



- 加入收藏
- 设为首页
- 联系站长

| 网站首页 | 文章中心 | 下载中心 | 图片中心 | 雁过留声 | 动易论坛 |

文章中心首页	新闻热点	自动控制	另类其他	电源技术	单片机	AVR单片机	PIC MSP430	CPLDFPGA	综合电子
ARM设计	电路板	技术文章	DSP数字信号处理	嵌入式系统	仪器仪表	网络文摘	模拟电路	数字电路	通信网络
显示技术	传感与控制	接口电路	存储器	测试测量	EMIEMC设计	家电音响	EDAIC设计	软件开发	芯片介绍
英文资料	消费类电子	电子制作	经验总结	无线通讯	半导体	行业观察	电子元器件	汽车电子	新品快递
计算机与外设	人物访谈	手把手学51单片机	手把手学PIC单片机						

» 您现在的位置: 杭州晶控电子有限公司 >> 文章中心 >> EDAIC设计 >> 正文

用SKILL语言对Concept HDL进行二次开发

【字体: 小 大】

用SKILL语言对Concept HDL进行二次开发

作者: 佚名 文章来源: 不详 点击数: 更新时间: 2009-8-27

收藏此页

摘 要: 本文简单介绍了在Cadence的Concept HDL平台上作二次开发的SKILL语言, 重点介绍了SKILL程序与Concept HDL平台的通信方式(MPS), 并给出了一个可运行的实例。

关键词: Skill语言; 消息传递系统; 应用程序接口; 回调函数

引言

Concept HDL工具是Cadence系列产品中画原理图的利器, 但针对一些特殊应用, Concept HDL还无法完成全部工作, 必须借助于SKILL 语言的开发。SKILL语言与通常接触到的计算机高级语言不太相同, 一是它的语法比较特别, 二是开发工具与运行平台之间的交互不易理解, 往往开发者经过一段时间的学习后, 熟悉了SKILL的语法, 也能编写SKILL程序, 但无法与Concept HDL工具衔接起来。Cadence 所提供的文档虽多, 但可操作性较差, 本篇文章是作者半年来进行开发的经验之作, 希望能与更多的人分享。



图 1 skill应用程序和Concept HDL的关系

SKILL简介

Cadence提供了对Concept HDL进行二次开发的SKILL语言, 它是一种基于通用人工智能语言——Lisp的交互式高级编程语言。SKILL语言支持一类C的语法, 大大降低了新手学习的难度, 同时高水平的编程者可以选择使用类Lisp语言的全部功能, 所以SKILL语言既可以用作最简单的工具语言, 也可以作为开发任何应用的、强大的编程语言。

SKILL可以与底层系统交互, 也提供了访问Cadence各个工具的丰富接口。它既可以在命令行下执行, 也可以在图形方式下运行。本文主要阐述在Concept HDL上如何使用SKILL提供的接口来完成Concept HDL不能直接完成的某些任务, 比如说从某个文本文件中加载器件信息等。

SKILL与Concept HDL的关系

针对Concept HDL, SKILL提供了专门的Concept-SKILL引擎, 即解释、运行Concept-SKILL程序的工具, 它支持访问Concept HDL环境及其中的对象, 如元器件、连线以及它们的属性等。但该引擎并没有与Concept HDL集成在一起, 它们之间利用MPS(Message Passing System)机制进行通信。MPS是一种远程过程调用机制, 一个MPS服务其实是一系列的MPS应用程序接口, 这些接口可作为进程的远程函数被调用, 提供这种被调用函数的一方称为server端; 相对应的, 调用这些函数的另一方称为client端。而Concept HDL与Concept-SKILL程序就是互为服务端和客户端的, 其关系如图1所示。

两者交互作用的实现

Concept HDL与Concept-SKILL相互通信主要依赖两种手段: 一种是通过Concept-SKILL提供的事件回调函数机制, 即在Concept HDL上发生某些事件, 可在程序中用特定的语句来捕捉, 并由相应的函数来处理; 另一种是充分利用Concept HDL提供的功能, 由其自身的命令集提供, 在程序中可根据需要, 向Concept HDL发命令, 控制其完成一系列相应操作。

在使用Concept HDL的过程中, 由于某些项目的特殊需要, 经常要完成这样一项功能, 即当在原理图中点击某元器件符号时, 和其相连的所有网格连线都显示高亮。这在Concept HDL中需要2步以上的步骤来完成, 为了使用起来更为方便, 本文在Concept HDL上开发了命令nets_extend来1步完成该功能。

```
;; file: ex_bodynets.il
;;-----;;
;;为鼠标单击的点选择一个器件符号, 并将其相连的网络连线合并入一个组中。创建的组中的所有元素自动高亮。
procedure(NearestComp (ConceptHandle, xy)
;; ConceptHandle参数是操纵Concept HDL的句柄。
;; xy参数 是鼠标单击的点坐标。
let( (command res opts msginfo group)
command=concat("SELECT BODIES", cnCreateMouse PressString(xy, 3), ";")
opts = cnSetRunOptions(?display cnOUT_VERBOSE, ?result cnOUT_VERBOSE)
res = cnSendCommand (ConceptHandle, command, ?runOptions opts)
;; 向Concept HDL发命令, 将鼠标单击的器件符号选入组
```

```

;; 向Concept HDL发出第一个命令,将鼠标单击的元件符号加入组。
group = cnGetGroupName(res)
command = strcat("INCLUDE ", group, " NETS;")
println(command)
opts = cnSetRunOptions(?display cnOUT_VERBOSE, ?result cnOUT_VERBOSE)
res= cnSendCommand (ConceptHandle,command, ?runOptions opts)
;; 向Concept HDL发出第二个命令,将与选定元件相连接的网络连线也加入同一组。
command = strcat("EXCLUDE ", group, " ", group, ";")
;; 将当前组释放。
cnSendCommand(ImportHandle, command)
); let
); proc
;; 在Concept HDL中每次单击鼠标,就重调用下面这个函数。
procedure( ex_nets (EvObject HandleSymbol)
    let( (res)
        NearestComp(symeval( HandleSymbol ),          cnGetMouseXY( EvObject ) )
cnRequestMousePress( ImportHandle, 'ex_nets, 'ImportHandle, "Pick a Pin22222")
    )
)
;; 一旦捕捉到Concept HDL关闭事件,就调用该函数,关闭MPS通信,并退出该应用。
procedure( ConceptExitHandler( Handle )
    printf("Concept exit detected\n")
    mpsClose( ExportHandle )
    exit( 0 )
)
;; MPS服务启动后,第一个调用的事件回调函数。
procedure( Requestcomm( Event, Command )
    cnRequestMousePress( ImportHandle, 'ex_nets, 'Import Handle,
        "Pick a Pin11111")
)
;; 每个MPS应用都需定义下面这个命令函数,该函数的结构是有固定模式的,在其中声明了一个命令,包括这个命令的名字、版本、及准备处理的
Concept HDL中的事件,和事件发生时要回调的函数名。
(defun cnCommands_nets_extendServer ()
    (list (list "nets_extend" ;; 命令名,此外可以有多个命令列表并列。
(list (list cnACTIVATE_EVENT)
'Requestcomm
    )
)
)
) ;; defun
;; 主程序开始
AppServiceName= "nets_extendServer"
AppServiceVersion = "1.0"
AppToolName=AppService Name
;; 调用mpsExport函数,通知Concept HDL本应用提供的远程函数。
ExportHandle = mpsExport( AppToolName, AppServiceName, AppServiceVersion,          '(cnCommands_nets_extendServer, Requestcomm),
?session cnmpsGetDefault SessionName() )
unless( cnmpsIsExported()
mpsSleep( ExportHandle, 1 )
) ;; unless
;; 取得操纵Concept HDL的句柄。
ImportHandle = cnmpsImport()
unless( ImportHandle
printf( "Could not import Concept mps Service\n" )
    exit( 1 )
)
;; 设置当Concept HDL退出时,调用conceptExitHandler函数。
ImportHandle->exitHandler = 'ConceptExitHandler
)

```

上面这个应用第一次被其中一个命令,如nets_extend命令触发时,从主程序开始运行,首先调用mpsExport函数,通知Concept HDL本应用提供的远程函数,其中最重要的是cnCommands_nets_extend Server函数,在其中列有每个命令应触发的函数名;然后,等待Concept HDL接收;接着调用cnmpsImport()取得操纵Concept HDL的句柄,它是应用程序向Concept HDL发命令的基础;最后设置当Concept HDL退出时需作的后处理工作函数。一个应用启动后,在关闭前都处于活动状态,再使用它的其它命令时,就直接由cnCommands_nets_extendServer函数中的列表找到命令对应的函数,调用即可。

通过cnCommands_nets_extendServer函数的定义,设置了在Concept HDL上启动nets_extend命令时应该调用的函数Requestcomm,在该函数中,当发生鼠标单击时,就调用函数ex_nets。在函数ex_nets中设置了这样一个无限循环: 首先将鼠标单击的点坐标作为参数,调用函数NearestComp,将单击的元件符号定义入group,并将与其相连的网络连线扩展入该group;然后再捕捉鼠标单击事件,再次调用自身。这样就可以连续使用该命令,以查看任一元件及其相关的连接,直到发出别的命令,该命令才结束。二次开发出来的命令可以挂接到Concept HDL的系统菜单上,使用起来和系统提供的其它功能一样方便。

编好的SKILL源程序必须放在Cadence安装目录下的tools\fet\skill-01.00\context目录中,不需要编译,可以直接解释运行。另外,需要将下面一段说明放入Cadence\share\ cdssetup\concept下的concepthdl.scr文件中。该文件在Concept HDL启动时进行加载,将每一个自定义命令都加载进去。

```

{evecho:decl.scr}
mpsdeclare eventecho
;; 命令名。
mpstool=eventEchoServer
;; 应用服务名。
mpsruncommand="cnskill -nograph evecho.il"
;; 如何加载SKILL程序。
mpstimeout=35
;; 设置服务超时不响应的时间。

```

这样,程序就可以在Concept HDL上运行了。打开Concept HDL,在它的Console Window中输入自定义的命令: nets_extend,然后点击已打开的某

页原理图中的一个元器件,就可以看到与其相连的网络连线也一起显示高亮了。
这个例子主要说明了二次开发的应用程序与Concept HDL的交互作用,实际上在二次开发的应用程序中可以作的工作很多,如读写文件、调用外部浏览器程序等。

结语
在Concept HDL上作二次开发的目的是完成Concept HDL上不提供或完成起来比较麻烦的功能。SKILL语言提供了读写文件、运算与判断的相关函数与相应控制,以及与Concept HDL相互通信的途径; Concept HDL的众多命令为最终在程序中完成绘图或将运算结果体现在绘图界面上提供了可能。在本文的设计实例中可以顺利地完成了在ConceptHDL上将特殊的网表自动加载为原理图,并随时去除或加载一个元器件的所有相关连线的功能。当然二次开发的内容远不止于此,它还包括完成更为复杂的界面和功能,如用到IPC编程、Java等,有兴趣的读者可以进行相关的研究。■

文章录入: admin 责任编辑: admin

- 上一篇文章: 三阶电荷泵锁相环系统级设计与仿真验证
- 下一篇文章: X结构成为主流

[【发表评论】](#) [【加入收藏】](#) [【告诉好友】](#) [【打印此文】](#) [【关闭窗口】](#)

最新热点	最新推荐	相关文章
<ul style="list-style-type: none">• 没有热点文章	<ul style="list-style-type: none">• 没有推荐文章	Zarlink针对付费电视推出QPS 利用STR-F6656设计34彩色电视 基于USB接口和智能卡的PKI客 用SPI实现dsPIC与ISD语音芯片 Skelmir 面向嵌入式设备的 C Hynix赔付东芝了旧怨 携手Sa 优百特采用Spansion MP4解决 OKI最新UV传感器ML8511无需广 NOKIA高通官司现转机 为巨额 Oki Electronics (HK) Ltd &

网友评论: (只显示最新10条。评论内容只代表网友观点,与本站立场无关!)

[| 设为首页](#) | [加入收藏](#) | [联系站长](#) | [友情链接](#) | [版权申明](#) | [网站公告](#) | [管理登录](#) |

杭州晶控电子有限公司 版权所有 COPYRIGHT2003——2009 HANGZHOU KinCony ELECTRONICS CO.,LTD All
rights reserved 站长: hificat