

【Saber 仿真技巧】C 语言的子程序调用

By : cumtzhangwang

Saber 是如何调用 C 语言子程序的，其过程基本分为两个步骤。

第一步，C 是编译性语言，因此要将写好的 C 语言子程序编译为动态链接库文件（DLL）之后才能被 Saber 调用。注意：编译好的 DLL 文件可放在工作目录下，以确保 Saber 能找到该文件。

第二步，如何让 Saber 知道要调用这个编译过的 C 语言子程序，这需要在 MAST 语言模型中声明并调用子程序函数，这样 Saber 仿真时就会将该子程序加载到内容中，在仿真运算中加以调用。

下面为具体实例：在 Saber 中调用具有求和功能的 C 语言子程序

1.编写具有求和功能的 C 语言子程序

```
__declspec(dllexport) void ADD  
( double *in,  
  int *nin,  
  int *ifl,  
  int *nifl,  
  double *out,  
  int *nout,  
  int *ofl,  
  int *nofl,  
  double *undef,  
  int *ier )
```

红框内的内容是Saber对C语言外部子程序的标准定义（除了函数名ADD，函数名要求大写）

```
{  
  double vin1,vin2,vsum;  
  vin1=in[0];  
  vin2=in[1];  
  vsum=vin1+vin2;  
  out[0]=vsum;  
}
```

绿框部分是函数主体，实现求和功能

2.将编写好的 C 语言子程序编译为动态链接库文件（.DLL）

用微软 VC 编译器很容易生成.DLL 文件（我用的是 VC++6.0），具体步骤可以百度或者谷歌，这里就不在介绍了。编译完成之后，生成的 DLL 文件（add.dll）要放在工作目录下。

3.利用 MAST 语言模型对 C 子程序进行调用

MAST 语言模型如下图所示：

```
element template add vin1 vin2 sum gnd  
electrical vin1, vin2, sum, gnd  
{  
  foreign add 调用C语言子程序  
  val v vi1,vi2,vosum  
  var i i1  
  values{  
    vi1=v(vin1)-v(gnd)  
    vi2=v(vin2)-v(gnd)  
    vosum=add(vi1,vi2)  
  }  
  equations{  
    i(sum->gnd) += i1  
    i1:vosum=v(sum)-v(gnd)  
  }  
}
```

MAST 语言具体语法可参考 Saber 自带帮助文档。

完成 MAST 模型之后可为模型建立符号并测试其功能是否匹配之前的输入输出函数关系定义。符号及测试电路如下图所示：

