行问题分析和排查, 关键的保护装置在内

部数据处理错误时, 可能会导致死机、响应能力变差等功能异常, 给电网的运行安全带来隐患^[5-9]。

对于多 CPU 内部通信测试, 本文针对继电保护装置的内部 CPU 之间的数据通信^[10-12], 开发了一种真实高效的测试方法, 设置专用内部检测模块, 加载在各 CPU 程序运行模块中, 该检测模块能够在不影响各 CPU 其他功能的情况下, 对实际工程应用的各种正常和异常数据处理能力进行充分验证, 不会对正常运行的其他数据造成任何干扰, 可以保证继电保护装置在各种运行环境下数据交互的可靠性^[13-18], 提高设备抗干扰和异常处理性能。

1 检测模块的工作原理及功能

1.1 嵌入式平台架构

智能变电站保护装置通常采用分层、模块化的嵌入式平台架构设计原则^[19-20], 主要分为硬件、软件平台和应用三层。硬件层有处理器、通信接口等通用模块组成, 能根

收稿日期: 2022-02-22; 修回日期: 2022-03-24。

作者简介: 韩悦(1986-), 女, 河南周口人, 大学本科, 工程师, 主要从事智能变电站控制保护产品研发测试技术方向的研究。

马全霞(1979-), 女, 河南新乡人, 大学本科, 高级工程师, 主要从事智能变电站控制保护产品研发测试技术方向的研究。

通讯作者: 张文(1984-), 男, 安徽阜阳人, 大学本科, 高级工程师, 主要从事智能变电站控制保护产品研发测试技术方向的研究。

引用格式: 韩悦, 张文, 马全霞, 等. 多 CPU 内部通信测试技术研究及应用[J]. 计算机测量与控制, 2022, 30(5): 49-54.

据不同需求进行配置；软件平台模块有硬件驱动、嵌入式操作系统、功能块库等组成；其中功能块库为应用程序提供人机接口、通信类等功能块，为开发人员调用；应用层是一个图形化的应用开发环境，提供设置、编译和在线调试等所需功能。

1.2 检测模块的工作原理

结合实际的嵌入式平台架构并针对智能变电站继电保护装置内部通信特点，设计了一个检测模块，嵌入到软件平台内部，通过它可以在不同的互联 CPU 板卡之间实现实时网络报文交换。

检测模块抽象、隐藏了各种板卡以及链路链接端口的多样性，应用程序仅需知道通讯对端的 CPU 板卡 ID，以及建立通讯的报文类型，就可以和目标板卡对应报文类型的应用程序建立点对点通讯，不用关心板子间的链路层、物理层等底层实现以及联网关系。

检测模块通讯服务承载于以太网链路层之上，和 TCP/IP 通讯共享以太网接口，通过使用不同的以太网报文类型来实现数据分流，和 TCP/IP 之间互不影响，没有耦合。TCP/IP 主要是用于板子和外部进行通讯，如和各种后台、各种调试工具之间的通讯等。检测模块是设计在程序内部的私有通讯服务模块，TCP/IP 协议栈各种运行状态，对装置内部基于检测模块之上的通讯不会产生任何影响，如图 1 所示。

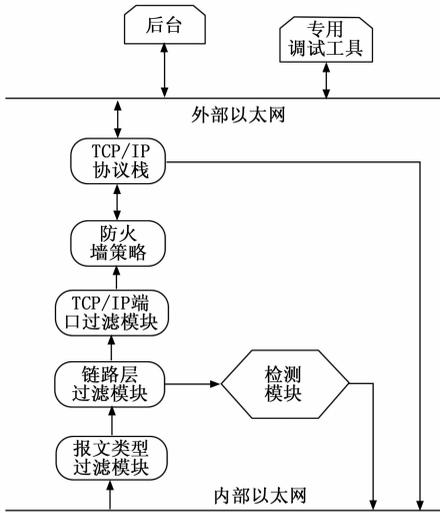


图 1 以太网通讯数据流示意图

在检测模块内部设计一个 ID 表，类似于交换机内部的 MAC 表，用于记录不同 ID 的板卡所在的通讯接口，该表是在系统初始化时创建的，工作机制和交换机的 MAC 表相同。

通讯报文类型为 16 位数，类似于 TCP/IP 中的端口，CPU 之间的应用程序如果需要通讯，需指定相同的报文类型，即两端的程序必须在相同的端口号上收发数据。发送报文可以指定优先级，优先级定义为 0~3，其中优先级 0 最低。优先级高的报文会被优先发送、转发和处理；

优先级高会抢占优先级低的报文的处理流程，从而保证实时性高的报文的通讯需求。

检测模块设置 TTL 参数，根据需求缺省值设置为 4，报文被转发一次，其生成周期会减 1，变为 0 时如果还没有到达目标板卡，报文会被丢弃，从而避免了风暴是回环报文的形成。

检测模块内部传输层报文头记录目标板卡 ID、源板卡 ID、TTL、报文类型、报文优先级等信息。链路层头部记录目的 MAC 地址、源 MAC 地址等信息。检测模块报文结构如图 2 所示。

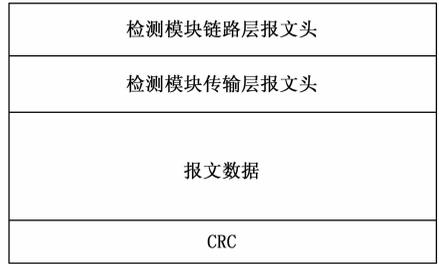


图 2 检测模块报文结构示意图

多 CPU 板卡内部通信均是通过背板直连，CPU 间的内部交互通过专用数据表进行交互^[12]，通过调试工具对测试状态进行控制和设置。设置在 CPU 内部的专用检测模块能够根据不同应用下的数据结构，自动识别内存数据区，从而按照测试需要对各 CPU 的通信交互数据进行修改，不会对正常运行的其他数据造成任何干扰。

CPU 专用检测模块由调试工具进行使能，能够重新初始化各种内部数据以及网络接口，网络连接方式等，每个 CPU 检测模块对收到的数据进行拷贝，然后根据调试工具预设后的数据进行替换，最后传递给调用接口进行发送。如图 3 为一个管理 CPU、两个应用 CPU 的典型应用结构。

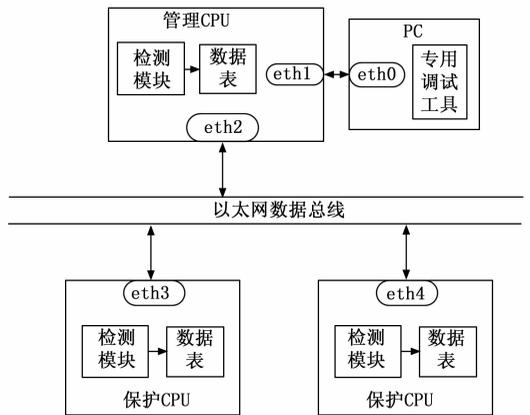


图 3 各 CPU 内部通信图

各 CPU 通过管理 CPU 进行统一控制，通过对外通信接口与 PC 端专用调试工具进行连接，测试时通过专用调试工具对各 CPU 内部通信数据进行修改和验证。专用调试工具通过加载检测模块对应用程序获取内部 CPU 号识别板卡信

息, 根据起始函数和全局数据表识别具体数据字段。每个 CPU 具备唯一的 ID, 每个管理 CPU 最多可以管理 8 个应用 CPU 板。每个应用 CPU 板可以处理 32 种报文类型, 发送报文时可以指定其优先级, 两个板卡之间通过订阅的报文类型和通信端口进行数据收发。

1.3 检测模块接口设计

检测模块由软件平台负责初始化, 相关接口不开放给应用层, 应用层看到的是一个已经可用的通讯系统。该模块提供了两组应用层可用的 API (Application Programming Interface) 接口, 分为基于回调模式的接口和基于句柄模式的接口, 两者接收报文的机制不同。

基于回调模式的 API。当本地收到报文后, 对于已注册处理函数的报文类型, 其处理函数会被直接调用, 这样做的优点就是能以最快的速度完成报文的接收处理, 适合对实时性有很高要求报文处理。

基于句柄模式的 API。该模式的 API 基于回调模式的 API 工作, 每创建一个句柄会添加一个对应类型报文处理函数回调接口, 收到报文时回调接口把报文取出放到句柄对应的报文缓冲区中, 然后应用程序可以从缓冲区把数据取走。

两种通讯模式的选择。基于回调模式处理的报文主要用于实时性能要求比较高的场合。回调函数要求要有尽量少的处理时间, 否则会影响相同优先级后续报文的处理。由于目前采用的 CPU 一般已经比较强劲, 这个过程几乎可以忽略不计, 报文处理的实时性主要取决于调用这些接口的任务优先级以及报文创建时的指定的报文优先级, 基于句柄的报文传输基本上能够满足大部分环境实时报文交互的需求。这种工作模式由于每个创建的句柄都有独立的缓冲区, 所以相同优先级的报文不会相互影响。

1.4 检测模块的功能

检测模块对 CPU 间的应用数据进行控制, 模拟各种应用工况下各 CPU 对数据处理的正确性。正常情况下测试各 CPU 间的心跳报文、突发机制、状态数据和非状态数据等, 异常情况下测试数据风暴、写字段被误改、连接状态不稳定等。在正常测试时, 各 CPU 应能正确处理交互信息并进行存储应用, 在异常测试时, 各 CPU 应对异常数据进行正确识别, 不影响正常通信数据, 同时不出现误发信号或误存储的情况。

2 多 CPU 通信机制及数据交互流程

2.1 通信数据类型及通信方式

为了保证数据处理的快速和高效, 多 CPU 通信数据仅设置了两种类型, 一种为状态量, 一种为非状态量, 两者的通信上送方式不同, 状态量数据需要在变化时突发, 非状态量数据为定时发送。状态量数据变位时连发 3 次, 间隔 2 ms, 正常时的心跳报文间隔为 5 s, 非状态量数据的发送间隔根据实际应用情况进行控制。

双 CPU 交互状态量通过数据表进行交互。在无变化时, 检测模块定时 (5 s) 发送数据把要发送的状态量状态

刷新到数据表, 发送接口检测到数据表变化时突发 3 次数据; 检测模块接收到另一块 CPU 发送的数据后, 把数据刷新到数据表, 后续保护装置的应用在数据表获取数据。

双 CPU 交互非状态量数据功能由检测模块通过两个元件完成, 一个元件为数据发送元件, 一个元件为数据接收元件。数据发送元件定时发送数据给另一块 CPU, 时间间隔通过元件的属性进行设置, 属性为空, 则认为间隔为 0; 数据接收元件连续获取另一块 CPU 的数据, 刷新到数据表。

通信异常判断是通过元件接收数据量进行判别, CPU 在接收元件连续 10 s 未收到对端 CPU 的数据且无心跳报文时, 置通信异常标志, 收到报文后立即返回。

2.2 多 CPU 内部通信的数据处理特征

多 CPU 内部通信的数据处理, 存在两个特征, 一是在交互数据方面, 内部通信数据仅在各板卡间进行交互, 外部不可见; 二是在空间和安全方面, 各 CPU 通过背板直连, 仅对通信数据内存进行安全高效处理, 不会对 CPU 的其他功能和性能造成影响。

根据这两个特征, 专用检测模块通过全局数据结构自动识别通信数据所属内存, 对内存中所有通信数据进行实时控制和修改, 从而按照测试需要对各 CPU 的通信交互数据的每个字段进行测试验证。

在实际应用过程中, 理论上存在装置收到干扰后, 数据被干扰导致误改或数据丢失的情况, 所以需要对内部通信数据的每个字段进行正常和异常测试, 保证在内部通信出现异常时, 不影响保护装置的其他功能。

2.3 内部数据的收发流程

内部数据的收发测试是在发送 CPU 加载检测模块后, 通过内部通信数据的结构体进行内存的自动解析和识别, 确定每个数据字段定位对应的内存地址。

查看当前内存中的数据内容, 同时记录修改前后的 3 组数据进行缓存, 能够完成记录单次内存修改的数据。根据测试数据类型是否为状态量, 状态量需要突发时按照 2 ms 间隔连续发送 3 帧测试数据, 若为非开关量则按照设定的周期要求进行发送。接收 CPU 收到内部数据后, 根据应用程序设定要求进行判别是否为需要的应用数据。当判定接收数据为需要的接收数据时, 开始按照内部协议进行数据解析, 同时将收到的数据刷新到应用数据表, 接收 CPU 总接收数据统计进行累加。如果经过数据判别, 数据类型或格式与需要的数据不一致, 则丢弃该通信数据, 错误统计累加, 同时总接收帧数累加。如果接收到的数据为周期发送数据初步判别后属于需要数据时, 总接收帧数累加, 同时更新接收数据表中的数据。内部数据收发流程图如图 4 所示。

3 多 CPU 通信验证结果及分析

以智能变电站线路保护装置为例, 使用专用调试工具对双 CPU 内部通信状况、通信数据、通信中断闭锁及双 CPU 之间的通信机制进行测试。检查 CPU 之间的通信状况是否正常、通信数据是否正确、CPU 异常导致发送中断后, 是否

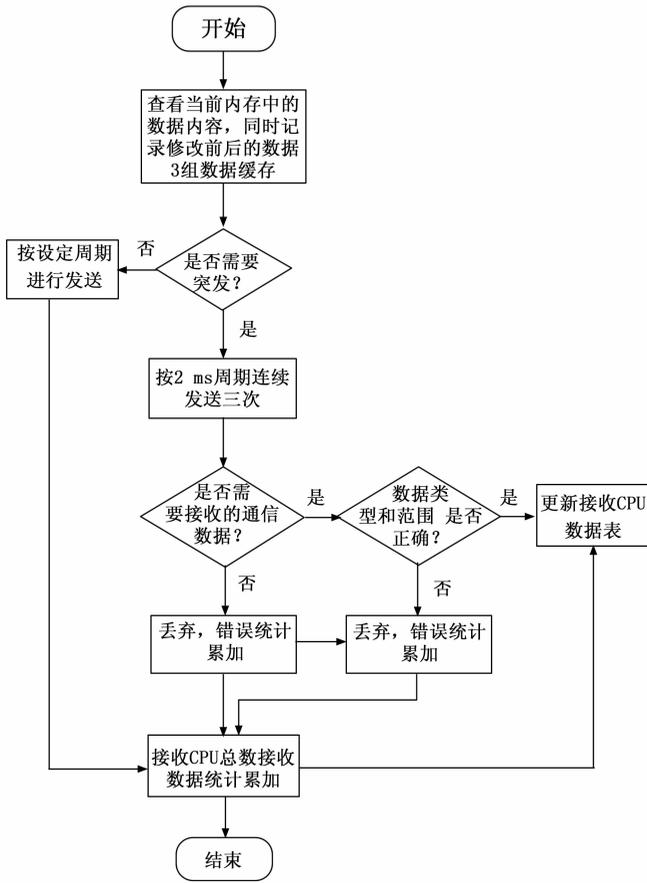


图 4 内部数据收发流程图

会导致保护误动及 CPU 之间的通信机制是否符合设计。

3.1 通信状态测试

检查双 CPU 之间的通信机制是否符合设计及通信状况是否正常。通过调试变量检查发送端发送计数与接收端接收计数是否一致, 采用增量判断, 统计 10 分钟内发送计数的增量与接收计数的增量差不超过 1%。

首先使用专用调试工具同时连接 CPU1 和 CPU2 的平台程序, 输入变量 CExchange, 该变量下面的 RxdMsgCount 为接收报文统计数据, TxdMsgCount 为发送报文统计数据如图 5 所示。

```

CExchange      0x0014DF80      0x0014DF80[...]      struct: CEXC...
  UserExchan... 0x0014DF80      8                      int
  PlatExchan... 0x0014DF84      2                      int
  CpuExchang... 0x0014DF88      0x02618FB0[...]      void *
  CpuExchang... 0x0014DF8C      0x02618FF4[...]      void *
  CpuTempSend... 0x0014DF90      0x02619038[...]      void *
  event_group... 0x0014DF94      0x00187BE8[...]      void *
  events        0x0014DF98      85536                  unsigned int
  TxdResetCount 0x0014DF9C      0                      unsigned int
  RxdResetCount 0x0014DFA0      44                     unsigned int
  RxdMsgCount   0x0014DFA4      10950                  unsigned int
  TxdMsgCount   0x0014DFA8      11183                  unsigned int
  receive_da... 0x0014DFAC      0x0007A7A0[...]      function *
  transmit_d... 0x0014DFB0      0x0007A768[...]      function *
  get_exchan... 0x0014DFB4      0x0007A848[...]      function *
  get_exchan... 0x0014DFB8      0x0007A854[...]      function *
    
```

图 5 CPU 变量收发数据

分别记录 CPU1 和 CPU2 当前的接收报文统计数据 and 发送报文统计数据;

10 分钟后再次统计 CPU1 和 CPU2 的接收报文统计数据 and 发送报文统计数据。

计算 10 分钟内测试通信数据是否符合设计, 查看 10 分钟后 CPU1 的接收报文数据增加是否与 CPU2 在 10 分钟内发送报文增加数据相等, CPU1 在 10 分钟的发送报文是否与 CPU2 在 10 分钟内接收报文的增加数据相等。

将验证数据计入表 1 中, 并对记录数据进行计算, 10 分钟后 CPU1 的接收报文数据增加 (7 575 - 6 975 = 600), 发送报文数据增加 (7 557 - 6 958 = 599)。CPU2 的接收报文数据增加 (11 549 - 10 950 = 599), 发送报文数据增加 (11 783 - 11 183 = 600)。从计算结果可以得出结论: CPU1 在 10 分钟内的接收报文数量等于 CPU2 在 10 分钟内的发送报文数量; CPU1 在 10 分钟内的发送报文数量等于 CPU2 在 10 分钟内接收报文数量, 经过验证可以看出双 CPU 之间的通信机制符合设计要求, 发送端发送计数与接收端接收计数增量差相同及通信状况正常, 满足 10 分钟内发送计数的增量与接收计数的增量差不超过 1% 的要求。若 10 分钟内 CPU1 接收报文数据与发送报文数据与 CPU2 数据增量不同, 则说明 CPU1 和 CPU2 通信状态异常。

表 1 10 分钟内 CPU1 和 CPU2 接收/发送数据

测试步骤	CPU1 接收/发送数据	CPU2 接收/发送数据
当前 CPU 收发报文数据	6 975/6 958	10 950/11 183
10 分钟后 CPU 收发报文数据	7 575/7 557	11 549/117 783
10 分钟内 CPU 收发数据增量	7 575 - 6 975 = 600 7 557 - 6 958 = 599	1 1549 - 10 950 = 599 11 783 - 11 183 = 600
测试结论	10 分钟内 CPU1 接收报文数据为 600 CPU2 发送报文数据为 599	10 分钟内 CPU2 接收报文数据为 599 CPU2 发送报文数据为 600
	10 分钟内 CPU1 接收报文数据等于 CPU2 发送报文数据	10 分钟内 CPU1 发送报文增加数量等于 CPU2 接收报文数据。

3.2 通信中断闭锁测试

检查当单个 CPU 异常导致发送中断后, 是否会导致保护误动, 验证该项目时可通过插拔 CPU 插件模拟单个 CPU 异常, 验证通信中断闭锁正确性。

测试步骤如下: 投入差动保护硬压板, 差动保护软压板, 差动保护控制字, 变化量启动电流定值 0.1 A, 差动动作电流定值 0.2 A, 通道自环。两个 CPU 均正常运行, 施加 A 相电流 0.11 A, 通过查看装置差动保护是否动作, 检查装置是否可正常跳闸出口, 验证通信中断闭锁功能正确性。

测试时, 在保护 CPU1 与启动 CPU2 通信中断前后,

分别模拟启动 CPU2 异常 (如拔掉该插件), 查看装置差动保护动作情况及装置是否跳闸出口; 模拟保护 CPU1 异常, 查看装置差动保护动作情况及装置跳闸出口是否异常。

拔掉 CPU2 插件, 在 CPU1 报出 CPU2 通信中断前, 施加 A 相电流 0.11 A, 测试正确结果应为 CPU1 报差动保护动作, CPU2 应不启动跳闸, 装置不应跳闸出口。

拔掉 CPU2, 在 CPU1 报出 CPU2 通信通断后, 施加 A 相电流 0.11 A, 测试结果应为 CPU1 应报差动保护动作, 测试正确结果应为 CPU2 不启动跳闸, 装置不应跳闸出口。

拔掉 CPU1, 施加 A 相电流 0.11 A, 测试正确结果应为差动保护不动作, 跳闸不出口。

若模拟以上测试项目的测试结果与上述结果不一致, 则说明该装置通信中断闭锁功能有问题。正确测试结果如表 2 所示。

表 2 通信中断闭锁测试

通信中断闭锁	测试结论	
	CPU1 差动保护动作情况	CPU2 启动跳闸情况
拔掉 CPU2 插件, 在 CPU1 报出 CPU2 通信中断前, 施加 A 相电流 0.11A	CPU1 应报差动保护动作	装置不应跳闸出口
拔掉 CPU2 插件, 在 CPU1 报出 CPU2 通信中断后, 施加 A 相电流 0.11A	CPU1 应报差动保护动作	装置不应跳闸出口
拔掉 CPU1 插件, 施加 A 相电流 0.11A	差动保护不动作	跳闸不出口

3.3 通信数据测试

检查 CPU 之间的通信数据是否正确, 通过修改一个 CPU 的数据, 检查另外一个 CPU 对应的数据是否有效。下面以 CPU1 传给 CPU2 数据和 CPU2 传 CPU1 状态为例验证 CPU 之间数据通信正确性。

3.3.1 CPU1 传给 CPU2 数据内容测试

CPU1 传给 CPU2 数据主要用于 CPU 的启动逻辑判断, 具体数据的内容如表 3 所示, CPU1 传给 CPU2 数据测试步骤为:

1) 修改 CPU1 的 CT 一次额定值为 2 500 A, CT 二次额定值为 1 A, PT 一次额定值 220 kV, 施加三相正常电压, 1 A 电流, 查看 CPU2 的采样值, 同时查看 CPU2 的录波文件中的采样数据是否正确。

2) 修改突变量启动定值, 施加电流突变量大于突变量启动定值, 同时施加故障, 查看保护跳闸是否出口。施加电流突变量小于突变量启动定值, 同时施加故障, 查看保护跳闸是否出口。

3) 修改零序启动定值, 施加零序电流大于零序启动定值, 同时施加故障, 查看保护跳闸是否出口。施加零序电流小于零序启动定值, 同时施加故障, 查看保护跳闸是否出口。

4) 修改振荡闭锁过流定值, 施加三相正序电流大于振荡闭锁过流定值, 同时施加故障, 查看保护跳闸是否出口。

施加三相正序电流小于振荡闭锁过流定值, 同时施加故障, 查看保护跳闸是否出口。

5) 修改过电压定值, 施加相电压大于过电压定值, 同时施加故障, 查看保护跳闸是否出口。施加相电压小于过电压定值, 同时施加故障, 查看保护跳闸是否出口。

6) 修改过压保护动作时间定值, 施加相电压大于过电压定值, 时间大于过压保护动作时间定值, 查看装置是否启动; 施加相电压大于过电压定值, 时间小于过压保护动作时间定值, 查看装置是否启动。

7) 投入过压远跳保护, 差动保护有效, 施加远传 1 开入, 查看装置是否收信启动, 远传是否出口。

8) 投入过负荷、过负荷跳闸, 施加相电流大于过负荷定值, 查看装置是否过负荷跳闸; 施加相电流小于过负荷定值, 查看装置是否过负荷跳闸。

9) 投入重合闸, 三相或单相重合, 单相跳位启重合闸, 三相跳位启重合闸, 采用跳位启重合闸, 查看装置重合闸是否出口。

10) 施加一相电流小于 0.04 A, 时间 13 s, 装置报出 CT 断线, 查看装置零序电流是否启动。

由于以上测试项目数据较多, 具体验证结果可通过专用调试工具查看。在此不再详述。

表 3 CPU1 传给 CPU2 数据内容

数据内容	CPU1 数据源位置	CPU2 数据目标位置
CT 一次额定值	CT 一次额定值	CT 一次额定值
CT 二次额定值	CT 二次额定值	CT 二次额定值
PT 变比	PT 变比	PT 变比
变化量启动电流定值	变化量启动电流定值	变化量启动电流定值
零序启动电流定值	零序启动电流定值	零序启动电流定值
振荡闭锁过流定值	振荡闭锁过流定值	振荡闭锁过流定值
过电压定值	过电压定值	过电压定值
过负荷跳闸电流定值	过负荷跳闸电流定值	过负荷跳闸电流定值
相间 III 段阻抗定值	相间 III 段阻抗定值	相间 III 段阻抗定值
正序阻抗角定值	正序阻抗角定值	正序阻抗角定值
负荷电阻定值	负荷电阻定值	负荷电阻定值
补偿系数实部	补偿系数实部	补偿系数实部
补偿系数虚部	补偿系数虚部	补偿系数虚部
过电压动作时间	过电压动作时间	过电压动作时间

3.3.2 CPU2 传给 CPU1 状态数据测试

CPU2 传给 CPU1 的状态数据主要用于触发装置自检, 如 FLASH 自检出错、开出击穿、开入击穿、开出断线、保护程序校验出错、通信中断等, 此类功能对于装置异常重要, 其中部分功能在装置运行中不易出现, 验证该项目时可利用专用装置调试分析工具加载相应平台程序, 通过修改变量, 模拟平台自检位发生改变来验证 CPU2 传给 CPU1 状态数据的正确性以及装置自检功能正确性。模拟装置 FLASH 自检出错、保护程序校验出错等自检功能后, 如图 6 所示, 查看装置是否弹出装置故障报告, 后台是否有装置故障报文, 自检报文, 运行是否灯灭。当装置出现自检出错时, 说

明装置装置已经有故障，装置运行灯应熄灭，相关自检出错报文在装置界面弹出，并将自检出错报文上送后台。

名称	地址	值	类型
Event_1_CB	0x00188268	0x00188268{...}	struct: EV_G...
ev_created	0x00188268	0x00188268{...}	struct: CS_N...
ev_id	0x00188274	0x45564E54	unsigned int
ev_name	0x00188278	0x00188278{...}	unsigned cha...
ev_current...	0x00188280	0x00002010	unsigned int
ev_tasks_w...	0x00188284	0x00000000	unsigned int
ev_suspens...	0x00188288	0x00000000{...}	struct: EV_S...

图 6 CPU2 传给 CPU1 状态数据

4 结束语

多 CPU 协同工作是常见的应用模式，对多 CPU 内部通信管理、数据异常处理需要进行充分验证，才能保证多 CPU 装置或系统的稳定运行。本文设计的在 CPU 内部设置的专用检测模块，不依赖外部设备，不改变原有接线布局，测试方法简单，可以模拟 CPU 间各类数据通信及对异常数据的处理，使用该专用检测模块能够及早发现和解决智能变电站相关产品在可靠性和长时间运行下可能存在的问题，改进了测试方法，提高了测试效率，保证了智能变电站继电保护装置的运行可靠性，在许继电气股份有限公司智能变电站继电保护产品的质量保障过程中发挥了重要作用。

参考文献:

[1] 谢黎, 周华良, 于同伟, 等. 一种智能变电站新型双网冗余设备及实现 [J]. 电力系统保护与控制, 2019, 47 (11): 151-156.

[2] 张颖, 曾翔, 文明, 等. 数字化变电站通信冗余实施方案的改进及可靠性分析 [J]. 电力系统保护与控制, 2012, 40 (5): 124-128.

[3] 冯世林. 智能路由多层交换机在变电站自动化系统中的应用 [J]. 四川电力技术, 2009, 32 (3): 73-76.

[4] 宋一丁, 王振华, 田巧红, 等. 一种以太网重复报文网络风暴处理方法 [J]. 电子设计工程, 2020, 28 (12): 140-143.

[7] 高萌萌, 孙志刚, 李硕, 等. 基于 NB-IoT 的区域空气质量监测系统设计与实现 [J]. 计算机测量与控制, 2020, 28 (10): 55-59.

[8] 乔琳君, 魏严锋. 基于 STC89C52 单片机的自动换气扇系统设计 [J]. 电子设计工程, 2021, 29 (8): 173-176, 181.

[9] 周泽文, 邵婷婷. 基于 STC89C52 单片机的指环脉搏测量仪设计与实现 [J]. 计算机测量与控制, 2021, 29 (2): 246-250.

[10] 程瑶, 田又源, 张果, 等. 基于颜色传感器的彩色铅笔自动分选系统 [J]. 仪表技术与传感器, 2021 (9): 70-73.

[11] 胡中南, 樊蓓蓓. 基于 LabVIEW 的空气质量监测系统设计与实现 [J]. 自动化仪表, 2021, 42 (7): 78-82.

[12] 吴付祥. 基于 STM32 的温湿度监控系统设计 [J]. 自动化与仪器仪表, 2020 (8): 111-113, 118.

[13] 陈星辰, 张丽萍. 基于无线传输的车载温湿度测量系统设计 [J]. 计算机测量与控制, 2017, 25 (5): 42-44, 48.

[5] 胡宝, 张文, 李先彬, 陈强林. 智能变电站嵌入式平台测试系统设计与应用 [J]. 电力系统保护与控制, 2017, 45 (10): 129-133.

[6] 刘雪优. 电力系统自动化技术的应用及发展分析 [J]. 科技创新与应用, 2020, 33: 164-165.

[7] 李喜龙, 黄馨. 浅析电力系统及其自动化的现状与发展 [J]. 电力系统装备, 2020 (3): 51-52.

[8] 王勋. 电力系统自动化技术的应用现状及发展趋势 [J]. 价值工程, 2017 (13): 202-204.

[9] 左群业, 胡宝, 刘天祥, 等. 智能变电站装置网络流量处理模式及其测试研究 [J]. 电气自动化, 2018, 40 (3): 8-10.

[10] 张楠, 黄益庄, 唐晓泉, 等. 多 CPU 结构综合保护装置内部通信的实现 [J]. 电力自动化设备, 2001, 21 (11): 40-43.

[11] 韩春江, 董贝, 唐晓泉, 等. 可视化通用继电保护软件平台设计 [J]. 电力自动化设备, 2011, 12 (下): 207.

[12] 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 [J]. 继电器, 2006, 34 (14): 11-14.

[13] 浮明军, 刘昊昱, 董磊超. 智能变电站继电保护装置自动测试系统研究和应用 [J]. 电力系统保护与控制, 2015, 43 (1): 40-44.

[14] 吴在军, 胡敏强, 杜炎森. 变电站通信网络实时性能仿真分析 [J]. 电力系统自动化, 2005, 29 (8): 45-49.

[15] 徐立子. 变电站自动化系统的可靠性分析 [J]. 电网技术, 2002, 26 (8): 68-72.

[16] 汪兴强, 丁明, 韩平平. 互联电力系统可靠性评估的改进等效模型 [J]. 电工技术学报, 2011, 26 (9): 201-207.

[17] 宋爽, 乔金星, 卜强生, 等. 智能变电站冗余数据处理方案的研究与应用 [J]. 电力工程技术, 2018, 37 (2): 83-88.

[18] 王永刚, 应战煌, 陈玉峰. 基于 ARM-Core4x 嵌入式软件平台的实现方法 [J]. 计算机测量与控制, 2017, 25 (5): 143-146.

[19] 冯立杰, 傅民仓, 李文波. 多 CPU 嵌入式系统的设计方法 [J]. 现代电子技术, 2006, 221 (6): 54-55.

[20] 乔文军, 万晓冬. 嵌入式软件覆盖测试工具的研究 [J]. 计算机测量与控制, 2007, 15 (9): 1238-1240.

[14] 周颖, 张海宁, 刘翔. 四旋翼飞行器的温湿度环境采集系统 [J]. 国外电子测量技术, 2019, 38 (10): 88-92.

[15] 黄克亚. 基于蓝牙技术的时间同步与无线监控系统实验设计 [J]. 实验技术与管理, 2021 (11): 64-69.

[16] 李云强. 基于智能手机的厨房环境监测系统的设计 [J]. 昆明理工大学学报 (自然科学版), 2019, 44 (4): 55-63.

[17] 屈哲, 刘龙, 孙雅欣, 等. 电驱式蒜薹收获机控制系统的设计与仿真 [J]. 农机化研究, 2021, 43 (12): 78-84.

[18] 谢毅. 面向安卓的移动终端操作系统防护技术 [J]. 电子设计工程, 2021, 29 (20): 57-60, 66.

[19] 吴韶波, 王明浩. 消费级物联网中的通信协议与标准 [J]. 物联网技术, 2017, 7 (12): 31-33, 37.

[20] 朱其祥, 徐勇, 蒋宗华, 等. 基于 Proteus 的井字棋游戏设计与仿真 [J]. 辽宁大学学报 (自然科学版), 2021, 48 (2): 121-124.