内容 – 第七章"遥控和编程范例"

7 遥控 – 编程范例

下列编程范例有一个等级体系结构,也就是说,后边的范例是建立在前边例子的基础之上的.这样,就可以用已给出范例的模块很容易地编制出操作程序.

IEC/IEEE总线编程的基本步骤

这些范例可以解释如何进行仪器编程,也可用作解决更复杂编程任务的基础.
VISUAL BASIC被用来做编程语言.然而,程序也可翻译为其它语言.

引入VisualBasic的IEC总线库

编程提示:

. **用"Print" 函数输出文本**

下列编程范例是建立在所有子程序都是窗体(文件扩展名:.FRM)部分的假设之上的.在这种情况下,语法

Print "Text"

是容许的.

然而,如果子程序被存为一个所谓的模块(文件扩展名:.BAS),则拥有所谓打印方法的窗体的名字应该在打印指令之前,例如,如果有一个名为"Main"的窗体,相关的打印指令应该象下面这样:

Main.Print "Text".

. **如何使用GPIB.DLL的函数**

要创建Visual Basic的控制应用程序,文件GPIB.BAS(出自VB 6.0 VBIB-32.BAS)被加入到项目中.这个文件包括关于处理错误，超时值的常量和定义.

. **将DLL函数声明为过程**

因为程序都返回整型值,文件GPIB.BAS中的函数都象如下格式声明为:

Declare Function xxx Lib "gpib.dll" (...) As Integer

当被访问/调用时,带有状态变量ibsta的函数值应该被分配一个变量.因为这个值也是通过函数的一个关联参数而被返回的,函数可以象下面这样声明为一个过程:

Declare Sub xxx Lib "rsib.dll" (...)

. **G生成一个应答缓冲区**

因为DLL在应答的情况下返回以零作为结尾的字符串,在访问ibrd()和ilrd()函数之前要先生成一个足够长度的字符串,这是由于Visual Basic在未被DLL所更新的字符串前面加一个长度值的前缀.
下列两个例子可用于为字符串产生一个长度值:
- Dim Rd as String * 100
- Dim Rd as String
Rd = Space$(100)

初始化和默认状态

所有子程序所用的变量应该保存在每个程序的开始部分.这样,在每个程序的开始部分，IEC/IEEE总线以及仪器的设置就被定义为默认状态.子程序" InitController"和"InitDevice"就用于产生这种效应.

创建全局变量

在Visual Basic中,全局变量被置于所谓的"模块"(文件扩展名:.BAS)中.这样,至少应该创建一个模块(例如"GLOBALS.BAS),这个模块包含被所有子程序使用的变量,比如说IEC/IEEE总线驱动程序所使用的器件地址.对于后边的编程范例,文件应该包含下列指令:
Global analyzer As Integer
Global boardId As Integer

初始化控制器:

```
注释 ------------------------------------------ 初始化控制器 ----------------------------------------------------------
Public SUB InitController()
iecaddress% = 20 ‘仪表的IEC/IEEE总线地址
CALL IBFIND("GPIB0", boardId%) ‘对控制器打开端口
CALL IBFIND("DEV1", analyzer%) ‘对仪表打开端口
CALL IBPAD(analyzer%, iecaddress%) ‘在仪表地址通知控制器
CALL IBTMO(analyzer%, 11) ‘反应时间为1秒
END SUB
REM ***************************************************************************************************************
```

初始化仪表

IEC总线寄存器和仪表设定被设置为默认状态.

```
注释 ---------------------------------------- 初始化仪表----------------------------------------------------------
Public SUB InitDevice()
CALL IBWRT(analyzer%, "*CLS") ‘重置状态寄存器
CALL IBWRT(analyzer%, "*RST") ‘重置仪表
END SUB
REM ***************************************************************************************************************
```

将显示置为ON/OFF

在默认状态下,所有的遥控指令都是在关掉显示的前提下来执行的,这样做是为得到最大的测量速度.然而，在控制程序的生成阶段,经常需要显示已编程的设置以及测试结果.下示函数就是如何用遥控打开或者关掉显示的例子::

```
注释 ---------------------------------------------- 打开显示 ----------------------------------------------------
Public SUB DisplayOn()
CALL IBWRT(analyzer%, "SYST:DISP:UPD ON") ‘打开显示
END SUB
REM ***************************************************************************************************************
注释---------------------------------------------- 关掉显示 ----------------------------------------------------
Public SUB DisplayOff()
CALL IBWRT(analyzer%, "SYST:DISP:UPD OFF") ¡¯Switch display off
END SUB
REM ***************************************************************************************************************
```

设置节能功能(显示永久性关闭)

在IEC/IEEE总线操作期间并不总是需要屏幕上的结果.尽管"SYSTem:DISPLAY:UPDate OFF"命令会关掉结果的显示,在遥控模式下这会在速度方面带来相当大的好处,但是显示本身尤其是背景照明仍然保持打开状态.

如果需要,显示应该通过节能功能的方式关掉,同时在激活之前反应时间应该按分钟来设定.

注意: *在仪表的前面板的任一键被按下的同时,显示就会打开..*

```
注释 ---------------------------------------- 设置节能功能 ----------------------------------------------------------
Public SUB PowerSave()
CALL IBWRT(analyzer%, "SYSTem:PSAVe:HOLDoff 1") ‘设置拖延时间为1分钟CALL
IBWRT(analyzer%, "SYSTem:PSAVe ON") ‘节能功能打开
END SUB
REM ***************************************************************************************************
```

简单仪表设定命令的传输

在这个例子中,设定仪表的中心频率,步长和参考电平.

```
注释 ---------------------------------------- 仪表设定命令----------------------------------------------------------
PUBLIC SUB SimpleSettings()
CALL IBWRT(analyzer%, "FREQUENCY:CENTER 128MHz")          ‘中心频率为128MHz
CALL IBWRT(analyzer%, "FREQUENCY:SPAN 10MHZ")             ‘步长为10MHz
CALL IBWRT(analyzer%, "DISPLAY:TRACE:Y:RLEVEL -10dBm")
                                                          ‘参考电平为-10dBm
END SUB
REM ***************************************************************************************************
```

返回到手动控制

```
注释 ---------------------------------- 将仪表转换为手动控制 ------------------------------------------------
CALL IBLOC(analyzer%)                    ‘设置仪表为本地状态
REM ***************************************************************************************************
```

读出仪表设定

上面例子中所做的设定在这里被读出.使用了缩写的命令.

```
注释 ------------------------------------ 读出仪表的设定----------------------------------------------------------
PUBLIC SUB ReadSettings()
CFfrequency$ = SPACE$(20)                ‘提供文本变量(20个字符)
CALL IBWRT(analyzer%, "FREQ:CENT?")      ‘询问中心频率
CALL IBRD(analyzer%, CFfrequency$)       ‘读值
CFspan$ = SPACE$(20)                     ‘提供文本变量(20个字符)
CALL IBWRT(analyzer%, "FREQ:SPAN?")      ‘查询步长
CALL IBRD(analyzer%, CFspan$)            ‘读值
RLevel$ = SPACE$(20)                     ‘提供文本变量(20个字符)
CALL IBWRT(analyzer%, "DISP:TRAC:Y:RLEV?")
                                         ‘查询参考电平
CALL IBRD(analyzer%, RLevel$)            ‘读值
注释 ------------------------------------ 在屏幕上显示值 ----------------------------------------------------
PRINT "Center frequency: "; CFfrequency$,
PRINT "Span: "; CFspan$,
PRINT "Reference level: "; RLevel$,
REM ***************************************************************************************************************
```

定位标记并显示值

```
注释 ------------------------------------ 标记功能的例子 ----------------------------------------------------
PUBLIC SUB ReadMarker()
CALL IBWRT(analyzer%, "CALC:MARKER ON;MARKER:MAX")
                                         ‘激活标记1并开始峰值检索
MKmark$ = SPACE$(30)                     ‘提供文本变量(30个字符)
CALL IBWRT(analyzer%, "CALC:MARK:X?;Y?") ‘查询频率和电平
CALL IBRD(analyzer%, MKmark$)            ‘读值
注释---------------------------------- 在屏幕上显示值----------------------------------------------------
PRINT "Center frequency / level "; MKmark$,
REM ***************************************************************************************************************
```

命令同步

在下面例子中执行的同步的可能事件在第五章"命令的次序和命令的同步"中讨论.

```
注释---------------------------------------- 命令同步的例子----------------------------------------------------
PUBLIC SUB SweepSync()
注释 如果命令INIT:CONT OFF在之前被发送,命令INITiate[:IMMediate]将起始一次扫描.
注释 必须保证当整个扫描都结束时,下个命令才被执行.
CALL IBWRT(analyzer%, "INIT:CONT OFF")
注释 ---------------------------------------- 第一个可能: 使用*WAI ----------------------------------------
CALL IBWRT(analyzer%, "ABOR;INIT:IMM; *WAI")
注释 ---------------------------------------- 第二个可能: 使用*OPC? ----------------------------------------
OpcOk$ = SPACE$(2)                    '*OPC?的间隔 – 提供反应
CALL IBWRT(analyzer%, "ABOR;INIT:IMM; *OPC?")
注释 ---------------------------------------- 控制器可以控制其他的仪表----------------------------------------
CALL IBRD(analyzer%, OpcOk$)          '由*OPC?等待"1"
REM -------- Third possibility: Use of *OPC ----------------------------
注释 为能够使用与国家仪表GPIB驱动想关联的服务请求功能,设置"Disable Auto Serial
注释 Poll"必须通过IBCONF!改为"yes".

CALL IBWRT(analyzer%, "*SRE 32")    '准许ESR的服务请求
CALL IBWRT(analyzer%, "*ESE 1")     '为操作完成位设置时间使能位
¡ operation complete bit
CALL IBWRT(analyzer%, "ABOR;INIT:IMM; *OPC") '起始扫描并同步至OPC
CALL WaitSRQ(boardID%,result%)      '等待服务请求
注释  继续主程序.
END SUB
REM ************************************************************************************************************
```

服务请求

服务请求程序需要仪器的扩展初始化过程,在这个过程中设置过渡位和使能寄存器.要使用与国家仪器的GPIB驱动相关联的服务请求功能, "Disable Auto Serial Poll"项必须通过IBCONF改为"yes".

初始化服务请求

注释 ---------- 在出错情况下SRO初始化的范例----------

```
PUBLIC SUB SetupSRQ()
CALL IBWRT(analyzer%, "*CLS")    ’重置状态报告系统
CALL IBWRT(analyzer%,"*SRE 168") ’容许STAT:OPER,STAT:QUES and ESR寄存器的服
                                 ’务请求
CALL IBWRT(analyzer%,"*ESE 60")  ’设置命令,执行,设备决定和查询错误的时间使能位
CALL IBWRT(analyzer%,"STAT:OPER:ENAB 32767") ’设置所有事件的操作使能位
CALL IBWRT(analyzer%,"STAT:OPER:PTR 32767") ’设置适当的操作转换使能位
CALL IBWRT(analyzer%,"STAT:QUES:ENAB 32767") ’设置所有事件的可疑使能位
CALL IBWRT(analyzer%,"STAT:QUES:PTR 32767") ’设置适当的可以转换位
END SUB
REM ***********************************************************************************************************
```

等待服务请求的到来

基本上有两种方法来等待服务请求的到来:

1. 阻塞 (用户不能输入):

直到信号被SRQ发信号通知时所等待的时间很短(比选定的超出时间还要短),或者在等待时间内无须对用户输入进行反应,或者-作为主要的原则-事件肯定要发生,只有在以上三个条件下，此方法才适用.

原因:

从函数WaitSRQ()被调用到指定事件的发生,在等待时间内,并不容许程序对鼠标点击和键盘输入的反应.而且,如果在预先定义的超出时间阶段内SRQ时间并没有发生，将引起程序退出.这样,在很多种情形下,这个方法并不适用于等待测量结果,尤其是在触发模式测量中.

需要下列函数的调用:

```
CALL WaitSRQ(boardID%,result%) ’等待服务请求
                               ’在等待期间用户无法输入!
IF (result% = 1) THEN CALL Srq ’如果识别到SRQ =>
                               ’赋值子程序
```

2. 非阻塞 (用户可以输入):

如果在事件被一个SRQ发信号通知之前的等待时间很长(比指定超出时间还长),或者在等待时间内使用户可以输入,或者事件不确定要发生,则推荐使用这种方法.这样,这种方法对于等待测量的结束(也就是说,结果的输出,尤其是在触发测量的情况下)来说就是很好的选择.这个方法必须需要一个等待循环,这个等待循环定时检查SRQ线的状态,并且在指定事件还未发生的期间内向操作系统返回一个控制.通过这种方法,在等待时间内,系统可以对用户的输入(鼠标点击,键盘输入)作出反应.

建议使用Hold()辅助函数,这个函数向操作系统返回一个可选的等待时间(见"无须阻断键盘和鼠标的等待"部分),这样就是用户在等待时间内可以输入.

```
result% = 0
For i = 1 To 10 'Abort after max. 10 loop
                                  '重复
CALL TestSRQ(boardID%,result%) '检查服务请求线路
If (result% <> 0) Then
CALL Srq                          '如果识别到SRQ =>
                                  '赋值子程序
Else
Call Hold(20)                     '调用20ms等待时间的保持函数,用户可以输入
Endif
Next i
If result% = 0 Then
PRINT "Timeout Error; Program aborted"' 输出错误信息
STOP                                    '停止软件
Endif
```

无须阻断键盘和鼠标的等待

使用Visual Basic中遥控程序的经常出现的问题是插入等待时间就阻断了键盘和鼠标.

如果在等待时间内程序也对用户的输入进行反应,在这个时间内程序事件的控制就必须返回到操作系统.在Visual Basic中,这是通过调用*DoEvents*函数来完成的.这个函数使键盘或者鼠标触发的事件被相关部分所执行.例如,在用户等待一个仪表设置结束的时间内,它容许按键和输入区域的操作.下面的编程例子描述*Hold()*函数,在以毫秒计的可选等待时间期间,它向操作系统返回一个控制.

```
Rem ******************************************************************
Rem The waiting function below expects the transfer of the desired
Rem waiting time in milliseconds. The keyboard and the mouse remain
Rem operative during the waiting period, thus allowing desired elements
Rem to be controlled
注释 下面的等待函数要求必要的以毫秒计的等待时间的传输.在等待期间,键盘和鼠标保持操作状
注释 态,这样就容许必要部分可以被控制
Rem ******************************************************************
Public Sub Hold(delayTime As Single)
Start = Timer ’保存计时器调用函数的计数
Do While Timer < Start + delayTime / 1000 ’核查计时器计数
DoEvents ’只要计时器还没有超时,将控制返回到操作系统来激活必要部分的控制
Loop
End Sub
Rem ******************************************************************
```

The waiting procedure is activated simply by calling *Hold*.

只要调用*Hold(<等待时间以毫秒计>)*就可以激活等待步骤.

服务请求程序

A service request is processed in the service request routine.

Note: the variables userN% and userM% must be pre-assigned usefully!

服务请求在服务请求程序中处理.

注意:变量N%和用户M%必须可靠地预先分配!

```
注释 ---------------------------------------- 服务请求程序 ----------------------------------------------------
Public SUB Srq()
ON ERROR GOTO noDevice      ’没有用户存在
CALL IBRSP(analyzer%, STB%) ’串行查询, 读状态字节
IF STB% > 0 THEN ’这个仪表设置STB中的位
SRQFOUND% = 1
IF (STB% AND 16) > 0 THEN CALL Outputqueue
IF (STB% AND 4) > 0 THEN CALL ErrorQueueHandler
IF (STB% AND 8) > 0 THEN CALL Questionablestatus
IF (STB% AND 128) > 0 THEN CALL Operationstatus
IF (STB% AND 32) > 0 THEN CALL Esrread
END IF
noDevice:
END SUB ’End of SRQ routine
REM ************************************************************************************************
```

在子程序中读出状态事件寄存器,输出缓冲器和错误/事件队列要受到影响.

读出输出缓冲器

```
注释 ---------------------------------- 单个STB位的子程序----------------------------------------------------
Public SUB Outputqueue() 'Reading the output buffer
result$ = SPACE$(100) '为反应留空间
CALL IBRD(analyzer%, result$)
PRINT "Contents of Output Queue : "; result$
END SUB
REM ***************************************************************************************************************
```

读出错误消息

```
注释 ---------------------------------- 读错误队列的子程序----------------------------------------------------
Public SUB ErrorQueueHandler()
ERROR$ = SPACE$(100) '给错误变量留空间
CALL IBWRT(analyzer%, "SYSTEM:ERROR?")
CALL IBRD(analyzer%, ERROR$)
PRINT "Error Description : "; ERROR$
END SUB
REM ***************************************************************************************************************
```

SCPI状态寄存器的赋值

```
注释 ---------------------------------- 不可靠状态寄存器的赋值----------------------------------------------
Public SUB Questionablestatus()
Ques$ = SPACE$(20) '给文本变量预留空白
CALL IBWRT(analyzer%, "STATus:QUEStionable:EVENt?")
CALL IBRD(analyzer%, Ques$)
PRINT "Questionable Status: "; Ques$
END SUB
REM ***************************************************************************************************************
注释 ---------------------------------------- 操作状态寄存器的赋值子程序-----------------------------------
Public SUB Operationstatus()
Oper$ = SPACE$(20) ' 给文本变量预留空白
CALL IBWRT(analyzer%, "STATus:OPERation:EVENt?")
CALL IBRD(analyzer%, Oper$)
PRINT "Operation Status: "; Oper$
END SUB
REM ***************************************************************************************************************
```

事件状态寄存器的赋值

```
注释 ----------事件状态寄存器的赋值子程序 ----------
Public SUB Esrread()
Esr$ = SPACE$(20) ’ 给文本变量预留空白
CALL IBWRT(analyzer%, "*ESR?") ’读ESR
CALL IBRD(analyzer%, Esr$)
IF (VAL(Esr$) AND 1) > 0 THEN PRINT "Operation complete"
IF (VAL(Esr$) AND 2) > 0 THEN PRINT "Request Control"
IF (VAL(Esr$) AND 4) > 0 THEN PRINT "Query Error"
IF (VAL(Esr$) AND 8) > 0 THEN PRINT "Device dependent error"
IF (VAL(Esr$) AND 16) > 0 THEN
PRINT "Execution Error; Program aborted"’ Output error message
STOP                              ’停止软件
END IF
IF (VAL(Esr$) AND 32) > 0 THEN
PRINT "Command Error; Program aborted"’ 输出错误消息
STOP                                    ’停止软件
END IF
IF (VAL(Esr$) AND 64) > 0 THEN PRINT "User request"
IF (VAL(Esr$) AND 128) > 0 THEN PRINT "Power on"
END SUB
REM ************************************************************************************************************
```

更多复杂的编程范例

FSP的默认设置

The following settings are an example of how to modify the default setting of the FSP.
下列设置是如何修改FSP默认状态的两个实例中的一个.需要注意的是,根据应用的范例不同,只有其中的一些设置是必要的.特别是对于解析带宽,视频带宽和扫描时间的设置常常是不需要的,因为在修改频率范围步长的默认设置中,这些参数被自动计算.插入损失同样也是根据参考电平自动计算的.在默认设置,电平检测器与选定轨迹模式自动耦合.在默认设置中自动计算的设置在以后的编程示例中被标以(*).

IEC/IEEE总线状态寄存器的设置

```
REM ***************************************************************************************************************
Public Sub SetupStatusReg()
'------------------------------------ IEEE 488.2 状态寄存器 ----------------------------------------------------
CALL IBWRT(analyzer%,"*CLS")        '重置状态寄存器
CALL IBWRT(analyzer%,"*SRE 168")    '服务请求使能
'for STAT:OPER-,STAT:QUES- and
'ESR registers
CALL IBWRT(analyzer%,"*ESE 61")     '给Operation Complete, Command-,
                                    'Execution-,Device Dependent-
                                    '和 Query Error 设置事件使能位
'------------------------------------------------ SCPI状态寄存器 ----------------------------------------------------
CALL IBWRT(analyzer%,"STAT:OPER:ENAB 0") '关闭操作状态寄存器
CALL IBWRT(analyzer%,"STAT:QUES:ENAB 0") '关闭不稳定状态寄存器
End Sub
REM ***************************************************************************************************************
```

测量的默认设置

```
REM ************************************************************************************************************
Public Sub SetupInstrument()
'-------------------- FSP default setting ---------------------------------
CALL SetupStatusReg '设置状态寄存器
CALL IBWRT(analyzer%,"*RST") '设置仪表
CALL IBWRT(analyzer%,"SYST:DISP:UPD ON") 'ON: 显示打开
                                         'OFF: 关闭(改善的性能)
CALL IBWRT(analyzer%,"DISP:FORM SINGle") '满屏
CALL IBWRT(analyzer%,"DISP:WIND1:SEL") '激活屏幕A
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描
'------------------------------------------- 设置频率 ----------------------------------------------------------
CALL IBWRT(analyzer%,"FREQUENCY:CENTER 100MHz") '中心频率
CALL IBWRT(analyzer%,"FREQ:SPAN 1 MHz")          '步长
'------------------------------------------- 设置电平 ----------------------------------------------------------
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:RLEV -20dBm") '参考电平
CALL IBWRT(analyzer%,"INP:ATT 10dB") '输入衰减(*)
'------------------------------------------- y轴刻度 ----------------------------------------------------------
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:SPAC LOG")    '对数电平轴
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:SCAL 100dB")  '电平范围
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:SCAL:MODE ABS") '绝对值标度
CALL IBWRT(analyzer%,"CALC:UNIT:POW DBM")            'y轴的单位
'------------------------------------------- 轨迹和探测器的设置 ------------------------------------------------
CALL IBWRT(analyzer%,"DISP:WIND:TRAC1:MODE AVER") '轨迹1的均值
CALL IBWRT(analyzer%,"AVER:TYPE VID") '线性的"LIN"的均值模式视频
CALL IBWRT(analyzer%,"SWE:COUN 10")    '扫描计数
CALL IBWRT(analyzer%,"DISP:WIND:TRAC2:STAT OFF") '轨迹2的空白
CALL IBWRT(analyzer%,"DISP:WIND:TRAC3:STAT OFF") '轨迹3的空白
CALL IBWRT(analyzer%,"CALC:MATH:STAT OFF") '轨迹算法关闭
CALL IBWRT(analyzer%,"DETECTOR1 RMS") '检测器轨迹1 (*)
CALL IBWRT(analyzer%,"DET2:AUTO ON") '检测器轨迹2 (*)
CALL IBWRT(analyzer%,"DET3:AUTO ON") '检测器轨迹3 (*)
'--------- Band width and sweep time ----------------------------------------
CALL IBWRT(analyzer%,"BAND:RES 100KHz") '解析带宽 (*)
CALL IBWRT(analyzer%,"BAND:VID 1MHz") '视频带宽(*)
CALL IBWRT(analyzer%,"SWE:TIM 100ms") '扫描时间(*)
END SUB
REM ************************************************************************************************************
```

使用标记和增量标记

标记检索功能,搜索范围的设置

下面是例子是建立在一个100MHz的调幅调制信号的基础上，这个信号有以下的参数：
. 载波信号电平：－30 dBm
.音频: 100 kHz
. 调制深度: 50 %
Marker 1 and delta marker 2 are set one after the other to the highest maxima of the measurement curve and then the frequency and level are read out. The default setting of the FSP can be used for the
following measurements (SetupInstrument).
标记1和增量标记2依次设置到测量曲线最大值的最高点，随后频率和电平被读出。分析仪的默认设置可用于下列测量（设置仪器）

```
REM ***************************************************************************************************************
Public Sub MarkerSearch()
result$ = Space$(100)
CALL SetupInstrument ¡¯Default setting
'---------------------------------------- 无检索限制的峰值检索 ----------------------------------------------------
CALL IBWRT(analyzer%,"INIT:CONT OFF") '切换至单扫描
CALL IBWRT(analyzer%,"CALC:MARK:PEXC 6DB") '定义峰值偏移
CALL IBWRT(analyzer%,"CALC:MARK:STAT ON") '打开标记1
CALL IBWRT(analyzer%,"CALC:MARK:TRAC 1") '把标记1分配给轨迹1
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
CALL IBWRT(analyzer%,"CALC:MARK:MAX;X?;Y?") '峰值的标记；读出
CALL IBRD(analyzer%, result$) '频率和电平
Print "Marker 1: ";result$
CALL IBWRT(analyzer%,"CALC:DELT2:STAT ON;MAX;MAX:LEFT")
'打开增量标记2
'峰值和随后的下个峰值左侧
CALL IBWRT(analyzer%,"CALC:DELT:MODE ABS") '增量标记2频率输出绝对值
CALL IBWRT(analyzer%,"CALC:DELT2:X?;Y?") '增量标记2-读出频率和电平
CALL IBRD(analyzer%, result$)
Print "Delta 2: ";result$
'------------------------------------ 在x轴方向有检索限制的峰值检索 ----------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:X:SLIM:STAT ON;LEFT 0Hz;RIGHt 100.05MHz")
'检索限制打开并且在右侧设置低频以下的部分
CALL IBWRT(analyzer%,"CALC:DELT3:STAT ON;MAX;MAX:RIGHt")
'增量标记3打开
'峰值和随后的下个右峰值
CALL IBWRT(analyzer%,"CALC:DELT3:X?;Y?") '增量标记3;读出频率和电平，两者必须读
                                          '出
CALL IBRD(analyzer%, result$) '有值0
Print "Delta 3: ";result$
```

```
'-----------------------------------------在x轴方向有检索限制的峰值检索----------------------------------------
CALL IBWRT(analyzer%,"CALC:THR:STAT ON")
CALL IBWRT(analyzer%,"CALC:THR –35DBM") '极限开并在低频以上设置
CALL IBWRT(analyzer%,"CALC:DELT3:STAT ON;MAX;MAX:NEXT")
                                  '增量标记3打开
                                  '峰值和随后下个峰值打开
                                  ' => 未发现
CALL IBWRT(analyzer%,"CALC:DELT3:X:REL?;:CALC:DELT3:Y?")
CALL IBRD(analyzer%, result$) '增量标记3；读出频率和电平，两者都是必须值为0
Print "Delta 3: ";result$
'------------------------------- 通过标记设置中心频率和参考电平 ----------------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK2:FUNC:CENT") '增量标记2 -> 标记和中心频率＝光
                                              '标2
CALL IBWRT(analyzer%,"CALC:MARK2:FUNC:REF") '参考电平=标记2
Call ibwrt(analyzer%,"INIT;*WAI") '执行同步扫描
END SUB
REM ***********************************************************************************************************
```

测量伪辐射

在辐射测量中，常常需要为多余的伪辐射寻找很大的频率范围.这可以用FSP的LIST PEAKS功能来做,它在一个预选的频率范围内发现多达50个峰值并且把它们输出为一个列表.搜寻范围既可以用频率和电平来定义,也可以用将要发现的峰值数目来定义.
在下面例子中,在预选频率范围内将发现10个最高的峰值.
只有在中心频率± 400 kHz附近的范围内>-60 dBm的信号才是我们感兴趣的,所以搜索范围也相应的受限制.发现的信号是以上升频率的顺序的输出.

```
REM ***************************************************************************************************
Public Sub SpuriousSearch()
powerlist$ = Space$(1000)
freqlist$ = Space$(1000)
count$ = Space$(30)
'---------------------------------------------- FSP的默认设置 ----------------------------------------------
CALL SetupInstrument ' 默认设置
CALL IBWRT(analyzer%,"INIT:CONT OFF") ' 默认设置
'---------------------------------------- 检索范围的定义 ----------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:X:SLIM:STAT ON")
CALL IBWRT(analyzer%,"CALC:MARK:X:SLIM:LEFT 99.6MHz;RIGHt 100.4MHz")
'激活检索范围并设置在中心频率附近±400 kHz 内
CALL IBWRT(analyzer%,"CALC:THR:STAT ON")
CALL IBWRT(analyzer%,"CALC:THR -60DBM") '激活并设置在-60 dBm
'---------------------------------------- 激活伪辐射的检索 ----------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:FPE:SORT X") '按频率排序
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:FPE 10") '检索10个最高的峰值
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:FPE:COUN?") '调用峰值的数量并核对
CALL IBRD(analyzer%, count$) 'and read it in
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:FPE:X?") '查询并读
CALL IBRD(analyzer%, freqlist$) '频率列表
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:FPE:Y?") '查询并读
CALL IBRD(analyzer%, powerlist$) '电平列表
Print "# of spurious: ";count$ '结果的输出数目
Print "Frequencies: ";freqlist$ '输出频率列表
Print "Power: ";powerlist$ '输出电平列表
END SUB
REM ***************************************************************************************************
```

频率计数

The following example is based on a signal with a level of at. The default setting of
the FSP can also be used for this measurement (SetupInstrument). The objective of frequency counting
is to determine the exact frequency of the signal at 100 MHz.
下面例子是以100 MHz–30 dBm电平的信号为基础的.分析仪的默认设置同样可用于这项测量(设置仪器).频率计数的目的是测定100MHz信号的确切频率.

```
REM ***************************************************************************************************
Public Sub MarkerCount()
result$ = Space$(100)
CALL SetupInstrument ’默认设置
’---------------------------------------- 用频率计数器测量信号频率 ----------------------------------------
CALL IBWRT(analyzer%,"INIT:CONT OFF") ’单扫描打开
CALL IBWRT(analyzer%,"CALC:MARK:PEXC 6DB") ’峰值偏移
CALL IBWRT(analyzer%,"CALC:MARK:STAT ON") ’标记1打开
CALL IBWRT(analyzer%,"CALC:MARK:TRAC 1") ’将标记1分配给轨迹1
CALL IBWRT(analyzer%,"CALC:MARK:X 100MHz") ’标记1设置在100 MHz
CALL IBWRT(analyzer%,"CALC:MARK:COUNT:RES 1HZ") ’频率计数器1 Hz
CALL IBWRT(analyzer%,"CALC:MARK:COUNT ON") ’频率计数器打开
CALL IBWRT(analyzer%,"INIT;*WAI") ’执行同步扫描
CALL IBWRT(analyzer%,"CALC:MARK:COUNT:FREQ?") ’查询测得的频率
CALL IBRD(analyzer%, result$) ’并读出
Print "Marker Count Freq: ";result$
END SUB
REM ***************************************************************************************************
```

固定参考点的操作 (参考固定)

下面例子以一个100MHz-20dBm电平的信号为基础.信号的谐波在200 MHz, 300 MHz等位置.在使用高品质信号源的前提下,这些谐波信号有可能处于FSP的动态范围之外.然而,为测量谐波抑制,电平应设置到更高的灵敏度以测量谐波;载波应由一个凹口滤波器进行抑制以避免分析仪射频输入的过载.

这样,在下面例子中用不同的电平设置执行两个测量,先在载波频率处用一个较高的参考电平,然后在三次谐波频率处用一个较低的参考电平测量的.

测量(设置仪器)用的分析仪的默认设置被用作起始点,随后调整(测量)设置.

```
REM ************************************************************************
Public Sub RefFixed()
result$ = Space$(100)
CALL SetupInstrument ’默认设置
’---------------------------------------- 测量参考点----------------------------------------------------------
CALL IBWRT(analyzer%,"INIT:CONT OFF") ’单扫描
CALL IBWRT(analyzer%,"CALC:MARK:PEXC 6DB") ’峰值偏移
CALL IBWRT(analyzer%,"CALC:MARK:STAT ON") ’标记1打开
CALL IBWRT(analyzer%,"CALC:MARK:TRAC 1") ’标记1分配给轨迹1
CALL IBWRT(analyzer%,"INIT;*WAI") ’执行同步扫描
CALL IBWRT(analyzer%,"CALC:MARK:MAX") ’设置标记1为100 MHz
CALL IBWRT(analyzer%,"CALC:DELT:FUNC:FIX ON") ’参考固定
’---------------------------------设置谐波测量的频率,电平和带宽-----------------------------------------
CALL IBWRT(analyzer%,"FREQ:CENT 400MHz;Span 1MHz" )’设置3次谐波的频率
CALL IBWRT(analyzer%,"BAND:RES 1kHz") ’和适当的RBW
CALL IBWRT(analyzer%,"SWEEP:TIME:AUTO ON") ’耦合扫描时间
CALL IBWRT(analyzer%,"INP:ATT:AUTO ON") ’优化电平
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:RLEV –50dBm")
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
CALL IBWRT(analyzer%,"CALC:DELT:MODE REL") '增量标记频率
                                            '相关
CALL IBWRT(analyzer%,"CALC:DELT:MAX;X?;Y?") '读出增量标记
Call ibrd(analyzer%, result$) '读出频率和电平
Print "Deltamarker 1: "; result$
END SUB
REM ************************************************************************************************
```

相位和相位噪声测量

在相位噪声测量期间,在1Hz内的噪声功率被调节为与一个邻接载波信号的功率成比例.在被测频率和载波频率之间所用的间隔通常为10kHz.
对于噪声测量,已经测量的电平绝对值是在1Hz带宽内测量的.
下面例子同样也以一个100MHz-30dBm电平的信号为基础.用两个标记来测定载波信号偏置10kHz的噪声和相位噪声.

```
REM ************************************************************************
Public Sub Noise()
result$ = Space$(100)
'--------- FSP默认设置 ----------------------------------------
CALL SetupStatusReg '配置状态寄存器
CALL IBWRT(analyzer%,"*RST") '重置仪表
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描
'--------- 设置频率----------------------------------------------
CALL IBWRT(analyzer%,"FREQUENCY:CENTER 100MHz") '中心频率
CALL IBWRT(analyzer%,"FREQ:SPAN 100 kHz") '步长
'--------- 设置电平 ---------------------------------------------
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:RLEV -20dBm") '参考电平
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
'--------- 设置参考点 -------------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:PEXC 6DB") '峰值偏移
CALL IBWRT(analyzer%,"CALC:MARK:STAT ON") '标记1打开
CALL IBWRT(analyzer%,"CALC:MARK:TRAC 1") '将标记1分配给轨迹1
CALL IBWRT(analyzer%,"CALC:MARK:MAX") '设置标记1为100 MHz
CALL IBWRT(analyzer%,"CALC:DELT:FUNC:PNO ON") '定义相位噪声参考点
'--------- 测量相位噪声 -----------------------------------------
CALL IBWRT(analyzer%,"CALC:DELT:X 10kHz") '设置增量标记
CALL IBWRT(analyzer%,"CALC:DELT:FUNC:PNO:RES?")'读出相位噪声测量的结果
Call ibrd(analyzer%, result$)
Print "Phase Noise [dBc/Hz]: "; result$
'--------- 测量噪声 ---------------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:X 99.96MHz") '设置标记1
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:NOIS:RES?")'读出结果
Call ibrd(analyzer%, result$)
Print "Noise [dBm/Hz]: "; result$
END SUB
REM ************************************************************************
```

波形因子的测量(使用n-dB down)

分析仪的n-dB-down功能被使用两次以测量滤波器的波形因子.(滤波器最大值60 dB和3 dB以下带宽的比值).测量30kHz解析带宽的波形系数.分析仪的默认设置被用于测量(设置仪器).

```
REM ************************************************************************
Public Sub ShapeFactor()
result$ = Space$(100)
'--------- FSP默认设置 ----------------------------------------
CALL SetupInstrument '默认设置
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描
'--------- 设置频率-------------------------------------------------
CALL IBWRT(analyzer%,"FREQ:SPAN 1MHz") '步长
CALL IBWRT(analyzer%,"BAND:RES 30kHz") '解析带宽
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
'--------- 测量 60 dB值 ---------------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:PEXC 6DB") '峰值偏移
CALL IBWRT(analyzer%,"CALC:MARK:STAT ON") '标记1打开
CALL IBWRT(analyzer%,"CALC:MARK:TRAC 1") '将标记1分配给轨迹1
CALL IBWRT(analyzer%,"CALC:MARK:MAX") '设置标记1为100 MHz
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:NDBD 60dB")'读出测得的带宽
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:NDBD:RES?")'在60 dB
CALL IBRD(analyzer%,result$)
result60 = Val(result$)
'--------- 测量3 dB 以下值-------------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:NDBD 3dB")'读出测得的带宽
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:NDBD:RES?")'在60 dB
CALL IBRD(analyzer%,result$)
result3 = Val(result$)
'--------- 读出波形因子-----------------------------------------
Print "Shapefaktor 60dB/3dB: ";result60/result3
END SUB
REM ************************************************************************
```

测量三阶拦截点

第三阶拦截点是两个邻接的有用信号的虚电平,在这个电平处,第三阶调制间信号的乘积与有用信号的电平值相同.

f_{S2}频率的调制间信号乘积通过混合一次谐波的有用信号P_{N2}和P_{N1}得到, f_{S1}频率的调制间信号乘积通过混合一次谐波的有用信号P_{N1}和P_{N2}得到.

$f_{S1} = 2 \times f_{n1} - f_{n2}$ (1)

$f_{S2} = 2 \times f_{n2} - f_{n1}$ (2)

下面例子是建立于分别在100MHz和110MHz频率的–30 dBm电平的两个邻接信号的基础上的.根据上面的公式,互调乘积处于90MHz和120MHz之间.频率被设定,这样,测得的混合积就在图中显示.否则,FSP的默认设置在测量中使用(设置仪器).

```
REM ***************************************************************************************************
Public Sub TOI()
result$ = Space$(100)
'---------------------------------------- FSP默认设置----------------------------------------------------
CALL SetupStatusReg '设置状态寄存器
CALL IBWRT(analyzer%,"*RST") '设置仪表
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描
CALL IBWRT(analyzer%,"SYST:DISP:UPD ON") 'ON: 显示打开
                                         'OFF: 关闭
'---------------------------------------- 设置频率 ------------------------------------------------------
CALL IBWRT(analyzer%,"FREQ:STARt 85MHz;STOP 125 MHz") '步长
'---------------------------------------- 设置电平------------------------------------------------------
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:RLEV –20dBm") '参考电平
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
'---------------------------------------- TOI 测量------------------------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:PEXC 6DB") '峰值偏移
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:TOI ON") '打开TOI测量
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:TOI:RES?") '并读出结果
CALL IBRD(analyzer%,result$)
'---------------------------------------- 读出结果 ------------------------------------------------------
Print "TOI [dBm]: ";result$
END SUB
REM ***************************************************************************************************
```

测量调幅调制深度

下面例子是建立于一个100MHz的调幅调制信号的基础之上的,此信号有如下参数:
. 载波信号电平: –30 dBm
. 音频频率: 100 kHz
.调制深度: 50 %
用于测量的分析仪的默认设置可用于下面描述的测量(设置仪器).

```
REM ***************************************************************************************************************
Public Sub AMMod()
result$ = Space$(100)
CALL SetupInstrument ’默认设置
’---------------------------------------------- 峰值检索 ----------------------------------------------------------
CALL IBWRT(analyzer%,"INIT:CONT OFF") ’单扫描
CALL IBWRT(analyzer%,"INIT;*WAI") ’执行同步扫描
CALL IBWRT(analyzer%,"CALC:MARK:PEXC 6DB") ’峰值偏移
CALL IBWRT(analyzer%,"CALC:MARK:STAT ON") ’标记1打开
CALL IBWRT(analyzer%,"CALC:MARK:TRAC 1") ’标记1分配给轨迹1
’---------------------------------------------- 测量调制深度----------------------------------------------------------
CALL IBWRT(analyzer%,"CALC:MARK:MAX;FUNC:MDEP ON") ’标记至峰值;
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:MDEP:RES?") ’测量调制深度
CALL IBRD(analyzer%, result$) ’Read out result
’---------------------------------------------- 读出结果 ----------------------------------------------------------
Print "AM Mod Depth [%]: ";result$
END SUB
REM ***************************************************************************************************************
```

极限线和极限测试

下面的例子显示屏幕A上的轨迹1和屏幕B上的轨迹2的新的极限线5的定义和使用,有如下参数:
.上极限线
.在频域的x绝对值轴
.5个参考值: 120 MHz / -70 dB, 126 MHz/-40 dB, 127 MHz/-40 dB, 128 MHz/-10 dB,
129 MHz/-40 dB, 130 MHz/-40 dB, 136 MHz / -70 dB
.用单位dB关联y轴
.无富余

```
REM ***************************************************************************************************
Public Sub LimitLine()
result$ = Space$(100)
'---------------------------------- FSP默认设置 ----------------------------------------------------------
CALL SetupInstrument '默认设置
CALL IBWRT(analyzer%,"FREQUENCY:CENTER 128MHz;Span 10MHz")'步长
Call ibwrt(analyzer%,"Diag:Serv:Inp Cal;CSO -30dBm") '信号Cal打开
'---------------------------------- 极限线的定义 ----------------------------------------------------------
CALL IBWRT(analyzer%,"CALC:LIM5:NAME 'TEST1'") '定义名字
CALL IBWRT(analyzer%,"CALC:LIM5:COMM 'Upper limit'") '定义内容
CALL IBWRT(analyzer%,"CALC1:LIM5:TRAC 1") '为屏幕A分配轨迹
CALL IBWRT(analyzer%,"CALC2:LIM5:TRAC 2") '为屏幕B分配轨迹
CALL IBWRT(analyzer%,"CALC:LIM5:CONT:DOM FREQ") '定义x轴范围
CALL IBWRT(analyzer%,"CALC:LIM5:CONT:MODE ABS") '定义x轴刻度
CALL IBWRT(analyzer%,"CALC:LIM5:UNIT DB") '定义y轴单位
CALL IBWRT(analyzer%,"CALC:LIM5:UPP:MODE REL") '定义y轴刻度
'---------------------------------- 数据点和极限的定义 ----------------------------------------------------
xlimit$ = "CALC:LIM5:CONT
120MHZ,126MHZ,127MHZ,128MHZ,129MHZ,130MHz,136MHz"
CALL IBWRT(analyzer%, xlimit$) '设置x轴的值
CALL IBWRT(analyzer%,"CALC:LIM5:UPP –70,-40,-40,-20,-40,-40,-70")
                                  '设置y轴的值
CALL IBWRT(analyzer%,"CALC:LIM5:UPP:THR -75DBM") '设置y轴极限(只对相关的y轴
                                                 '可能)
'---------------------------------------------------------------------------------------------------------
'一个空白或者 x /y 偏置可以在这里定义.
'---------------------------------------- 激活并赋值屏幕A中的极限线----------------------------------------
CALL IBWRT(analyzer%,"CALC1:LIM5:UPP:STAT ON") '激活屏幕A中的5号线
CALL IBWRT(analyzer%,"CALC1:LIM5:STAT ON") '激活屏幕A中的极限核对
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
```

```
CALL IBWRT(analyzer%,"CALC1:LIM5:FAIL?") ’查询极限核查结果
CALL IBRD(analyzer%, result$) ’Result: 1 (= FAIL)
’---------------------------------------- 读出结果 ----------------------------------------
Print "Limit Result Line 5: ";result$
’---------------------------------- 通过状态寄存器在屏幕A上估算极限线 ----------------------------------
CALL IBWRT(analyzer%,"*CLS") ’重置状态寄存器
’------------------------------------------------ 测量 ------------------------------------------------
CALL IBWRT(analyzer%,"INIT;*OPC") ’执行同步扫描
CALL WaitSRQ(boardID%,status%) ’等待服务请求
’------------------------------------------------ 读出结果 ------------------------------------------------
IF (status% = 1) THEN
CALL IBWRT(analyzer%,"STAT:QUES:LIM1:COND?") ’读出 STAT:QUES:LIMit
CALL IBRD(analyzer%, result$)                ’寄存器
IF ((Val(result$) And 16) <> 0) THEN
Print "Limit5 failed"
ELSE
Print "Limit5 passed"
END IF
END IF
END SUB
REM ************************************************************************************************************
```

测量信道和邻信道功率

在下面例子中,信道和邻道功率首先在0dBm电平800MHz IS95信号上测量的.随后使用快速ACP测量935.2MHz的GSM信号的信道和邻道功率.
此外,极限测试也被激活.

```
REM ***************************************************************************************************
Public Sub ACP()
result$ = Space$(100)
'---------------------------------------- FSP的默认设置 ----------------------------------------------------
CALL SetupStatusReg                          '设置状态寄存器
CALL IBWRT(analyzer%,"*RST")                 '重置仪表
CALL IBWRT(analyzer%,"INIT:CONT OFF")    '单扫描
CALL IBWRT(analyzer%,"SYST:DISP:UPD ON" )'ON: 显示打开
                                             'OFF: 关闭
'------------------------------------------ 设置频率 ----------------------------------------------------
CALL IBWRT(analyzer%,"FREQ:CENT 800MHz") '设置频率
'------------------------------------------ 设置电平 ----------------------------------------------------
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:RLEV 10dBm") '参考电平
'---------------------------------------- 例1: 配置CDMA的CP/ACP ------------------------------------------
CALL IBWRT(analyzer%,"CALC2:MARK:FUNC:POW:SEL ACP") 'ACP measurement on
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:POW:PRES F8CDMA") 'Select CDMA800 FWD
CALL IBWRT(analyzer%,"SENS:POW:ACH:ACP 2") 'Select 2 adjacent channels
CALL IBWRT(analyzer%,"SENS:POW:ACH:PRES ACP") 'Optimize settings
CALL IBWRT(analyzer%,"SENS:POW:ACH:PRES:RLEV") 'Optimize reference level
CALL IBWRT(analyzer%,"SENS:POW:ACH:MODE ABS") 'Absolute measurement
CALL IBWRT(analyzer%,"SENS:POW:HSP ON") 'Fast ACP measurement
'---------------------------------------- 执行测量和查询结果 ----------------------------------------------
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
CALL IBWRT(analyzer%,"CALC2:MARK:FUNC:POW:RES? ACP") '查询结果
CALL IBRD(analyzer%, result$)
'---------------------------------------- 读出结果 ----------------------------------------------------
Print "Result (CP, ACP low, ACP up, Alt low, Alt up): "
Print result$
```

```
'---------------------------------------- 例 2:手动配置GSM的CP/ACP ----------------------------------------
result$ = Space$(100)
CALL IBWRT(analyzer%,"FREQ:CENT 935.2MHz")          '设置频率
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:POW:SEL ACP") 'ACP 测量打开
CALL IBWRT(analyzer%,"SENS:POW:ACH:ACP 1")          '第一个邻道
CALL IBWRT(analyzer%,"SENS:POW:ACH:BAND 200KHZ")    '信道带宽200 kHz
CALL IBWRT(analyzer%,"SENS:POW:ACH:BAND:ACH 200KHZ")'邻道带宽200 kHz
CALL IBWRT(analyzer%,"SENS:POW:ACH:SPAC 200KHZ")    '信道间隔200 kHz
CALL IBWRT(analyzer%,"SENS:POW:ACH:PRES ACP")       '优化设置
CALL IBWRT(analyzer%,"SENS:POW:ACH:PRES:RLEV")      '优化参考电平
CALL IBWRT(analyzer%,"SENS:POW:ACH:MODE ABS")       '取测量绝对值
'---------------------------------------- 起始测量并查询结果 ----------------------------------------
CALL IBWRT(analyzer%,"INIT;*WAI") '执行同步扫描
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:POW:RES? ACP") '查询结果
CALL IBRD(analyzer%, result$)
'---------------------------------------- 读出结果 ----------------------------------------
Print "Result (CP, ACP low, ACP up): "
Print result$
'---------------------------------------- 激活极限核查 ----------------------------------------
result$ = Space$(100)
CALL IBWRT(analyzer%,"CALC:LIM:ACP:ACH 30DB, 30DB") '设置相对极限
CALL IBWRT(analyzer%,"CALC:LIM:ACP:ACH:ABS –35DBM,-35DBM")
                                                     '设置绝对极限
CALL IBWRT(analyzer%,"CALC:LIM:ACP:ACH:STAT ON")    '相对极限核查打开
CALL IBWRT(analyzer%,"CALC:LIM:ACP:ACH:ABS:STAT ON") '绝对极限核查打开
CALL IBWRT(analyzer%,"CALC:LIM:ACP ON")              '极限核查打开
'---------------------------------------- 起始测量并查询结果 ----------------------------------------
CALL IBWRT(analyzer%,"INIT;*WAI")                   '执行同步扫描
CALL IBWRT(analyzer%,"CALC:LIM:ACP:ACH:RES?")      '查询极限核查的结果
CALL IBRD(analyzer%, result$)
'---------------------------------------- 读出结果 ----------------------------------------
Print "Result Limit Check: ";result$
END SUB
REM ***************************************************************************************************************
```

占用带宽测量

```
REM ************************************************************************************************
Public Sub OBW()
result$ = Space$(100)
'---------------------------------------------- FSP 的默认设置 ----------------------------------------------
CALL SetupStatusReg                          '设置状态寄存器
CALL IBWRT(analyzer%,"*RST")                 '重置仪表
CALL IBWRT(analyzer%,"INIT:CONT OFF")  '单扫描
CALL IBWRT(analyzer%,"SYST:DISP:UPD ON") 'ON: 显示打开
                                              'OFF: 关闭
'---------------------------------------- 针对GSM的OBW配置FSP ----------------------------------------
CALL IBWRT(analyzer%,"FREQ:CENT 935.2MHz") '设置频率
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:POW:SEL OBW") 'OBW测量打开
CALL IBWRT(analyzer%,"SENS:POW:ACH:BAND 200KHZ") '信道带宽 200 kHz
CALL IBWRT(analyzer%,"SENS:POW:BWID 95PCT") '功率的百分比
CALL IBWRT(analyzer%,"SENS:POW:ACH:PRES OBW") '设置频率并优化参考电平
CALL IBWRT(analyzer%,"SENS:POW:ACH:PRES:RLEV")
CALL IBWRT(analyzer%,"SENS:POW:NCOR OFF")    '噪声纠正
                                             'OFF: 关闭
                                             'ON: 打开
'---------------------------------------- 执行测量并查询结果 ----------------------------------------
CALL IBWRT(analyzer%,"INIT;*WAI")            '执行同步扫描
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:POW:RES? OBW") '查询结果
CALL IBRD(analyzer%, result$)
Print result$
END SUB
REM ************************************************************************************************
```

时域功率测量

在下面例子中,100MHz 300kHz带宽信号的平均功率将被测量.此外,峰值功率,rms值和标准偏移也被测量.要完成这些,使用了时域功率测量功能.

```
REM ***************************************************************************************************************
Public Sub TimeDomainPower()
result$ = Space$(100)
'---------------------------------------- FSP默认设置 ----------------------------------------------------------
CALL SetupStatusReg          '设置状态寄存器
CALL IBWRT(analyzer%,"*RST") '重置仪表
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描
CALL IBWRT(analyzer%,"SYST:DISP:UPD ON") 'ON: 显示打开
                                         'OFF: 关闭
'---------------------------------------- 针对时域功率测量设置FSP ----------------------------------------------
CALL IBWRT(analyzer%,"FREQ:CENT 100MHz;SPAN 0Hz") '设置频率
CALL IBWRT(analyzer%,"BAND:RES 300kHz")           '解析带宽
CALL IBWRT(analyzer%,"SWE:TIME 200US")            '扫描时间
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:SUMM:PPE ON") '峰值测量打开
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:SUMM:MEAN ON") '平均测量打开
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:SUMM:RMS ON") 'RMS测量
CALL IBWRT(analyzer%,"CALC:MARK:FUNC:SUMM:SDEV ON") '标准偏移打开
'---------------------------------------- 执行测量并查询结果 ---------------------------------------------------
CALL IBWRT(analyzer%,"INIT;*WAI")                   '执行同步扫描
'查询结果:
query$ = " CALC:MARK:FUNC:SUMM:PPE:RES?;"          '峰值测量
query$ = query$ + ":CALC:MARK:FUNC:SUMM:MEAN:RES?;" '平均测量
query$ = query$ + ":CALC:MARK:FUNC:SUMM:RMS:RES?;" 'RMS测量
query$ = query$ + ":CALC:MARK:FUNC:SUMM:SDEV:RES?" '标准偏移
Call IBWRT(analyzer%, query$)
CALL IBRD(analyzer%, result$)
Print result$
END SUB
REM ***************************************************************************************************************
```

在功率斜线上的快速功率测量

在移动无线电测试中,一项常见的任务就是在最高可能速度下在不同的功率控制电平处,对于DUT的测量.对于这个任务,FSP提供两个测试功能,它们可根据信号的参数不同来使用.在下面,通过两个例子来介绍这两个方法.

使用多和标记的功率测量

多和标记功能适合于测量具有下列参数的脉冲序列的功率:
.例如,以相同时间间隔出现的脉冲,这对以时隙形式的GSM传输来说是很典型的.
.第一个信号的电平高于极限值
.随后的脉冲可以有各种电平值
此功能将第一个脉冲作为触发信号.将专门通过针对脉冲序列选定的时间选择方式来测量后续脉冲的功率.这样,这个功能就适合在DUT输出功率变化很大并且不可靠的处于触发开始之上时进行调整.
测量的精度是由脉冲周期于总测量时间的比值来决定的,这个比值不应该低于1:50.
这个功能经常使用选定屏幕的轨迹1.

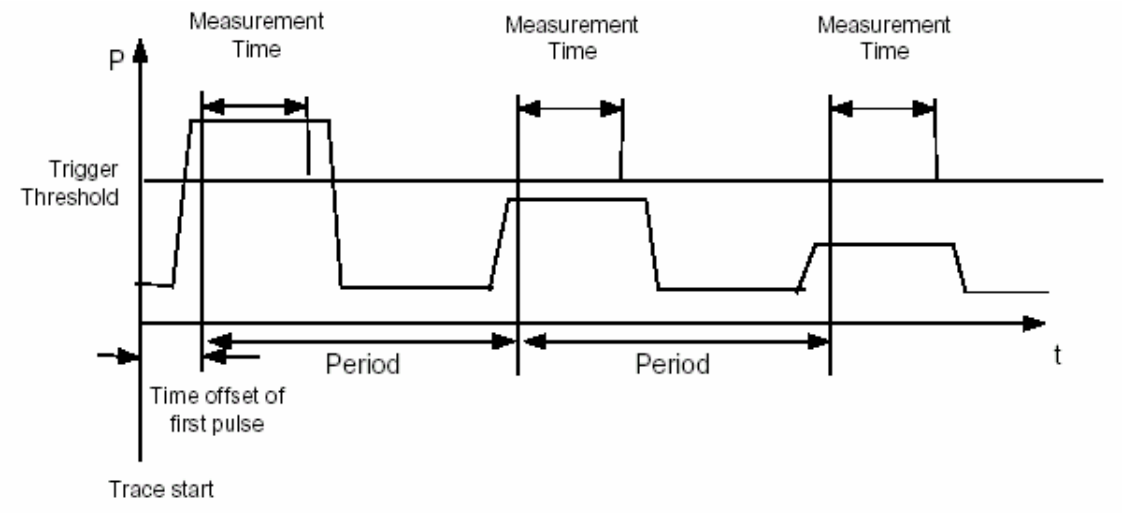

图 7-1 信号特性和要测量的信号时间选择方式

在下面例子中,在对第一个脉冲50⌈s的偏移,450⌈s的测量时间每个脉冲和576.9⌈s的脉冲周期的条件下,测量一个8脉冲的序列.

```
REM ***************************************************************************************************************
Public Sub MultiSumMarker()
result$ = Space$(200)
'---------------------------------------------- FSP默认设置----------------------------------------------------------
CALL SetupStatusReg           '设置状态寄存器
CALL IBWRT(analyzer%,"*RST") '重置仪表
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描模式
CALL IBWRT(analyzer%,"SYST:DISP:UPD ON") 'ON: 打开显示
                                          'OFF: 关闭显示
'---------------------------------------- 针对时域功率测量配置FSP ----------------------------------------
CALL IBWRT(analyzer%,"FREQ:CENT 935.2MHz;SPAN 0Hz") '频率设置
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:RLEV 10dBm") '设置参考电平为10 dB
CALL IBWRT(analyzer%,"INP:ATT 30 dB")          '设置输入衰减为30dB
CALL IBWRT(analyzer%,"BAND:RES 1MHz;VID 3MHz") '带宽设置
CALL IBWRT(analyzer%,"DET RMS")                '选择RMS探测器
CALL IBWRT(analyzer%,"TRIG:SOUR VID")          '触发源:视频
CALL IBWRT(analyzer%,"TRIG:LEV:VID 50 PCT")   '触发开始: 50%
CALL IBWRT(analyzer%,"SWE:TIME 50ms")         '扫描时间ε 1 帧
'---------------------------------------- 执行测量并查询结果----------------------------------------------------------
CALL IBWRT(analyzer%,"INIT;*WAI")  '执行同步扫描
                                   '查询结果:
cmd$ = "CALC:MARK:FUNC:MSUM? "
cmd$ = cmd$ + "50US,"              '首脉冲偏移
cmd$ = cmd$ + "450US,"             '时间测量
cmd$ = cmd$ + "576.9US,"           '脉冲周期
cmd$ = cmd$ + "8"                  '脉冲数量
CALL IBWRT(analyzer%,cmd$)
CALL IBRD(analyzer%, result$)       '读结果
Print result$
END SUB
REM ***************************************************************************************************************
```

多脉冲功率测量

多脉冲功率测量功能适合于测量以下参数的脉冲序列的功率:
.以不同时间间隔出现的脉冲
.序列所有脉冲的电平值都高于触发起始值,或者使用外触发信号.
这个功能对每个脉冲都需要一个触发事件.这意味着如果使用视频触发或者中频功率触发,则所有脉冲的电平都要高于触发起始值.
这样,这个功能就尤其适合于重新测量已经调整好的DUT,DUT的输出功率是处于指定的范围之内的.
这个测量针对相对实际测量时间最小的开销进行优化.

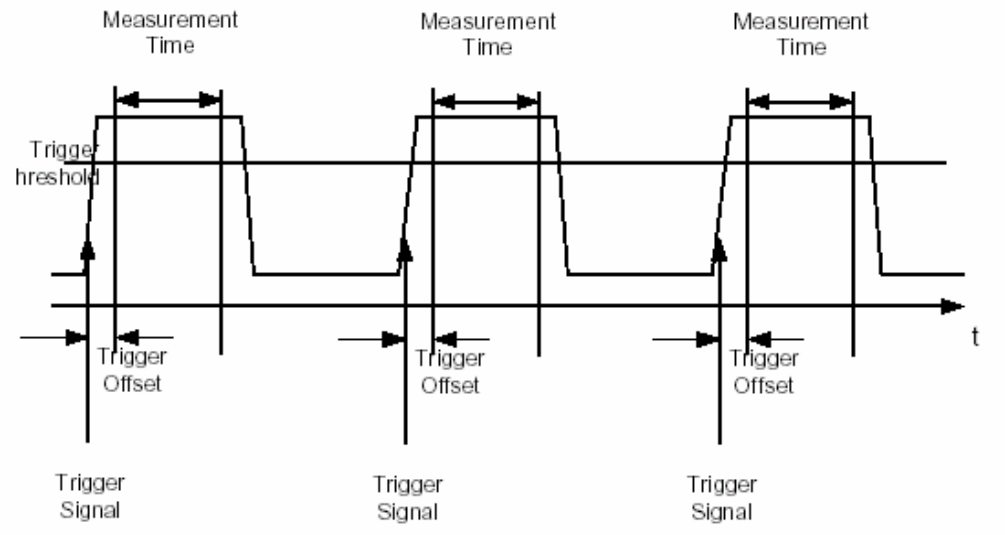

图 7-2信号特性和要测量的信号时间选择方式.

军方根功率和峰值功率都被测量,这主要取决于是选择RMS检测器还是峰值检测器.这个功能经常使用选定屏幕的轨迹1.
对这个测量,要设定下列参数:
. 分析仪频率
. 解析带宽
. 每个单脉冲的测量时间
. 触发源
. 触发起始值
. 触发偏移
. 功率测量的类型 (峰值, 平均)
. 要测量脉冲的数量
在测量期间,每个脉冲被映射为屏幕内的一个象素,也就是说,轨迹的任何改变只有在屏幕的左手边沿才能检测到.在显示关掉的情况下经常可以得到最大测量速度.
In the example below, a GSM pulse sequence of 8 pulses is measured with 5 ⌈s trigger offset, 434 ⌈s measurement time/pulse, video trigger with 50% trigger threshold, and peak detection:
在下面例子中,在5⌈s触发偏移,434⌈s测量时间每脉冲,50%触发起始值的视频触发,和峰值检测条件下,一个8脉冲的脉冲序列被测量.

```
REM ***************************************************************************************************************
Public Sub MultiBurstPower()
result$ = Space$(200)
'---------------------------------------------- FSP默认设置 ----------------------------------------------------
CALL SetupStatusReg                 '设置状态寄存器
CALL IBWRT(analyzer%,"*RST")         '重置仪器
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描模式
CALL IBWRT(analyzer%,"SYST:DISP:UPD OFF") 'OFF: 显示关闭
'---------------------------------------------- 执行测量并查询结果----------------------------------------------
cmd$ = "MPOW? "
cmd$ = cmd$ + "935.2 MHZ,"          '中心频率
cmd$ = cmd$ + "1MHZ,"               '解析带宽
cmd$ = cmd$ + "434US,"              '测量时间
cmd$ = cmd$ + "VID,"                '触发源
cmd$ = cmd$ + "50PCT,"             '触发起始值
cmd$ = cmd$ + "1US,"                '触发偏移, 必须> 125 ns
cmd$ = cmd$ + "PEAK,"               '峰值探测器
cmd$ = cmd$ + "8"                   '脉冲数量
CALL IBWRT(analyzer%, cmd$)
CALL IBRD(analyzer%, result$)       '读结果
Print result$
END SUB
REM ***************************************************************************************************************
```

使用频率列表的快速电平测量

FSP的一个典型任务就在一些频率点处测量功率,例如,在基频的整数倍处(谐波测量),或者在由移动无线标准所定义的频率处(例如,在一个GSM信号的载波频率周围± 200 kHz, ± 400 kHz等处的瞬时值所产生的频谱).在很多种情况下,为适配信道间隔并且符合动态范围的要求,对不同的频率点需要不同的电平和/或带宽设置.

尤其是对于这项应用,FSP提供一些遥控功能(在SENSe:LIST子系统中的可用命令),它们允许电平测量,这些测量是建立在不同仪器设置分配给不同频率的一个频率列表的基础之上的.不仅频率列表可以编程,同时执行的测量类型列表(PEAK,RMS,AVG)也是可选的.

下面例子描述在双带放大器上的谐波测量.总的来说,谐波电平随频率的增加而减少.这样,要提升测量灵敏度,参考电平比三次谐波还降低10dB.

使用到下列设置:

参考电平: 10.00 dBm 向上至二次谐波, 距三次谐波0 dBm

射频衰减: 20 dB

电子衰减: 0 dB

RBW: 1 MHz

VBW: 3 MHz

滤波器类型: NORMal

测量时间: 300 ⌠s

触发延时: 100⌠s

触发视频, 45 %

频率	类型
935.2 MHz	GSM 900 fundamental
1805.2 MHz	GSM 1800 fundamental
1870.4 MHz	GSM 900 2nd harmonic
2805.6 MHz	GSM 900 3rd harmonic
3610.4 MHz	GSM 1800 2nd harmonic
3740.8 MHz	GSM 900 4th harmonic
5815.6 MHz	GSM 1800 3rd Harmonic

The frequencies are selected in ascending order to minimize system-inherent waiting times resulting
from frequency changes.
At each frequency point the peak power and the rms power are measured. The peak power and the rms
power values are stored alternately in the results memory.

频率是按上升的次序选择的,这样是为了使系统固有的由频率变化所引起的等待时间最小化.在每个频率点检测峰值功率和rms功率. 峰值功率和rms功率在结果存储器中互换的存储.

```
REM ***************************************************************************************************************
Public Sub FrequencyList()
result$ = Space$(500)
'---------------------------------------------- FSP默认设置 ----------------------------------------------------
CALL SetupStatusReg                          '配置状态寄存器
CALL IBWRT(analyzer%,"*RST")                 '重置仪表
CALL IBWRT(analyzer%,"INIT:CONT OFF")        '单扫描模式
CALL IBWRT(analyzer%,"SYST:DISP:UPD OFF") '显示关闭
'--------------------------------------针对频率列表上的功率测量配置FSP ----------------------------------
Call IBWRT(analyzer%, "TRIG:LEV:VID 45PCT") '视频触发起始值
Call IBWRT(analyzer%, "LIST:POWer:SET ON,ON,OFF,VID,POS,100us,0")
'---------------------------------------- 执行测量并查询结果----------------------------------------------------
cmd$ = "LIST:POWer? "
cmd$ = cmd$ + "935.2MHZ,10dBm,20dB,OFF,NORM,1MHz,3MHz,300us,0,"
cmd$ = cmd$ + "1805.2MHZ,10dBm,20dB,OFF,NORM,1MHz,3MHz,300us,0,"
cmd$ = cmd$ + "1870.4MHZ,10dBm,20dB,OFF,NORM,1MHz,3MHz,300us,0,"
cmd$ = cmd$ + "2805.6MHZ,0dBm,20dB,OFF,NORM,1MHz,3MHz,300us,0,"
cmd$ = cmd$ + "3610.4MHz,10dBm,20dB,OFF,NORM,1MHz,3MHz,300us,0,"
cmd$ = cmd$ + "3740.8MHz,0dBm,20dB,OFF,NORM,1MHz,3MHz,300us,0,"
cmd$ = cmd$ + "5815.6MHz,0dBm,20dB,OFF,NORM,1MHz,3MHz,300us,0"
Call IBWRT(analyzer%, cmd$)
Call IBRD(analyzer%, result$)
Print result$
END SUB
REM ***************************************************************************************************************
```

变换器的电平纠正 (变换器系数的定义)

在更复杂的测试系统中,为避免从DUT以外的其他源引入任何测量错误,就必须考虑测试设置的频率响应.FSP提供可能来定义一个由频率决定的衰减纠正因数(变换器因数).
在下面例子中,定义了一个带有下列参数的因数:
名字: Transtest
单位: dB
缩放比例: lin
注释: 模拟线缆纠正
频率电平
10 MHz 0 dB
100 MHz 3 dB
1 GHz 7 dB
3 GHz 10 dB
因数可以定义,也可以根据需要激活.

```
REM ***************************************************************************************************
Public Sub TransducerFactor()
'------------------------------------------ 定义变换器因数 ----------------------------------------------
CALL IBWRT(analyzer%,"CORR:TRAN:SEL 'TRANSTEST'")
                                  '定义 "Transtest"
                                  '变换器因数
CALL IBWRT(analyzer%,"CORR:TRAN:UNIT 'DB'") '单位 'dB'
CALL IBWRT(analyzer%,"CORR:TRAN:SCAL LIN") '线性频率轴
CALL IBWRT(analyzer%,"CORR:TRAN:COMM 'Simulated cable correction'")
cmd$ = "CORR:TRAN:DATA "                    '输入频率和电平
cmd$ = cmd$ + "10MHz, 0,"                   '无单位的电平值
cmd$ = cmd$ + "100MHz, 3,"
cmd$ = cmd$ + "1GHz, 7,"
cmd$ = cmd$ + "3GHz, 10"
CALL IBWRT(analyzer%,cmd$)                  '输入频率和电平值
'------------------------------------------------ 激活变换器----------------------------------------------
CALL IBWRT(analyzer%,"CORR:TRAN:STAT ON") '激活变换器因数
END SUB
REM ***************************************************************************************************
```

读轨迹数据

在下面例子中,在默认设置下一起记录的轨迹数据由仪器中读出并以列表形式显示于屏幕上.读出是以二进制格式和ASCII格式读出的,步长分别是大于0和等于0.
在二进制,消息头被赋值为长度标识值并被用于计算x轴的值.
在ASCII格式下,只输出电平值列表.
二进制数值经过三个步骤读出:

1. 读出长度标识值的位数值
2. 读出长度标识值
3. 读出轨迹数据

由于头和数据在二进制数据方面的数据类型不同,这个步骤对于只支持类似数据类型(矩阵)的编程语言来说就是必要的.(比如Visual Basic)

***注意:** 轨迹矩阵被分成多维,以这种方式它们可以容纳FSP的轨迹点(501点).*

```
REM ***************************************************************************************************************
Public Sub ReadTrace()
'---------------------------------------- 定义变量 ----------------------------------------------------------------
Dim traceData(1250) As Single '浮点二进制数据的缓冲
Dim digits As Byte             '长度信息数值的位数
Dim traceBytes As Integer      '以字节表示的轨迹数据的长度
Dim traceValues As Integer    ' 缓冲中的数值数
asciiResult$ = Space$(25000)  ' ASCII轨迹数据的缓冲
result$ = Space$(100)         '简单结果的缓冲区
startFreq$ = Space$(100)      '起始频率的缓冲区
span$ = Space$(100)           '步长的缓冲区
'---------------------------------------- FSP的默认设置 ----------------------------------------------------------
CALL SetupInstrument                    '默认设置
CALL IBWRT(analyzer%,"INIT:CONT OFF") '单扫描
CALL IBWRT(analyzer%,"INIT;*WAI")      '执行同步扫描
'------------------------------------ 定义读出的步长 ----------------------------------------------------------
Call ibwrt(analyzer%,"FREQ:STARt?") '读出起始频率
Call ibrd(analyzer%,startFreq$)
startFreq = Val(startFreq$)
Call ibwrt(analyzer%,"FREQ:SPAN?") '读出步长
Call ibrd(analyzer%,span$)
span = Val(span$)
```

```
'---------------------------------- 以二进制格式读出 ----------------------------------------------------------
Call ibwrt(analyzer%, "FORMAT REAL,32") '选择二进制格式
Call ibwrt(analyzer%, "TRAC1? TRACE1")  '读出轨迹1
Call ilrd(analyzer%, result$, 2)        '读出并存储
digits = Val(Mid$(result$, 2, 1))       '长度信息的数字位数
result$ = Space$(100)                   '重新初始化缓冲区
Call ilrd(analyzer%, result$, digits)   '读出
traceBytes = Val(Left$(result$, digits)) ' 并存储长度信息
Call ibrd32(analyzer%, traceData(0), traceBytes) '将轨迹数据读入缓冲区
Call ilrd(analyzer%, result$, 1) 'Read the terminator <NL>
'-------------------------------- 以频率/电平值对的形式读出二进制数据----------------------------------
traceValues = traceBytes/4    '单精度 = 4 bytes
stepsize = span/traceValues   '计算频率步长宽度
For i = 0 To traceValues – 1
Print "Value["; i; "] = "; startFreq+stepsize*i; ", "; traceData(i)
Next i
'------------------------------------ 时域默认设置----------------------------------------------------------
Call ibwrt(analyzer%,"FREQ:SPAN 0Hz") '切换至时域
CALL IBWRT(analyzer%,"INIT;*WAI")     '执行同步扫描
'---------------------------------- 以ASCII格式读出----------------------------------------------------------
Call ibwrt(analyzer%,"FORMAT ASCII") '选择ASCII格式
CALL ibwrt(analyzer%,"TRAC1? TRACE1") '读出轨迹1
CALL ibrd(analyzer%, asciiResult$)
Print "Contents of Trace1: ",asciiResult$ '输出
END SUB
REM ***********************************************************************************************
```

测量信号的幅度和相位

(I/Q数据获得)

由于FSP的内部结构,除了信号的功率值以外,FSP还能够测量并输出信号的幅度和相位.这使更深入的分析成为可能.(FFT,解调等等).

图 7-3 显示分析仪的从中频到处理器的硬件.中频滤波器使频谱分析仪的解析滤波器,其带宽由300kHz到10MHz可选.

模/数转换器以32MHz的采样率对中频信号(20.4MHz)进行采样.数字信号被向下转化为更复杂的基带,经过低通滤波的,而且采样也被减少,也就是说,通过将原来的采样率被2除,现在输出的采样率被设置在15.625 kHz和32 MHz.这避免了在窄的带宽中过采样,这样,就节省了处理时间和增加了最大记录时间.

I/Q被写入单独的存储器,每个存储器由128k字.存储器使硬件触发的.

数据获得硬件部分

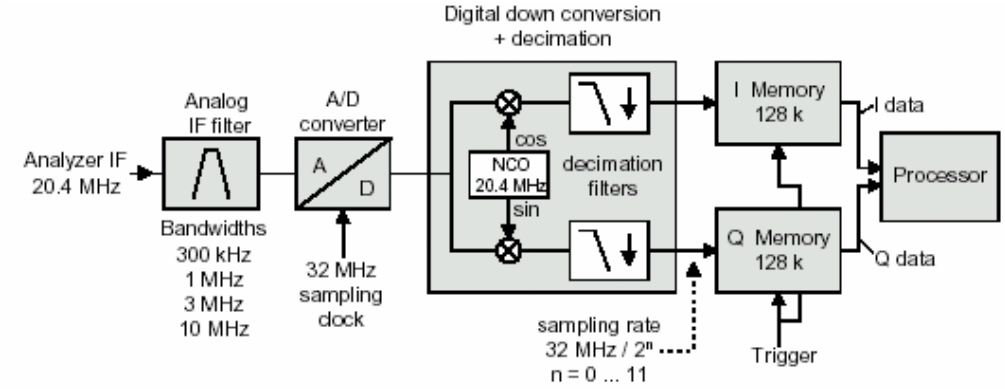

Fig. 7-3 说明分析仪信号处理的方块图

根据选定的取样率,在这个测量中可以得到下列最大的带宽:

取样率	最大带宽	备注
32 MHz	9.6 MHz	
16 MHz	7.72 MHz	
8 MHz	4.8 MHz	由于反卷积滤波器的特性,指定带宽之外的信号可被反卷积到有用带宽之内.
4 MHz	2.8 MHz	
2 MHz	1.6 MHz	
1 MHz	800 kHz	
500 kHz	400 kHz	
250 kHz	200 kHz	
125 kHz	100 kHz	
62.5 kHz	50 kHz	
31.25 kHz	25 kHz	
15.625 kHz	12.5 kHz	

由于仪器的采样概念(21.4 MHz中频, 32 MHz采样率),镜象频率使受模拟10MHz滤波器作用而带限的.

对于一个受限于10MHz带宽(在中心频率+ 5 MHz之上)的输入信号,可得到一个高于输入信号800kHz的镜象频率.

以MHz维单位的镜像频率可按如下计算:

$f_{image}=2 \cdot (f_{center}+5.4MHz)- f_{signal}$

这里

f_{image} = 以MHz计的镜像频率

f_{center} = 以MHz计的中心频率

f_{signal} = 以MHz计的测得信号频率

要得到正确的测量,射频输入信号必须是带限的.距离中心频率超过5.4MHz的信号被映射到10MHz滤波器的通带中.可用模拟滤波器(带宽大于等于300kHz)来提供附加的对已测信号的带限作用.

下面例子显示如何以预先定义的采样率收集数据并由I/Q存储器中读出来.

数据是以参考分析仪输入的形式,以电压值的形式进行输出.数据可以以二进制或ASCII的格式读入.

在二进制格式下,在信息头加载的长度信息被赋值并用于计算x轴的值.

在ASCII格式,只有一个电压列表被输出.

二进制数据以三个步骤读入:

1.加载长度信息的数字的数目被读入.

2.长度信息本身被读入.

3. 轨迹数据被读入.

对于象Visual Basic这样的只支持相同数据类型(矩阵)的结构的编程语言,这个步骤是必要的,然而二进制数据格式在头部和数据部分使用不同的数据类型.

注意: *已测的矩阵被分成多维,这样它们就可以容纳FSP的I/Q数据(2 * 128 k * 4 byte).*

```
REM ***************************************************************************************************
Public Sub ReadIQData()
'--------- 定义变量-----------------------------------------------
Dim IData(131072) As Single '浮点缓冲区I数据 (= 32*1024 byte)
Dim QData(131072) As Single ' 浮点缓冲区Q数据 (= 128*1024 byte)
Dim digits As Byte ' 信息长度数字的位数
Dim IQBytes As Long '以字节计的轨迹数据的长度
Dim IQValues As Long '以字节计的已测数值的数目
asciiResult$ = Space$(6553600) 'ASCII I/Q数据的缓冲区 (= 25*2*1024 byte)
'--------- FSP默认设置 ----------------------------------------
CALL SetupInstrument '默认设置
CALL IBWRT(analyzer%,"TRAC:IQ:STAT ON") '激活I/Q数据获得模式;必须在
'TRAC:IQ:SET!之前完成,选择测试点的最大数目(= 128 * 1024 – 512)在10 MHz RBW,
                '32 MHz采样率,自由触发,正触发边沿和0 s触发延时.
CALL IBWRT(analyzer%,"TRAC:IQ:SET NORM,10MHz,32MHz,IMM,POS,0,130560")
'--------- 以二进制格式读数据-------------------------------------
Call ibwrt(analyzer%, "FORMAT REAL,32")    '选择二进制格式
Call ibwrt(analyzer%, "TRAC:IQ:DATA?") '测量并读I/Q数据
Call ilrd(analyzer%, result$, 2) '读并存储长度信息数字的位数
digits = Val(Mid$(result$, 2, 1))
result$ = Space$(100) '重新初始化缓冲区
Call ilrd(analyzer%, result$, digits) '读并存储
IQBytes = Val(Left$(result$, digits)) '长度信息
IQBytes = IQBytes / 2                  '每缓冲区字节数目而等分
Call ibrd32(analyzer%, IData(0), IQBytes) 'I数据读入缓冲区
Call ibrd32(analyzer%, QData(0), IQBytes) 'Q数据读入缓冲区
Call ilrd(analyzer%, result$, 1) 'Read delimiter <NL>
'--------- 以频率/电平对的形式输出二进制数据----------------
IQValues = IQBytes/4 '单精度= 4 bytes
For i = 0 To IQValues – 1
Print "I-Value["; i; "] = "; IData(i)
Print "Q-Value["; i; "] = "; QData(i)
Next i
'--------- 以ASCII格式读数据-------------------------------------
Call ibwrt(analyzer%,"FORMAT ASCII") '选择ASCII格式
Call ibwrt(analyzer%, "TRAC:IQ:DATA?") '重测并读I/Q数据
CALL ibrd(analyzer%, asciiResult$)
CALL IBWRT(analyzer%,"TRAC:IQ:STAT OFF") '如果不再执行测量关闭I/Q数据获得模
                                         '式
END SUB
REM ***************************************************************************************************
```

I/Q数据取均值

对于I/Q测量,FSP也由均值功能,也就是说,I/Q可以在几次测量的基础上取均值.这是受下列条件支配的:

1. 对于数据测量,一个外部触发信号必须可用,而且相对于待测信号,触发信号必须是相位锁定的.
2. 对DUT和FSP来说,必须使用相同参考信号的电平.
3. 采样率必须是32MHz,因为只有用这样的采样率,测量才能于触发信号同步执行.

如果以上的条件都满足了,在连续的测试重,不会发生相位滑移.
相位滑移将会使测得的均值无效,这样,在极端的情形下,会得到一个0值.
不取均值的数据测量的仪器默认设置将被迫改为如下:

```
'---------------------------------------- FSP的默认设置 ----------------------------------------------------
CALL SetupInstrument                        '默认设置
CALL IBWRT(analyzer%,"TRAC:IQ:STAT ON") '激活I/Q数据获得模式;这必须在
                                            'TRAC:IQ:SET!之前完成
                  '在10 MHz RBW选择测试点的最大数目(= 128 * 1024 – 512),
                  '32 MHz采样率,外部触发,正触发边沿和0 s 的触发延时.
CALL IBWRT(analyzer%,"TRAC:IQ:SET NORM,10MHz,32MHz,EXT,POS,0,130560")
CALL IBWRT(analyzer%,"TRAC:IQ:AVER ON")      '打开I/Q均值
CALL IBWRT(analyzer%,"TRAC:IQ:AVER:COUN 10") '设置10次测试操作
'--------------------------------------- 以二进制格式读数据 ------------------------------------------------
...
```

设备设置的保存和载入

保存仪器的设置

在下面例子中,测定要存储的设置/测量数据;只存储硬件设置.维完整性考虑,其余设置的选择命令用状态OFF表示.

```
REM ************************************************************************************************
Public Sub StoreSettings()
'这个子程序选择要保存的设置,并且在D:\USER\DATA目录下创建数组"TEST1".它使用默认
'设置,并且在保存这个设置之后重置仪器.
'---------------------------------------- FSP默认设置 ----------------------------------------------------
Call SetupInstrument
CALL IBWRT(analyzer%,"INIT:CONT OFF")    '单扫描
CALL IBWRT(analyzer%,"INIT;*WAI")        '执行同步扫描
'------------------------------------ 选择要存储的项 ----------------------------------------------------
CALL IBWRT(analyzer%,"MMEM:SEL:HWS ON") '保存硬件设置
CALL IBWRT(analyzer%,"MMEM:SEL:TRAC OFF") '不保存轨迹
CALL IBWRT(analyzer%,"MMEM:SEL:LIN:ALL OFF") '只保存激活的极限线
'------------------------------------ 定义注释 ----------------------------------------------------
CALL IBWRT(analyzer%,"MMEM:COMM 'Test Setup'")
'--------- 保存选定项 ----------------------------------------
CALL IBWRT(analyzer%,"MMEM:STOR:STAT 1,'D:\USER\DATA\TEST1'")
'------------------------------------------ 重置仪器 ----------------------------------------------------
CALL IBWRT(analyzer%,"*RST")
END SUB
REM ************************************************************************************************
```

载入仪表设置

在下面例子中,在D:\USER\DATA目录下保存的数组"TEST1"被载入到仪表中.

```
REM ***************************************************************************************************
Public Sub LoadSettings()
'这个子程序在目录D:\USER\DATA中载入数组"TEST1".
'---------------------------------- 状态寄存器的默认设置 ----------------------------------------------
Call SetupStatusReg                          '配置状态寄存器
'--------------------------------------- 载入数组 ------------------------------------------------------
CALL IBWRT(analyzer%,"MMEM:LOAD:STAT 1,'D:\USER\DATA\TEST1'")
'--------------------------------使用载入的数组起始测量-----------------------------------------------
CALL IBWRT(analyzer%,"DISP:TRAC1:MODE WRITE") '设置轨迹为Clr/Write
CALL IBWRT(analyzer%,"INIT;*WAI")                '起始扫描
END SUB
REM ***************************************************************************************************
```

设置起始记忆的数组

在下面例子中,分析仪首先重置.随后,存储在D:\USER\DATA目录下的数组TEST1为STARTUP RECALL函数选中,也就是说,数组针对每个*RST, PRESET和每个仪表起始而设置.作为说明, *RST命令被再执行一次.

```
REM ***************************************************************************************************
Public Sub StartupRecallSettings()
'---------------------------------- 重置 FSP ----------------------------------------------------------
CALL IBWRT(analyzer%,"*RST")
'---------------------------------- 状态寄存器的默认设置 ----------------------------------------------
Call SetupStatusReg                          '配置状态寄存器
'--------- Select startup recall data set---------------------------------
CALL IBWRT(analyzer%,"MMEM:LOAD:AUTO 1,'D:\USER\DATA\TEST1'")
'--------- 激活起始记忆数组 ------------------------------
CALL IBWRT(analyzer%,"*RST")
END SUB
REM ***************************************************************************************************
```

读写文件

从仪表中读文件

再下面例子中,存储于D:\USER\DATA目录下的文件TEST1.SET由仪表中读出并存储于控制器中.

```
REM ***************************************************************************************************
Public Sub ReadFile()
'---------------------------------------- 生成变量 ----------------------------------------------------
Dim digits As Byte            '长度信息的数字的位数
Dim fileBytes As Long         '以字节计的轨迹数据的文件的长度
result$ = Space$(100)         '简单结果的缓冲区
'-------------------------------------- 状态寄存器的默认设置 ------------------------------------------
Call SetupStatusReg           '配置状态寄存器
'---------------------------------------- 读出文件 ----------------------------------------------------
Call ibwrt(analyzer%, "MMEM:DATA? 'D:\USER\DATA\TEST1.SET'")
                                          '选择文件
Call ilrd(analyzer%, result$, 2)        '读长度信息的数字的位数并存储
digits = Val(Mid$(result$, 2, 1))
Call ilrd(analyzer%, result$, digits)   '读并存储长度信息
fileBytes = Val(Left$(result$, digits))
FileBuffer$ = Space$(fileBytes)          '文件缓冲区
Call ilrd(analyzer%, FileBuffer, fileBytes) '将文件读入缓冲区
Call ilrd(analyzer%, result$, 1) 'Read terminator <NL>
'---------------------------------------- 将文件存储入控制器 ------------------------------------------
Open "TEST1.SET" For Output As #1
Print #1, FileBuffer; ' ; 在文件结尾避免线馈
Close #1
END SUB
REM ***************************************************************************************************
```

在仪表中创建文件

在下面例子中,在控制器中可用的文件TEST1.SET被存储在仪表中的
D:\USER\DATA\DUPLICAT.SET.

```
REM ***************************************************************************************************
Public Sub WriteFile()
'---------------------------------- 生成变量 ----------------------------------------------------------
FileBuffer$ = Space$(100000)       '文件缓冲区
Dim digits As Long                 '长度信息的数字的位数
Dim fileBytes As Long              '以字节计的文件的长度
fileSize$ = Space$(100)            '字符串形式的文件长度
result$ = Space$(100)               '简单结果的缓冲区
'---------------------------------- 状态寄存器的默认设置----------------------------------------------
Call SetupStatusReg                '配置状态寄存器
'---------------------------------- 准备有限长度数据块------------------------------------------------
fileBytes = FileLen("H:\work\vb\test1.set") '确定文件长度
fileSize$ = Str$(fileBytes)
digits = Len(fileSize$) – 1        '确定长度信息的数字的位数
fileSize$ = Right$(fileSize$, digits)
FileBuffer$ = "#" + Right$(Str$(digits), 1) + fileSize$
                                    '在文件缓冲区中存储长度信息
'---------------------------------- 从控制器中读文件--------------------------------------------------
Open "H:\work\vb\TEST1.SET" For Binary As #1
FileBuffer$ = FileBuffer$ + Left$(Input(fileBytes, #1), fileBytes)
Close #1
'---------------------------------- 写文件 ------------------------------------------------------------
-
Call ibwrt(analyzer%, "SYST:COMM:GPIB:RTER EOI") '在仪表上设置接受结束位
Call ibwrt(analyzer%, "MMEM:DATA 'D:\USER\DATA\DUPLICAT.SET'," +
FileBuffer$)                        '选择文件
END SUB
REM ***************************************************************************************************
```

设置并起始一个打印输出

下面例子中显示针对打印输出测量屏幕,如何配置输出格式和输出设备.
按下列步骤进行:

1. 设置打印输出所需要的测量
2. 查询可用的输出设备
3. 选择一个输出设备
4. 选择输出介面
5. 设置输出格式
6. 起始一个全程同步的打印输出

这里假设所需要的设置是100MHz –20dBm功率的信号,同样,所需的打印机是可用打印机的第六个.打印先交给选定的打印机来完成,然后再交给一个文件.

```
REM ************************************************************************************************************
Public Sub HCopy()
DIM Devices(100) as string          ’打印机名缓冲区
FOR i = 0 TO 49
Devices$(i) = Space$(50) ’给打印机名预分配缓冲区
NEXT i
’---------------------------------- FSP默认设置 ----------------------------------------------------------
CALL SetupStatusReg                         ’配置状态寄存器
CALL IBWRT(analyzer%,"*RST")                ’重置仪器
CALL IBWRT(analyzer%,"INIT:CONT OFF")       ’单扫描
CALL IBWRT(analyzer%,"SYST:DISP:UPD ON") ’显示打开
’---------------------------------------- 配置测量 ------------------------------------------------------
CALL IBWRT(analyzer%,"FREQ:CENT 100MHz;SPAN 10MHz") ’设置频率
CALL IBWRT(analyzer%,"DISP:WIND:TRAC:Y:RLEV -10dBm") ’参考电平
CALL IBWRT(analyzer%,"INIT;*WAI")                    ’执行测量
’---------------------------------------- 查询关于可用的输出设备 ----------------------------------------
CALL IBWRT(analyzer%,"SYST:COMM:PRIN:ENUM:FIRSt?") ’读出第一个输出设备,并

CALL IBRD(analyzer%,Devices$(0))
PRINT "Drucker 0: "+Devices$(0)                  ’指示名字
For i = 1 to 99
CALL IBWRT(analyzer%,"SYST:COMM:PRIN:ENUM:NEXT?") ’读出下一个打印机的
CALL IBRD(analyzer%,Devices$(i))                 ’ 名字
IF Left$(Devices$(i),2) = "’’" THEN GOTO SelectDevice ’再列表的末端终止
PRINT "Drucker"+Str$(i)+": " Devices$(i)          ’指出打印机的名字
NEXT i
```

选择设备:

```
'---------------------------------------- 选择设备,打印机语言和输出介面----------------------------------
CALL IBWRT(analyzer%,"SYST:COMM:PRIN:SEL "+ Devices(6))'选择#6打印机
8 CALL IBWRT(analyzer%,"HCOP:DEST 'SYST:COMM:PRIN'") '配置:
                                                     '"输出至
                                                     '打印机接口"
CALL IBWRT(analyzer%,"HCOP:DEV:LANG GDI")           '输出语言'GDI'
'---------------------------------------- 选择方向 (肖像/风景) 和彩色/黑白-------------------------------
CALL IBWRT(analyzer%,"HCOP:PAGE:ORI PORTrait")       '肖像
CALL IBWRT(analyzer%,"HCOP:DEV:COL OFF")             '黑和白
'---------------------------------------- 配置并起始打印输出-----------------------------------------------
CALL IBWRT (analyzer%,"HCOP:ITEM:ALL")                '选择全屏
'CALL IBWRT (analyzer%,"HCOP:ITEM:WIND1:TRAC:STAT ON") '二中择一:
                                                      '只有
'CALL IBWRT (analyzer%,"HCOP:ITEM:WIND2:TRAC:STAT ON") '在屏幕A或B中的轨
                                                       '迹
CALL IBWRT (analyzer%,"*CLS")                         '重置状态寄存器
CALL IBWRT (analyzer%,"HCOP:IMMediate;*OPC")          '起始打印输出
CALL WaitSRQ(boardID%,result%)                        '等待服务请求
IF (result% = 1) THEN CALL Srq                        '如果识别出SRQ =>
                                                      '赋值子程序
'----------------------------- 以WMF格式打印输出到文件中(BMP)格式 ---------------------------------
CALL IBWRT(analyzer%,"HCOP:DEST 'MMEM'")              '配置:
                                                     '"打印到文件"
CALL IBWRT(analyzer%,"HCOP:DEV:LANG WMF")            '文件格式WMF
'CALL IBWRT(analyzer%,"HCOP:DEV:LANG BMP")           '文件格式BMP
CALL IBWRT(analyzer%,"MMEM:NAME 'D:\USER\DATA\PRINT1.WMF'") '确定
                                                      '文件名
CALL IBWRT (analyzer%,"*CLS")                        '重置状态寄存器
CALL IBWRT (analyzer%,"HCOP:IMMediate;*OPC")         '起始打印输出
CALL WaitSRQ(boardID%,result%)                       '等待服务请求
IF (result% = 1) THEN CALL Srq                      '如果识别到SRQ =>
                                                    '赋值子程序
END SUB
REM ************************************************************************************************
```