

基于 NI-VISA 的晶圆测试探针台远程控制软件的设计与实现

杜开元^{1,2}, 袁俊¹, 卢旭坤¹

(1. 广东利扬芯片测试股份有限公司, 广东 东莞 523041;

2. 西安电子科技大学 微电子学院, 西安 710071)

摘要: 在晶圆级芯片测试过程中, 晶圆探针台是测试正确进行的关键实施设备, 探针台的使用与输出的 map 图等数据将直接反应晶圆 (wafer) 测试情况; 目前探针台 (prober) 设备存在型号多、指令类型繁杂和必须现场操作的问题, 增加了测试人员对测试方案开发以及量产测试监测的难度; 为此提出了一种基于 NI-VISA 与网络地址转换 (network address translation, NAT) 内外网穿透的软件设计, 通过将 NI 底层动态链接库嵌入到软件函数中, 并集成为人机交互界面, 实现测站终端与探针台快速连接控制, 并通过快速反向代理 (fast reverse proxy, FRP) 技术实现内外网 NAT 穿透, 实现远程控制监控探针台; 该软件设计在解决芯片测试方案远程调试困难的同时大幅缩短了测试方案开发周期; 在提高了工作效率的基础上, 减少了不必要的人力成本, 有助于晶圆级芯片测试方案开发以及探针台设备监控的工作。

关键词: 晶圆测试; Prober 探针台; 远程控制; 内外网 NAT 穿透

Design and Implementation of Remote Control Software for Wafer-Test Prober Equipment Based on NI-VISA

Du Kaiyuan^{1,2}, Yuan Jun¹, Lu Xukun¹

(1. Guangdong Leadyo IC Testing Co., Ltd., Dongguan 523041, China;

2. School of Microelectronics, Xidian University, Xi'an 710071, China)

Abstract: The wafer prober is the key implementation equipment for correct testing in the wafer testing process. The use of the prober and the output die map and other data will directly reflect the wafer testing conditions; At present, the fact about wafer prober is that there are many types of prober, multifarious commands and necessary to operate in the field, which makes it more difficult for test engineer to develop the test plan and monitor the mass production test; For this reason, a software design based on NI-VISA and NAT-DDNS is proposed in this paper. By embedding the NI bottom dynamic link library into the software function and integrating it into the human-computer interaction interface, the terminal of the measuring station and the prober station can be quickly connected and controlled. Moreover, the internal and external network NAT can be penetrated through the FRP technology to realize the remote control of the monitoring prober station. The software design not only solves the difficulty of remote debugging of the chip test scheme, but also greatly shortens the development cycle of the test scheme. On the basis of improving work efficiency, unnecessary labor cost is reduced, which is helpful for wafer testing scenarios developing and probe equipment monitoring.

Keywords: wafer-test; prober; remote control; NAT-DDNS

0 引言

近年来, 晶圆测试在整个芯片制造流程中有着重要的地位, 测试的成本占芯片制造总成本的 30%, wafer 中性能异常的 die 通过人工测试是难以实现的, 而探针台 (prober) 作为一种在晶圆级芯片测试时必须的探针测试设备, 通过探针测试, 大大减少了测试时间以及后续流程中的成本与加工耗费^[1]。随着仪器自动化技术的发展, wafer 测试设备的自动化程度越来越高, 手动或半自动测试设备慢慢被淘汰, 使用全自动探针台的探针测试成为了晶圆测试的主要

手段^[2]。由于晶圆测试对工作环境和无尘条件的苛刻要求以及主流 Prober 探针台只能通过自带触摸显示屏进行操作的限制, 使得芯片测试研发工程师对测试方案进行调试的过程变得繁琐, 并且耗费了大量时间在不必要的环节。

对 GPIB 通讯协议进行应用编程时需要使用美国国家仪器 NI 公司所开发的 NI-VISA 接口。该接口适用于常用的多种仪器总线通信, 并且自身有独特的软件架构, 保证在驱动程序时不受设备平台、通信总线和外部环境的限制。使用 NI-VISA 进行仪器控制的应用开发可以通过一模一

收稿日期: 2020-05-06; 修回日期: 2020-06-16。

作者简介: 杜开元 (1996-) 男, 云南大理人, 硕士, 主要从事集成电路设计方向的研究。

引用格式: 杜开元, 袁俊, 卢旭坤. 基于 NI-VISA 的晶圆测试探针台远程控制软件的设计与实现[J]. 计算机测量与控制, 2021, 29(1):

样的方法与仪器通信，而不用考虑接口具体是哪一种类型，大大简化仪器的控制应用^[3]。

Python 编程语言有着丰富的库以及 GUI 开发环境，Pyvisa 库搭配 NI 公司的 gpib-32.dll 动态链接时库文件作为开发底层文件，再加上 PyQt5 作为 GUI 开发工具进行人机交互界面开发，可以实现实时控制测试设备仪器的功能^[4]。

本文通过 python、NI-VISA 以及 GPIB 提出探针台无法进行远程化、自动化操作现状的解决方案，以本地局域网内计算机测试终端为核心，GPIB 接口作为端口，python 编程实现友好的程控人机交互界面，NAT 内网穿透技术作为远程控制桥梁，实现选择仪器、探针台的 map 图显示、仪器控制以及仪器状态监控等处理功能，控制探针台 Prober 进行晶圆探针测试，实现研发中心对车间探针台仪器的集中统一管理。

1 系统硬件结构设计

1.1 系统硬件结构

系统测试站部分由探针台 prober、GPIB-USB 转换接口、测试站终端、路由器组成，探针台与测试站终端之间需要安装连接 GPIB-USB 转换接口来实现设备与终端的 GPIB 通信（如图 1），由于探针台本身与测试站终端的 GPIB 连接还承担着与 ATE 通讯的任务，所以在软件进行连接的时候会先行释放 ATE 与探针台之间的 GPIB 连接句柄，以免造成 GPIB 总线忙的状态。

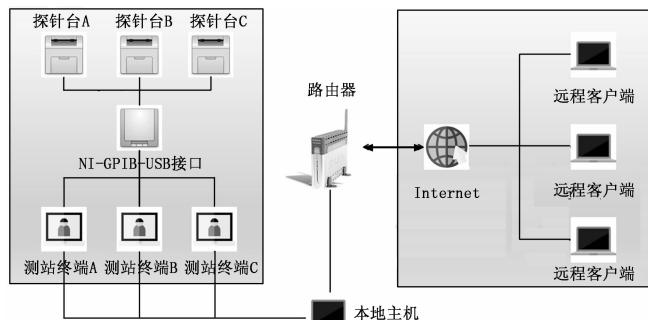


图 1 系统硬件结构图

系统远程控制部分由远程控制服务器及远程客户端组成，使用 frp 内网穿透的技术，通过在远程客户端进行访问 IP 和密钥等设置，可以穿透连接控制内网环境中的测试站终端^[5]。

1.2 与探针台连接

NI GPIB-USB-HS 是 NI 公司开发的一款 IEEE 488 控制器和便携式分析仪，用于采用 USB 插槽的计算机。使用该设备可将仪器集成到系统中，以及排除和解决 GPIB 硬件和软件问题，由于不需要 GPIB 电缆，因此可使用高速 USB 端口控制最多 14 个 GPIB 仪器。

GPIB 连接线可以连接测试站终端与探针台 Prober，适用于从 Lab 到工厂车间等不同环境内的仪器控制应用，多个 GPIB 连接线可以连接到同一测试终端，通过 GPIB 连接

模块的选择来进行选择性控制与监测。

2 系统软件设计

2.1 功能简介

软件设计分为测试站终端与客户端两部分。测试站终端部分通过 GUI 人机交互界面对 GPIB 总线上连接的设备进行连接、监控、控制及测试结果显示。测试站终端模块中的 GPIB 控制子模块主要是实现测试站终端与当前总线上 prober 设备的连接、控制权限句柄的获取。测试进程获取子模块主要用于实时 map 图的获取以及测试结果的获取。仪器状态监控子模块主要用于实时监控探针台状态与异常，便于及时处理。

远程客户端部分能够使用 frp 内网穿透技术通过内网中连接了外网的主机作为中转，将测试站终端的端口连接到客户电脑，再使用 Windows 自带 MSTAC 远程桌面连接进行访问，就实现了远程操控未连接外网的电脑并控制本地探针台的功能。

2.2 测试站模块设计

如图 2 所示，客户端人机交互界面能够达到和实际操作探针台一样的交互体验。

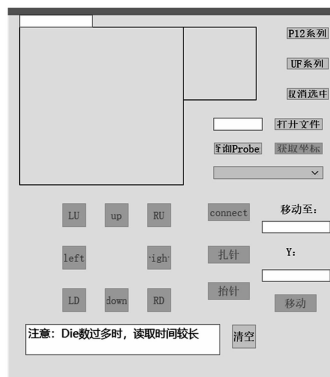


图 2 测试站模块界面

显示模块是通过 python 的 pillow 库显示 Map 图信息、BIN 项信息、坐标信息、测试结果以及状态监控显示。

仪器查询连接模块，能够选择不同公司不同型号探针台，并通过 GPIB 连接线进行测试站终端与 prober 的连接。

仪器参数设施模块通过界面修改 OD 等探针台测试中的重要参数，设置 prober 参数。

仪器交互控制模块，通过人机交互界面的按钮，对探针台进行移动、扎针以及拾针等操作。

2.3 远程客户端模块设计

使用 frp 内网穿透的技术和 Python 编写远程客户端 GUI 模块，将 frp 的程序和 config 设置嵌入到远程客户端中，将内网穿透需要的参数输入框放置在远程客户端 GUI 界面，通过在远程客户端进行访问 IP 和密钥等设置，可以通过公网端口穿透内网并与内网环境中的测试站终端 PC 端口连接。

2.4 软件设计流程图

软件基于 Python 编程语言来进行设计，启动程序后通



图 3 远程客户端模块界面

过选择不同设备类型来选择设备指令集, 该选择将决定后续收发指令阶段的底层内容, 通过 NI-VISA 动态链接库底层文件中的查询函数查询当前 GPIB 总线上连接的设备地址, 选取 prober 设备地址后发送 ibdev 命令获取 handler, prober 进入等待接收命令阶段。读取待测 wafer 的信息文件来提取 map 图坐标系以及 wafer 基本信息, 获取坐标系后自动发送查询指令并接受坐标信息, 将坐标值返回到基于 pillow 库的 cavans 模块进行分析作图, 作图完成后对 GUI 显示框刷新并更新当前 map 图, 完成刷新后返回指令发送准备状态等待新的操作。具体软件设计流程图如图 4。

2.5 远程控制流程

远程控制流程由客户端主机组织, 配置完成并完成点对点内网穿透连接后, 由 SSH 技术实现本地主机端口与测站主机端口的互连, 客户端主机借用本地主机端口向测站探针台发送指令进行测试, 并使用同样的端口接受并返回测试与控制的结果, 具体流程如图 5 所示。

3 系统关键技术实现

3.1 Python 编程实现仪器连接与控制

使用 NI 公司开发的 GPIB 连接线, 能够满足测站终端与多台设备连接并进行控制的要求, 同时将 NI 驱动文件中的 gpib32.dll 动态链接库与 Python 自带的 Py-Visa 库结合并作为主程序中 GPIB 通讯协议的底层文件, 再使用 Python 编程构造 GUI 人机交互界面, 将设备的程序控制模块与 GUI 的模块进行连接。

使用 Python 在 Pycharm 开发环境下与探针台控制流程如下。

1) 调用动态链接库与 Py-Visa 库;

编写底层文件, 通过 import 方法将两个库引用并封装到函数中。

2) 查询并选择仪器对象:

rm = pyvisa.ResourceManager (); //查询并返回当前 GPIB 连接的仪器信息

Device = self.rm.list_resources (); //查询到的仪器存入 list

3) 获得仪器连接地址:

GPIB:: XX:: INSTR 是被连接仪器的代号, 其中

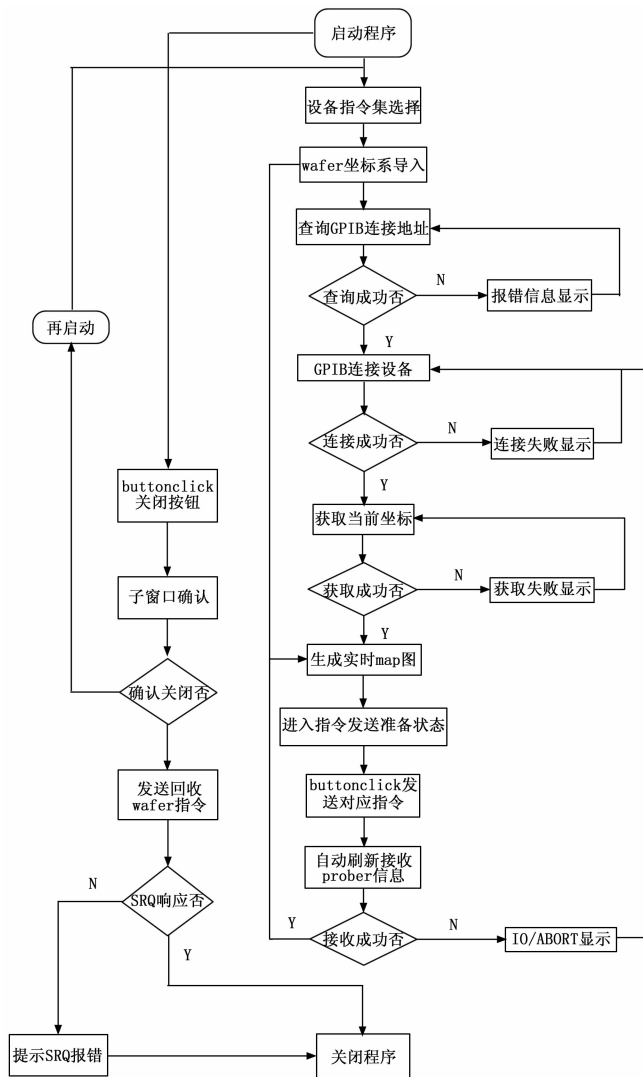


图 4 远程客户端模块界面

“XX”就是 GPIB 通讯中的 address, 所以只要使用 “address = int (machine. split ('::') [1])” 代码获得 address。

4) 仪器连接:

将获得的 address 传入封装好的 connect 函数中:

Gpib.Gpib (0, self. address, 0, timeout=12, send_eoi=1, eos_mode=0);

检查返回值, 如果正确则完成仪器的连接。

Connect 函数具体实现代码如下:

```

if isinstance(name, str):
    self.id = gpib.find(name)
    self._own = True
else:
    self.id = gpib.dev(name, pad, sad, timeout, send_eoi, eos_mode)
    self._own = True

```

其中 gpib.find 与 gpib.dev 就是将 gpib32 动态链接库与 Py-visa 对应功能模块封装起来的函数, 例如:

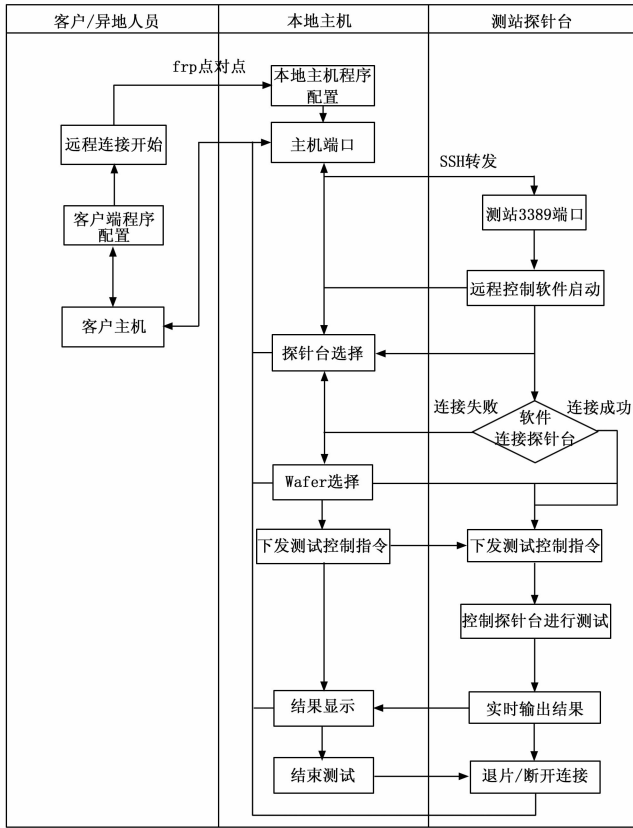


图 5 远程控制流程图

```
def dev(board, pad, sad=NO_SAD, tmo=T30s, sendei=1, eos=0):
```

```
ud = _lib.ibdev(board, pad, sad, tmo, sendei, eos)
```

```
if ud < 0:
```

```
raise GpibError("dev")
```

```
return ud
```

5) 仪器控制:

仪器控制通过 GPIB 通讯协议结合仪器控制标准指令来实现, 使用底层函数综合出的 write 与 read 函数, 配合不同类型 prober 的标准 CMD 即可实现仪器控制, 例如 TEL 公司 UF 系列 prober 的 chunk 移动代码如下:

```
up_d = b'Y+01\r\n'
```

```
self.my_instrument.write(up_d)
```

```
self.my_instrument.write(self.A_b)
```

```
try:
```

```
cCoor = self.my_instrument.read(1000)
```

```
except GpibError as e:
```

```
self.gpib_read_error()
```

6) 接收仪器信息并显示:

Prober 设备在 GPIB 控制过程中接收到指令后会根据指令进行相应动作, 在操作结束后返回操控者期望的信息, 同时根据返回信息中所包含的坐标和 BIN 项等信息, 并在 Pillow 作图模块中刷新当前 Map 图, 然后将最新 Map 图更新至显示模块。由于优化了 Pillow 作图模块的代码, 使得在更新 Map 图时只针对发生改变的 Die 进行改动, 而不是

重新读取 Map 图信息, 大大的加快了显示模块更新的速度。

7) 仪器状态状态监控:

为了使测站终端能够监控 prober 设备状态, 同时 prober 能向终端返回异常报警、发送准备完成等通知, 为此 GPIB 通讯协议设置了 SRQ (服务请求) 功能。当已连接的设备向测站终端发出 SRQ 请求, 那么 GPIB 总线上 SRQ 线为 "TURE", 通知测站终端有连接在总线上的设备发出了服务请求。但是仅凭总线上的 SRQ 信号无法确定发出请求设备的地址, 于是需要测站终端进行轮询处理, 查询发出请求的地址并处理请求内容。

在程序中状态监控分为两部分, 使用多线程的工作模式, 其中监控模块通过 while 循环定时在总线上一台一台地查询当前 SRQ 状态, 被查询的设备向测站终端反馈 9 bit 的数据, 其中包含最高位的状态字节以及余下八位 SRQ 请求数据。测站终端通过分析返回的数据, 监控哪一台设备发出了 SRQ, 提出何种请求。而指令反馈模块则是将 SRQ 请求查询设置在向设备发送指令之后, 这样既有准确的设备地址和指令内容, 又可以快速监控设备对指令的反应情况。状态监控模块的内容都将实时显示在 GUI 界面中的显示模块, 便于使用者查看。具体 SRQ 请求如表 1。

表 1 P12 机台 SRQ 请求表

SRQ	Contents	Related command
41(H)	X Index Complete	I, X, X/Y
42(H)	Y Index Complete	J, Y, X/Y
43(H)	Z UP Complete	Z, Z+
44(H)	Z DOWN Complete	D
45(H)	Marking Complete	R, R?
46(H)	Initial Die, Reference Die	B, U, u
47(H)	Index Transfer Complete	b, M, Q, r
48(H)	Lot End	(U, u)
49(H)	Test Preparation Complete	T
4A(H)	Initial Wafer	
4B(H)	Wafer End	B, b, I, J, M, Q, r, X, Y
4C(H)	Alignment Error	
4D(H)	Continuous Fail Error	
4E(H)	Prober Assist	
4F(H)	Outside Probe Area(Wafer End)	b, I, J, X, Y, X/Y
50(H)	Prober Error	
51(H)	Test Preparation Not Complete	T
52(H)	Parameter Receive Preparation Complete	H, i, N
53(H)	Z Limit	Z, Z+
59(H)	Process Complete	C, c, e, H, h, i, j, L, m, N, h
60(H)	Waiting I Command	I
62(H)	Stop Status	
65(H)	Waiting I Command	I
68(H)	Manual Test	t
FE(H)	P arameter Error	
FF(H)	Command Error	

3.2 实现人机交互界面的设计

远程控制人机交互界面 GUI 的设计通过使用 Python 提供的 QtDesigner 集成式套件。设计步骤如下:

在工具界面放置一个主窗口 Mainwindow 控件作为 GUI 的框架。

根据提前设计规划好的功能模块, 通过 layout 布局合理的放置 Frame 作为模块框架。

在相应的功能模块 Frame 中加入 ListWidget、pushbutton 和 combox 等控件。

通过信号槽的方式将 button_click 等事件与程序中的功能函数连接起来。

完成 GUI 的设计后会生成 ui 文件, 通过配置 external tools 中的 pyuic 工具对 ui 文件进行转换生成 py 文件, py 文件中的 Mainwindow 就是 GUI 主界面, 然后在软件设计主程序中 import 主界面 py 文件, 就可以将主界面中的各个模块与主程序中的程序模块通过信号槽连接, 实现人机交互界面的设计。

3.3 内外网 NAT 穿透的实现

从 20 世纪 90 年代开始, 一种被称作网络地址转换 (NAT) 的技术诞生, 也叫做网络地址伪装或者 IP 地址伪装 (IP masquerading), 这种技术广泛应用在解决由 IPv4 地址不足引起的保存 IP 地址困难问题。解决的方法是在 IP 数据包通过路由器或防火墙时重新编写发出数据包的 IP 地址或被访问的 IP 地址。这种技术的最佳使用场景是在内网有多台主机但只有一个公有 IP 地址访问外网的情况。

NAT 穿透, 就是要在两个处于不同类型 NAT 网络下的节点之间建立起稳定的直接连接, 要实现这种直接连接, 必须的条件是互连双方互知道对方的公网 IP 与端口, 这样才能互相发送数据包来建立连接^[7]。通过分析可知, 本软件所在的公司网络为圆锥形地址限制 NAT (Address-Restricted cone NAT), 也就是内网各主机的客户端必须首先发送数据包到对方 IP 地址, 然后才能接收来自对方 IP 地址的数据包。在内外网访问方面, 唯一的限制是要求数据包是来自对方 IP 地址。内部地址和端口映射到外部地址和端口, 所有发自内网主机的数据包都经由外部地址和端口向外发送。外网主机地址 (不限制端口) 能通过给外部地址和端口发包到达内网。具体内外网访问类型如图 6 所示。

要想实现圆锥形地址受限 NAT 的穿透, 需要能轻松获得对方公网 IP 与端口, 为了实现这种功能, 选用 FRP 来安全地向对方暴露自身 IP 地址与端口。FRP (fast reverse proxy) 是一个高效安全的可用于内外网 NAT 穿透的反向代理 (reverse proxy) 应用, 支持 tcp, udp 协议, 为 http 和 https 应用协议提供了额外的能力, 且尝试性支持了点对点穿透。对于公司来说, 某些服务如果直接暴露在公网上将会存在一定的安全隐患, 所以需要安全地暴露内网服务。使用 xtcp 类型的代理可以避免让任何人都能访问要穿透的

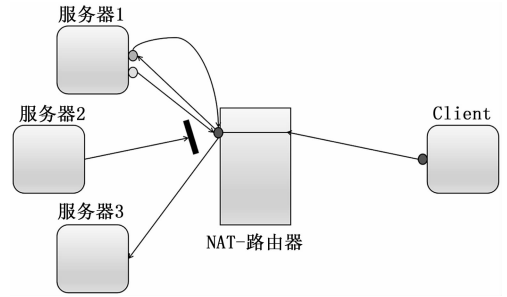


图 6 圆锥形地址限制 NAT

服务, 所以需要在客户端访问者也运行一个 frp client。

Frp 使用方法为在客户端与本地端进行 config 的配置, 并且搭建一个辅助服务器来使客户端与本地端互相交换 config 中配置的 IP 和端口, 服务器有着固定的公网 IP 与端口, 所有本地端与客户端都能连接到辅助服务器, 服务器就获得了所有本地端与客户端的 IP 和端口。客户端只要请求想穿透的本地端, 服务器验证双方 config 配置中用户名和密钥然后就可以返回目标本地端的外网 IP 和端口, 同时通知目标本地端即将进行 FRP 连接并发送客户端的外网 IP 与端口, 再运行 frpc 与 frps 时能够通过本地主机转发内网测站终端的 SSH 服务, 客户端就可以通过在两边都部署上 frpc 建立起客户端与本地主机之间直接连接, 就实现了穿透 NAT 设备的需求, 具体结构如图 7 所示。

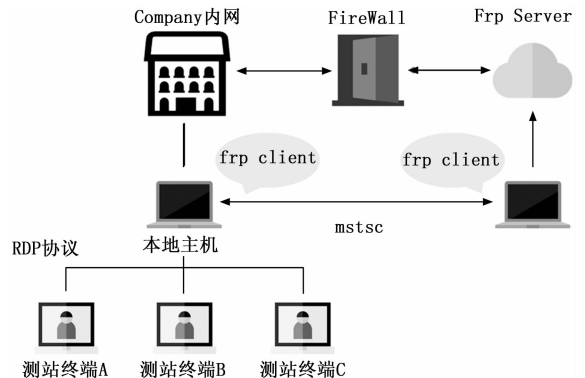


图 7 Frp 穿透模式

上述应用情景是基于穿透双方的 NAT 类型一致时, 服务器仅作为辅助来实现数据中转。但遇到 P2P 穿透的应用情景时, 往往双方 NAT 类型并不相同, 这个时候客户端到服务器与客户端到本地端的 IP 和端口都是不同的, 但本地端的外网 IP 与端口却没有改变, 所以客户端到本地端的连接是可以实现的, 只要通过 UDP 通讯中的 recvfrom address 结构体, 就能得到客户端的外网 IP 和端口, 即可实现 P2P 穿透的应用。

4 系统应用效果

该软件针对晶圆测试探针台远程化自动化操作需求开
(下转第 199 页)