



文件分類：

- ✧ **0.18um UMC/Faraday**
- ✧ **ICC**
- ✧ **Design Compiler**
- ✧ **SOC Encounter**
- ✧ Calibre
- ✧ Virtuoso

文件標題：

0.18um UMC/Faraday Design Kit Memory 使用注意事項

文件大綱：

1. Memory 種類。
2. 產生 Memory
3. Synopsys™ Design Compiler synthesis mode
4. ICC library
5. 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	2015/10/14	盧彥均

1 Memory 種類：

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

1.1 Synchronous Single-Port SRAM

1.1.1 目錄：

CBDK018_UMC_Faraday_v1.0/CIC/Memory/ftclib_200901.2.1/EXE/
memaker

1.1.2 文件：

CBDK018_UMC_Faraday_v1.0/CIC/Memory/ftclib_200901.2.1/EXE/mem
aker.package/doc/Memmaker_UserGuide_200901.2.1.pdf

1.2 Synchronous Dual-Port SRAM

1.2.1 目錄：

CBDK018_UMC_Faraday_v1.0/CIC/Memory/ftclib_200901.2.1/EXE/
memaker

1.2.2 文件：

CBDK018_UMC_Faraday_v1.0/CIC/Memory/ftclib_200901.2.1/EXE/mem
aker.package/doc/Memmaker_UserGuide_200901.2.1.pdf

2 產生 Memory：

2.1 執行 Memory Compiler 程式

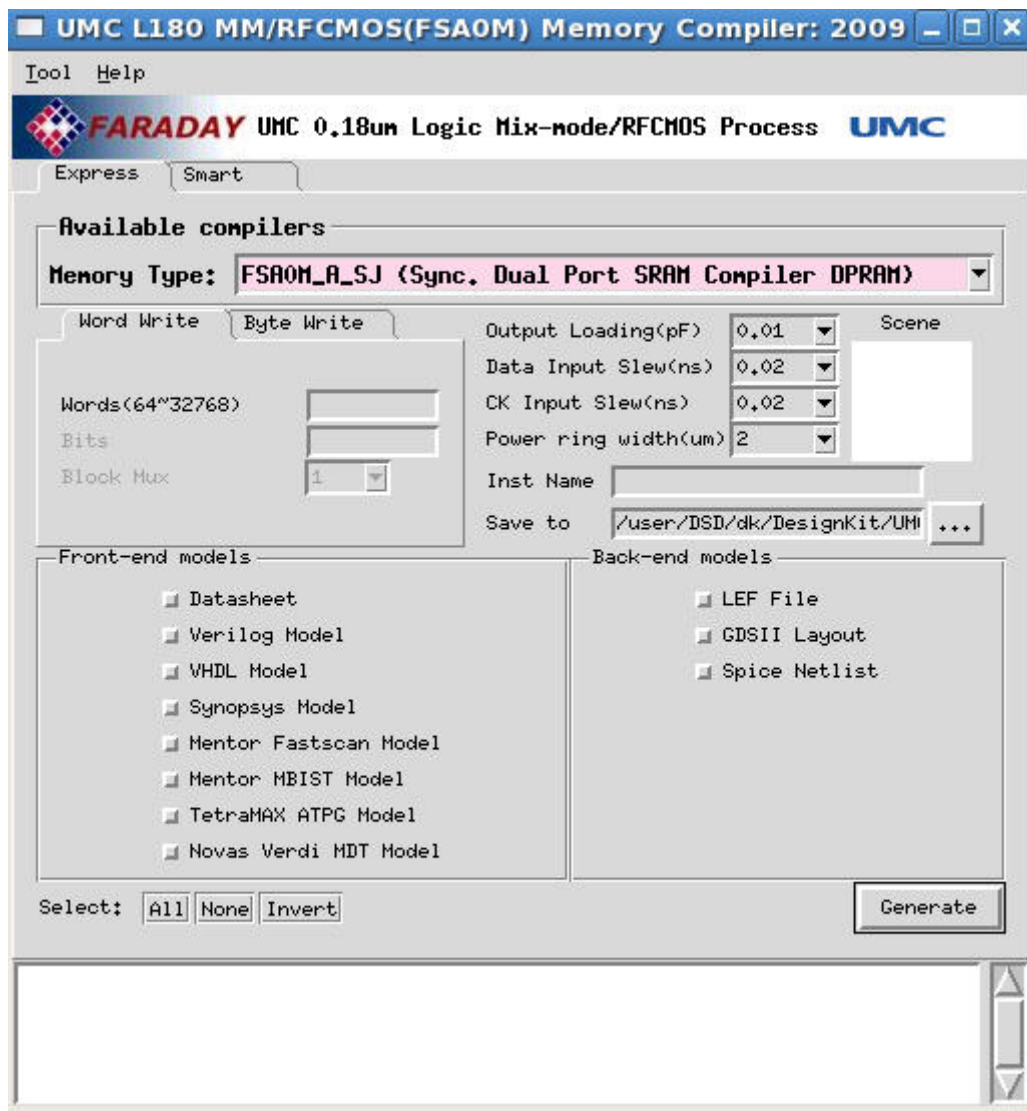
例如：

unix%

CBDK018_UMC_Faraday_v1.0/CIC/Memory/ftclib_200901.2.1
/EXE/memaker &

2.2 設定 Memory 種類：

2.2.1 在 Memory Type 選擇所需的 Memory 種類



2.2.2 填入所需 Memory 規格

2.3 點選 **Select > All**：請點選 Generate 產生 Memory

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 Milkyway library，請按照下列步驟產生，相關檔案請至 CBDK018_UMC_Faraday_v1.0/CIC/ICC/ 尋找。

- 4.1 利用 Memory Generator 產生 .vclef 和 .lef 檔案。

- 4.2 把 Design Kit 所附的 ICC technology file (umc_018_1p6m_mk_20ka_cic.tf) 拷貝至工作目錄下。

例如：

```
unix% cp umc_018_1p6m_mk_20ka_cic.tf ram_icc/
```

- 4.3 把 Design Kit 所附的 lef2fram.scm 和 Memory Generator 產生 .lef 拷貝至工作目錄下。

例如：

```
unix% cp lef2fram.scm ram_icc/
```

```
unix% cp memory.lef ram_icc/
```

- 4.4 進入工作目錄 ram_icc。利用文字編輯器把 lef2fram.scm 打開，檔案內容如下：

```
define lib_name    "cell_name"  
define tech_file  "dk.tf"  
define data_path  "ram.lef"  
cmCreateLib  
setFormField "Create Library" "Library Name" lib_name  
setFormField "Create Library" "Technology File Name" tech_file  
setFormField "Create Library" "Set Case Sensitive" "1"  
formOK "Create Library"  
.....  
.....
```

4.5 編輯 lef2fram.scm 內容。

4.5.1 將 **cell_NAME** 取代成 Memory 的 Instance Name。

例如：**RAM_NAME** 改成 hdsramsp512x32

4.5.2 將 **dk.tf** 改成 umc_018_1p6m_mk_20ka_cic.tf。

4.5.3 將 **ram.lef** 改成 memory.lef。

4.6 執行下列指令。

```
unix% Milkyway -galaxy -nogui -log read_lef.log -load lef2fram.scm
```

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 umc18_core_lvs.v -l umc18_io_lvs.v  
-l umc18_pad_lvs.v -l hdsramsp512x32_lvs.v -s umc18_core_lvs.spi -s  
umc18_io_lvs.spi -s umc18_pad_lvs.spi -o CHIP.spi -s1 VCC -s0 GND
```

- 5.6 執行 Calibre LVS 時不需額外指定任何參數。