

國立清華大學 電機工程學系

實作專題研究成果摘要

Low-Density Parity-Check Codes
and Applications

錯誤更正碼編解碼器實作
及其應用

專題領域：通訊

組 別：A113

指導教授：趙啟超 教授

組員姓名：龔柏丞

研究期間：109年 9月14日至 110年 6月底止，計 10個月

一、報告摘要

低密度奇偶校驗碼(Low-Density Parity-Check Code, LDPC code)是現代常用的一種錯誤更正碼，其目的為更正訊息在傳輸時，因各種雜訊而產生的錯誤。本次專題的目的正是要藉由 LDPC 編碼器/解碼器(encoder/decoder)的實作，探討均等錯誤保護碼(Equal error protection code, EEP code)與不均等錯誤保護碼(Unequal error protection code, UEP code)對於錯誤的更正能力。

除此之外，本次的實作成果也被應用於快閃記憶體(Flash memory)當中的 NAND flash，探討錯誤更正碼對於記憶體儲存的更正能力，並藉由 UEP code 讓不同位元之間具有不相等的更正能力，藉此對較容易出錯的訊息做出較強的保護。

在本次的專題研究當中，C 語言是主要的模擬工具，而 MATLAB 則作為數據圖像化的媒介，藉此模擬訊息在通道中的傳輸過程、以及在 memory cells 裡的存取過程。

藉由模擬所得出的數據可發現：UEP code 的架構確實能有效的針對不同 level 給予具有明顯區別的保護程度；而將 UEP code 應用在 NAND flash 的各個 page 上也能得到很好的訊息保護能力、並針對 BLER 較高的部分進行有效的保護，讓 memory cells 的錯誤率在環境相同的情況下顯著降低。

UEP code 針對不同情境有著獨特的架構，且經過實驗模擬後發現其確實能拉開不同 level 之間的保護能力，因此對於諸如 memory cells 等不同位元錯誤率不相等的情境，UEP code 是一種好的選擇。

二、報告內容

(一)前言

低密度奇偶校驗碼(Low-Density Parity-Check Codes, LDPC code)是現代常用的一種錯誤更正碼，同時也是目前所有錯誤更正碼當中，更正能力最佳的一種。

本次專題的目的正是要藉由 LDPC 編碼器/解碼器(encoder/decoder)的實作，了解 sum-product algorithm (SPA)在解碼時所扮演的角色，以及探討不同特性的 parity-check matrix 對於錯誤的更正能力。

我也將實作成果應用在快閃記憶體(Flash memory)當中的 NAND flash，由此探討錯誤更正碼對於記憶體儲存的更正能力，並利用不均等錯誤保護碼(Unequal error protection code, UEP code)使得不同位元之間具有不相等的更正能力，藉此對較容易出錯的訊息做較強的保護。

(二)原理分析與系統設計

2.1 原理分析

線性區段碼(Linear block code)

奇偶校驗矩陣(Parity-check matrix)為線性區段碼中用於描述特定線性關係的矩陣，一般以 H 作為表示；矩陣當中的所有元素皆為 0 或 1，亦即在 $GF(2)$ 的結構之下；若 H 為一個 $r \times n$ 大小的矩陣，則當一個 $1 \times n$ 的編碼訊息 x 之轉置陣列 x^T 滿足矩陣乘法 $Hx^T = 0$ 時，我們稱 x 為一個 codeword，亦即一組有效的編碼訊息，而所謂的"code"則是指所有 codeword 所形成的向量空間(vector space)。

接著，我們能利用高斯消去法將 H 對應到一個生成矩陣(generator matrix)，一般以 G 作為表示，並將其作為把原始訊息轉換成編碼訊息的媒介；若 G 為一個 $k \times n$ 大小的矩陣，則我們將這組 code 稱為 (n, k) code；若 u 代表一個 $1 \times k$ 的原始訊息，則編碼訊息 x 的產生方式即為矩陣乘法 $x = uG$ ，此時的 x 會是一個 codeword。

訊息傳輸的流程中，Encoder 的功能就是把輸入的原始訊息 u 與生成矩陣 G 進行矩陣相乘，得到編碼訊息 $x = uG$ ；Decoder 的任務則是把收到的訊息 $y = z + x$ 乘上 H 檢查有無錯誤發生，其中 z 為通道中的雜訊；若 y 並非滿足條件的 codeword，則我們必須利用演算法找出一個 codeword y' 使得 $Hy'^T = 0$ ，且 y' 是距離 y 最近的 codeword，而該演算法的設計會在之後介紹。

低密度奇偶校驗碼(Low-Density Parity-Check Codes, LDPC code)

LDPC code 是線性區段碼的其中一種，其在 parity-check matrix 的架構上具有幾點特色：矩陣中每一行(resp. 列)有 1 的地方之數目必須相同，且這個數目必須遠小於矩陣的總行數(resp. 總列數)；根據上述的特色，我們可以將 parity-check matrix 以二分圖(bipartite graph)的方式表示。以 $H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$ 為例，便能做出如圖 2-1 的二分圖。

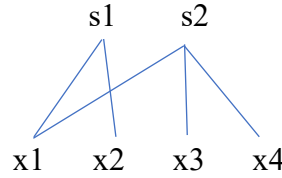


圖 2-1

其中，H 的列對應的是 check node，也就是上排的 s1、s2，而行對應的是 variable node，也就是下排的 x1、x2、x3、x4；藉由二分圖的表示，我們可以根據機率的運算把 decoding 的步驟完成。

定義 Log likelihood ratio (LLR)及兩種運算：

$L(p_0, p_1) = \ln p_0/p_1$ ，其中 $p_0 = P(x=0 | y)$ ， $p_1 = P(x=1 | y)$

$\text{VAR}(L_1, L_2) = L_1 + L_2$

$\text{CHK}(L_1, L_2) = \text{sgn}(L_1)\text{sgn}(L_2)\min(|L_1|, |L_2|) + \Delta(L_1, L_2)$ ，其中 $\Delta(L_1, L_2)$ 在實作時直接用查表的方式即可。

接著利用 sum-product algorithm (SPA)實作出 decoder；以下為 SPA 概述。

定義 $\mathcal{L}(i) = \{j : H_{i,j} = 1\}$

$\mathcal{M}(j) = \{i : H_{i,j} = 1\}$

$q_{i,j}$ = codeword 當中第 j 個 bit 的 LLR，由 i 以外的 check nodes 計算獲得

$u_{i,j}$ = codeword 當中第 j 個 bit 的 LLR，由 j 以外的 variable nodes 計算獲得

初始化：對於所有 $H_{i,j} = 1$ 的 (i, j) ，設定 $q_{i,j} = L_j$ ，其中 $L_j = \ln \frac{P(y_j|x_j=0)}{P(y_j|x_j=1)} = \left(\frac{4E_s}{N_0}\right) y_j$

步驟一：Bottom-up，對於所有 $H_{i,j} = 1$ 的 (i, j) ， $u_{i,j} = \text{CHK}_{j' \in \mathcal{L}(i) \setminus j} (q_{i,j'})$

步驟二：Top-down，對於所有 $H_{i,j} = 1$ 的 (i, j) ， $q_{i,j} = \text{VAR}(\text{VAR}_{i' \in \mathcal{M}(j) \setminus i} (u_{i',j}), L_j) = L_j + \sum_{i' \in \mathcal{M}(j) \setminus i} u_{i',j}$

步驟三：對於所有 $j = 1, 2, \dots, n$ ， $q_j = \text{VAR}(\text{VAR}_{i \in \mathcal{M}(j)} (u_{i,j}), L_j) = L_j + \sum_{i \in \mathcal{M}(j)} u_{i,j}$

步驟四：對於所有 $j = 1, 2, \dots, n$ ， $x_j = \begin{cases} 0, & q_j \geq 0 \\ 1, & q_j < 0 \end{cases}$

步驟五：若藉由步驟四得出的 x 滿足 $Hx^T = 0$ ，則該 x 為我們找到的 codeword；若不滿足則回到步驟一繼續計算。

上述的流程為一次疊代(iteration)，我們在 decode 之前會先設定一個疊代次數的上限，若疊代次數抵達這個上限時還沒找到滿足條件的 x ，則稱為解碼失敗(decoding failure)。

不均等錯誤保護碼(Unequal error protection code，UEP code)

UEP code 的原理為：藉由將 parity-check matrix 設計成具有特殊架構的方式，讓 code 可以分成許多個 stage，而每個 stage 內的訊息保護程度皆不相同，因此我們可以把重要的訊息放入保護能力較佳的 stage、把相對不重要的訊息放入保護能力較差的 stage，進而實現資源的有效分配。

UEP code 通常會在影像傳輸、資料儲存等應用上使用，例如 NAND flash 的 TLC cells 以 LSB page 的誤塊率(Block error rate，BLER)最高，因此需要有更強的保護。

2.2 系統設計

模擬方式與流程

在本次的專題中，0 所代表的電位為+1，1 所代表的電位為-1，因此在訊息接收端以電位 0 做為分界線。

對於通道雜訊的模擬，我以 C 語言產生加性高斯白雜訊(Additive white Gaussian noise，AWGN)進行模擬，藉由設定不同的訊雜比(Signal-to-noise ratio，SNR)來調整雜訊標準差的大小，可以得出在各個 SNR 值的情況下的位元誤碼率(bit error ratio，BER)以及誤塊率，分別是傳輸訊息中錯誤位元數與總位元數的比例、錯誤塊數與總塊數的比例，進而畫出該 code 的 SNR-BER 的曲線圖。

至於 NAND flash 的部分，由於電壓在 cells 裡儲存時會因為干擾或是雜訊而使電壓偏移，所以整個存儲的過程也可看成是訊息在通道內的傳輸，因此我對於雜訊模擬的方式同樣是以 C 語言產生 AWGN，並加入到即將存入 cells 的電壓之中；若電壓因為雜訊過大而偏移至其他的電壓區間，則對於編碼訊息的判讀便會出現錯誤。

圖 2-2 為本次專題對於 LDPC code 的模擬流程圖；encoder 主要負責產生生成矩陣 G 以及進行矩陣乘法 $x = uG$ ，進入 channel 時會添加雜訊，decoder 則是執行 SPA 找出滿足 $Hx^T = 0$ 條件的 codeword。



圖 2-2 模擬流程圖

Parity-Check Matrix 之架構

準循環碼(quasi-cyclic code, QC code)是指任意一組 codeword x 經過特定次數的循環移位(circular shift)後仍是一組 codeword；若將 QC code 應用於 LDPC code，則這種 code 的 parity-check matrix 通常會由數個大小相等的排列矩陣(permutation matrix)組合而成，而其中的排列矩陣為單位矩陣經過循環移位後所形成。

在這次的專題中，我在主要實作的部分使用了一種 EEP code 以及兩種 UEP code，其中一種 UEP code 有 3/8 是保護程度較佳的 stage，另一種 UEP code 則有 1/3 是保護程度較佳的 stage。

(三)實驗結果

3.1 UEP 應用於 bits 之間錯誤率均等的 codewords

圖 3-1 是將 UEP code 使用在每個 bits 錯誤率均等的 codeword 後，所畫出來的 E_b/N_0 (亦即 SNR)對 BER 的作圖；兩條曲線分別代表著保護程度不同的兩個 stage。

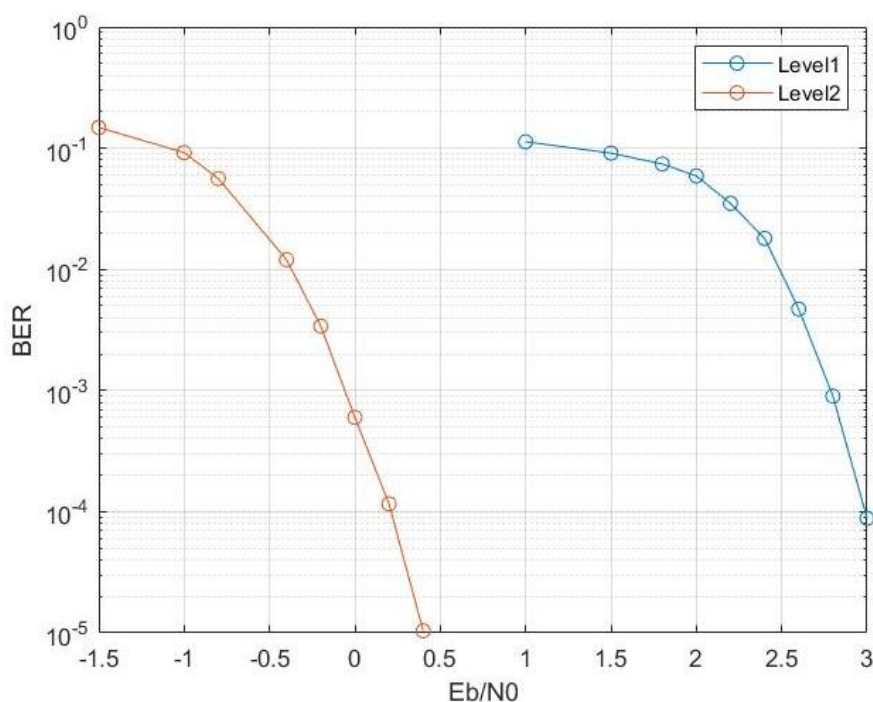


圖 3-1 UEP 應用於錯誤率均等的 codewords

隨著 SNR 的增加，BER、BLER 皆會跟著下降，因此圖3-1的兩條曲線皆是斜率為負；歸因於 parity-check matrix 在結構上的設計，level2的保護程度比 level1的保護程度更好，同時我們也可以藉由圖3-1得出：在相同 BER 的情況下，level2所需的 SNR 遠小於 level1所需的 SNR，代表實驗結果符合預期成效，UEP code 確實能在不同 stage 中做出具有差異化的保護程度。

3.2 EEP 與 UEP 應用於 NAND flash

圖3-2是將 EEP code 以及 UEP code 使用在 NAND flash 後，所畫出來的 average GRBER 對 BER 的作圖，其中 average GRBER 是 decode 之前的原始誤碼率。

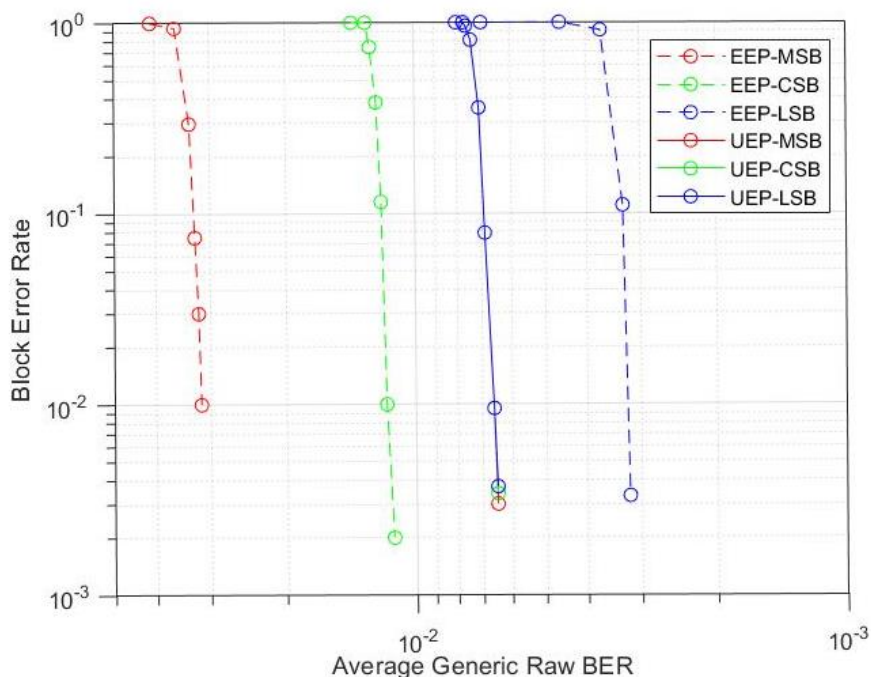


圖3-2 EEP 與 UEP 對於 NAND flash 更正能力之比較

Average GRBER 越低代表通道中的雜訊較低，所以 decode 完成後的 BLER 也會較低，兩者呈現正相關；根據 NAND flash 中電壓與三個 bits 之間的對應關係，我們得知 LSB 是最容易發生錯誤的 page，因此在 EEP 的部分，若要達到相同的 BLER，LSB page 所需的 average GRBER 遠小於 MSB page，CSB page 則是介於 MSB 與 LSB page 之間，整體實驗數據符合預期成果，需要藉由 UEP code 來提高保護的效率。

至於 UEP 的模擬結果，雖然 LSB page 發生錯誤的機率遠大於另外兩個 page，然而藉由 UEP code 的架構賦予 LSB page 較高的保護程度之後，三個 page 在 decode 完成後的 BLER 幾近相同，因而達成了保護效率的最佳化。

若是比較 EEP 與 UEP 的更正能力便會發現：雖然使用 UEP code 之後，三條曲線的 BLER 在相同 average GRBER 之下都高於使用 EEP code 的 MSB 以及 CSB page，但是由於 NAND flash cells 整體的錯誤率只取決於最容易發生錯誤的 LSB page，因此只要 LSB page 的錯誤率下降，便達到了降低整個 cell 錯誤率的目標，實現了本次實驗的目標。

(四)結論

為了探討 UEP code 對於不同 stage 保護程度上的差異，我將 UEP code 應用在每個 bits 錯誤率均等的 codeword 後，可以發現 UEP code 確實能給每個 stage 賦予不同程度的保護效果。

而在 NAND flash 當中，歸因於 bits 與電壓之間的 mapping 方式，我使用 EEP code 對其進行模擬後，確實發現了三個 page 在 BLER 上的顯著差異，其中又以 LSB page 的錯誤率最高、MSB page 的錯誤率最低。

由於 NAND flash cells 的整體錯誤率取決於 LSB page，因此我將 UEP code 套用在 NAND flash 上，使三個 page 的 BLER 接近相同、且比原先使用 EEP code 時的 LSB page 錯誤率更低，藉此確實達成了降低錯誤率、同時提高保護效率的效果。

三、心得感想

首先我想感謝當初願意收我作為專題生、並在 meeting 中給予耐心指導的趙啟超教授，讓我得以在喜愛的通訊領域中進行實作專題，並收穫我以前未曾學習過的許多知識與領悟。

在進行專題的過程中，審視過往學習成效的機會是我在專題裡的收穫之一。由於本次專題的內容牽涉到了機率、線性代數、通訊系統等多門課程，因此在研讀文獻或是設計實驗架構時常常會需要用到之前學過的知識；然而正是在這個過程中，我發現雖然自己過往的學習成績不錯，但卻缺乏些許跨科目的、融會貫通的知識運用能力，這也不禁讓我對於自己的學習方式有了重新的評估，並期許自己在未來吸收新知的同時，能將其跟已有的觀念進行整合，達成知識層面的融會貫通。

我的另一個大收穫是深刻理解到了分工合作的重要性。由於我是自己一組進行實作專題，所以若是在文獻閱讀上遇到不懂的概念、或是在模擬流程設計上遇到瓶頸的話，便常常會一個人陷入某個特定觀念的鬼打牆之中，也因此耗費了不少時間；如果此時有個跟我一起進行專題的組員，我們便能互相討論文獻中的觀念以及正確的模擬方式，減少許多不必要的思考時間。因此雖然我是獨自進行實作專題，卻深刻地體會到了有組員能一起分工合作的重要性。

最後，我非常感謝實驗室的學長姐們，不僅會在我遇到瓶頸時詳細地解說其中的細節，也時常會跟我閒聊研究生的生活、以及對於升學上的經驗分享，讓我度過了充實且有趣的兩個學期。