

適用於高數位用戶迴路之 DMT 數位 IP 模組設計及實現 (II)

Design and Implementation of Digital IP for DMT Engine in High-Speed DSL Applications (II)

計畫編號：NSC 90-2218-E-002-040

執行期限：90/08/01 ~ 91/07/31

主持人：吳安宇 副教授 Email: andywu@cc.ee.ntu.edu.tw

執行機構：國立台灣大學電機工程學系

一、中文摘要

在 DMT Modulation Engine 中，高點數的反快速傅利葉轉換/快速傅利葉轉換 (IFFT/FFT)、多維格子迴旋碼 (Multi-dimensional TCM) 編解碼器以及李德-所羅門 (Reed-Solomon) 編解碼器都是重要的核心模組。由於這些模組都具有高計算複雜度 (Computational Complexity) 的特性；若利用數位信號處理器 (DSPs) 來加以實現，這些模組的運作將會佔據太多的系統資源，而且無法達到即時運算的目的 (Real-time Processing)。因此利用超大型積體電路 (VLSI) 來實現這些模組是比較適合的作法。因此，本子計畫研究的重點在於針對這些 DMT Modulation Engine 中的重要核心模組來設計高效能/低功率的數位 IP (Intellectual Property)。

在這個子計畫中，首先我們將對各個模組做演算法上的分析，以期在演算法階層 (Algorithmic Level)，以設計空間搜尋 (Design Space Exploration) 方式，改進計算複雜度及節省記憶體空間/頻寬。接著針對其 VLSI 架構作推導，更進一步改善模組之速度/功率/面積，最後將落實於 VLSI 電路實現。計畫目標為建立一組高效能/低功率的數位 IP 模組，供子計畫二之 DMT 基頻架構使用。同時，我們並以可重設組態的 (Reconfigurable) IP 為研究之重點，以達到 IP 再使用 (Reuse) 及快速原型設計 (Rapid Prototyping) 之目的。

關鍵詞：

離散多頻調變，反快速傅利葉轉換/快速傅利葉轉換，多維格子迴旋碼編解碼器，李德-所羅門編解碼器，矽智慧區塊。

二、英文摘要

High-point IFFT/FFT, 4D-TCM codec and Reed-Solomon codec are the kernel modules in DMT modulation Engine. Due to the massive computational complexity, the implementation of these modules by DSP processor will dominate the computational complexity and cannot achieve real-time data processing in practical implementations. Hence, using VLSI to implement those digital IPs would be a better solution. The main goal of this project is to design high-performance /low-power digital IP modules in the DMT engine.

In this project, we will first analyze the algorithms of each IP module. By applying "design space exploration", we seek to find optimized design to reduce the computational complexity and memory space/bandwidth at the algorithmic level. At the architectural/circuit level, we will derive effective VLSI architectures and circuits to further improve the area/speed/power performance. By the end of the project, we will implement these IP modules down to ASIC level. The final goal is to create a set of high-speed/low-power digital IP modules for the DMT baseband architecture developed in sub-project 2, and link with other modules of the group project. Also, to achieve the goal of IP reuse and rapid prototyping, we will also explore the reconfigurable structures for these IPs.

Keywords :

DMT, IFFT/FFT, TCM, Reed-Solomon, Intellectual property (IP).

三、計畫緣由與目的

最近網際網路(Internet)的進步導致迫切需要較高的資料傳輸率，為了解決傳統雙絞電話線的傳輸瓶頸，一些調變/解調的方法被提出，包括 CAP、DMT 和 QAM，離散多頻(DMT)調變/解調方法是非對稱數位傳輸系統(ADSL)上的標準傳輸技術，更對於非常高速數位用戶迴路(VDSL)提出類似的 SDMT 技術。DMT 利用大量的前瞻性 DSP 技術來達到調適性速率(rate-adaptive)的資料傳輸，但是它的計算複雜度卻遠超過其他調變/解調的方法。

在 DMT Modulation Engine 中，高點數的反快速傅利葉轉換/快速傅利葉轉換(IFFT/FFT)、多維格子迴旋碼(Multi-dimensional TCM)編解碼器以及李德-所羅門(Reed-Solomon)編解碼器都是重要的核心模組。實際上，這些模組有具有高計算複雜性的共同特性。在硬體實現中，利用 DSPs 來實現這些模組的即時處理是幾乎不可能，因此，更好的解決辦法將是利用 VLSI 來實現這三個核心模組。

四、研究方法與成果

在這個子計畫中，首先我們將對各個模組做演算法上的分析，以期在演算法階層(Algorithmic Level)，以設計空間搜尋(Design Space Exploration)方式，改進計算複雜度及節省記憶體空間/頻寬。接著針對其 VLSI 架構作推導，更進一步改善模組之速度/功率/面積，最後將落實於 VLSI 電路實現。計畫目標為建立一組高效能/低功率的數位 IP 模組：

● IFFT/FFT 模組

1. 低複雜度快速傅立葉轉換器架構

我們對於 DMT[1]系統，提出一套低複雜度 FFT/IFFT 演算法及其硬體實現架構，能大幅降低 FFT/IFFT 處理器的硬體實現成本。演算法如下：

$$X(0) = X(N) = 0$$

$$X(k) = X^*(2N-k) \quad \text{for } k=1,2,\dots,N-1 \quad (1)$$

$$x(n) = \frac{1}{2N} \cdot \left[\sum_{k=0}^{2N-1} X(k)W_{2N}^{nk} \right], \quad \text{for } n=0,1,\dots,2N-1 \quad (2)$$

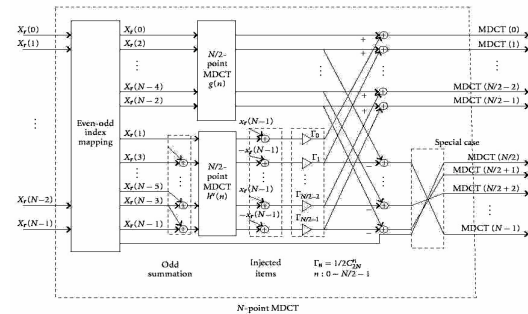
$$W_{2N}^{nk} = \exp(-j \frac{2\pi nk}{2N}) = \cos(\frac{2\pi nk}{2N}) - j \sin(\frac{2\pi nk}{2N}) \quad (3)$$

$$x(n) = \frac{1}{2N} \cdot \left[\sum_{k=0}^{N-1} X(k)W_{2N}^{-nk} + \sum_{k=N}^{2N-1} X(k)W_{2N}^{-nk} \right] \quad (4)$$

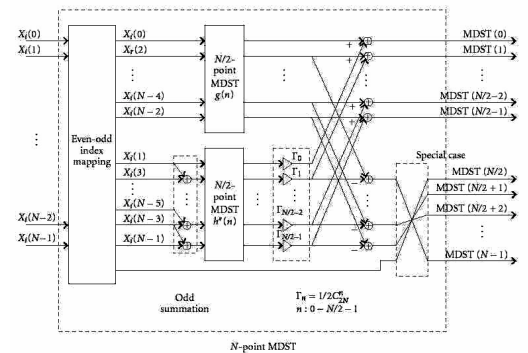
將(4)中的(3)，可以簡化(4)為

$$x(n) = \frac{1}{N} \cdot \left[\sum_{k=0}^{N-1} X_r(k) \cos \frac{2\pi nk}{2N} - \sum_{k=0}^{N-1} \sin \frac{2\pi nk}{2N} \right] \\ = \frac{1}{N} \cdot [MDCT(n) - MDST(n)], \quad (5) \\ \text{for } n = 0,1,\dots,2N-1$$

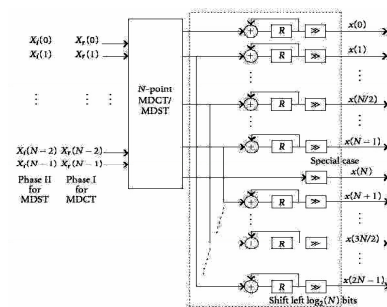
從(5)中，IFFT 被分解成兩個實數運算。其一近似離散餘弦轉換(Discrete Cosine Transform)，另一則為近似離散正弦轉換(Discrete Sine Transform)。分別命名為 Modified DCT(MDCT)及 Modified DST(MDST)。MDCT 及 MDST 只作實數的運算。其硬體架構為圖一、圖二、圖三。



圖一、N-point MDCT 架構



圖二、N-point MDST 架構



圖三、IFFT 處理器架構圖

2. 硬體實現結果

下圖為以 MDCT 及 MDST 來發展之 FFT/IFFT 處理器與 Cooley-Tukey[2]之硬體實現比較表。圖四為乘法運算次數的比較表，圖五為加法運算次數的比較表。

| N | IFFT | | | FFT | | |
|------|-----------------------------------|-------------------------------|-------|-----------------------------------|-------------------------------|-------|
| | Cooley-Tukey (O ₁) | Our (O ₂) | CR | Cooley-Tukey (O ₁) | Our (O ₂) | CR |
| 256 | 4N log ₂ 2N | N log ₂ N - 2N + 2 | | 4N log ₂ 2N | N log ₂ N - 2N + 2 | |
| 256 | 9216 | 1538 | 0.169 | 9216 | 1538 | 0.169 |
| 512 | 20480 | 3586 | 0.175 | 20480 | 3586 | 0.175 |
| 1024 | 45056 | 8194 | 0.182 | 45056 | 8194 | 0.182 |
| 2048 | 98304 | 18434 | 0.188 | 98304 | 18434 | 0.188 |
| 4096 | 212992 | 40962 | 0.192 | 212992 | 40962 | 0.192 |
| 8192 | 458752 | 90114 | 0.196 | 458752 | 90114 | 0.196 |

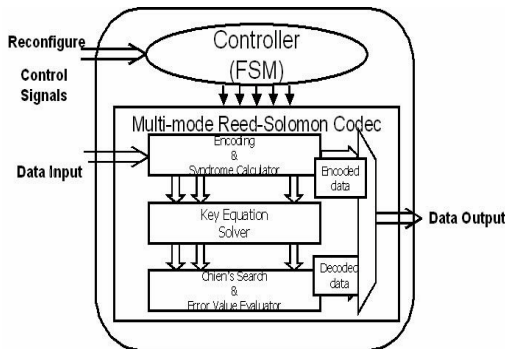
圖四、乘法運算次數比較表

| N | IFFT | | | FFT | | |
|------|-----------------------------------|-----------------------------------|-------|-----------------------------------|-------------------------------|-------|
| | Cooley-Tukey (O ₁) | Our (O ₂) | CR | Cooley-Tukey (O ₁) | Our (O ₂) | CR |
| 256 | 6N log ₂ 2N | (9/2)N log ₂ N + N + 1 | | 6N log ₂ 2N | (9/2)N log ₂ N + N | |
| 256 | 13824 | 9473 | 0.685 | 13824 | 9472 | 0.685 |
| 512 | 30720 | 21249 | 0.692 | 30720 | 21248 | 0.692 |
| 1024 | 67584 | 47105 | 0.697 | 67584 | 47104 | 0.697 |
| 2048 | 147456 | 103425 | 0.701 | 147456 | 103424 | 0.701 |
| 4096 | 319488 | 225281 | 0.705 | 319488 | 225280 | 0.705 |
| 8192 | 688128 | 487425 | 0.708 | 688128 | 487424 | 0.708 |

圖五、加法運算次數比較表

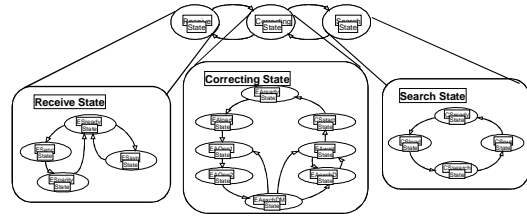
● Reed-Solomon FEC Codec

我們提出了一個符合多重通訊系統規格之可重設組態多模式 RS(n, k, t)架構 [3]，其錯誤的更正能力 $0 \leq t \leq 8$ ，可變的碼字(codeword)為 $0 \leq n \leq 255$ ，其完整架構如圖六所示。此種設計最大的優點乃在於縮短重新設計不同規格 RS codec 的時程，以達到 IP reuse 及快速雛型設計(rapid prototyping)的目的。



圖六、所提出之可重設組態多模式 RS 架構

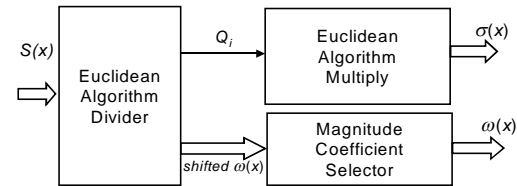
在我們所提出的 RS 架構中，主要分為 softcore 和 hardcore 兩部份。在 softcore 裡，輸入參數 n、k、t 的值，利用 Finite State Machine (FSM)產生控制訊號來控制 hardcore 中的編碼、解碼運算單元。Softcore 示意圖如圖六所示。



圖七、編碼器與 Syndrome 計算器整合架構圖

在 Hardcore, RS 編碼器是以 a(x)-based 來作設計，其相較於 g(x)-based 而言，具有較佳的規律性，與解碼器中的 syndrome calculator 可以達到硬體共享 (hardware sharing) 的功用，因此較適合來作可重設性電路的設計。

致於 RS 解碼器的部份，我們選擇使用 Euclidean GCD Algorithm 來解 key equation，此架構主要分為 Euclidean Algorithm Divider、Euclidean Algorithm Multiply 及 Magnitude Coefficient Selector 三個部份。如圖八。



圖八、RS 解碼器架構圖

由於此演算法採用的是疊代(iterative)的解碼方式，因此其錯誤更正能力彈性較大。如此較能符合我們所提出的可重設組態多模式 RS 架構。圖九所示為實現後與其他架構所作的比較表。

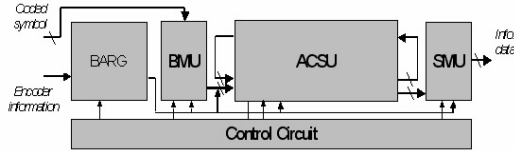
| | RS(255,239) [6] | RS(n, k, t) for ADSL [7] | Our Multi-mode RS(n, k, t) Codec |
|---------------------------|--------------------------------|-----------------------------------------------------|--------------------------------------------------|
| Gate count | 122,630 Tr./4 ≅ 30,658 gate | 220,841 tr./4 ≅ 55,210 gate | 34,647 gate (289x8 FIFO register) |
| Latency | 287 | 321 | 3m+4m+4m (Max: 7508) n+3t+10 (Max: 289) |
| Clock Rate | 41 MHz (0.25um) | 75 MHz | 48 MHz 100 MHz (0.35um) |
| Data Rate | 330 Mbps | 600Mbps | 48 Mbps 800 Mbps |
| Reconfigurable Capability | n = 255 t = 8 | n = 2 ^m -1 m ≤ 8, t ≤ 8 bit-serial | 0 ≤ n ≤ 255 0 ≤ t ≤ 8 |

圖九、硬體與效能比較表

● Trellis-Coded Modulation Codec

在 TCM 迴旋碼解碼器的設計中，使用最大相似演算法的唯特比解碼器已被廣泛地使用。然而在不同的應用中，不同的參數往往導致在唯特比解碼器時，必須重頭來過，使得設計的過程耗時又耗力。因此我們提出一個可規劃式的唯特比解碼器，希望只需更改模組間的控制電路，便可應用到不同規格的設計上；如圖十，

我們加上一個稱為 BARG(BM-to-ACS Routing Generator)的模組，利用改變傳統唯特比解碼器中BMU和ACS之間的一些線路更動和加上適當的邏輯電路，即可適用於不同參數之應用。



圖十、可規劃式唯特比解碼器架構

然而，和傳統的唯特比解碼器一樣，BMU、ACSU、SMU 等三個模組亦為設計上重要的考量[7]；其中在 BMU 裡，我們採用 soft decision 的方式來計算其值，這樣一來可以較使用 hard decision 的方式得到 2.2dB 的編碼增益值；而在 ACSU 這個模組裡，處理單元(PE)的數目將會影響其面積、速度等；我們在比較了各種方法之後(如圖十一)，採取較平行處理、適合高速運作的方式，即該表中的“ Full PEs ”方法。最後是 SMU 模組的設計[8]，這裡的重點是在於資料在記憶體存取、運作的方式，調查所得的各項方法如圖十二所示，我們採取的是 One-P 的方法。

| | Single PE | Full PEs | | N_p PEs | |
|---------------------|-----------|----------|------------------|---------------|------------------|
| | | Medium | Medium | One-Time Step | In place |
| Routing | Medium | Medium | Medium | Complex | Simple |
| Path Memory | 2^*N^*B | N^*B | $(2N-2^{2b})^*B$ | N^*B | $(2N-2^{2b})^*B$ |
| PE Using Percentage | 100% | 100% | 100% | 100% | 50% |
| Through-put Rate | $2/N$ | 1 | $2N_p/N$ | $2N_p/N$ | $2N_p/N$ |
| Area ratio | 1 | $N/2$ | N_p | N_p | N_p |

圖十一、處理單元數目對 ACSU 的影響

| Method | Reg Exch | Traceback | | | Trace-Forward | Sliding Windows | Hybrid RE & TB |
|-----------------------|-------------------------|-----------------------|------------------------------------|-------------------------------------|-------------------------------------------------|-----------------|---------------------------------|
| | | One-P | k-P even | k-P odd | | | |
| Memory size | D^*N | $\frac{k+1}{k-1}DN$ | $\frac{2k}{k-1}DN$ | $\frac{2k-1}{k-1}DN$ | $2^*D^*N^*m$ | D^*N | $D^*N^*L^*N$ |
| Type of cell required | Dual-port mux + wiring | Single-port SRAM | | | | | SRAM + DP + mux + wiring |
| latency | D | $\frac{k+1}{k-1}D$ | $\frac{2k}{k-1}D$ | $\frac{2k-1}{k-1}D$ | $2D$ | D^* | $D+L+DL$ |
| Write Bandwidth | D^*N | | N | N | N | N^*D | L^*N+N |
| Read Bandwidth | D^*N | | k | k | 1 | 1 | L^*N+N |
| Total | D^*2N | | $k+N \approx N$ | $k+N \approx N$ | $1+N \approx N$ | $1+N^*D$ | $2(L+1)^*N$ |
| Others | Large power consumption | Simple address coding | k=1 point 1-directional addressing | k=1 point Bi-directional addressing | When k=3 in TB, TF's latency is short than TB's | | Suitable for processor approach |

N : number of PEs D : trace back depth k : memory bank - 1 L : state width

圖十二、SMU 的不同實現方法比較

在決定了各個模組的實現方法之後，我們使用 Matlab 來做功能上的驗證。我們以一個參數為(2,1,5)的迴旋碼為例，可以得到下圖的模擬結果，確認其結果為正確的。

```

Circle 1
input 0 0 1 0 0 0 0 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1 1
output 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
error: 10 00 00 00 00 00 01 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00

Circle 2
input 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0
output 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
error: 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 11 10

Circle 3
input 1 1 0 0 1 1 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 1 0 0 1 0
output 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
error: 01 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10

Circle 4
input 1 0 0 0 1 0 1 0 1 0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1
output 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
error: 00 00 01 10 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10

Circle 5
input 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 1 1 0
output 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0
error: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00
    
```

圖表四、Matlab 模擬結果

五、結論與討論

在本子計劃中，我們針對這些 DMT Modulation Engine 中的重要核心模組來設計高效能/低功率的數位 IP 模組。透過演算法分析降低硬體實現的複雜度，並藉由 VLSI 架構的推導，進一步改善模組之速度/功率/面積，最後將落實於 VLSI 電路實現。

六、參考文獻

- [1] ANSI Standard T1.143, "Network and customer installation interface-Asymmetric digital subscriber line (ADSL) metallic interface," 1995.
- [2] J.W. Cooley and J. W. Tukey, "A algorithm for the machine calculation of the complex Fourier series," Math. Comp., vol. 19, pp. 297-301, April 1965.
- [3] H. Y. Hsu and A. Y. Wu, "VLSI Design of a Reconfigurable Multi-mode Reed-Solomon Codec for High-Speed Communication Systems," In Proceedings. IEEE Asis-Pacific Conference on ASICs, pp.359-362, 2002.
- [4] G. Fettweis, M. Hassner, "A Combine Reed-Solomon Encoder and Syndrome Generator with Small Hardware Complexity," *Circuits and Systems, ISCAS 92 Proceedings*, vol. 4, pp. 1871-1874, 1992.
- [5] H. Lee, M.L. Yu, and L. Song, "VLSI Design of Reed-Solomon Decoder Architecture," *ISCAS 2000 Proceedings Circuits and Systems*, pp. v-705-708, 2000.
- [6] L. F. Wei, "Trellis-coded modulation with multi-dimensional constellations," in *IEEE Trans. Information Theory*, vol. 33, pp. 483-501, July 1987
- [7] H. L. Lou, "Implementing the Viterbi algorithm," in *IEEE Signal Processing Mag.*, vol. 12, no.5, pp. 42-52, Sept. 1995.
- [8] G. Feygin and P. G. Gulak, "Survivor sequence memory management in Viterbi decoders," in *IEEE Trans. on Communication*, vol. 41, no.3, pp. 425-429, Mar.1993.