

# 106-2 實用數位系統設計期末專題報告

## <16-bit 有號數乘法器之設計與實現>

期末專題 乘法器—書面報告

—組長—

資工系 范真瑋

—組員—

資工系 周家池

資工系 吳俊忻

—目次—

前言

分工情況與貢獻百分比

實現之乘法器演算法與架構說明

驗證流程說明

邏輯合成與驗證結果與說明

心得與討論

## —前言—

我們在設計乘法器時，有搜尋相當多的資料，其中對於乘法器的基本種類有一定的認知，不過在面對效能上似乎沒有可靠的參考，撰寫方式的不同以及選用製程的差異，都會帶來不小的偏差。

最終在 4 類乘法器中，我們選擇了我們最容易理解的陣列式乘法器來進行嘗試，不僅是容易實踐，有較多明顯可優化的地方，在呈現的效能上也有部分資料顯示陣列式乘法器在該次專題評分標準上絕對不會是特別差的選擇，基於這樣的認知我們開始了對陣列式乘法器的一系列實踐。

最終我們也感受到了雖然在速度上可以相當不錯，而且還有許多在影響面積不大的情況能小幅度改善速度的可能，可是減少面積的手段幾乎沒有，我們最終實做出來的乘法器，除了結構單純，速度良好外，在面積方面就相當痛苦，此外結構上在不同的週期有不同的加減法結構，也令其 pipeline 化等常見優化手段幾難以實現，若今天的位元數增加，整份 code 幾乎得重寫，讓修改程式碼的工作變得較不輕鬆。

—分工情況與貢獻百分比—

	姓名	負責項目	貢獻百分比	簽名
1	范真璋	乘法器主要架構撰寫、 加法器架構撰寫、 資料蒐集、 投影片報告主製作	33.3 %	范真璋
2	吳俊忻	乘法器主要架構撰寫、 加法器架構撰寫、 資料蒐集、 書面報告撰寫	33.3 %	吳俊忻
3	周家池	乘法器主要架構撰寫、 Testbench 撰寫、 coding style 改寫、 資源蒐集、 繪製表格與架構圖、 合成模擬	33.3 %	周家池

## —實現之乘法器演算法與架構說明—

### 1. 基本的陣列式乘法器(無號數)、carry save 結構:

一開始我們先嘗試實作基本的陣列式乘法器，這裡我們只考量低位元運算、無號數，此處參考老師講義架構，為 carry save 架構。

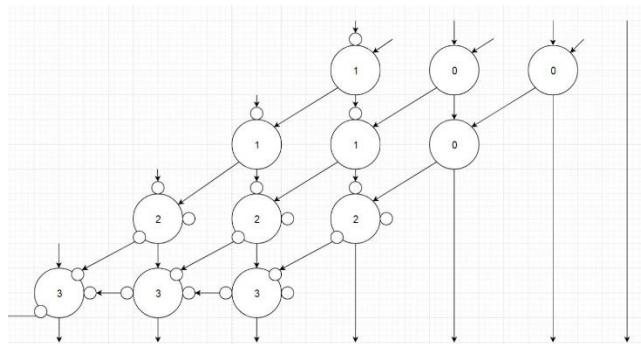
carry save 架構中，同一層的 carry out 並非輸出給同層的下一位加法器，而是輸出給下一層 pp 的下一位加法器，此種在 delay 上表現比較良好，因為同一層的各位元運算可同時進行。

### 2. 直接補數陣列式乘法器(有號數):

完成第一步結構之後開始考量有號數的處理，此處我們參考網路上看到的特殊結構，被稱作「直接補數陣列式乘法器」。

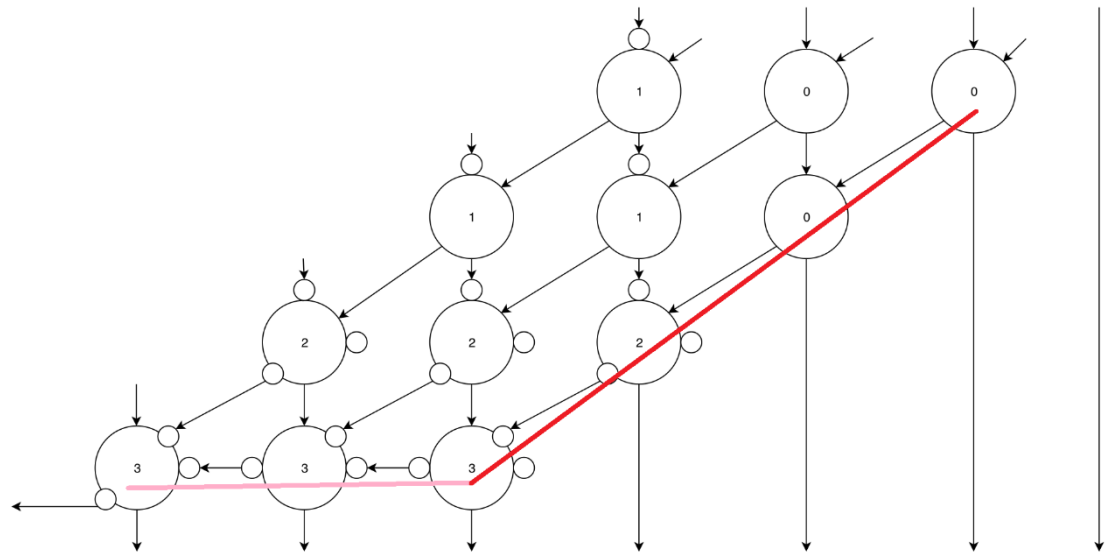
直接補數陣列式乘法器，改變一般陣列式乘法器內容，上一步的陣列式乘法器是由  $n * (n-1)$  個 full adder(以下簡稱 FA) 所串接而成，每個 FA 都是一個加法的小型單位，而在直接補數陣列式乘法器中混入了減法邏輯，有號數中首位元為 signed bit，signed bit 代表著後續的運算位元擁有了(負號)的概念，因此在更改後續的 FA，另其由加法改為減法性質，諸如  $0 - 1$  這樣的運算，輸出 SUM 為 1，而 carry out 為 帶有負號概念的 1 ( $= -1$ )，利用這種性質達到借位運算的效果，利用減法的運算方式，使其運算有號數成為可能。

直接補數陣列式電路架構(僅以 4bits \* 4bits 乘法結構 表示)如下圖:



上圖中每個大圈都可以視為一個一般的 FA，小圈則代表負權的注入。

### 3. 改善的嘗試(skip adder / select adder)



圖中畫線部份為該結構的 critical path。

由圖中架構看出可以看出，我們的運算過程分割兩個部分，分別是上層  $((n-1) * (n-1))$  個 FA 的加法，總共向下進為了  $n-1$  次 ( $n-1$  個週期)，另一部分則是最下層由一組 RCA 將上層的 carry out 回收，向左進位  $n-1$  次 ( $n-1$  個週期)。由上述分析可得知：

上半部 carry save 架構，佔據總運算時間的一半 (紅色線)，佔據大部分面積。

最下層的 RCA 幾乎佔據總運算時間的一半 (粉紅色線)，而其面積只佔據總面積  $1/n$ 。

#### a Carry skip adder

將最下層 RCA 更換成 carry skip adder (以下簡稱 CSKA)，藉此加快最下層的計算時間，並且不會增加太多面積，然而無法改善 critical path 的 delay。

#### b Carry select adder

將最下層 RCA 更換成 carry select adder (以下簡稱 CSEA)，藉此加快最下層的計算時間，並且由於最下層 RCA 只佔據了 (總面積  $1/n$ )，所以就算採用了犧牲面積換取時間的 CSEA 也依然不會增加太多總面積，但是由於這部分佔據了總費時的一半，經過這樣的調整我們讓我們成功壓縮將近總延遲的  $1/4$ 。

### c 模組化

由於上半層有著大面積且重複性的結構，我們考慮著通過將其改寫，改由更大的 module 去改善大面積的反覆加法，並讓合成軟體有更好的優化空間，但由於時間上的問題，我們這部分的最終仍沒能完整實踐。

### d Booth 編碼

由於陣列式的主架構實在是太消耗面積，在面積的部分我們幾乎無法達到更加優化，在費時的部分上半部分我們已經盡我們所能去研究，最終我們感受到陣列式乘法器的有限性，所以我們開始著手研究 booth 編碼，並有一部分的實作，但暫時還無法應用於我們舊有的成品，純 booth 的部分也還遠遠無法達到我們前面陣列式乘法器的效能。

## —驗證流程說明—

撰寫完成 verilog code 後，首先進行 ncverilog 模擬並給予測資，確認乘法結果無誤，在 testbench 中我們可以直接撰寫條件運算結果與預期結果不符合會輸出錯誤，當遇到錯誤則檢查 code 並在 simvision 確認波形。

每個 part 完成後我們都會進行合成，並觀察效能，所以最終我們完成的乘法器為直接補數陣列式乘法器及 carry select 版本擁有較好的效能

## 1. 電路面積

```
*****
Report : area
Design : multiplier
Version: N-2017.09-SP2
Date   : Mon Jul  2 02:00:12 2018
*****

Information: Updating design information... (UID-85)
Library(s) Used:

    slow (File: /mnt3/Process/CBDK_IC_Constest_v2.1/SynopsysDC/db/slow.db)

Number of ports:                1443
Number of nets:                 2736
Number of cells:                1587
Number of combinational cells:  1293
Number of sequential cells:     33
Number of macros/black boxes:   0
Number of buf/inv:              273
Number of references:           233

Combinational area:             10310.007401
Buf/Inv area:                   926.780386
Noncombinational area:          840.213015
Macro/Black Box area:           0.000000
Net Interconnect area:          undefined (No wire load specified)

Total cell area:                 11150.220415
Total area:                      undefined

***** End Of Report *****
```

由上圖的 Total cell area 我們小組所做出來的 array 乘法器面積為  $11150.220415 \text{ um}^2$ ，而.13 製程的 NAND2 闌為  $5 \text{ um}^2$ ，所以換算出來的 gate count 約為 2230 個 NAND2 gate。

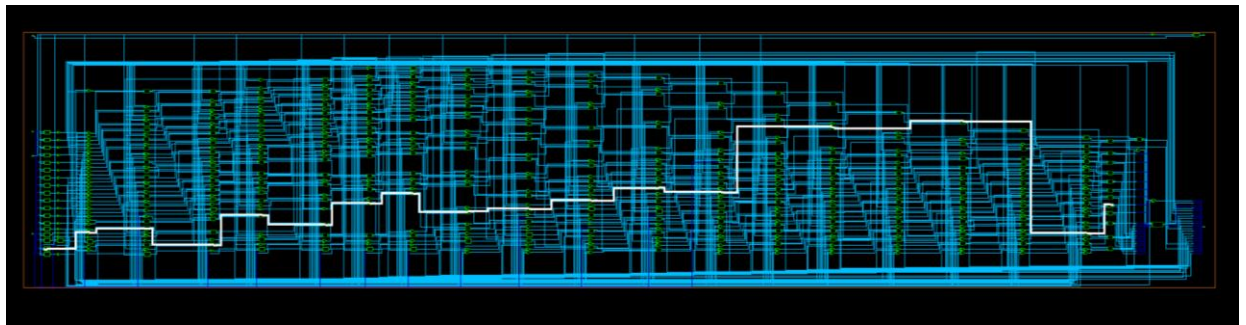
## 2. 完成乘法運算時間

A228/U1/Y (0A21XL)	0.34	8.39 f
A228/C_out (FA3_16)	0.00	8.39 f
csea/Cin (CSEA)	0.00	8.39 f
csea/U2/Y (A021X1)	0.35	8.74 f
csea/U1/Y (A021X1)	0.33	9.07 f
csea/mux2/Sel (MUX3)	0.00	9.07 f
csea/mux2/U2/Y (A022X1)	0.26	9.33 f
csea/mux2/Z[1] (MUX3)	0.00	9.33 f
csea/Sum[9] (CSEA)	0.00	9.33 f
prod[29] (out)	0.00	9.33 f
data arrival time		9.33

-----  
(Path is unconstrained)

由上圖的 data arrival time 可得知我們的乘法器走 critical path 需要 9.33 ns，也就是說 worst case 所花的時間是 9.33 ns。

### 3. Critical path

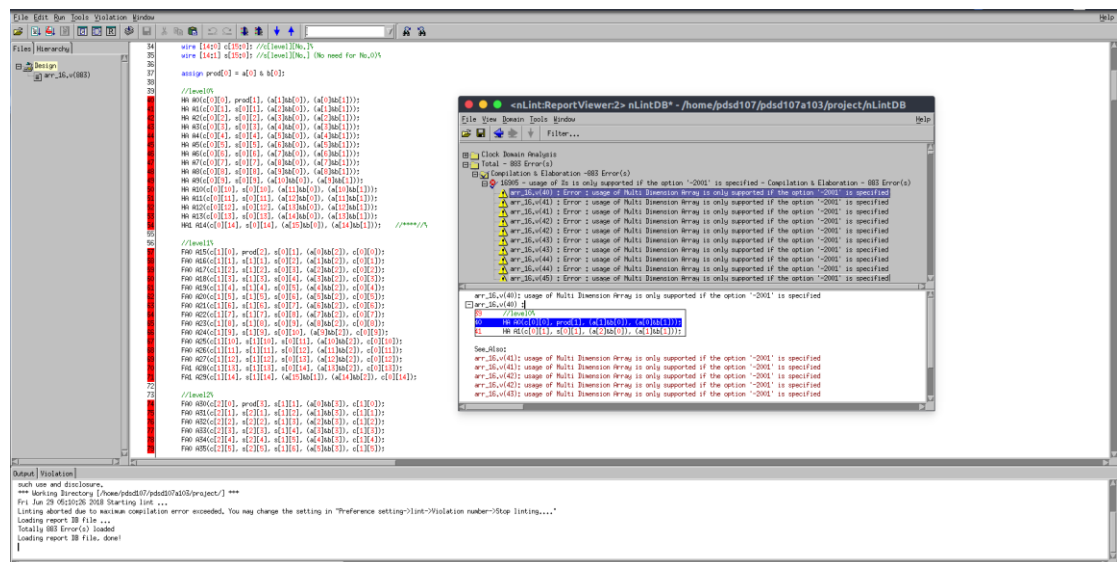


由於我們這組是 Combinational circuit，所以可以藉由軟體找出電路的 Critical path，上圖為 DV 標示出來的 critical path。

### 4. 電路面積 \* 完成乘法運算時間

$$2230(\text{gate count}) * 9.33(\text{ns}) = 20805.9$$

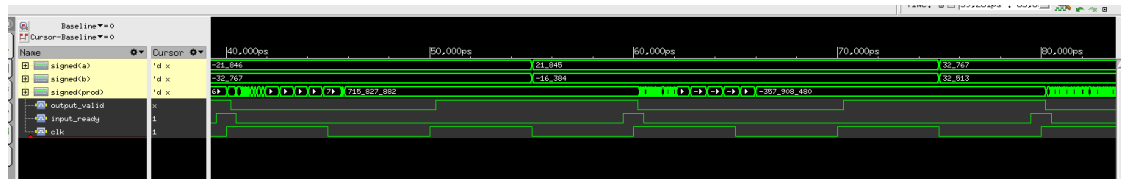
### 5. nLint 分析 RTL code



圖中的 Error 為:usage of Multi Dimension Array is only supported if the option '-2001' is specified，且點選 Error 並按 F1 卻沒有其他說明出現，經由小組討論過後我們決定保持原狀，因 Error 出現應該是選項中的“-2001”沒有選，而我們從講義與網路上卻沒有找到相關的資訊可以解決這個問題，最後合成與跑波形圖也沒有錯誤。



## 6. 波形圖



- a 當 ready=1，代表準備好新輸入值，input 被寫入。
- b clk 正元觸發時，若 ready 為 1:代表 input 被改變，valid 改 0。
- c clk 正元觸發時，若 ready 為 0: valid 會被設為 1，表示先前輸入值的計算已完成。