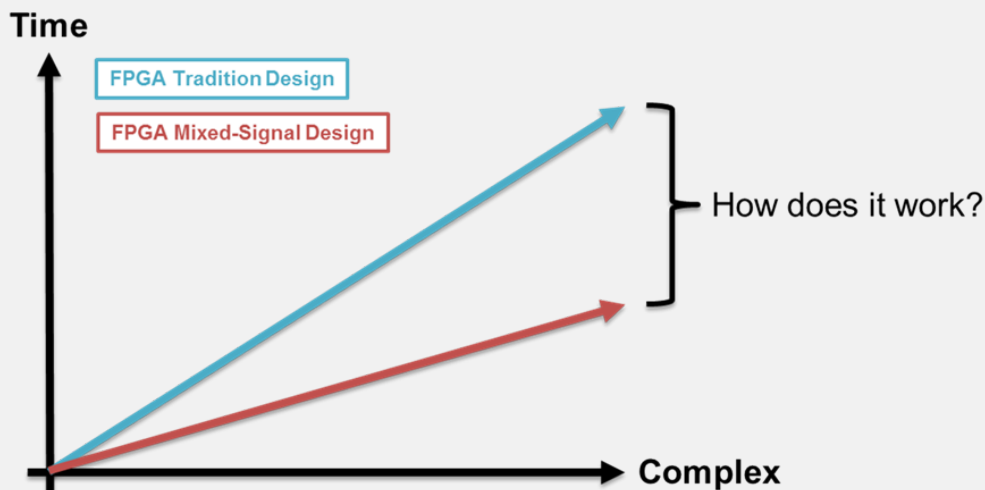


FPGA 混和設計與模擬

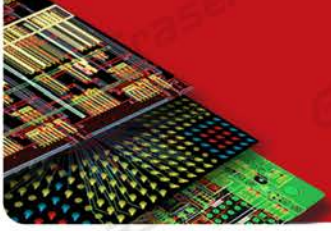
(上篇) 整合 PSpice 到 Incisive / Virtuoso 設計流程

FPGA 混和設計與模擬流程專為 FPGA 設計工程師量身訂做，以讓 FPGA 設計工程師能夠在短時間內了解 FPGA 與周邊電路 Co-Work 的狀況。為了減少重複性的動作與時間，上篇我們將探討如何將周邊電子電路的零件，導入到 Incisive / Virtuoso 混和電路設計流程。



(下篇) 加入 IBIS Mode 提升混和模擬準確度

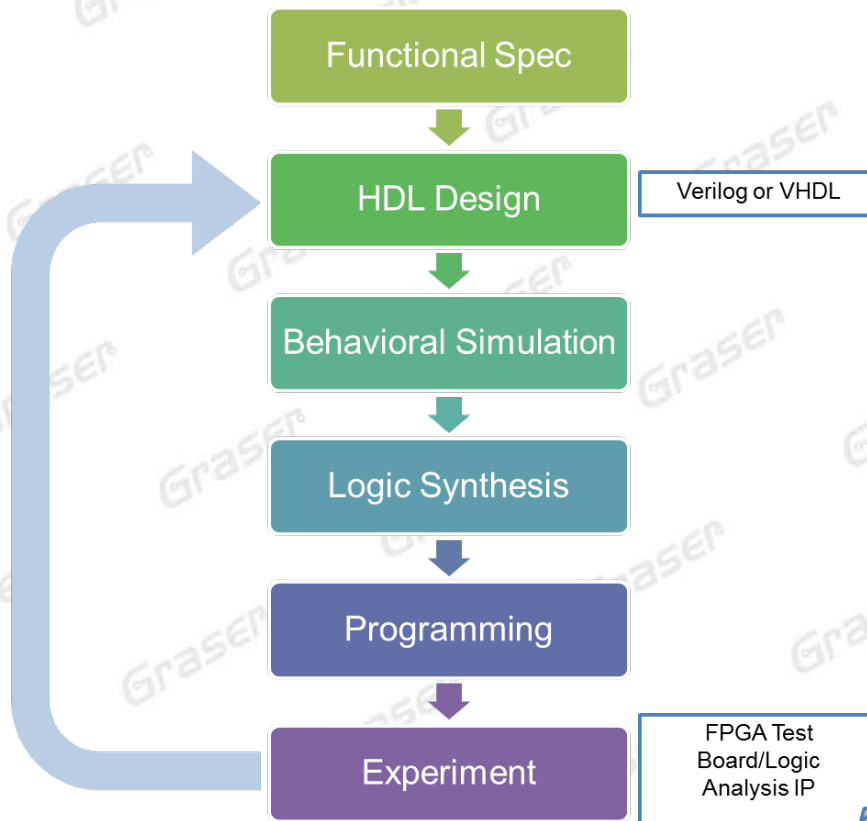
- Date : 2016 / 04 / 27
- Author : Paine
- Revision : 1.0
- Version : IES15.1, IC617, MMSIM15.1
- 備註:



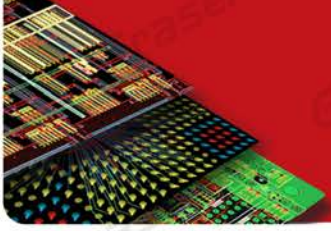
FPGA 混和設計與模擬流程

(上篇) 整合 PSpice 到 Incisive / Virtuoso 設計流程

以往 FPGA 設計工程師需將 HDL 燒入到 FPGA 晶片，才能了解 FPGA 與外部周邊電路的 Co-Work 狀況，假若 FPGA 與周邊電路發生功能上的問題，那麼則需要回到 HDL 設計階段進行修改，修改完畢則需再花費時間進行合成、實現與燒入的動作，假若設計複雜度提升，那麼上述流程進行的次數也增加許多。為了能夠減少重複性的動作與時間，我們將探討如何將周邊電子電路的零件，導入到 Incisive/Virtuoso 混和電路設計流程。

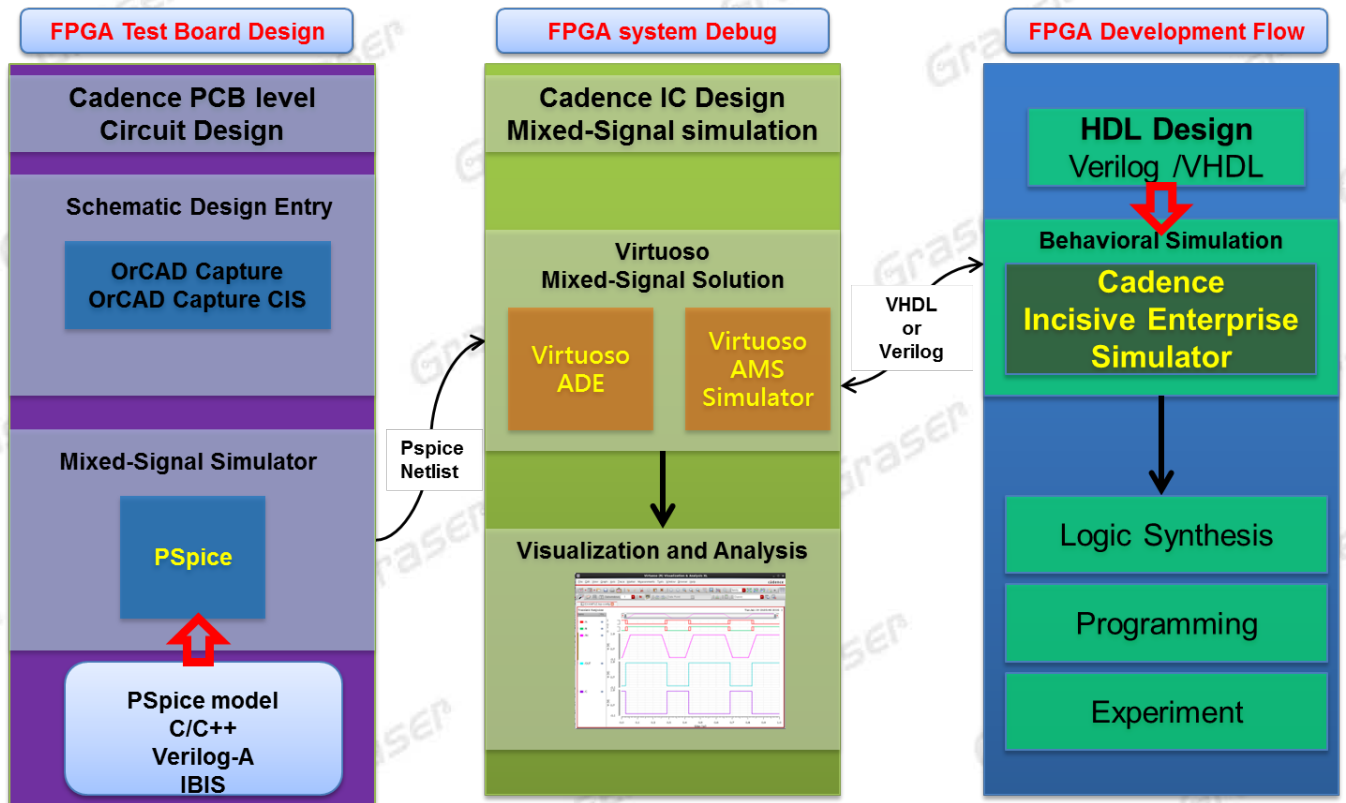


圖一、傳統 FPGA 設計流程

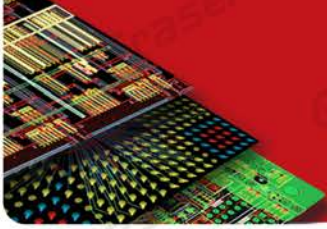


設計流程說明

為了能夠避免修改 HDL 而需花費過多時間在合成、實現與燒入的重複性動作，我們不妨可以嘗試將 Board-level 與 Chip-level 的電路進行整合、模擬與偵錯。在 Board-level 主要採用 OrCAD PSpice 工具軟體將電子電路的元件轉換成 PSpice 語言，接著將 Chip-level 的 HDL 語言與 Board-level 的 PSpice 語言整併在 AMS 設計平台並且進行模擬與偵錯；假若 Chip-level 與 Board-level 有功能上不匹配，那麼我們可以在 AMS 設計平台上直接修改 HDL 或者是 PSpice，無須再花費過多時間等待合成、實現與燒入的重複性動作。最後，若完成功能偵錯與修改，則再進行合成、實現與燒入的動作。

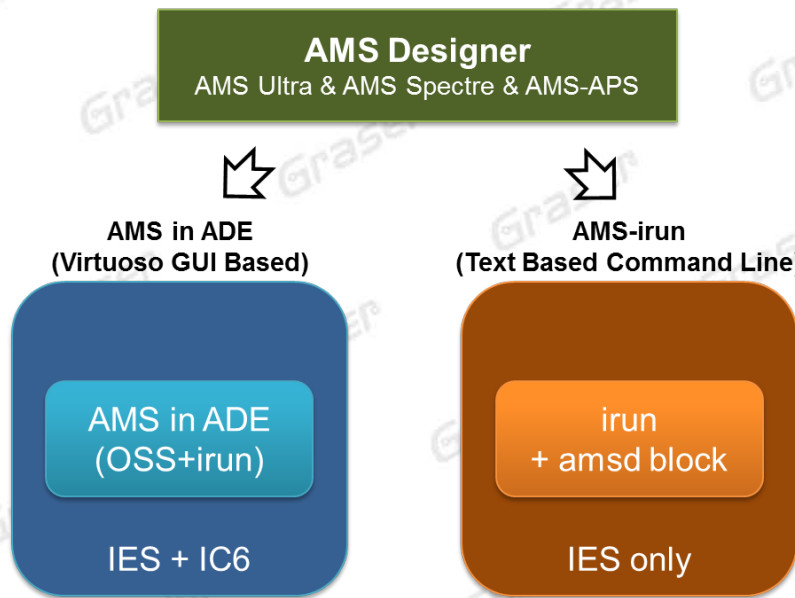


圖二、混合模擬設計流程

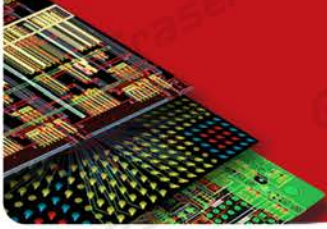


設計方法種類

當我們要與 PSpice 進行混和模擬，目前 Cadence 有提供兩種方法 AVUM(Virtuoso GUI Based)與 AIUM(Text Based Command Line)。AVUM(AMS Virtuoso Use Model)主要是採用 Virtuoso 圖形化介面再搭配 IES + MMSIM 的工具軟體來進行電路設計，若使用者時常使用圖形化電路設計或者是曾經使用過 Virtuoso 系列產品，那麼 AVUM 設計流程則適合該使用者；AIUM(AMS INCISIVE Use Model)主要是以文字介面再搭配 IES + MMSIM 的工具軟體進行電路設計，若使用者時常撰寫 HDL 語言以及經常使用 Shell 或者是 Makefile 建立設計流程，那麼 AIUM 設計流程則非常適合該使用者。

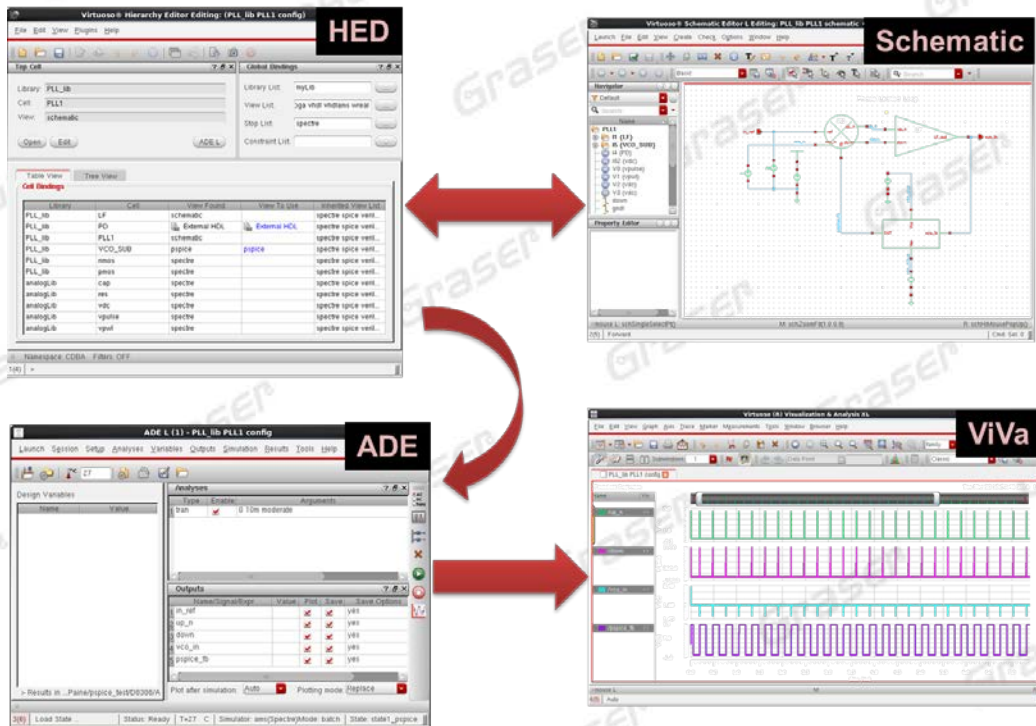


圖三、混和模擬設計方法

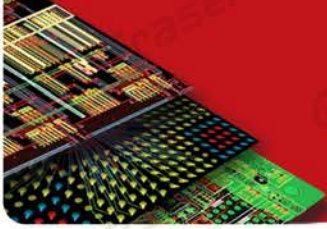


AVUM 設計流程 (IC6 + IES + MMSIM)

AVUM 設計流程主要以 Virtuoso 為 AMS 設計平台，在一開始設計電路時，主要以 Hierarchal Editing(HED)與 Virtuoso Schematic Editor(VSE)的介面，HED 主要是透過 hierarchal 的方式來快速切換每一個 cell 的 view，VSE 則是協助使用者進行圖形化電路設計。



圖四、AVUM 設計流程

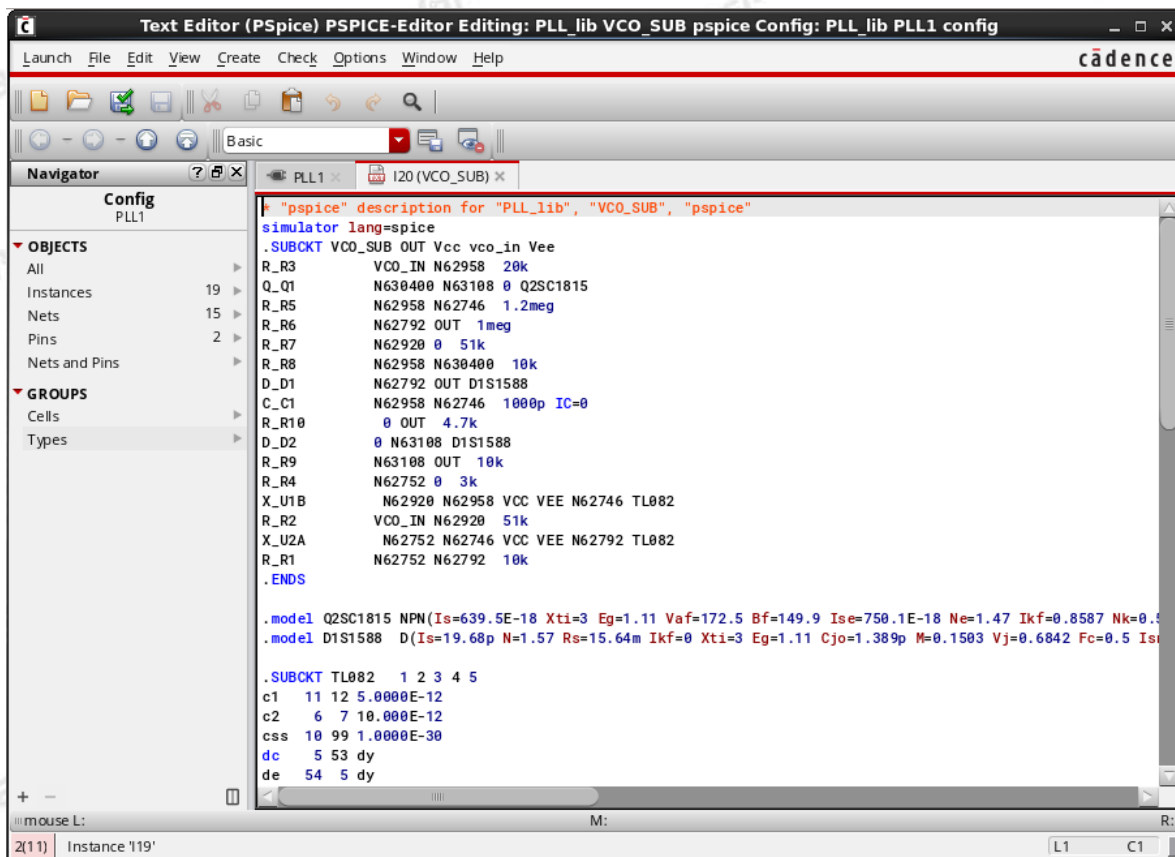


當我們要從導入 PSpice 到 VSE 時，我們則先需要建立 PSpice view 與 Symbol view。

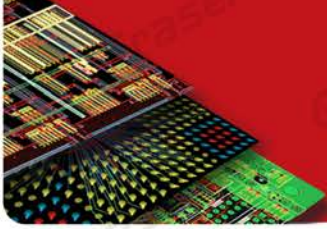
操作流程：Library Manager -> File -> New -> Cell View -> Type 點選 Pspice(圖五) -> 按 OK -> 貼上 PSpice Netlist(圖六)



圖五、PSpice Cell View 設定



圖六、PSpice Cell View 編輯視窗



當完成 PSpice 的 cell view 與 symbol 時，則可以依照我們的需求，開始建立起系統級的混和模擬電路(圖七)，完成之後，則使用 HED 檢查每一個 cell view 是否符合設計者的需求。接著，在 HED 的介面上開啟 ADE(Analog Design Environment)的介面開始設定模擬環境(圖八)，除了設定 Analog Solver 與 Outputs 之外，以下三點(1)(2)(3)需再多加留意。模擬完成之後，則會自動開啟 Viva 協助使用者偵錯(圖九)，假若使用者想要改變 Debugging 工具軟體為 SimVision，則需切換到 Interactive(debugger) 選項(4)。

ADE 操作注意事項：

1. 選擇模擬器：

Setup > Simulator / Directory / Host > 在 Simulator 項目選擇”ams”

2. 選擇 PSpice Model：

Setup > Model libraries > 載入 design.pspice

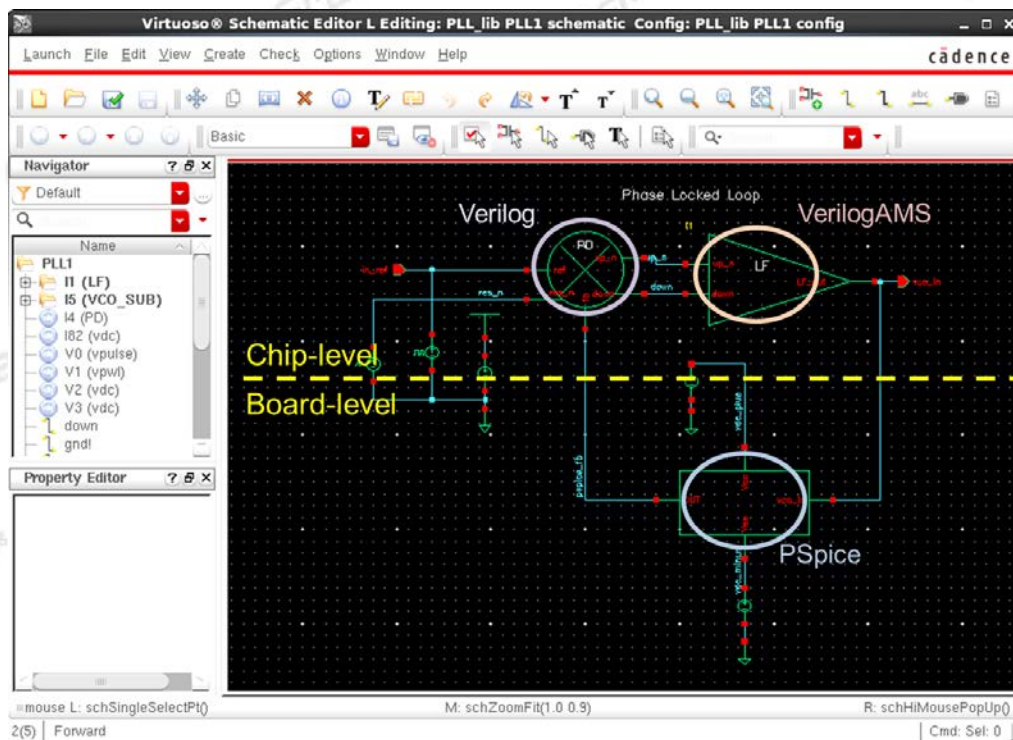
3. 選擇 RUN Mode：

Simulation > Netlist and Run Options > 在 NETLIST AND RUN MODE 項目選擇 OSS-based netlister with irun

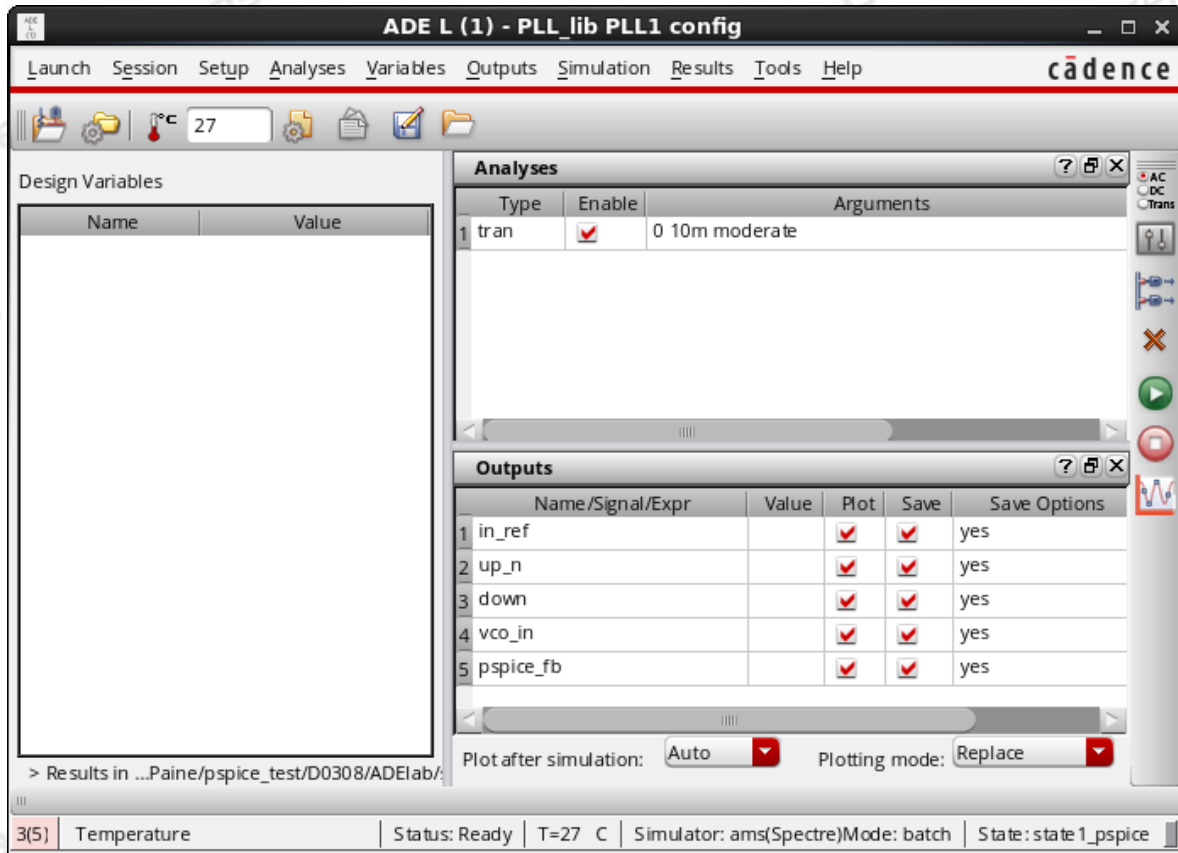
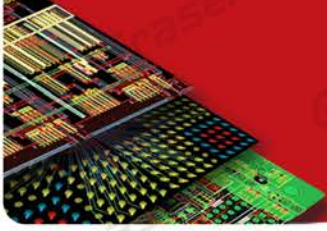
4. 切換 Debugging tool 操作：

Simulation > Netlist and Run Options > 在 Simulation Mode 項目選擇

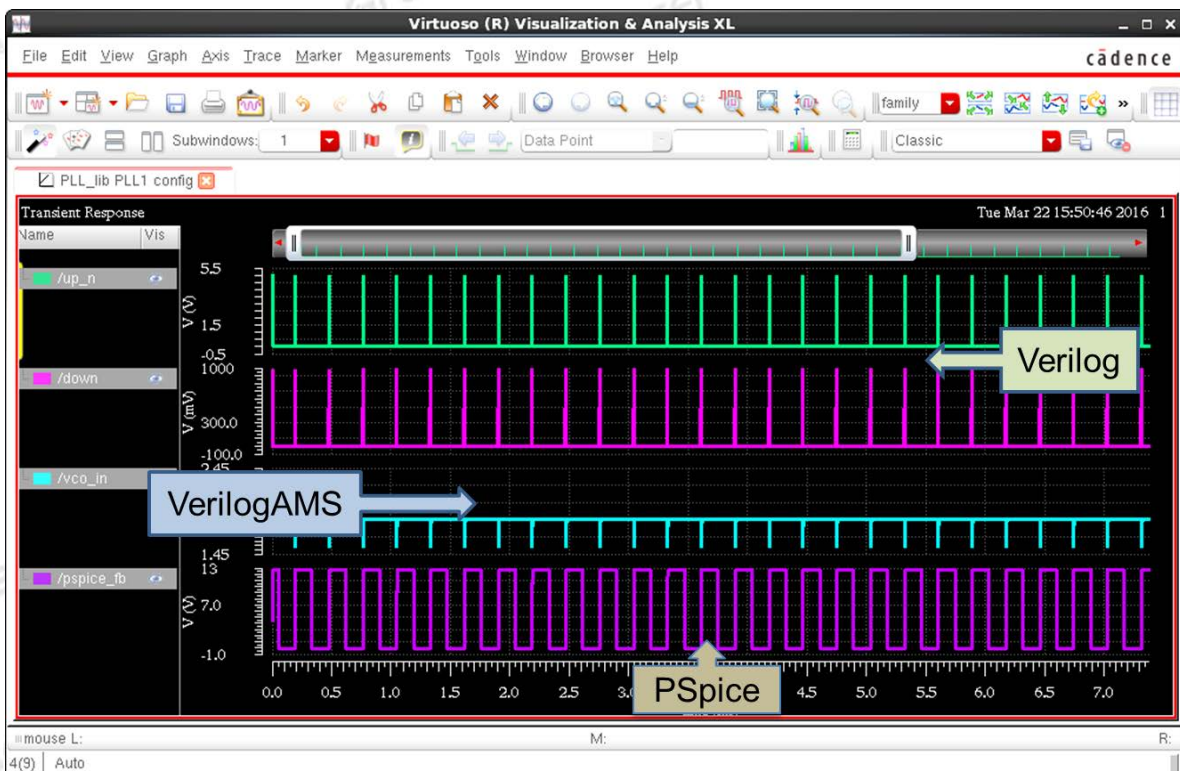
- Viva(預設)：Batch (normal)
- SimVision：Interactive (debugger)



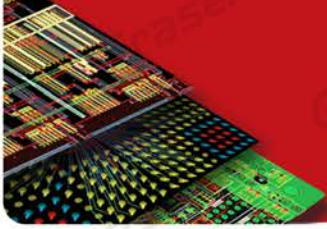
圖七、使用 VSE 導入 PSpice 電路並做整合



圖八、ADE 操作介面

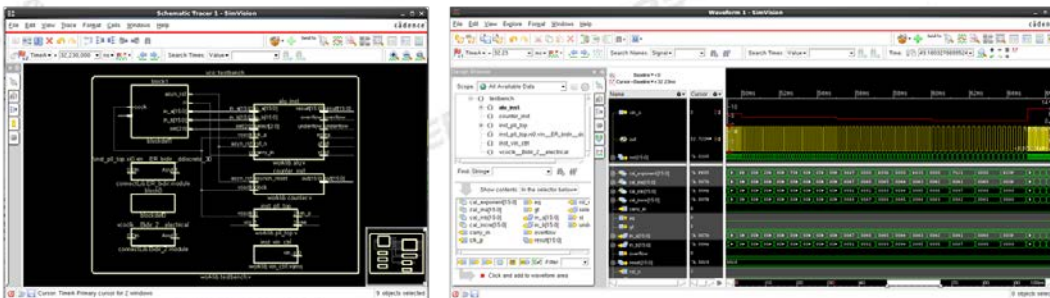


圖九、使用 VIVA 查看模擬結果



AIUM 設計流程 (IES + MMSIM)

AIUM 設計流程主要以 IES 為 AMS 設計平台，IES 為數位模擬器，主要負責 Verilog、VHDL、...、等相關數位電路設計與驗證相關的語言，假若遇到 Spice、Spectre 或者是 PSpice 相關的類比電路設計語言，那麼則由 Spectre 或者是 APS 類比模擬器進行模擬，模擬完成之後則採用 SimVision 協助使用者進行偵錯。另外 acf.scs(圖十一)主要設定 Analog Solver(tran,dc,ac 與 noise)、option 與 probe 相關內容；amscf.scs(圖十二)則是設定 amsd block(portmap、config 與 ie)、載入 spice 檔、載入 model 與載入 acf.scs 設定檔，其中 portmap card 主要掌管 spice 與 hdl 之間的連結，config card 掌管 cell/instance 所使用的 view(spice 與 hdl)，ie card 則是掌管 connect module。



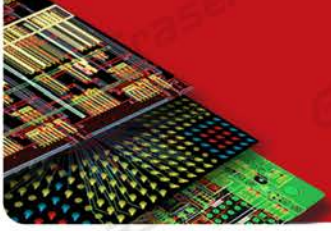
圖十、AIUM 設計流程

```
simulator lang=spice lookup=spectre
.....
.tran 1ns 10ms method=gear2
.probe v(*)
```

圖十一、acf.scs 設定內容

```
include "./source/analog/PLL.sp"
include "./acf.scs"
amsd {
    portmap subckt=pll_top busdelim="_"
    config cell=pll_top use=spice
    ie vsup=30 vss=-15
}
```

圖十二、amscf.scs 設定內容



接著，當我們要載入 PSpice 的電路到 AMS 設計平台時，則需使用

- Spectre 格式：pspice_include "filter.lib"
- SPICE 格式：.pspice_include "filter.lib"

的語法將 PSpice 做載入。載入完成之後須將 PSpice 做成一個 sub-circuit 並提供相對應的電源(圖十三)，完成 sub-circuit 則可以讓頂層的 Verilog 語言進行運用與模擬(圖十四)。模擬完成則採用 SimVision 進行偵錯(圖十五)。

```
simulator lang=spice

simulator lang=spectre
pspice_include "./pspice/vco.lib"
simulator lang=spice

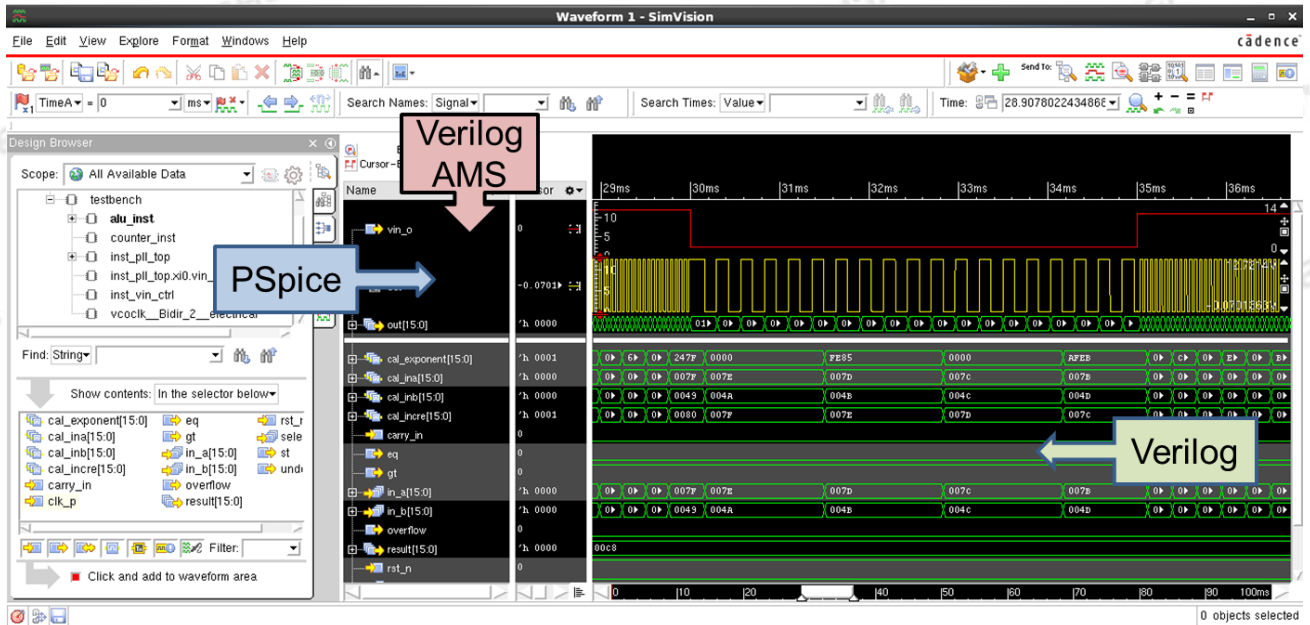
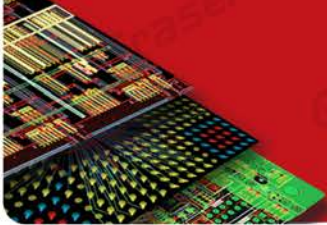
.subckt pll_top out vcc vin vee
xi0 out vcc vin vee VCO_SUB
V1 vcc 0 dc 15
V3 vee 0 dc -15
.ends pll_top

simulator lang=spectre
```

圖十三、載入 PSpice 電路

```
...
counter counter_inst(.asynch_reset(asyn_rst), .clock(vclock), .out(out));
alu alu_inst (
    .result(result),
    .overflow(overflow),
    .underflow(underflow),
    .eq(eq), .gt(gt), .st(st),
    .clk_p(vclock), .rst_n(asyn_rst),
    .carry_in(ci),
    .in_a(in_a), .in_b(in_b),
    .select(sel)
);
pll_top
inst_pll_top(.out(vclock),.vcc(vcc),.vin(inst_vin_ctrl.vin_o),.vee(vee));
vin_ctrl inst_vin_ctrl(.vin_o(inst_pll_top.xi0.vin));
...
```

圖十四、Verilog 頂層設計內容



圖十五、與 PSpice 模擬結果

總結

經由上述的說明，我們可以了解到 AVUM 與 AIUM 設計流程，可以協助使用者在模擬階段看到 FPGA 與周邊的電路兩邊互相工作狀況，而無須因為修改 HDL 來源碼，需不斷重複進行合成、實現與燒入的動作，若 HDL 來源碼複雜度越高，那麼重複性的動作也相對需花費更多時間。最後，下一篇 FPGA 混和模擬文章將會撰寫有關如何將 HDL、IBIS 與 PSpice 進行模擬，以更貼近實體訊號運作狀況。