

## 如何快速地合并 S 参数（上）

摘要：为了提高 S 参数合并的效率、减少繁琐的人为操作，芯和半导体 SnpExpert 提供了四种快速实现 S 参数合并的方法：1) 通过在软件界面手动配置实现 S 参数合并；2) 由 Python 脚本自动实现 S 参数合并；3) 通过导入 csv 文件自动实现 S 参数合并；4) 按命名规则自动实现 S 参数合并。本文先介绍前三种方法，然后进行结果的对比验证。通过本文的介绍，可知此三种方式均可成功实现 S 参数的快速合并，用户可根据实际应用场景选择相应的解决方案。

### 前言

在高速电路设计中，不仅需要考虑到单通道的损耗带来的影响，还需考虑到通道与通道间的串扰所带来的影响。而传统的矢量网络分析仪一般只能测得 4 个端口的 S 参数，尽管市场上已有多端口的网络分析仪，但价格偏贵。那么如何将 4 端口的 S 参数合并成多端口的 S 参数成为了一个急待解决的问题。以 8 个差分信号组成的串扰通道为例，通过 VNA 测得 8 个直通 (Thru)，56 个近端 (NEXT) 与 56 个远端 (FEXT) 的 4 端口的 S 参数，获得 4 端口 S 参数后，需要快速准确实现合并。针对上述问题，芯和半导体 SnpExpert 根据不同应用场景提供了四种实现合并的解决方案，本文将先介绍前三种解决方案，第四种解决方案将在以后的《如何快速地合并 S 参数（下）》中具体介绍。

## 合并 S 参数的方法

### 1. 通过软件界面手动配置实现 S 参数合并

在 SnpExpert 中，选择 S-Parameter Process 下的 Combine，弹出图 1 所示界面。首先，选择 Port Direction (Sequence 或者 Odd-Even)，即是合并后的 S 参数的 Port 顺序。其次，选择 Number of Diff Pairs，本示例中，可选择 N Diff Pair，然后在下方 Number of Ports 中输入 32，即合并后的 S 参数的端口数。接下来，在 Input 界面里，选择相应的 S 参数，在 Output Port Mapping 里配置与实际 VNA 测试中对应的 Port 顺序。最后点击 Combine 即可完成合并。

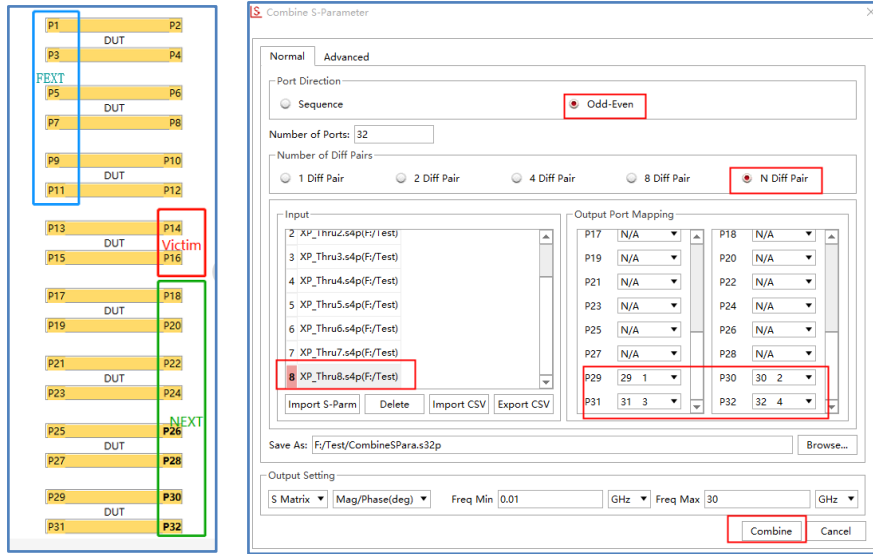


图 1 SnpExpert 中合并 S 参数配置界面

## 2. 由 Python 脚本自动实现 S 参数合并

在 SnpExpert 中，选择 Run Script。或者通过命令行后台调用 SnpExpert，无需打开界面的方式 Run Script。针对图 1 中串扰示例的设置，编辑 Python 脚本如下图 2 所示。其中关键是配置输入端口与输出端口顺序，也就是 vecCombineInfo 的信息配置。合并 S 参数命令解释如下：

- 1) vecCombineInfo: 表示端口配置信息，格式为: SnpID-OutputPortID-InputPortID;
- 2) freqMin/ freqMax: 表示输出 S 参数的最小和最大频率值，单位为 Hz，其默认值分别为 0 和 20e9;
- 3) portNum: 表示输出 S 参数的 port 数，默认为 4;
- 4) fileSavePath: 表示输出 S 参数路径;
- 5) strParaType: 表示输出 Matrix 的类型，支持 S/Y/Z，默认为 S;
- 6) strFuncType: 表示输出 Matrix 的 format，支持 Mag/Phase(deg)、Real/Imaginary、dB/Phase(deg)，默认为 Mag/Phase(deg)；
- 7) blsOldEven: 表示输出 S 参数的 Port Direction 是否为 Odd-Even，默认为 true;

```

snpf = []
snpf.append("F:\\Test\\XP_Thru1.s4p") #SnpID 1
snpf.append("F:\\Test\\XP_Thru2.s4p") #SnpID 2
snpf.append("F:\\Test\\XP_Thru3.s4p") #SnpID 3
snpf.append("F:\\Test\\XP_Thru4.s4p") #SnpID 4
snpf.append("F:\\Test\\XP_Thru5.s4p") #SnpID 5
snpf.append("F:\\Test\\XP_Thru6.s4p") #SnpID 6
snpf.append("F:\\Test\\XP_Thru7.s4p") #SnpID 7
snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 8

snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 9
snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 10
snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 11

snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 12
snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 13
snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 14
snpf.append("F:\\Test\\XP_Thru8.s4p") #SnpID 15

snpIDList =snpf.LoadSnpFileList(snpf) #导入s参数 SnpID即是append的顺序

vecCombineInfo=[
    '1_1_1','1_2_2','1_3_3','1_4_4',
    '2_5_1','2_6_2','2_7_3','2_8_4',
    '3_9_1','1_10_2','1_11_3','1_12_4',
    '4_13_1','2_14_2','2_15_3','2_16_4',
    '5_17_1','1_18_2','1_19_3','1_20_4',
    '6_21_1','6_22_2','6_23_3','6_24_4',
    '7_25_1','7_26_2','7_27_3','7_28_4',
    '8_29_1','8_30_2','8_31_3','8_32_4',
    #Thru
    '9_1_1','9_2_2','9_14_3','9_16_4',
    '10_5_1','10_7_2','10_14_3','10_16_4',
    '11_9_1','11_11_2','11_14_3','11_16_4',
    #NEXT
    '12_18_1','12_20_2','12_14_3','12_16_4',
    '13_22_1','13_24_2','13_14_3','13_16_4',
    '14_26_1','14_28_2','14_14_3','14_16_4',
    '15_30_1','14_32_2','14_14_3','14_16_4',
    #NEXT
]
#vecCombineInfo输入格式为:SnpID-OutputPortID-InputPortID
snpf.CombineSParameter(vecCombineInfo,1e9,9.5e9,16,"F:\\Test\\Combine.s32p","S Matrix","Mag/Phase(deg)",False)
    
```

图 2 Python 脚本示例

## 3. 通过导入.csv 文件自动实现 S 参数合并

SnpExpert 里通过导入 csv 文件实现 S 参数合并，有两种方式：用户可以在界面导入 csv

文件（图 3）或者通过 Run Python 脚本调用 csv 文件进行操作（图 4）。

图 3 是针对图 1 串扰示例所配置的 csv 文件。csv 文件格式为：第一行是新合成 S 参数端口数，其余行为配置信息。配置信息每行 4 列：第一列为 S 参数路径，若与上一行路径相同，可省略；第二列为 S 参数名称；第三列为输入端口号，可缺省，若缺省，则默认输入端口为其上部的，距离其最近的非空行的端口号；第四列为输出端口号，可缺省，若缺省，输出端口依次累加，第三列和第四列从前往后依次对应。

完成 csv 文件配置后，打开 SnpExpert，选择 S-Parameter Process 下的 Combine，弹出下图所示界面。首先，选择 Port Direction 为 Odd-Even，即是合并后的 S 参数的 Port 顺序。其次，选择 Number of Diff Pairs 为 N Diff Pair。最后，选择 Import CSV 导入配置好的 csv 文件进行合并。在当前界面中，如果修改了配置信息，也支持导出新的 csv 文件，以便下次使用。

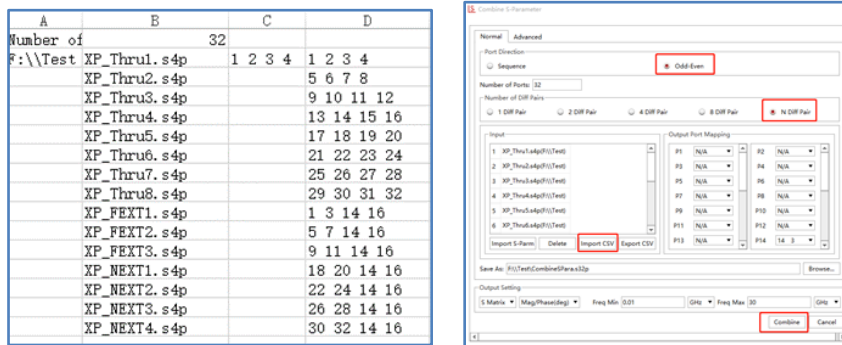


图 3 CSV 配置信息及由界面导入 CSV 文件实现合并

另外，通过 Python 脚本也可以调用 csv 配置文件，这样进一步简化了操作，避免错误的发生。具体格式如下图 4 所示：

```

pyConfig.EnableLog(True)           #日志
snp = SnpExpert()                  #创建对象
snp.CreateProject('combine')       #创建工程
csvFile=r'F:\Test\combine.csv'
snp.CombineSParameter(csvFile, 1e9, 9.5e9, "C:\Test\CascadeSpara_CSV_Python.s32p", "S Matrix",
"Mag/Phase(deg)", True)
    
```

图 4 通过 Run Python 脚本调用 CSV 文件

## 4. 结果对比验证

将前面三种方法合并的 s32p 文件结果跟未合并前的 s4p 的 S 参数进行对比。下图是比较结果：黑色是未合并前的 FEXT 及 NEXT；红色是通过导入 csv 文件合并后的 FEXT(Diff1, Diff8) 及 NEXT(Diff12, Diff8)；绿色是通过界面手动设置合并后的 FEXT(Diff1, Diff8) 及 NEXT(Diff12, Diff8)；蓝色是通过 Python 脚本合并后的 FEXT(Diff1, Diff8) 及 NEXT(Diff12, Diff8)。由图可知，三者的结果是完全重合的，说明以上三种方法都可以实现 S 参数的合并。

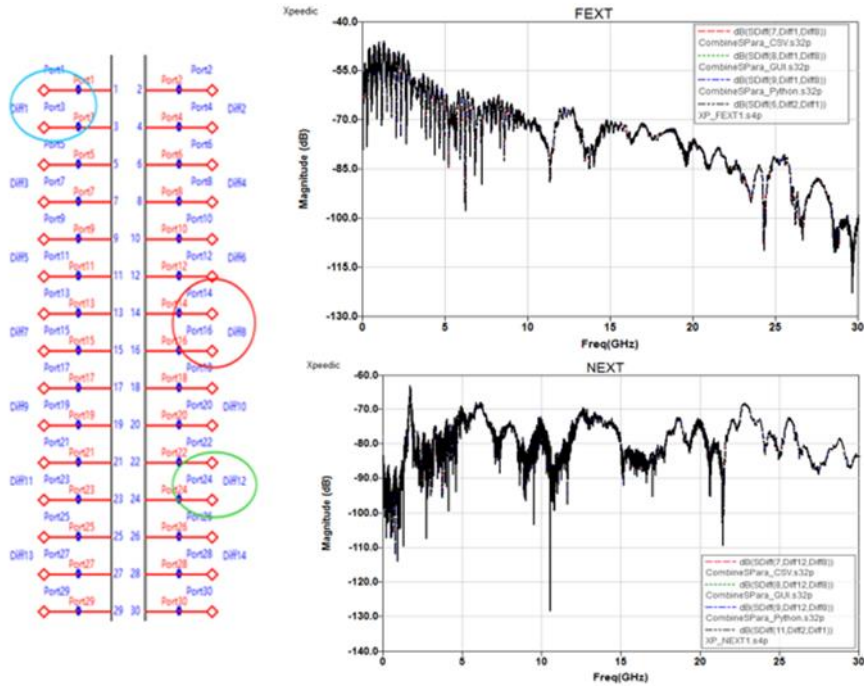


图 5 合并后的 S 参数与原始 FEXT、NEXT 的对比

## 总结

芯和半导体 SnpExpert 提供的这三种解决方案都可以成功实现 S 参数的合并，用户可根据实际应用场景选择相应的解决方案。对于端口数较少的 S 参数，可采用界面配置的方式，比较快捷、直观；对于端口数较多且想实现后续循环利用的 S 参数，用 csv 文件配置参数信息更高效。若采用由 Python 脚本调用 csv 文件的方式合并，一是端口配置比较直观，二是操作步骤少，可以降低出错的概率。

## 关于芯和半导体

芯和半导体是 EDA 软件、集成无源器件 IPD 和系统级封装领域的领先供应商。公司致力于为半导体芯片设计公司和系统厂商提供差异化的软件产品和芯片小型化解决方案，包括射频 IC 设计、模拟混合信号设计、系统级封装设计和高速数字系统设计等。这些产品和方案在 5G、智能手机、物联网、人工智能和数据中心等领域得到广泛应用。

芯和半导体凭借以客户需求驱动发展的理念，赢得了众多客户的青睐。随着公司自有知识产权的不断开发，芯和半导体已经成为中国集成电路自动化软件技术和微电子技术行业的标杆企业。

芯和半导体前身为芯禾科技，创建于 2010 年，企业总部位于上海浦东张江，并在美国硅谷、中国北京、深圳、苏州、成都、西安设有销售和技术支持中心。如欲了解更多详情，敬请访问 [www.xpeedic.cn](http://www.xpeedic.cn)。