

# 行政院國家科學委員會專題研究計畫成果報告

## 通訊與訊號處理晶片用模組庫之發展

### Development of Module Library for Communication/ Signal Processing System Chips

計畫編號: NSC 88-2215-E-002-039

執行期限: 87年8月1日至88年7月31日

主持人: 闕志達 國立臺灣大學電機學院電機系

電子信箱: chiueh@cc.ee.ntu.edu.tw

#### 一、中文摘要

本計劃在於建立一適用於基頻訊號處理單元設計之參數化模組庫與晶胞庫。設計者設定參數於這些參數化模組後，可直接進行功能階層模擬並得到 Verilog 闡階層硬體描述碼。藉由這種設計方式我們可大幅縮短系統設計時間並避免設計人員撰寫闡階層硬體描述時可能發生的錯誤。我們提出參數化模組類別及資料結構並據此實作參數化模組如各種邏輯晶胞陣列、加/乘法器與 FIR 濾波器。在晶胞庫設計部份，我們利用 TSMC 0.6 / 0.35 $\mu$ m 兩製程設計晶胞，兩晶胞庫各提供 56 種常用晶胞。我們調整電晶體使其在低電壓可操作，並對不同狀況下晶胞效能進行廣泛測試。

關鍵詞: 模組庫、晶胞庫、參數化模組

#### 二、英文摘要

The project is focused on developing the parametric module library and the standard cell libraries suitable for digital baseband signal processor design. The designers assign the parameters to parametric modules, perform function level simulations, and immediately get the corresponding gate-level Verilog

code. With this design methodology we could drastically reduce the system design time and prevent errors made by designers. We purposed a parametric module class and several data structure. Based on these classes, we wrote a series of parametric modules including many kinds of logic arrays, high speed adders/multipliers, and FIR filters. The library cells are designed using TSMC 0.6/0.35 $\mu$ m technologies. Each standard cell library contains 56 frequently used cells. We sized the transistors in order to operate the cells in low-voltage conditions and finally performed an exhaustive test to guarantee the cells meet the function/performance specifications.

Keywords: module library, cell library, parametric module

#### 三、計劃緣由與目的

目前無線通訊、寬頻通訊與訊號處理產業已成為成長性最高之高科技產業之一。吾人可預期未來設計者需要在更短期間設計效率及功能更豐富、品質更優良的產品以爭取市場。然而一般無線通訊、寬頻通訊與訊號處理之積體電路系統複雜度非常的高。以通訊系統而言，基頻電路由各種常用訊號處理模組構成。設計者視

調變、編碼方式設計各種規格之訊號處理模組。而因基頻訊號處理仍需極高的運算速率，因此視要求設計專用晶片或選用數位訊號處理器實作。

本計劃為「訊號處理系統單晶片設計環境」總計劃項下子計劃三，其主要目的在於發展一數位通訊系統與訊號處理系統晶片使用之重要模組及自動產生模組系統。其中模組庫除參數化數位訊號處理模組庫設計外尚包含金氧半製程之高效能低功率晶胞庫之設計。按原計劃參數化數位訊號處理模組庫與模組產生系統只產生各種規格之正確模組之電路檔案。但配合目前 IP 潮流與測試考量，我們為各模組內建驗證功能。晶胞庫設計部份包含利用國科會晶片製作中心提供 TSMC 0.6 $\mu$ m 1P3M / 0.35 $\mu$ m 1P4M 兩製程設計電路、完成實體與抽象佈局設計。此外需針對各晶胞進行完整功能模擬與驗證、參數萃取並對製程變異 / 溫度變化進行廣泛測試。

因計劃目的為設計適用在數位基頻通訊與訊號處理晶片設計之模組庫，故晶胞 / 模組設計應於多重電壓下均滿足時序及扇出能力之要求。預計於計劃執行完成後，使用者利用模組產生系統設定模組參數需求之後，系統即可自動輸出閘階層電路模組與對映功能模擬之驗證函數。該驗證函數可配合較高階層之系統模擬使用，而閘階層電路模組配合上述晶胞庫，再利用目前商用自動佈局軟體即可迅速完成晶片之實體設計。

## 四、研究方法與成果

### 4.1 晶胞庫之建立

晶胞庫為晶片實體設計之底層，各晶胞之時序 / 功率規格將作為模組產生系統建構模組之依據，並影響實際模組甚或晶片之整體效能。對於一般邏輯晶胞，我們選取功率 - 延遲乘

積較小之靜態電路架構，調整邏輯或加入緩衝器，反覆進行佈局與時序驗證直至時序滿足需求為止。對於高扇入之邏輯單元，視需要採用 SFPL 架構。對於最常用之全 / 半加器與進位預處理 (carry look-ahead) 等運算單元，我們選取多種邏輯架構分別設計模擬，期能適用於不同種類之要求，然如 SFPL[2] / SRPL / CPL / DPL 等邏輯具高度不對稱性，亦多以雙端訊號操作，其佈局面積之使用效率較低。對於記憶單元，為保證自動佈局之電路工作及測試狀況穩定，目前晶胞庫包含之正反器為靜態雙相位正 / 負 / 雙緣觸發。動態邏輯及動態單 / 雙相位正反器位準因受電容比例影響，不適合於導線負載及延遲高度變異之自動繞線環境使用故不列入。此二晶胞庫內容見表(一)，計各有 56 個晶胞可供使用，圖(一)列出 TSMC 0.6 $\mu$ m 1P3M 晶胞庫部份晶胞之佈局圖。

### 4.2 各晶胞佈局設計考量

在晶胞佈局之設計上，為求佈局適用於不同繞線引擎，高層導線與接點使用方式需與製程參數檔定義之方向、線寬配合，位置與間距對齊格子點(2.4 $\mu$ m)，俾使各晶胞於自動繞線時滿足多孔性 (porosity, 與自動繞線成功機率相關) 之需求，並提高繞線引擎之演算效率。其餘與佈線工具相關之佈局細節不列舉。這些設計方式之代價多為增加晶胞佈局面積。

### 4.3 晶胞時序要求及測試考量

隨製程之尺寸縮減，導線間寄生電容值將大符提高。考慮圖(二) TSMC 0.6 $\mu$ m 製程導線層雜散電容比例[1]，第一層導線之層間寄生電容、邊緣電容已大於異層間平版電容，而高層導線情況差異則更大。為此估計晶胞之扇出能力時應將雜散電容列入考慮，

晶胞之佈局亦應使雜散電容總合較小。我們估計一般晶胞輸出端真實負載及雜散電容合計數量級可至 10~100 fF 之譜,故以 100fF 作為晶胞設計之扇出能力要求,而高推力之反向器及緩衝器扇出定為 500fF。另為求晶胞進行完整之時序驗證,需自行撰寫程式,計算各晶胞內不同輸入產生輸出邏輯轉態之所有組合,換成波型、進行電路模擬以求延遲時間之最大值。我們設定晶胞訊號延遲限制條件為:

- } 各種反向器、緩衝器、(解)多工器所有輸出轉態均於 **1.0ns** 以內完成
- } 各種扇入之簡單邏輯、AOI / OAI 所有輸出轉態均於 **2.0ns** 以內完成
- } 各種複雜運算邏輯所有輸出轉態均於 **4.0ns** 以內完成
- } 各種正反器觸發 / 重置訊號至資料輸出轉態於 **1.5ns** 以內完成

TSMC 0.35 $\mu$ m 1P4M 製程晶胞標準定為在 Slow-Slow 模型、75°C、電源電壓 2.0V、負載 100fF 狀況下之所有邏輯轉態均滿足上述標準。TSMC 0.6 $\mu$ m 1P3M 製程因臨限(threshold)電壓高,低電源電壓狀況下電晶體推力明顯不足,故晶胞標準設定為負載 20fF,其餘條件同 0.35 $\mu$ m 1P4M 製程。完成佈局設計後,我們電源電壓模擬 2.0V / 3.3V / 5.0V、負載模擬 0fF / 20fF / 100fF、製程模擬 Typical / Slow-Slow 模型、溫度模擬 25°C / 75°C 之所有組合狀況,並將各組合下延遲時間及平均消耗功率製表。

#### 4.4 參數化模組架構設計

圖(三)為基頻通訊與訊號處理系統晶片設計之典型流程。給定系統規格後,設計者首先進行功能模擬,即利用 C/C++ 語言描述演算法並模擬。此時訊號精確度多設為浮點數,時間

精確度亦以時脈(Clock)周期為單位,故處理資料量可遠大於電路模擬。功能模擬中我們關心系統演算法的正確性。待演算法確定後,將訊號精確度改為定點數模擬,確定效能損失在可接受範圍內。定點模擬完畢後設計者以閘階層硬體描述語言表述系統並進行閘階層模擬。此時除訊號精確度為定點數外,模擬採用事件驅動之機制以檢查邏輯延遲等时序特性。

本計劃參數化模組產生器部份針對設計流程中 C/C++系統描述至閘階層電路模組產生進行研究,期能自動化以減少設計時間並消除人為錯誤發生機率。我們的構想是把閘階層輸出及模擬等功能全部整合在 C/C++參數化模組中。而目前已定義參數化模組撰寫規則,將支援任意長度定點數功能模擬、位元定址能力資料結構及輸出硬體描述語言等功能內建於抽象電路模組中,並用以實作部份常用電路模組,其細節詳述於后:

#### 4.5 功能階層模擬方式

功能階層模型與模擬方式必需可模擬多時脈系統。如圖(四),我們認為模組的功能描述可區分成一無延遲時間功能區塊及一延遲元件。輸入訊號在經過運算之後,其結果將延遲整數時間周期後反映在輸出端。在系統模擬迴圈內呼叫預先定義之時脈觸發及重置函數,各模組重置、計算及更新輸出值的函數將依序被執行。此模擬的進行相當類似兩相位時脈之同步電路操作,系統模擬迴圈在一個時脈週期內將各子模組之運算函數執行完畢並進行所有同步子 (synchronizer, 通常指正反器) 與子模組同步子更新。

#### 4.6 模組變數資料結構

C/C++ 並非為硬體描述設計,為描述任意模組及線路連結時,我們希

望它提供額外的能力如：

- } 可變精確度定點數(位元寬度至 64 位元以上)具備一般運算及比較大小等功能
- } 可自由轉換成 C/C++ 基礎資料型態
- } 可作位元階層定址
- } 支援變數之連結語法，即可由不同訊號的排合組合產生新的訊號

圖(五)為模組變數之類別繼承關係。為達成任意長度定點數且支援自由型態轉換及運算，我們撰寫 Regs/Wires 兩類別作為一般模組使用的訊號型態。一般訊號類別之成員函數功能包含：

- } 物件建構解構、自動記憶體管理
- } 命名、型態判別、註冊機制
- } 功能模擬、傾印 Verilog HDL
- } 精確度(位元寬度)及電路參數更改
- } 參數估計如電路延遲、管線級數的計算

為達成位元階層定址與變數之連結語法，我們設計 NetSeg 虛類別用以自動處理定址與連結運算(設計者不需自行呼叫此類別)。此處需利用 C++ 參考語法(reference)、運算子重載(operator overloading)、複製建構子(copy constructor)及自動型別轉換等四項特性。設計者利用這些類別設計 C/C++ 參數化電路模組，概念與 Verilog HDL 的使用頗近似。

#### 4.7 參數化模組資料結構

模組資料結構共通性較訊號資料結構為高，因此規劃一基礎類別 BaseModule 供其於參數化模組繼承。如圖(六)，一模組由輸出入埠、子模組與內部節點三個部份定義。輸出入埠以陣列型態儲存，子模組與內部節點變化較大，以鍵結串列儲存。鍵結串列上節點皆俱 parent 指標。模組

類別成員函數功能包含：

- } 物件建構解構、自動記憶體管理
- } 配置子模組、內部節點/匯流排、節點電器特性設定
- } 命名、註冊、模擬、傾印 Verilog HDL
- } 管線(Pipeline)設計
- } 參數估計如電路延遲、管線級數的計算

目前設計參數化模組可調整的參數舉例如模組各輸出入埠寬度、常數輸出入埠之埠值、模組管線級數等，其餘參數則隨模組功能、型式而有差異。

圖(七)為參數化模組物件之工作流程。模組物件之建構子進行輸出入埠設定與註冊工作，參數化模組需再設定參數值。模組初始化後可進行子模組與模組間連結的建立、管線設計與資料流計算，然後作模擬、參數估測與與閘階層 Verilog 輸出，並視需要對模組參數做調整，重複進行上述工作。資料流計算區塊目前並未實作，因此內定功能模擬函數與參數估測函數均採重覆執行方式待結果收斂。

以功能模擬函數為例，如圖(八)，因並未假設各模組按資料流順序執行，若資料流的方向(白色箭頭)為由右至左，而子模組執行計算先後為反向(按子模組 1、2、3、4 執行)，則每個子模組計算一次後，輸出仍不會正確。每個子模組都要傳回自己是否將輸出埠值改變。模組每次執行子模組的運算函數後，要用子模組的傳回值來判定是否再計算一次(如果有任何子模組改變輸出埠值，則所有子模組要再計算一次)。若模組內無組合電路迴路，此方式最後可收斂。就功能模擬函數而言，這種方式雖慢，但對所有模組均適用，因此可用來判別模組功能(子模組連結)正確與否。一旦確定功能正確，設計者可以用自行撰寫之精簡模擬函數覆蓋內定模擬函數。參數

估計如管線級數與邏輯延遲時間估計方式均採用類似方式達成。

#### 4.8 常用參數化模組設計

我們目前已完成參數化模組有：

- } 邏輯晶胞陣列 (底層模組)
- } CPA、CSA、Wallace Tree 加法器
- } 樹狀 / 陣列平行乘法器 (係數分成變數 / 常數，有號數 / 無號數)
- } Direct / Transposed Form FIR 濾波器 (係數亦分成變數 / 常數兩種)

這些模組參數沒有預設範圍，範圍限制主要來自系統記憶體大小。以簡單乘法器為例，圖(九)為一 5x5 無號數乘法器宣告範例，圖(十)為系統合成之閘階層 Verilog HDL。注意一般模組與訊號名稱以流水號編排，且為區分參數化模組，所有參數都在模組名稱中出現。而圖(十一)為 8x8 有號數陣列乘法器的實作實例：在系統輸出閘階層 Verilog 模組後將檔案匯入 Cadence 整合環境，再利用 Cell Ensemble 自動佈局引擎完成繞線之佈局圖(使用 TSMC 0.6 $\mu$ m 1P3M 製程)。

#### 五、未來的工作

下一年度中將視需要擴充原晶胞庫晶胞數量並提昇晶胞效能之要求。就模組庫部份而言，目前模組資料結構還有功能未完成處待完成。預定繼續完成之參數化模組計有：

- } Counter、Shifter、FIFO
- } Complex multiplier
- } SRAM、ROM、PLA
- } MAC、Correlator、FPU

其中 MAC、Correlator、FPU 具較複雜電路結構，將利用既有參數化模組構成。除此之外應可開始設計參數化模組之圖形使用者介面，使設計者可以更方便的使用這些參數化模組。

#### 六、結論與討論

本計畫發展常用基頻訊號處理之參數化模組與底層晶胞庫之建立。電

路設計者在宣告模組及設定參數後即可進行功能模擬並得到閘階層硬體描述，再利用底層晶胞庫進行自動佈局。此外設計者可方便的利用既有模組建構複雜度更高的參數化模組。因此預期可大幅縮短常用基頻訊號處理模組設計與驗證時間，並減少設計者發生的錯誤。

#### 七、參考文獻

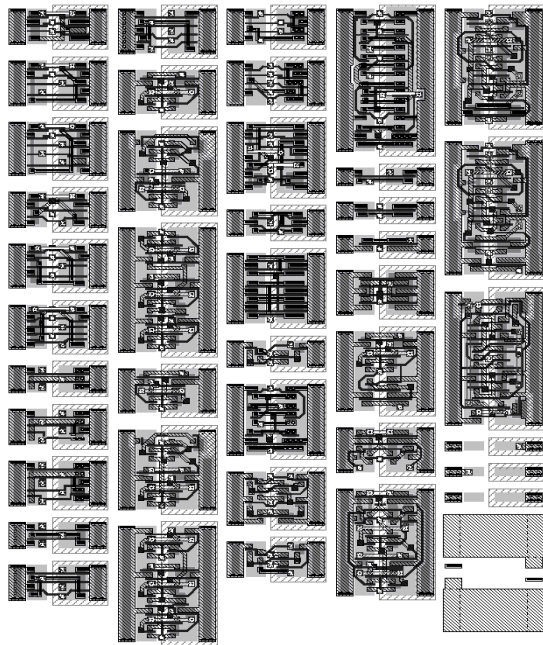
1. TSMC 0.6- $\mu$ m Logic HSPICE Models (0.6U-TW-16L-T1), 1996.
2. Neil H. E., *Principle of CMOS VLSI design a systems perspective 2<sup>nd</sup> ed.*, 1992

#### 八、圖表

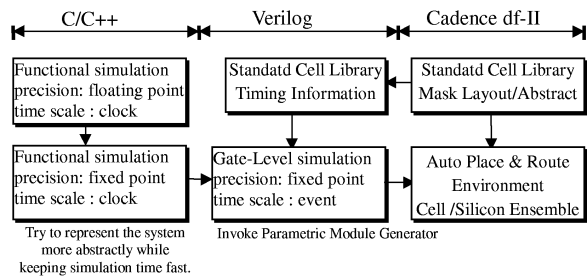
分類	晶胞名稱	架構
簡單邏輯	NAND(2~4), NOR(2~4), AND(2~4), OR(2~4), OAI(3~4), AOI(3~4)	Static CMOS *SFPL
	XOR(2~4), XNOR(2~4), Multiplexer (2-1/4-1) Transmission Gate	Tran. Gate
高扇出邏輯	AND5, OR5	SFPL
運算單元	Half adder, Carry, 4-2 compressor, Carry-lookahead cell, Booth encoder	Static CMOS Tran. Gate
	Full adder	Tran. Gate CPL / DPL SRPL / Hybrid
驅動單元	Inverter(small/large), Driver(small/large), Buffer with enable	Static CMOS
記憶單元	DFF (positive edge), DFF (negative edge), DETFE (double edge)	Static CMOS
佈局連結單元	LEFTCAP, RIGHTCAP, GLUE, HIGH, LOW, FEEDTHU,POWERCE LL, LEFTCAP, RIGHTCAP,	-

表(一)、晶胞庫內容一覽表

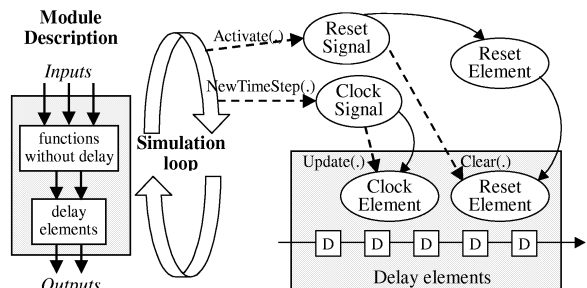
\* 註：TSMC 0.6 $\mu$ m 晶胞庫之 NAND4 / NOR4 / AND4 / OR4 晶胞因低電壓狀況下推力不足，改採 SFPL 架構



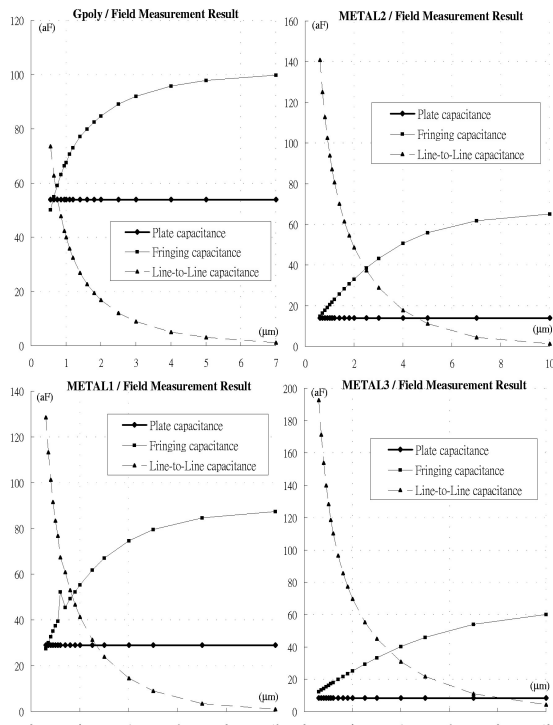
圖(一)、TSMC 0.6μm 晶胞佈局(部份)



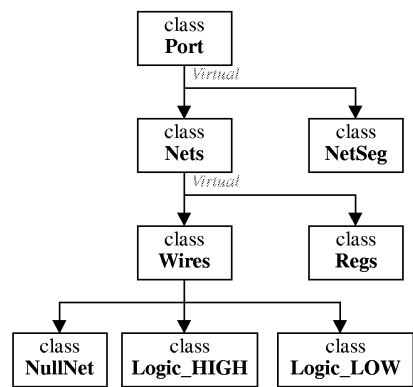
圖(三)、基頻通訊與數位訊號處理系統之晶片設計流程



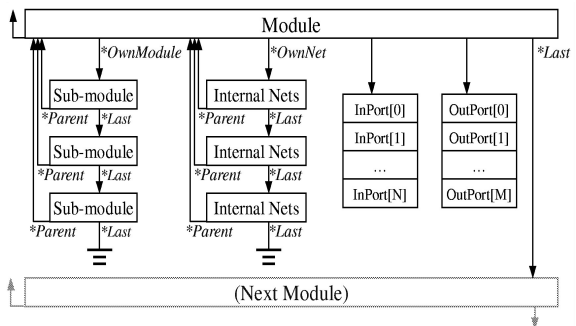
圖(四)、C/C++模型與功能模擬方式



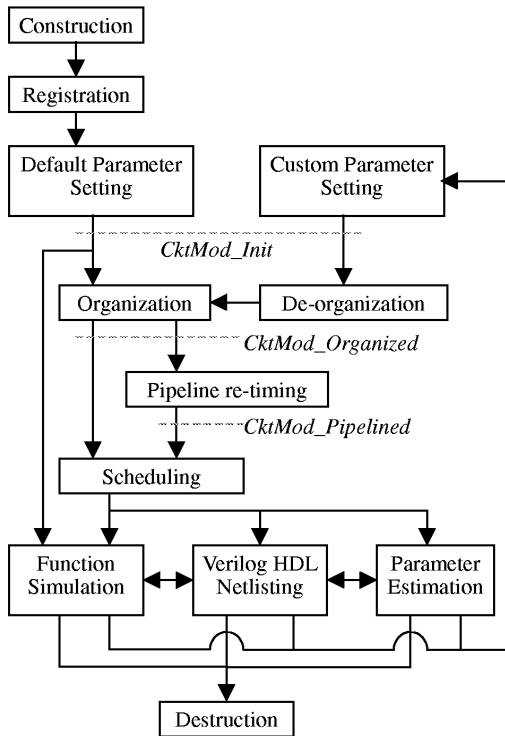
圖(二)、導線層雜散電容比例圖  
(摘自 TSMC 0.6μm 1P3M 製程參數)



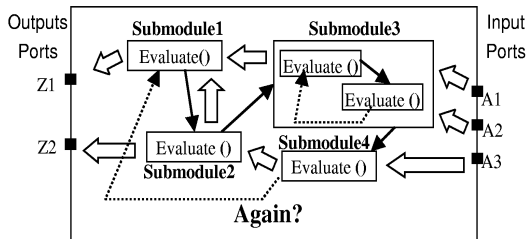
圖(五)、模組變數之類別關係



圖(六)、參數化模組資料結構



圖(七)、模組物件工作流程



圖(八)、遞迴式功能模擬示意圖

```

Regs          z(10), a(5), b(5);
UTreeMUL     TestModule1(z, a, b);

```

圖(九)、C++之訊號與參數化模組宣告

```

module UMUL_5_5_10 (n3,n4,n5);
output  [9:0]  n3;
input   [4:0]  n4;
input   [4:0]  n5;

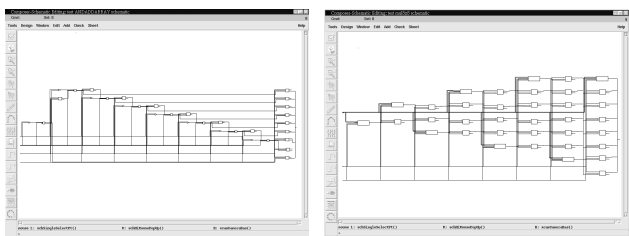
wire [24:0]  n6;
wire [9:0]  n9;
wire [7:0]  nA;

AN2 and21_0 (n6[0],n4[0],n5[0]);
AN2 and21_1 (n6[1],n4[0],n5[1]);
AN2 and21_2 (n6[2],n4[0],n5[2]);
AN2 and21_3 (n6[3],n4[0],n5[3]);
AN2 and21_4 (n6[4],n4[0],n5[4]);
AN2 and21_5 (n6[5],n4[1],n5[0]);
AN2 and21_6 (n6[6],n4[1],n5[1]);
AN2 and21_7 (n6[7],n4[1],n5[2]);
AN2 and21_8 (n6[8],n4[1],n5[3]);
AN2 and21_9 (n6[9],n4[1],n5[4]);
AN2 and21_10 (n6[10],n4[2],n5[0]);
AN2 and21_11 (n6[11],n4[2],n5[1]);
AN2 and21_12 (n6[12],n4[2],n5[2]);
AN2 and21_13 (n6[13],n4[2],n5[3]);
AN2 and21_14 (n6[14],n4[2],n5[4]);
AN2 and21_15 (n6[15],n4[3],n5[0]);
AN2 and21_16 (n6[16],n4[3],n5[1]);
AN2 and21_17 (n6[17],n4[3],n5[2]);
AN2 and21_18 (n6[18],n4[3],n5[3]);
AN2 and21_19 (n6[19],n4[3],n5[4]);
AN2 and21_20 (n6[20],n4[4],n5[0]);
AN2 and21_21 (n6[21],n4[4],n5[1]);
AN2 and21_22 (n6[22],n4[4],n5[2]);
AN2 and21_23 (n6[23],n4[4],n5[3]);
AN2 and21_24 (n6[24],n4[4],n5[4]);
WTA_5_6_7_8_9_0_2 wta2 (n9,nA,{n6[4],n6[3],n6[2],n6[1],n6[0]},{n
6[9],n6[8],n6[7],n6[6],n6[5],n0},{n6[14],n6[13],n6[12],n6[11],n6[10],n0,
n0},{n6[19],n6[18],n6[17],n6[16],n6[15],n0,n0,n0},{n6[24],n6[23],n6[2
2],n6[21],n6[20],n0,n0,n0,n0});
CSA_3_3_4 csa3 ({n3,{n9[9],n9[8],n9[7],n9[6],n9[5],n9[4],n9[3],n9
[2],n9[1],n9[0]},{nA[7],nA[6],nA[5],nA[4],nA[3],nA[2],nA[1],nA[0]},1'b0,1
'b0});

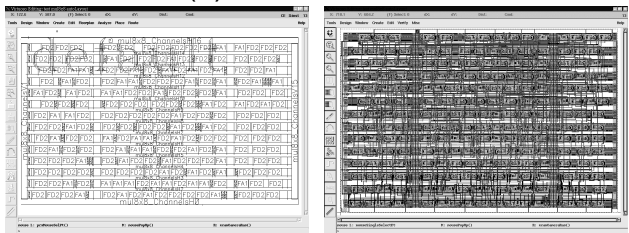
endmodule

```

圖(十)、系統合成 Verilog HDL 輸出  
(未列出 WTA 子模組)



(a) 階層模組匯入



(b) 自動佈局結果

圖(十一)、利用模組庫及晶胞庫合成之  
訊號處理模組佈局 (8x8 陣列乘法器)