

於 FPGA 平台製作具有疊代停止之渦輪解碼器

Implementation of a Turbo Decoder with Iteration Stopping by FPGA Platform

黃樹林 張創然 周泰祥

Shu-Lin Hwang Chuang-Jan Chang Tai-shaung Chou

摘要

渦輪碼(Turbo code)通道編碼機制提供一個接近通道容量的編解碼實作技術。由於渦輪編解碼於傳遞訊息方面的優異性，適用在第三代通訊(3G)的應用上。因其電路複雜度過高，使其無法輕易應用到無線行動系統上。降低電路複雜度對現今的渦輪碼來說是一個相當重要的研究課題，因此本文以 SW-Log-MAP(slide window-logarithm-maximum a posteriori)架構於 FPGA 上實現渦輪解碼器並探討加入疊代停止機制所能降低的複雜度。使用 Cyclone II 家族 EP2C70F896 晶片建構渦輪解碼器，合成後其 t_d 為 29.958ns 而 f_{max} 為 13.51 MHz。由軟體驗證分析顯示，於固定點模型疊代停止設計中，硬式規則與傳統固定疊代次數相比，大於 1db 以上明顯可減少約平均 70% 以上之疊代次數，可有效降低運算複雜度與時間。

關鍵詞：渦輪碼、疊代停止機制、複雜度

ABSTRACT

The channel coding using turbo code is close to the capacity of channel in 3G applications. However, the circuit is complexity; we can not easy to use for wireless communication systems. Reducing the circuit complexity of turbo decoder is very important. The architecture of decoder has been implemented with structural SW-Log-MAP(slide window-logarithm-maximum a posteriori) by FPGA and uses the iteration stopping rules to reduce circuit complexity in this thesis. Our decoder has been verified in hardware with Altera Cyclone II Family EP2C70F896C6 Device, and the t_d is 29.958ns and f_{max} is 13.51MHz. Simulation results show that the proposed architecture reduces average number of iterations about 70 % comparing with the other turbo coders used a fixed number of iterations for signal noise rate over 1.3db.

Keywords: Turbo code, iteration stopping rules, complexity.

1.前言

由於渦輪編解碼於傳遞訊息方面的優異性是有目共睹，特別適用在第三代通訊(3G)的應用上。但其電路複雜度高與功率消耗大，使其無法輕易應用到無線行動系上。因此本文著重在於複雜度降低的技術與方法，並以 Visual C++ 建立基本軟體驗證平台，進行模擬實驗並分析渦輪碼之疊代次數、訊號雜訊比 SNR 與位元錯誤率三者之間之關係，並以停止疊代規則之軟式規則與硬式規則進行效能與複雜度之比較，尋找最佳之降低複雜度方案。於本文中，模擬驗證結果以軟式規則效能較為凸出，但是在同時考量複雜度與效能時，則以硬式規則為最佳。

2. 渦輪碼基本架構

1993年由Berrou、Glavieux和Thitimajshima提出一個並行串接的通道編碼機制 [1]、[2]。他們確認此種技術在AWGN (Additive White Gaussian Noise)通道上使用BPSK(Binary Phase Shift Keying)的調變進行傳輸，並把這編解碼技術稱做渦輪編解碼。此技術利用解碼器輸出來改進解碼過程，再加上使用疊代(Iteration)的解碼架構及軟入軟出SISO(Soft-in/Soft-out)之最大事後機率MAP(Maximum a posteriori)演算法，於一可接受計算複雜度下，幾乎可達到接近通道容量之效能。如利用兩個解碼速率(code rate)為1/2的迴旋碼並聯而成渦輪碼解碼器，在信號雜訊比為0.7dB時，位元錯誤率可達 10^{-5} 。因當時只是數值模擬結

果，大多數人持懷疑態度，但往後幾年相關研究重複了上述結果，而且有更多相關理論和實際應用的發展，因此很快成為國際資訊編碼理論界之研究重點[3]。

2.1 渦輪碼編碼器

渦輪編碼器之基本型態包括兩個遞迴系統迴旋RSC (Recursive systematic convolutional)編碼器以及一個交錯器(interleaver)，如圖1所示[1]。其中輸入位元為 u_k ，可編碼出系統位元 x_k^s 與同位元 $1 x_k^{p1}$ 與同位元 $2 x_k^{p2}$ ， k 為時間級。

其中最為常見之基本架構如圖2[4]，擁有3個移位暫存器且限制長度 K (constraint length) 為 4，編碼速率為 $1/2$ 。限制長度 K 所代表的意義為每次輸出受到幾次的輸入影響，並可定義為：

$$K = m + 1 \tag{1}$$

其中 m 表示移位暫存器最大儲存量，於圖2中 $K=4$ ，表示是由3個移位暫存器所組成。限制長度越大則編碼位元前後的相關性越大，而編碼的效能也就越好，然而限制長度越大會使得解碼的時間變長，其所需要儲存資訊的記憶體也會越大。通常渦輪碼所需之限制長度不需要太長即可達到良好效能，一般之限制長度大約為3~5之間。

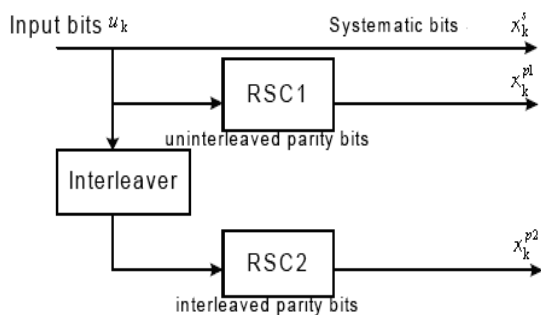


圖 1. 渦輪編碼器架構

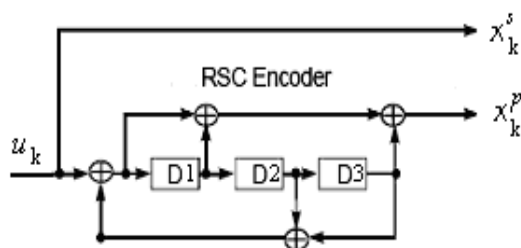


圖 2. RSC 編碼器架構

當輸入位元為 u_k 時，可編碼出系統位元 x_k^s 與同位元 x_k^{p1} ，同時移位暫存器內容(D1、D2、D3)也會跟著輸入位元變化。一般要表示這類結構特性可由三種等效圖表示，分別為編碼樹，格狀圖與狀態圖表示，以下我們使用圖3之格狀圖進行說明[4]。

圖3之 $S_0 \sim S_7$ 代表移位暫存器內容，如圖2之 D1、D2、D3。另外於圖3中，實線代表編碼器輸入位元為 u_k 為 0，虛線則代表為 1，每個輸入訊息於格狀圖中都有其相對應之位置。因此可反推回編碼器輸入與移位暫存器內容之關係，再加以計算原本輸入訊號。

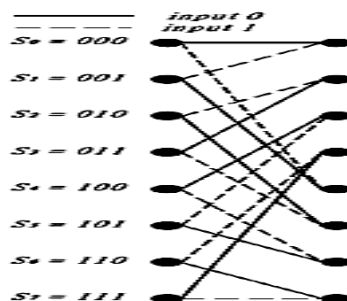


圖 3. RSC 編碼器之格狀圖

2.2 渦輪碼解碼器架構

一般所使用的遞迴渦輪解碼器如圖4所示 [2]，內部構造有兩個系統解碼器(RSC Decoder)，還有兩個交錯器(Inv)與一個解交錯器(Deinv)連接著兩個解碼器以進行資料交換。

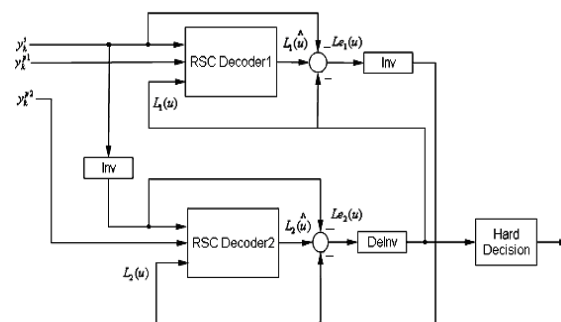


圖 4. 渦輪解碼器架構

典型之RSC解碼器如圖5共有三種輸入，分別為系統位元 y_k^s 、同位元 y_k^p 與事前資訊 $L(u)$ (prior information)，輸出則是事後資訊(posteriori information) $L(\hat{u})$ ， $L_e(u)$ 為外部資訊。RSC解碼器內部架構共有4個方塊，分別為分支路徑計算

器BMC (branch metric calculator)、前向狀態值計算器FSMC (forward state metric calculator)、後向狀態值計算器BSMC (backward state metrics calculator) 與對數相似比計算器 LLRC (log-likelihood ratio calculator)。

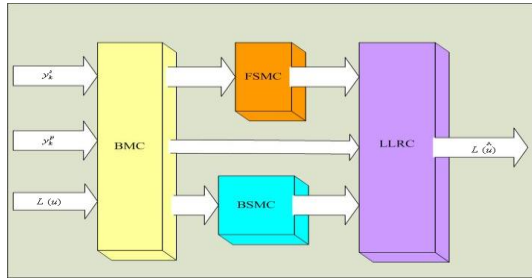


圖 5. RSC 解碼器方塊圖

2.2.1 分支路徑計算器(BMC)

其功能為計算從狀態 s_{k-1} 至 s_k 之分支路徑值 $G_k(s_{k-1}, s_k)$ 。由系統解碼器之三個輸入系統位元 y_k^s 、同位元 y_k^p 、與事前資訊 $L(u)$ ，可以(2)式表示[6]：

$$G_k(s_{k-1}, s_k) = \frac{1}{2} (L_c(u_k) + L_c y_k^s u_k + L_c y_k^p x_k^p) \quad (2)$$

其中 L_c 代表通道可靠值。

2.2.2 前向狀態值計算器(FSMC)

計算出分支路徑值 $G_k(s_{k-1}, s_k)$ 之後，對每個時間級 k ，從編碼器現在之狀態 s' 到下一個狀態 s 的前向狀態值用(3)式表示[7]：

$$A_k(s_k) = MAX^*(A_{k-1}(s_{k-1}) + G(s_{k-1}, s_k)) \quad (3)$$

$k = 1 \dots N - 1$

於MAP演算法中，需使用到大量乘法與指數運算非線性的函數，以實際觀點而言這些非線性函數並不適合於VLSI電路實現。因此轉個方向來思考，在對數領域中乘法會變成加法，指數會消失。在1989年Erfanian 和Pasupathy 利用Jacobi對數理論化簡MAP演算法[7]，在1995年Robertson, Villebrn 及Hoeher 針對MAP演算法在Turbo Code 上的應用提出Log-MAP演算法[8]。

Log-MAP演算法實際上是把MAP演算法中相似值運算全部用對數相似值表示，這樣可將乘法運算就變成加法運算，並且可預先將補償值做

成表而利用查表來完成運算。如下式(4)：

$$MAX^*(x, y) = \ln(e^x + e^y) = MAX(x, y) + \ln(1 + e^{-|x-y|}) \quad (4)$$

由(4)式之定義為取 x 與 y 的最大值再加上一個修正的偏移量，其為變數 $X-Y$ 的函數，可預先計算並儲存在預先建立的尋查表(Lookup Table)裡。因此(4)式中之 MAX^* 我們稱之為 MAX^* 運算子，其基本架構圖如圖6，以此建立基本單元於圖7。

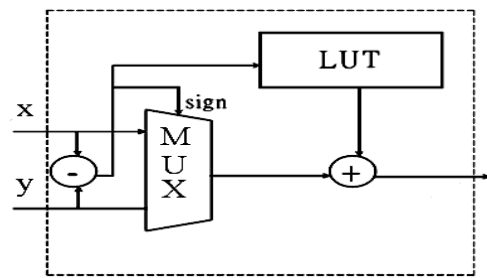


圖 6. MAX*運算子架構圖

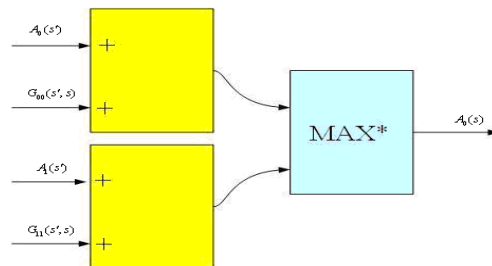


圖 7. 前向狀態值之基本單元

2.2.3 後向狀態值計算器(BSMC)

如同FSMC對每個時間級 k ，從編碼器現在狀態 s' 到下一個狀態 s 的後向狀態值可用(5)式表示[6]：

$$B_{k-1}(s_{k-1}) = MAX^*(B_{k-1}(s_{k-1}) + G(s_{k-1}, s_k)) \quad (5)$$

$k = N \dots 2$

2.2.4 對數相似比計算器(LLRC)

當前向與後向狀態值與路徑值都計算出後，則解碼器找出最大可能性事後機率LLR，其中LLR 可以數學式(6)表示[6]：

$$L(u_k) = MAX_{u_k=+1}^*(A_{k-1}(s') + G_K(s', s) + B_k(s)) - MAX_{u_k=-1}^*(A_{k-1}(s') + G_K(s', s) + B_k(s)) \quad (6)$$

因使用編碼器限制長度為4，因此其格狀圖共有八個狀態。LLRC之輸入增加至20個，分別為

BMC之4個路徑值、FSMC之8個前向狀態值以及BSMC之8個前向狀態值。因此 (4)式必須由(7)式改進MAX*計算方式成(8)式，使其可用於任意偶數輸入情況下。

$$\log(e^a + e^b + e^c + e^d) = \log(e^{\log(e^a + e^b)} + e^{\log(e^c + e^d)}) \quad (7)$$

$$MAX^*(a,b,c,d) = MAX^*[MAX^*(a,b), MAX^*(c,d)] \quad (8)$$

2.3.基礎疊代停止方式

渦輪碼編解碼於先天上因為需要計算每個位元的LLR值，對於如此複雜之計算是不容易實現的，因此有許多學者投入如何降低複雜度的行列。降低複雜度方法可分為許多方式，本文主要著重於疊代停止架構，其基本原理來自Matache等人於2000年所提出之硬式規則(Hard rule)與軟式規則(Soft rule)為基礎[5]。

2.3.1 Hard rule

硬式規則是藉由於每次疊代中比較硬式位元(Hard bit)順序是否完全相同，其中硬式位元之取得方式為判斷 LLR 值之正負，若 LLR 為正時其值為 1，反之則為 0。

(1) Hard rule 1

在這規則中，於每次疊代後進行檢查兩個解碼器的硬式位元的一致性，因此可以於兩者完全相同時提早停止疊代在 $n \leq N_{\max}$:

$$u_{i,1}^n = u_{i,2}^n \quad \forall i, 1 \leq i \leq K \quad (9)$$

其中 $u_{i,1}^n$ 與 $u_{i,2}^n$ 分別為第 n 次疊代的第一個與第二個解碼器之硬式位元， K 是訊息區塊的大小，因此在半次的疊代中就比較一次兩者的硬式位元是否一致，進而將疊代停止，並且將此硬式位元送到解碼器的輸出。

(2) Hard rule 2

在這規則中，疊代停止的方式是依第二個解碼器硬式位元與其前一次疊代硬式位元是否完全相同，若完全相同時可以提早停止疊代在 $n \leq N_{\max}$:

$$u_{i,2}^n = u_{i,2}^{n-1} \quad \forall i, 1 \leq i \leq K \quad (10)$$

這樣的規則不會像 Hard rule 1 在兩個連續的半次疊代中就比較一次硬式位元，而是需要兩個

連續疊代中才做比較。

(3) Hard rule 3

這個規則要滿足停止疊代的條件是需要解碼器硬式位元與前一次及前前一次完全一致，才會於 $n \leq N_{\max}$ 提早停止疊代

$$u_{i,2}^n = u_{i,2}^{n-1} = u_{i,2}^{n-2} \quad \forall i, 1 \leq i \leq K \quad (11)$$

在這規則下，我們可以輕易預估說 Hard rule 3 將會比 Hard rule 2 更加難以停止疊代。

2.3.2 Soft rule

軟式規則是基於比較位元可靠度對臨界值(Threshold)之關係進行判斷此次疊代是否停止，其中臨界值為可調值。令 $\lambda_{i,1}^n$ 為從第一個解碼器的第 n 次疊代、第 i 個位元所產生的位元可靠度， $\lambda_{i,2}^n$ 為第二個解碼器的第 n 次疊代、第 i 個位元所產生的位元可靠度。我們將基於平均與最小的位元可靠度，發展出下列的軟式位元規則。

(1) Soft rule 1

從解碼器二所產生的位元平均的可靠度絕對值與一個門檻值 θ_1 比較，若滿足下列條件則此次的疊代次數 $n \leq N_{\max}$ 將提早停止。

$$\frac{1}{K} \sum_{i=1}^K |\lambda_{i,2}^n| \geq \theta_1 \quad (12)$$

(2) Soft rule 2

從解碼器二所產生的位元最小的可靠度絕對值與一個門檻值 θ_2 比較，若滿足下列條件則此次的疊代次數 $n \leq N_{\max}$ 將提早停止。

$$\min_{0 < i \leq K} \left[|\lambda_{i,2}^n| \right] \geq \theta_2 \quad (13)$$

(4) Soft rule 3

解碼器一與二所產生位元平均可靠度絕對值與一個門檻值 θ_3 比較，若這個規則滿足則提早停止於 $n \leq N_{\max}$ 。

$$\min_{0 < i \leq K} \left[\frac{1}{2} |\lambda_{i,1}^n + \lambda_{i,2}^n| \right] \geq \theta_3 \quad (14)$$

3. 模擬結果與分析

3.1 實驗平台

本文設計流程採用由上而下設計法如圖 8 所示，先從定義系統規格開始，再發展高階演算法模型。演算法確立後，使用 Visual C++ 6.0 進行演算法測試，以此為基礎再使用 QUARTUS II 5.0 發展平台及 VHDL 語言進行硬體實作並驗證其正確性。

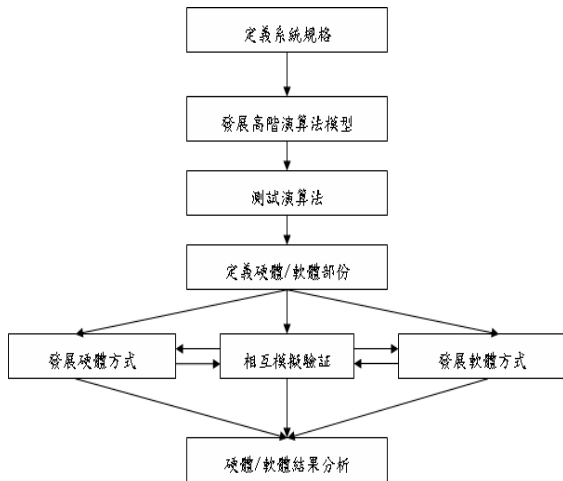


圖 8. 發展設計流程圖

本文之研究方法分為硬體與軟體兩部份，兩者之間相互驗證。硬體方面利用 Altera 公司的 Quartus II 5.0 進行各項模擬與分析，軟體驗證平台則是使用 Visual C++ 6.0 進行演算法之驗證。

3.2 參數介紹

我們針對所使用之規格作設定，模擬在之前所提出的架構，表 1 為模擬的參數設定：

表 1 效能模擬參數設定表

實驗平台	Quartus II 5.0 與 Visual C++ 6.0
計算方式	Fixed point 與 Float point
調變方式	BPSK
雜訊通道	AWGN Channel
Code rate	1/3
限制長度	4
訊框大小(Frame size)	512
移動視窗大小	64
解碼器種類	SW-Log-MAP
信號雜訊比	0~2 db
疊代停止方式	Hard rule1、Hard rule2 Soft rule1~ Soft rule3

3.3 slide window 之渦輪解碼器架構

由前述渦輪碼演算法可知，若要計算出 LLR 值，則需先求解出後向狀態值，但是後向狀態值之計算需要從最後之位元逆推所得。因此在製作渦輪碼解碼器上，不僅需要大量的記憶體儲存前向與後向狀態值，同時在實作上也要等待最後位元接收完畢後才可解碼，這會嚴重地影響解碼的效率。

針對以上兩個問題，Dawid 等人於 1995 年提出了移動視窗(slide window; SW)架構 [10]，此架構可以有效解決渦輪碼解碼器記憶體消耗以及解碼效能的問題，其解決方式為將解碼長度由原本的訊框大小 N 切割成許多長度 L 之區塊，再分段進行解碼。圖 9 為渦輪碼之 slide window 示意圖，前向狀態值之計算從 0 開始往後計算，後向狀態值則需分開計算初始值 $B1$ 及計算值 $B2$ ，因此若要獲得 $L \sim 0$ 之計算值 $B2$ ，必須先從時間 $2L$ 開始往回推出初始值 $B1$ ，再從 L 開始計算 $B2$ 。

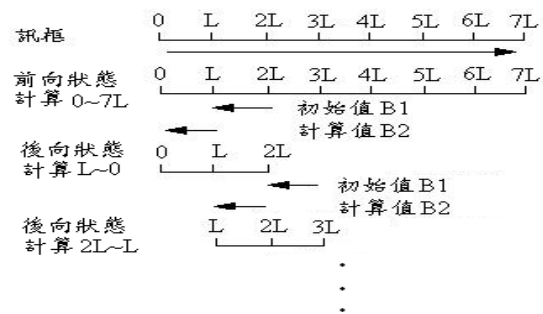


圖 9. 渦輪碼之 slide window 示意圖

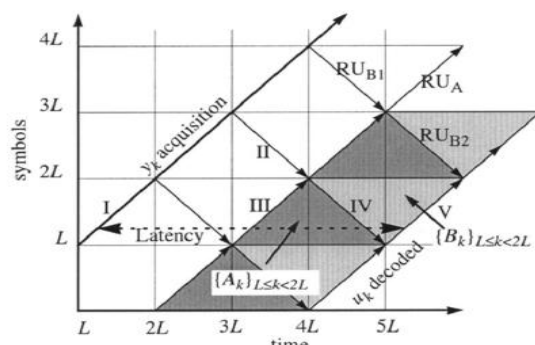


圖 10. SW-Log-MAP 架構表示圖

依照 slide window 架構，我們可設計出 slide window 時間對應圖如圖 10。其訊框之 $L \sim 2L$ 部份解碼動作如下：

- I. 於時間 $L \sim 2L$ 時，儲存系統位元 y_k^s 儲存於

Frame Memory。

- II. 於時間 $3L \sim 4L$ 時，進行 $3L \sim 2L$ 之初始值 $B1$ 之計算。
- III. 於時間 $3L \sim 4L$ 時，計算 $L \sim 2L$ 之前向狀態值並儲存於前向狀態記憶體。
- IV. 於時間 $4L \sim 5L$ 時，計算 $2L \sim L$ 之後向狀態值 $B2$ 同時解出 LLR 值並儲存於 LIFO 記憶體。
- V. 由 LIFO 讀出 $L \sim 2L$ 之 LLR 值。

參考圖10，修改圖5架構成圖11。其內部結構可劃分為Frame Memory、FSMC、前向狀態記憶體(α RAM)、BMC、BSMC1、BSMC2、LLRC 以及LIFO。

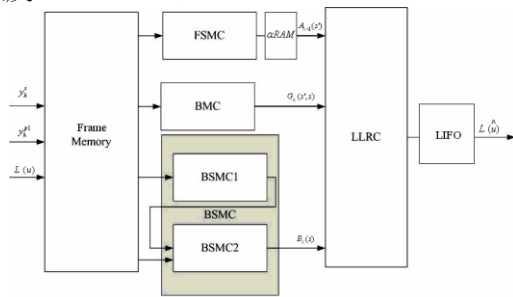


圖 11. SW-Log-MAP 渦輪解碼器架構圖

3.4 渦輪碼解碼器硬體實現與效能分析

使用固定點模型於 FPGA 上設計，合成晶片為 Cyclone II 家族 EP2C70F896，基本規格如表 2。以 Quartus II 進行合成後如圖 12，其資料位元如表 3。

表 2 EP2C70F896C6規格表

Feature	EP2C70
LEs	68,416
M4K RAM Blocks	250
Total RAM Bits	1,152,000
Embedded 18x18 Multipliers	150
PLLs	4
Maximum User I/O Pins	622
Available Packages	672-pin FBGA 896-pin FBGA

表 3 固定點模式SW-Log-MAP架構之位元寬度

	整數位元	小數位元
接收位元 y_i^s, y_i^p	8	3
事前資訊 $L(u)$	11	3
分支路徑 $G(s)$	8	3
前向狀態值 $A(s)$	11	3
後向狀態值 $B(s)$	11	3
對數相似比 LLR	11	3

利用 Timing Analyzer Tool 分析 t_d 與 f_{max} ，compiler tool 分析影響晶片面積 Logic element 使用個數、佔用晶片記憶體大小，以及使用 PowerPlay Power Analyzer Tool 分析使用中元件功率消耗，結果如表 4 所示。

表 4 FPGA 元件之效能

	LOGIC ELEMENT	MEMORY BITS	t_d	f_{max}	POWER DISSIPATION
Memory	216	6812	1553 ns	260.0 1 #Bz	220.31 mW
BMC	77	0	14.21 ns		217.06 mW
FSMC	1077	14336	12.833 ns	31.14 #Bz	210.82 mW
BSMC	1461	0	0.447 ns	58.84 #Bz	220.16 mW
LLRC	854	0	27.234 ns		224.81 mW
Decode r	3545	21248	20.958 ns	13.51 #Bz	214.47 mW

由表 4 可得出解碼器之 t_d 為 29.958ns 而 f_{max} 為 13.51 MHz。其中影響效能最為深遠之元件為 LLRC，其 t_d 大小為 27.234ns，不但遠比其他元件之 t_d 來得大，且接近解碼器之 t_d 。LLRC 之架構主要以 adder、MAX*、sub 這三種電路方塊所組成，最終計算結果需經 4 次電路方塊之運算。因此管線(pipeline)應為有效之改善方式，於電路方塊間加入暫存器並加以管線化後，應可改善 t_d 提升 f_{max} 。

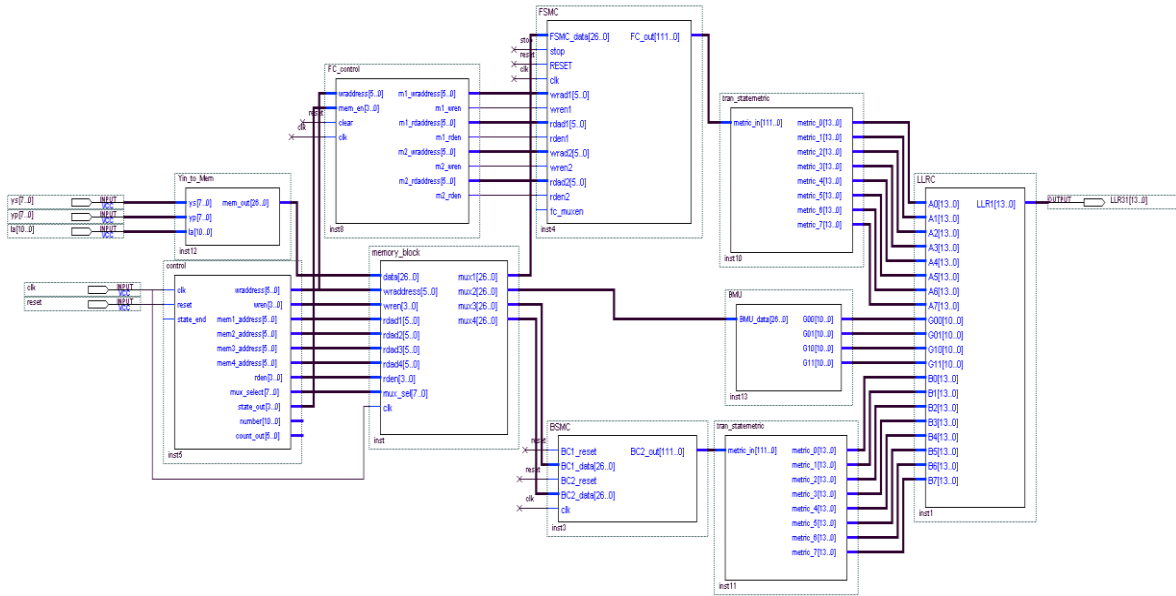


圖 12. 已模組化之SW-Log-MAP渦輪解碼器

3.5 Log-MAP 浮點數模型停止法則分析

以疊代次數固定 20 次所模擬出結果如圖

13:

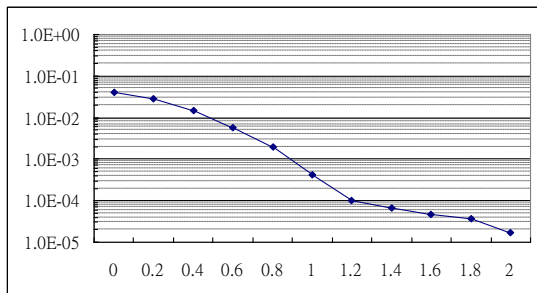


圖 13. 浮點數模型 SNR 與 BER 關係圖

由上圖可知，於 SNR(訊號雜訊比)極小時其 BER(位元錯誤率)接近 0.5，但隨著 SNR 增加信號強度也隨之增強，因而易於解碼其 BER 便遞減，於 SNR 為 2.4 之後，BER 模擬結果為 0。

依據 2.3 節所述的規則進行模擬與比較，觀察其 BER 與必要疊代次數，圖 14、15 為硬式規則 BER 與疊代次數圖，圖 16、17 為軟式規則 BER 與疊代次數圖。

由圖 14 得知，在硬式規則部份 HARD1 與 HARD2 於 SNR 在 1.5 之前 BER 大約相同，在 1.5

之後 HARD1 表現稍差但仍低於 fixed 20，而在降低疊代次數上 HARD1 比 HARD2 平均降低 4.57%，HARD1 最多比 HARD2 少 0.56 次，兩者都接近於 BEST(於 BER 為 0 時馬上停止疊代)疊代次數。因此 HARD1 方式將優先考慮加入固定點模型設計中。

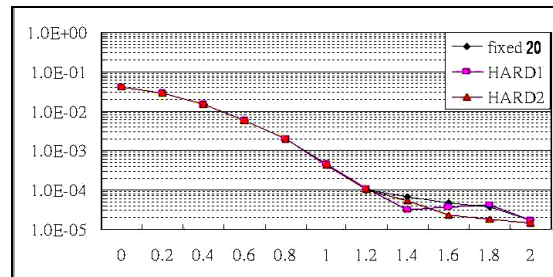


圖 14.浮點數模型硬式規則之 SNR 與 BER 關係圖

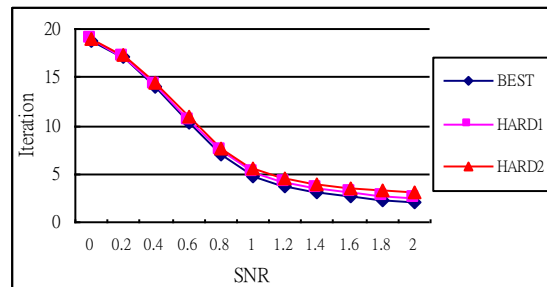


圖 15.浮點數模型硬式規則 SNR 與疊代次數關係圖

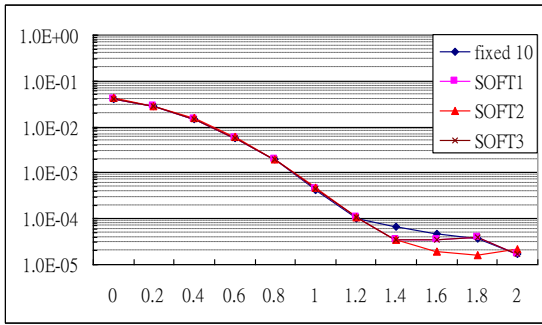


圖 16.浮點數模型軟式規則 SNR 與 BER 關係圖

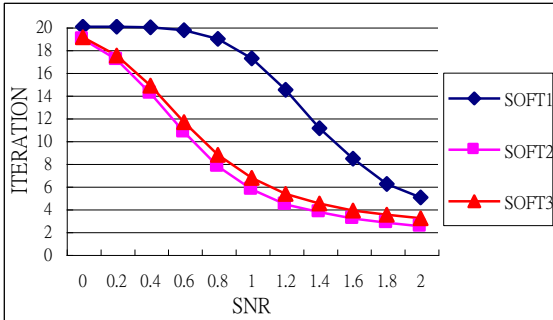


圖 17.浮點數軟式規則 SNR 與疊代次數關係圖

3.6 SW-Log-MAP 固定點模型設計與分析

圖18為固定點模型硬式規則解碼效果比較。可以清楚看到，硬式規則BER略比固定疊代10次較為提高，但是它可以有效降低疊代次數。如圖19所示，大於1db以上可減少約平均70%以上之疊代次數，可有效降低運算複雜度與時間，而小於1db時仍有降低約0%~25%。

圖(20)~(23)為軟式規則於不同臨界值(threshold)解碼效果比較。我們可以發現，軟式規則降低疊代次數略比硬式規則好，但軟式規則2之BER則表現不佳。

就以上分析比較，可看出以軟式規則1改善程度最好，硬式規則1比硬式規則2較佳。但在製作疊代停止器時，以硬式規則之方式會較為簡易且運算複雜度較低，不會增加太多額外成本。

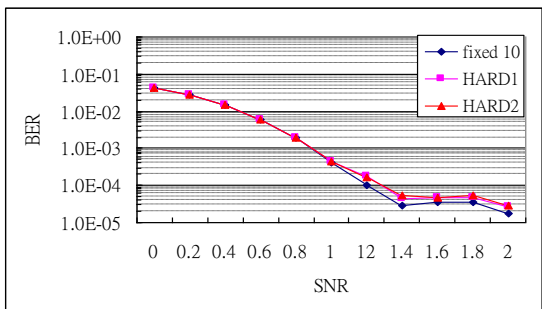


圖 18.固定點模型設計硬式規則解碼效果

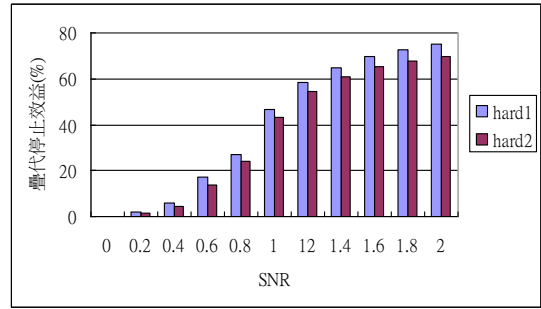


圖 19.固定點模型設計硬式規則降低疊代次數

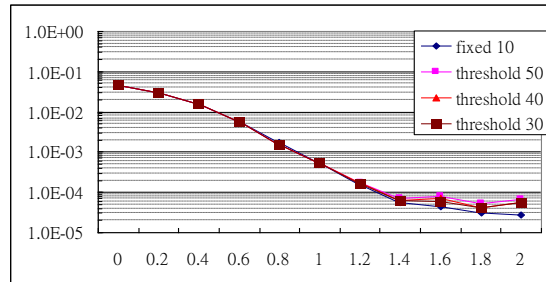


圖 20.固定點模型設計軟式規則 1 之解碼效果

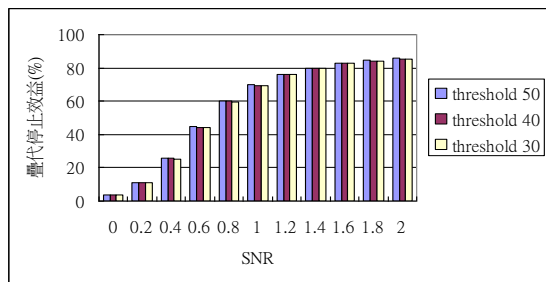


圖 21.固定點模型軟式規則 1 降低疊代次數

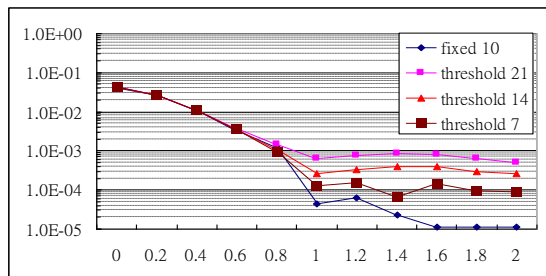


圖 22.固定點模型軟式規則 2 之解碼效果

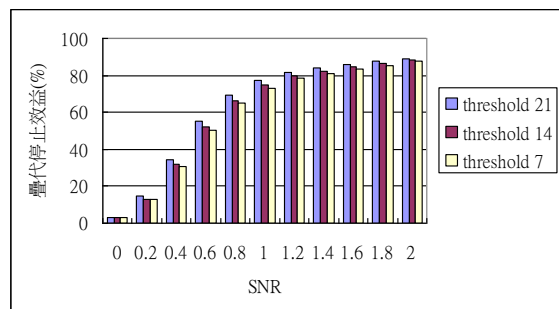


圖 23.固定點模型軟式規則 2 降低疊代次數

4. 結論

渦輪碼近年來已被使用在第三代行動電話上，由於傳統 MAP 解碼器必須使用大量記憶體來暫存計算結果，在實際應用上通常使用移動視窗方式減少 MAP 架構對記憶體之需求，對於行動通訊上面功率消耗也是一項相當重要的環節。

因此於本文中，我們使用 SW-Log-MAP 的常用 VLSI 架構，並加入疊代停止為基礎，進行硬體與軟體之驗證分析。硬體使用 Cyclone II 家族 EP2C70 晶片建構渦輪解碼器，其 t_d 為 29.958ns 而 f_{max} 為 13.51 MHz。由軟體驗證分析顯示，於固定點模型疊代停止設計中，硬式規則與固定疊代 10 次相比，大於 1db 以上明顯可減少約平均 70% 以上之疊代次數，有效降低運算複雜度與時間，小於 1db 時仍有降低約 0%~25%。

參考文獻

- [1] 吳松翰，民國93年，低功率渦輪碼解碼器之設計與實現，碩士論文，國立中山大學資訊工程所。
- [2] 黃修進，民國94年，對數MAP 解碼器的實現架構設計，碩士論文，國立高雄第一科技大學電腦與通訊工程所。
- [3] Bernard Sklar，民國 93 年 9 月，數位通訊：原理與應用，全華圖書。
- [4] 3GPP TS 25. 212, "Multiplexing and channel coding" Release 1999, v3.6.0, 2001.
- [5] A. Matache, S. Dolinar, and F. Pollara, "Stopping Rules for Turbo Decoders", JPL TMO Progress Report 42-142, 2000.
- [6] Claude Berrou, Alain Glavieux and Punya Thitimajshima, "Near shannon limit error – correcting coding and decoding : turbo-codes (1)" Proc. ICC '93, pp. 1064-1070, 1993.
- [7] Claude Berrou, and Alain Glavieux, "Near optimal error correcting coding and decoding: Turbo-codes," IEEE Trans. Commun, pp.1261-1271, Oct. 1996.
- [8] Emmanuel Boutillon, Warren J. Gross, Student Member, "VLSI Architectures for the MAP

Algorithm" IEEE Trans. on Commu., vol. 51, No. 2, Feb. 2003.

- [9] H. Dawid, H. Meyr, "Real-time Algorithms and VLSI Architectures for Soft Output MAP Convolutional Decoding," in Proc. Personal, Indoor and Mobile Radio Communications, PIMRC'95. Wireless: Merging onto the Information Superhighway. Vol. 1, pp. 193-197, 1995.
- [10] J. A. Erfanian and S. Pasupathy, "Low-complexity parallel -structure symbol bysymbol detection for ISI channels," in Proc. IEEE Pacific Rim Conf. on Commu, Computers and Signal Processing, pp. 350-353, 1989.

