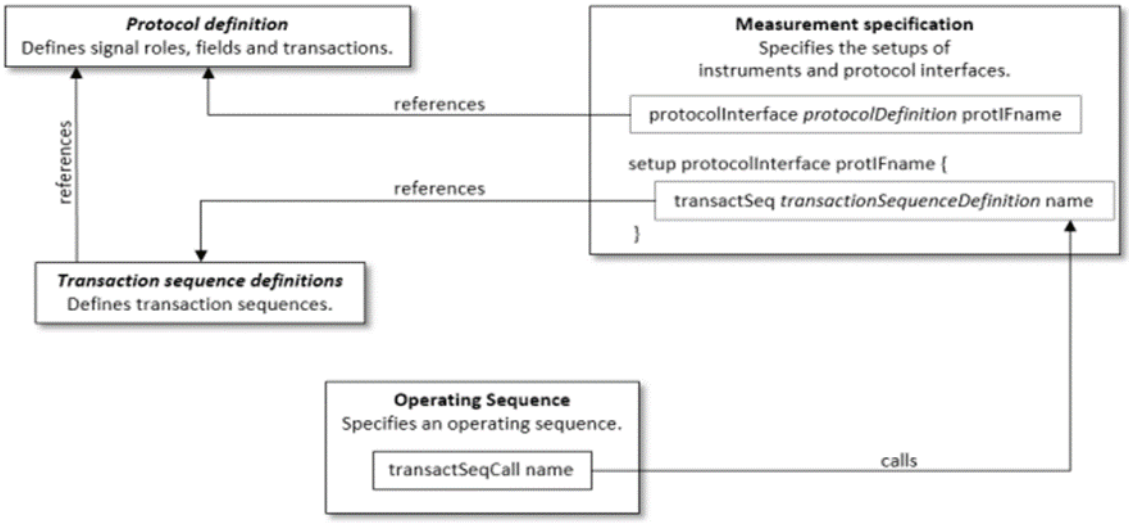
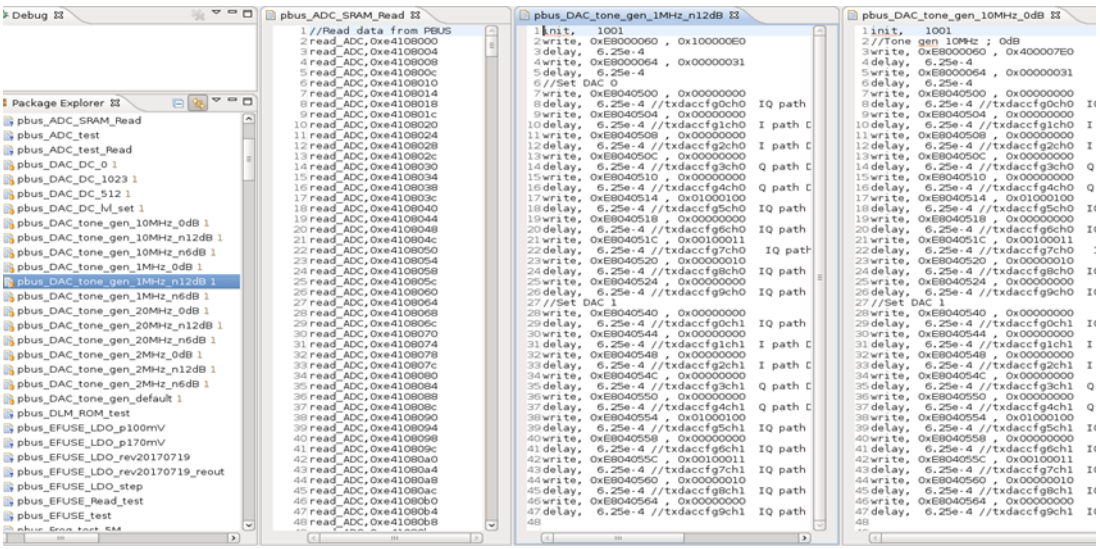
关于 PAAutoGen Tool

我们知道， V93000 测试平台推出 WaveScale RF 和 MX 测试方案，同时 SMT8 作为配套软件应用，推出了多项新功能，而功能亮点之一的 Protocol Aware（PA）更是被许多的客户所使用。

PA 是一种基于协议（诸如 I2C、PCIe）的测试，配合以 DS-API 方法，它可以大大简化编写测试流程中对应的测试向量，并能让用户灵活地控制各种读写指令。下图所示的各方框所代表的就是 PA 测试所需的相关文件。



在这些文件中，Transaction Sequence Definition(Trseq) 往往是最让人头疼的。一个 Trseq 文件中会包含大量的读写操作, SMT8 提供了两种方法去生成这个文件，一种就是按照该文件的语法格式人工“硬写”，参见下图。

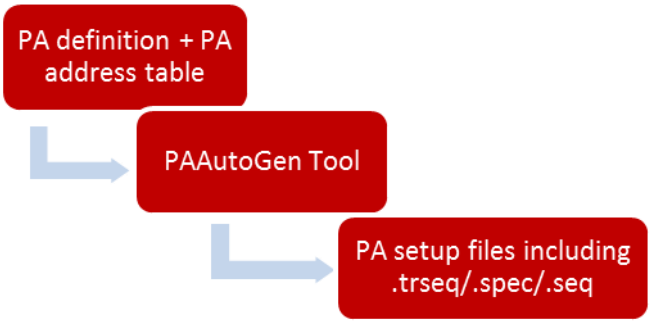


另外一种就是在测试项中使用 DS-API 方法去编写操作

```
tsdef.addTransaction("write", 14, 0x9C000, 14, 0x9C000);
tsdef.addTransaction("write", 13, 0x88881, 13, 0x88881);
tsdef.addTransaction("write",  2, 0xBC001,  2, 0xBC001);
tsdef.addTransaction("write", 15, 0x05800, 15, 0x05800);
tsdef.addTransaction("write", 75, 0x80000, 75, 0x80000);
tsdef.addWait(20e-6);
// myRegister.read(78 ,"CW78_V_RO_VBG_THADC"+ testband);//
tsdef.addTransaction("read", String.valueOf(78),arrOut.get

tsdef.addTransaction("write", 15, 0x01800, 15, 0x01800);
tsdef.addTransaction("write", 64, 0x16120, 64, 0x16120);
tsdef.addTransaction("write", 76, 0x00200, 76, 0x00200);
tsdef.addTransaction("write", 75, 0x60000, 75, 0x60000);
tsdef.addWait(120e-6);
tsdef.addTransaction("write", 64, 0x06120, 64, 0x06120);
//  myRegister.read(78 ,"CW78_V_RO_CAL_FINAL_THADC"+ testb
tsdef.addTransaction("read", String.valueOf(78),arrOut.get

tsdef.addTransaction("write", 15, 0x00800, 15, 0x00800);
tsdef.addTransaction("write", 15, 0x05800, 15, 0x05800);
tsdef.addTransaction("write", 75, 0x80000, 75, 0x80000);
tsdef.addWait(20e-6);
// myRegister.read(78 ,"CW78_V_RO_VBG_TTG_302"+ testband);
tsdef.addTransaction("read", String.valueOf(78),arrOut.get

tsdef.addTransaction("write", 15, 0x01800, 15, 0x01800);
tsdef.addTransaction("write", 46, 0xE0198, 46, 0xE0198);
tsdef.addTransaction("write", 56, 0x64010, 56, 0x64010);
tsdef.addTransaction("write", 76, 0x02000, 76, 0x02000);
tsdef.addTransaction("write", 75, 0x40000, 75, 0x40000);
tsdef.addWait(140e-6);
tsdef.addTransaction("write", 56, 0x60010, 56, 0x60010);
// myRegister.read(78 ,"CW78_V_RO_CAL_FINAL_TTG"+ testband
tsdef.addTransaction("read", String.valueOf(78),arrOut.get

tsdef.addTransaction("write", 15, 0x00800, 15, 0x00800);
tsdef.addTransaction("write", 46, 0x00198, 46, 0x00198);
//*********Purpose: POR RX_LDO RCAL (OrionP_Sleep_RXLDO_CA
tsdef.addTransaction("write", 367, 0x0FD13, 367, 0x0FD13);
tsdef.addTransaction("write", 368, 0x003FF, 368, 0x003FF);
tsdef.addTransaction("write", 369, 0x0037F, 369, 0x0037F);
tsdef.addTransaction("write", 15, 0x07800, 15, 0x07800);
tsdef.addTransaction("write", 75, 0x80000, 75, 0x80000);
tsdef.addWait(20e-6);
// myRegister.read(78 ,"CW78_V_RO_VBG_RXLDO_333"+ testband
tsdef.addTransaction("read", String.valueOf(78),arrOut.get

tsdef.addTransaction("write",  15, 0x01800,  15, 0x01800);
tsdef.addTransaction("write", 390, 0x00C90, 390, 0x00C90);
tsdef.addTransaction("write",  76, 0x01000,  76, 0x01000);
```

两种方法都涉及大批量数据的操作，劣势也是显而易见的。如果用户待测案例的协议部分含有

大量的读写操作需要人工输入，则对应的 PA 测试文件数量会非常庞大，极易出错且难以维护。爱德万测试中国研发中心的 TML 团队研发了一款自动生成 PA 测试文件的工具 PAAutoGen Tool。该工具只需用户输入 PA definition file 和标准格式的地址表文件（address table），即可一键生成 Spec、Opseq 和 Trseq 这些测试所需的文件。地址表（address table）则是该工具重要的输入，格式基本类似 Excel，方便易学，用户可以通过一个很简单的转换脚本实现自有地址表和标准地址表文件的无缝切换。



这样，通过自动化的方式轻松对应到 SMT8 的 PA 框架下，在贴合客户芯片测试流程中，完成各种向量的激励反馈和结果处理。同时，该工具还支持十六进制/十进制数据转换、DebugRead、DynamicWrite 等功能，极大地提高了用户使用 PA 开发测试程序的效率。

如对具体的使用案例及相关信息感兴趣，请随时联系我们咨询详情~