文件分類：

- ✧ **0.18um TSMC/Arm**
- ✧ **ICC**
- ✧ **Design Compiler**
- ✧ **SOC Encounter**
- ✧ Calibre
- ✧ Virtuoso

文件標題：

0.18um TSMC/Arm Design Kit Memory 使用注意事項

文件大綱：

1. Memory 種類。
2. Transistor-level Post-layout Simulation 及晶片下線注意事項
3. Synopsys™ Design Compiler synthesis mode
4. DFII library
5. ICC library
6. Calibre Black Box LVS 注意事項

更新紀錄：

更新說明	更新時間	更新人員
● 初版	2004/10/27	盧彥均
● 二版	2008/02/25	莊宗桓
● 三版	2009/10/27	盧彥均

1 Memory 種類：

此 Design Kit 所附之 Memory Compiler 由 Arm 公司提供，可產生之 Memory 種類如下。

1.1 High-Speed Single-Port SRAM

1.1.1 目錄：`CBDK_TSMC018_Arm/CIC/Memory/ra1shd`

1.1.2 文件：`CBDK_TSMC018_Arm/CIC/doc/ra1sh.pdf`

1.2 High-Speed Dual-Port SRAM

1.2.1 目錄：`CBDK_TSMC018_Arm/CIC/Memory/ra2sh`

1.2.2 文件：`CBDK_TSMC018_Arm/CIC/doc/ra2sh.pdf`

1.3 High-Speed Single-port Register File

1.3.1 目錄：`CBDK_TSMC018_Arm/CIC/Memory/rf1sh/`

1.3.2 文件：`CBDK_TSMC018_Arm/CIC/doc/rf1sh.pdf`

1.4 High-Speed Two-port Register File

1.4.1 目錄：`CBDK_TSMC018_Arm/CIC/Memory/rf2sh/`

1.4.2 文件：`CBDK_TSMC018_Arm/CIC/doc/rf2sh.pdf`

1.5 High-Speed Single-port ROM

1.5.1 目錄：`CBDK_TSMC018_Arm/CIC/Memory/rodshd`

1.5.2 文件：`CBDK_TSMC018_Arm/CIC/doc/rodsh.pdf`

2 Transistor-level Post-layout Simulation 及晶片下線注意事項：

由於 Design Kit 內附的 Memory Generator 無法產生真實 Layout，如要進行 Transistor-level Post-layout Simulation 或透過 CIC 下線，必須在 CIC 重新產生真實的 Layout。因此使用 Memory Compiler 之後請務必點選上方選單 **Utilities > Write Spec** 方式將 Memory 的規格檔案儲存下來，以作為往後 CIC 在 Post-layout Simulation 及晶片下線時的依據。倘若缺少任何一個你所用到的 Memory Spec，你將無法進行 Post-layout Simulation 及晶片下線。

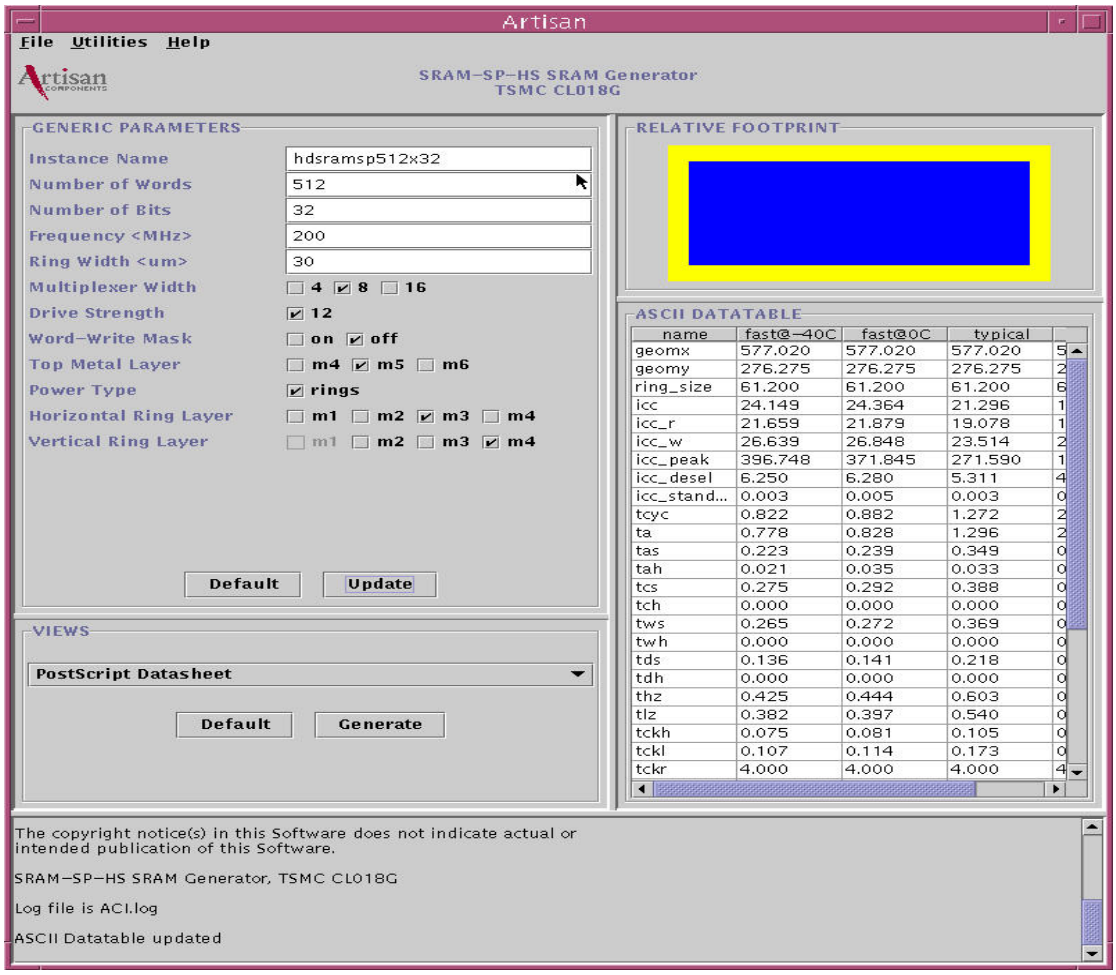
2.1 執行 Memory Compiler 程式

例如：

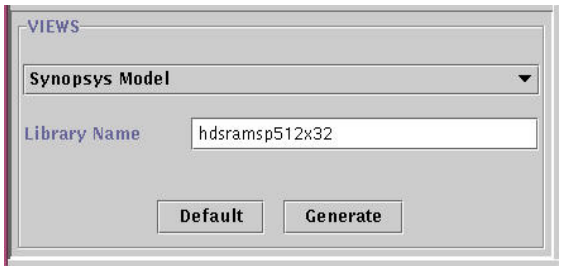
```
unix% ~CBDK_TSMC018_Arm/CIC/Memory/ra1shd/bin/ra1shd &
```

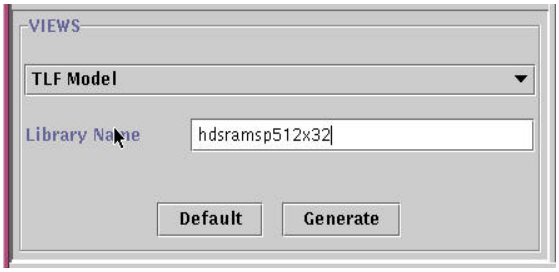
2.2 設定 Memory 規格：

2.2.1 在 GENERIC PARAMETERS 填入所需的 Memory 規格

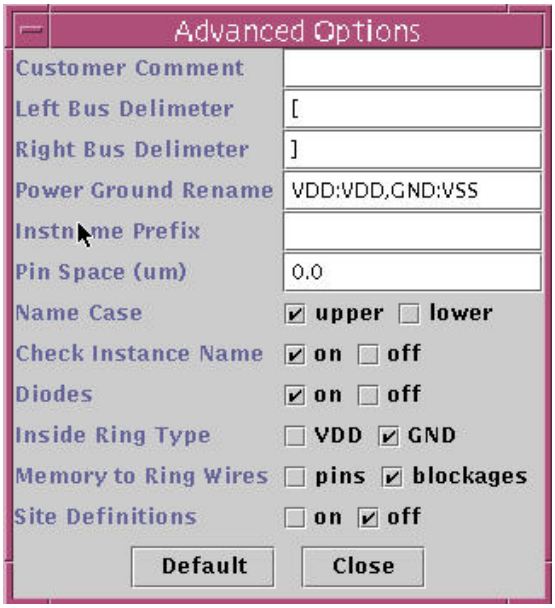


2.2.2 點選 VIEWS 的"**Synopsys Model**"和"**TLF Model**"的 Library Name 改為和 Memory 的 Instance Name 一樣



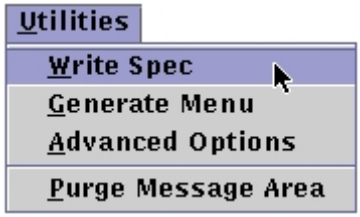


2.2.3 點選上方選單 **Utilities > Advanced Options**：TSMC 的 Design Kit 的 Power Name 為 VDD，Ground Name 為 VSS。 Memory Compiler 內定的 Power/Ground Name 也是 VDD 與 VSS，所以不須更動 Power Ground Name。

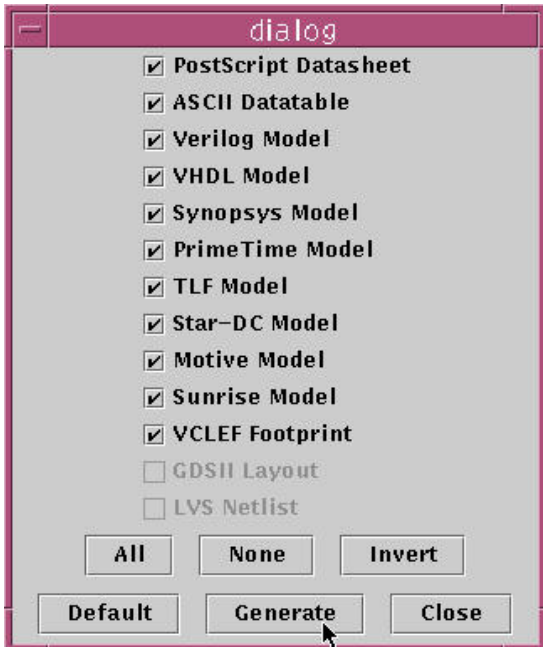


2.2.4 設定完後記得點選 Update。

2.3 輸出 Memory Spec 檔：點選 **Utilities > Write Spec**



2.4 點選 **Utilities > Generate Menu**：請點選 Generate 產生 Memory



2.5 File > Exit

3 產生 synthesis model for Synopsys™ Design Compiler：

Synopsys™ Design Compiler 無法直接使用 Memory Generator 產生出來的 .lib 檔，請依照下列步驟將 .lib 檔轉為 Design Compiler 可使用之 .db 檔。

3.1 使用 Memory Generator 產生 .lib 檔案。Memory Generator 會針對 4 種操作狀況分別產生不同的 .lib 檔(fast，slow，typical)。

例如：

```
hdsramsp512x32_fast@0C_syn.lib
hdsramsp512x32_fast@-40C_syn.lib
hdsramsp512x32_typical_syn.lib
hdsramsp512x32_slow_syn.lib
```

3.2 Compile .lib 檔案成為 .db 檔。先確定你的 Synopsys 軟體設定環境設定正確，然後再 UNIX command prompt 下執行 **dc_shell-t**。接這在 de_shell 中執行以下步驟：

```
dc_shell-xg-t> read_lib NAME.lib
dc_shell-xg-t> write_lib USERLIB –output OUTPUT_FILE_NAME
```

其中 NAME.lib 可以是 4 種操作狀況的任一種

USERLIB 是在 Memory Generator 裡產生 Synopsys library model (.lib) 時所指定的 Library Name，如果你 compile 時沒指定 Library Name，內定為 USERLIB。

例如：

```
unix% dc_shell-t
dc_shell-xg-t> read_lib hdsramsp512x32_slow_syn.lib
dc_shell-xg-t> write_lib hdsramsp512x32 –output
                hdsramsp512x32_slow_syn.db
```

3.3 上步驟產生的.db 檔為 Synopsys™ Design Compiler 可以使用的 synthesis model。使用方式和 Core Cell 和 IO Cell 的.db 檔使用方式一樣，只要在 ***search_path*** 中設定.db 檔放置的目錄路徑以及在 ***link_library*** 和 ***target_library*** 中加入.db 檔檔名即可。

4 ICC library：

Memory Generator 無法直接產生 ICC 可以使用的 library，請按照下列步驟產生，相關檔案請至 CBDK_TSMC018_Arm/CIC/ICC/memory 尋找。

4.1 利用 Memory Generator 產生.vclef 和.lib 檔案。

4.2 依照第 4 點的步驟產生 4 個.db 檔案。

4.3 建立工作目錄並把.vclef 和.db 檔案拷貝至工作目錄下。

例如：

```
unix% mkdir ram_icc
unix% cp /mem/hdsramsp512x32.vclef ram_icc/
unix% cp /mem/hdsramsp512x32_*.db ram_icc/
```

4.4 把 Design Kit 所附的 ICC technology file（tsmc18_CIC.tf）拷貝至工作目錄下。

例如：

```
unix% cp tsmc18_CIC.tf ram_astro/
```

4.5 把 Design Kit 所附的 astrogen_dk.cmd 及 ram.map 拷貝至工作目錄下。

例如：

```
unix% cp astrogen_dk.cmd ram_icc/
unix% cp ram.map ram_icc/
```

4.6 進入工作目錄 ram_icc。並在工作目錄下產生 lef2Arcs.map 檔案，指定 Cell 為 Macro，檔案內容如下：

```
gdsMacroCell
```

剛才產生的 Memory 的 Instance Name

例如：

```
gdsMacroCell
hdsramsp512x32
```

4.7 編輯 astrogen_dk.cmd。將 ***RAM_NAME*** 取代成 Memory 的 Instance Name。

例如：***RAM_NAME*** 全部改成 hdsramsp512x32

4.8 開啟 **Astro**，在 Message/Input Area 鍵入下列指令。

```
load “astrogen_dk.cmd”
```

4.9 檢查有無錯誤訊息。

4.10 開啟產生的 ICC library 及 Memory Cell，檢查 pin 的位置是否有異常，並確認有 text 加在 pin 上。

5 Calibre Black Box LVS 注意事項

5.1 利用 Memory Generator 產生 Verilog 檔案。

5.2 編輯上步驟產生之 Verilog 檔案，只留 module 和 input output 宣告，其餘刪除。此檔案會在執行 **v2lvs** 程式時用到。

例如將編輯後的檔案存成 hdsramsp512x32_lvs.v，其內容為：

```
module hdsramsp512x32 (
   Q,
   CLK,
   CEN,
   WEN,
   A,
   D,
   OEN
);
   output [31:0] Q;
   input CLK;
   input CEN;
   input WEN;
   input [8:0] A;
```

```
    input [31:0] D;
    input OEN;

endmodule
```

5.3 除了 Verilog 檔案之外，Black Box LVS 還需要 Pseudo SPICE Netlist。以上述的例子來說，假設 Pseudo SPICE Netlist 檔名為 hdsramsp512x32_lvs.spi，其內容應為：

```
.SUBCKT hdsramsp512x32
Q[0] Q[1] Q[2] Q[3] Q[4] Q[5] Q[6] Q[7] Q[8] Q[9] Q[10] Q[11] Q[12]
Q[13] Q[14] Q[15] Q[16] Q[17] Q[18] Q[19] Q[20] Q[21] Q[22] Q[23]
Q[24] Q[25] Q[26] Q[27] Q[28] Q[29] Q[30] Q[31] CLK CEN WEN A[0]
A[1] A[2] A[3] A[4] A[5] A[6] A[7] A[8] D[0] D[1] D[2] D[3] D[4]
D[5] D[6] D[7] D[8] D[9] D[10] D[11] D[12] D[13] D[14] D[15] D[16]
D[17] D[18] D[19] D[20] D[21] D[22] D[23] D[24] D[25] D[26] D[27]
D[28] D[29] D[30] D[31] OEN
.END
```

5.4 有了上述之 Verilog 和 Pseudo SPICE Netlist 檔案之後，還必須在 Calibre LVS 的 rule file 中加入 Memory 的 Black Box 宣告才可以進行 Black Box LVS。格式如下：

```
LVS  BOX  Memory_cell_name
```

例如：

```
LVS  BOX hdsramsp512x32
```

5.5 執行 **v2lvs** 時需用 **-l option** 指定 Memory 的 Verilog 檔案。

例如：

```
unix% v2lvs –v CHIP_pr_lvs.vg –l tsmc18_lvs.v –l tpz973gv_lvs.v -l
hdsramsp512x32_lvs.v –o CHIP.spi –s tsmc18_lvs.spi –s
tpz973gv_lvs.spi
```

5.6 執行 Calibre LVS 時不需額外指定任何參數。