

適用於 5G NR 接收機的 LDPC Codes

平行處理解碼器之研究

專題生：蘇沛錦

指導教授：李昌明

國立中正大學 通訊工程學系



大綱 (Outline)

- 一、摘要
- 二、最小和解碼演算法
- 三、Column by Column 平行處理方式
- 四、避免資料存取衝突之資料結構
- 五、分析結果
- 六、結論



一、摘要

- 低密度同位檢查碼 (LDPC Codes) :
 - 更正資料在傳輸時受到通道雜訊干擾所造成的錯誤
 - 具有非常接近Shannon極限的卓越性能和保護大量資料傳輸之優勢
 - 為5G NR標準中數據傳輸的編碼方式
- Column by Column平行處理方式:
 - 檢測矩陣內同一列的偏移值相同→記憶體存取衝突→解碼時間延遲
 - 標定LDPC檢測矩陣中同一列發生衝突的位置
 - 將這些會產生衝突的Column分成多個Column Set進行處理



※低密度同位檢查碼

- LDPC Codes為線性區塊碼的一種，表示成 (n, k) 的形式。
- 碼字內的 $(n - k)$ bits 定義為查核位元。

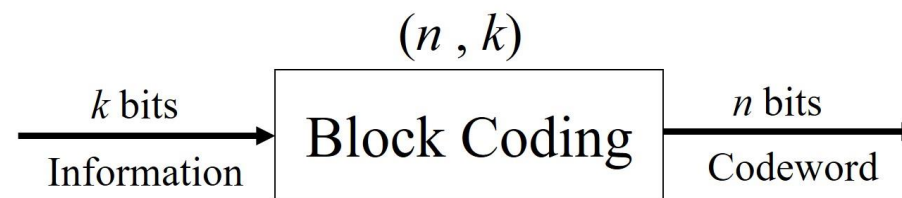


Fig. 1 (n, k) 線性區塊編碼

- m : 輸入訊息 $m = [m_0 \ m_1 \ \cdots \ m_{k-1}]_{1 \times k}$

生成矩陣

(Generator Matrix)

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}_{k \times n}$$
$$c = m \times G$$

同位檢測矩陣

(Parity Check Matrix)

$$H = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{(n-k)-1} \end{bmatrix}_{(n-k) \times n}$$
$$\hat{c} \times H^T = 0$$

- c : 傳送碼字 $c = [c_0 \ c_1 \ \cdots \ c_{n-1}]_{1 \times n}$

- \hat{c} : 解算碼字 $\hat{c} = [\hat{c}_0 \ \hat{c}_1 \ \cdots \ \hat{c}_{n-1}]_{1 \times n}$



※低密度同位檢查碼

- 檢測矩陣 H 為多數個0和少數個1所組成的稀疏矩陣
 - 透過二分圖表示→ Tanner Graph
 - 上排: 檢測節點 (Check Node)
 - 下排: 變數節點 (Variable Node)

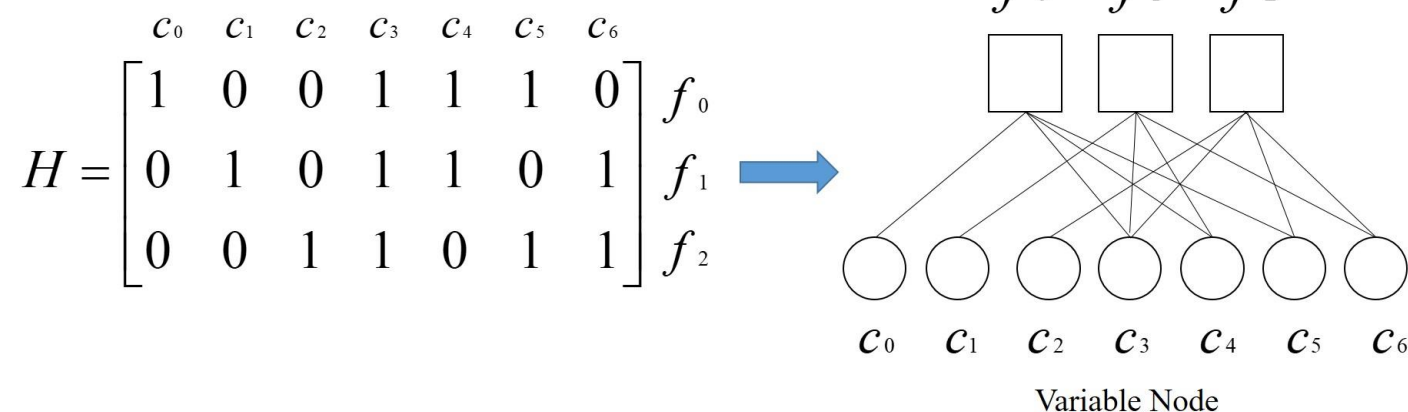


Fig. 2 (7,4) LDPC Codes的檢測矩陣 H 和Tanner Graph表示法

二、最小和解碼演算法

- 訊息傳遞演算法 (MPA)
 - LDPC Codes 常見的解碼方式
 - 針對接收碼字 y_i 進行事後機率之對數相似比例 (LLR) 計算
- AWGN channel (with mean $\mu = 0$ and variance σ^2):
 - c : 傳送碼字 $c = [c_0 \ c_1 \ \cdots \ c_{n-1}]_{1 \times n}$
 - y : 接收碼字 $y = [y_0 \ y_1 \ \cdots \ y_{n-1}]_{1 \times n}$

$$L(c_i) \equiv \log(l(c_i)) = \log\left(\frac{Pr(c_i = 0|y_i)}{Pr(c_i = 1|y_i)}\right) = 2 \frac{y_i}{\sigma^2} \quad (1)$$



二、最小和解碼演算法

- 最小和演算法 (MSA)

- 將對數域之MPA的機率運算過程再進行化簡、近似處理:

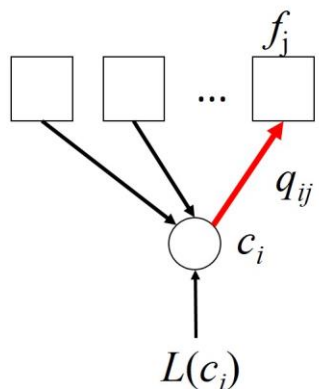
1. 初始化訊息 $L(c_i)$ ，且 $L(q_{ij}) = L(c_i)$ 。

2. 檢測節點向下更新訊息 $L(r_{ji})$ 。

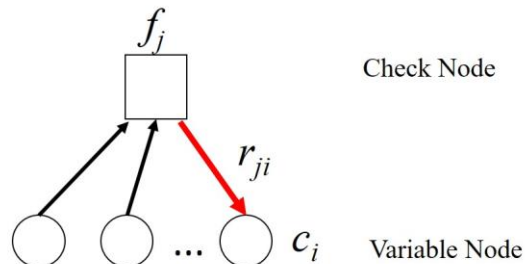
3. 變數節點向上傳遞訊息 $L(q_{ij})$ 。

4. 觀察是否達到終止條件或最大迭代次數，否則回到步驟2.進行迭代運算。

Variable-to-check Message, $L(q_{ij})$



Check-to-variable Message, $L(r_{ji})$



Normalized MSA:

$$L(r_{ji}) = \left(\prod_{i' \in R_j \setminus i} \alpha_{i'j} \right) \times k \times \min_{i' \in R_j \setminus i} \beta_{i'j} \quad (2)$$

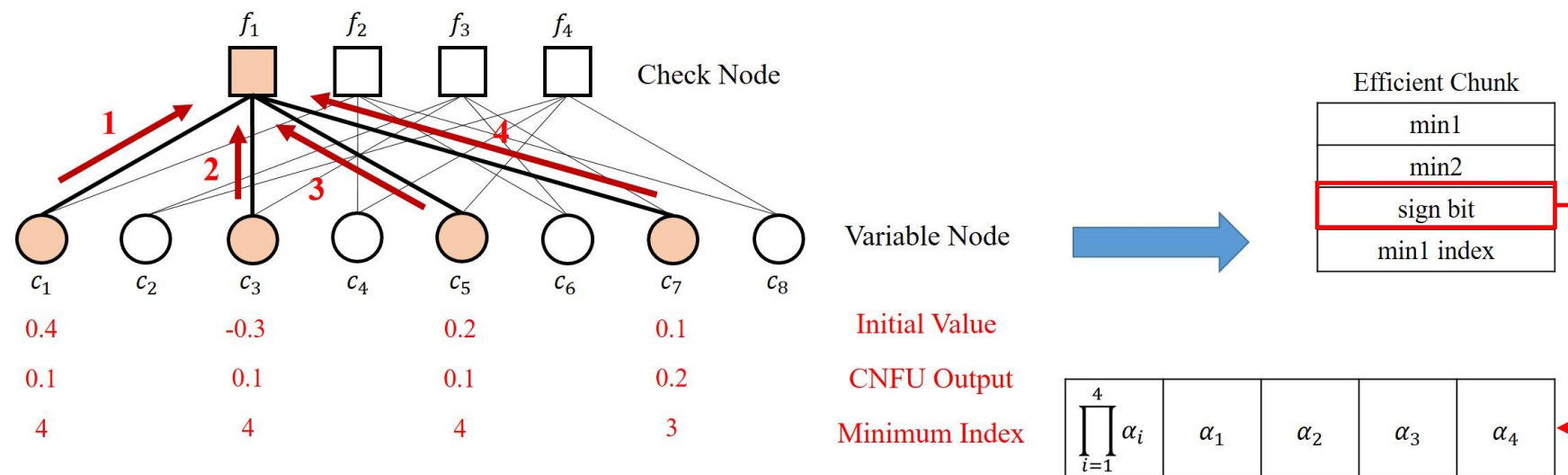
$$\alpha \equiv \text{sign}(L(q_{ij})), \beta \equiv |L(q_{ij})|$$

k : Normalization factor ($0.6 \leq k \leq 0.9$)

$$L(q_{ij}) = L(c_i) + \sum_{j' \in C_{i \setminus j}} L(r_{j'i}) \quad (3)$$

三、Column by Column 平行處理方式

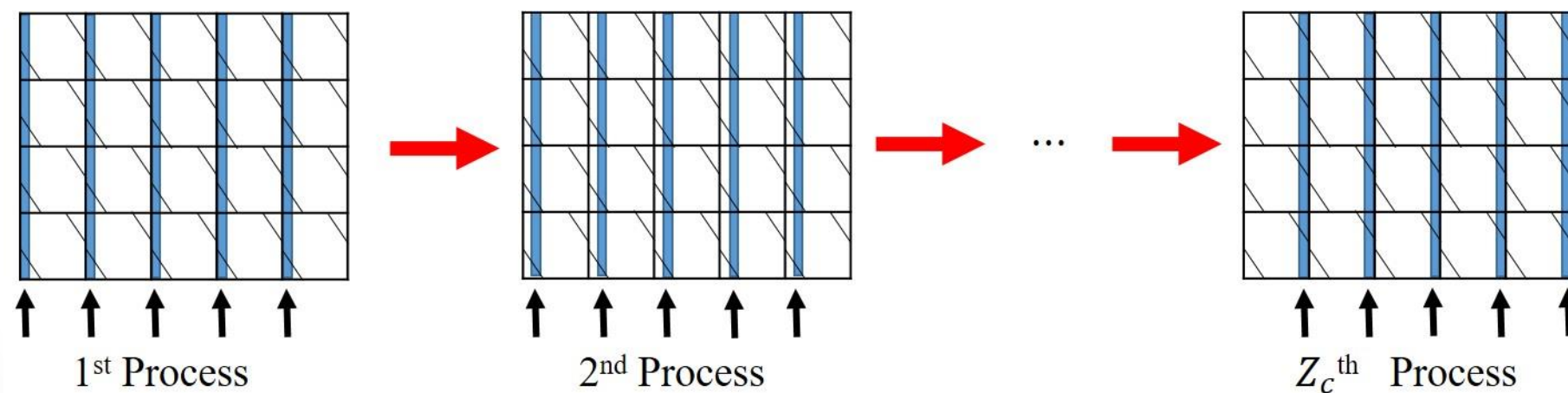
- 解碼器的記憶體空間(Efficient Chunk)只需少量儲存 $L(r_{ji})$ 訊息
 - 最小值(min1)、第二小值(min2)、最小值索引、正負符號位元
- 後續進行 $L(q_{ij})$ 訊息更新時，可同時預測下一次迭代計算所使用的 $L(r_{ji})$



圖(一) (8,4) 檢測矩陣 H 更新檢測節點上的 $L(r_{ji})$ 訊息和簡化儲存之Efficient Chunk

三、Column by Column平行處理方式

- NR LDPC Codes在解碼流程上，可使用Column by Column方式進行平行處理：
 - 初始化時先更新全部的 $L(q_{ij})$ ，並且簡化儲存 $L(r_{ji})$ 。
 - 進行 $L(q_{ij})$ 運算時，先還原出正確的 $L(r_{ji})$ ，再更新記憶體。
 - 經由 Z_c 次的處理後，所有記憶體皆更新完成，提供給下一次的迭代運算使用。



圖(二) Column by Column平行處理方式

✂ NR LDPC Codes

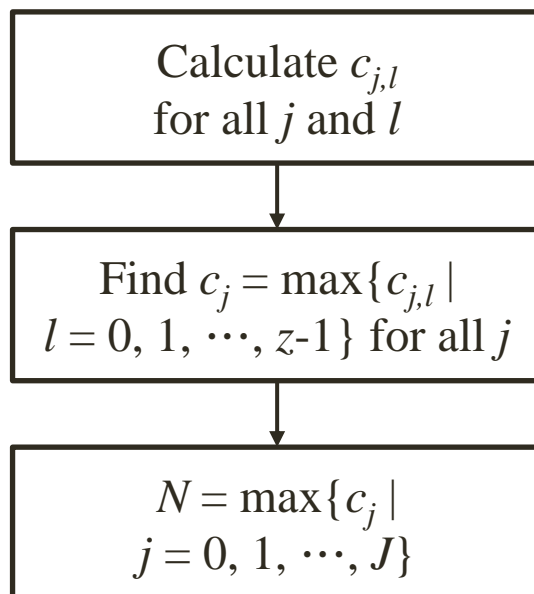
- 類循環 (Quasi-Cyclic)結構
- 二種基本矩陣: BG1、BG2
- 檢測矩陣內部: $Z_C \times Z_C$ 的零矩陣或循環平移之單位矩陣

Fig. 3 NR LDPC BG2 ($Z_C = 7$)的檢測矩陣 H

四、避免資料存取衝突之資料結構

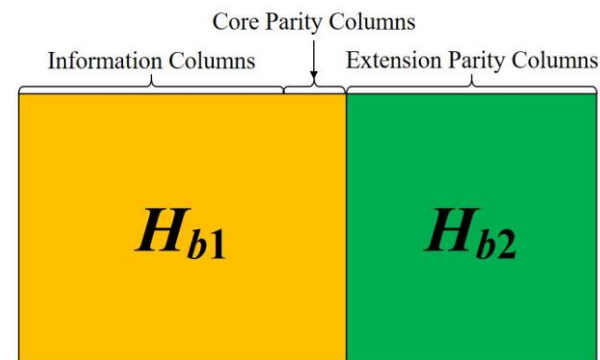
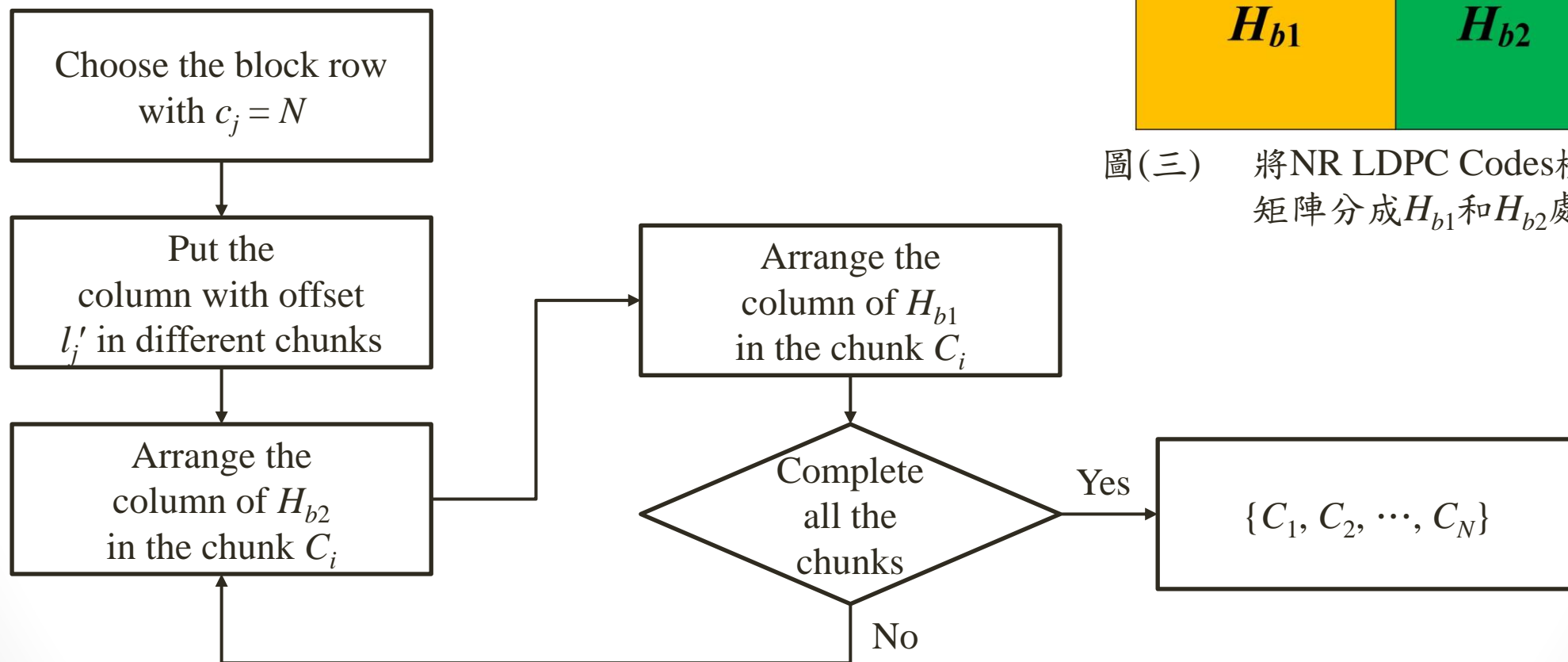
- 檢測矩陣內同一列偏移值相同→資料存取衝突
- 平均分配多組的Column Set→避免解碼延遲問題、加快解碼器之MSA運算速度
- 針對NR LDPC Codes尋找平行處理之Column Set的流程如下：

Step 1. Find the number of column sets (chunks) N



四、避免資料存取衝突之資料結構

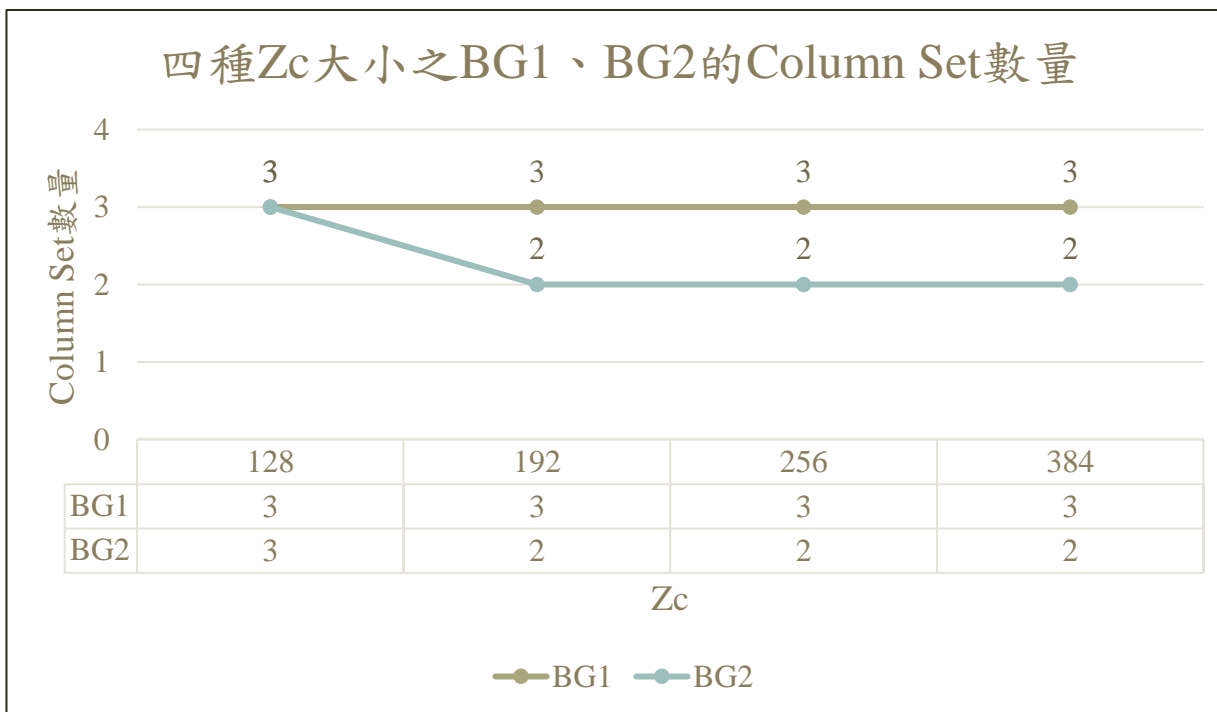
Step 2. Construction of column sets $\{C_1, C_2, \dots, C_N\}$



圖(三) 將NR LDPC Codes檢測矩陣分成 H_{b1} 和 H_{b2} 處理

五、分析結果 (5G NR)

- 兩種矩陣針對不同 Z_C 大小的平行化處理程度:
 - BG1在四種 Z_C 條件下的MSA平行解碼加速，理論上皆可提升為原本的三倍。
 - BG2除了 $Z_C = 128$ 外，其餘三種 Z_C 針對MSA運算速度，理論上可加快二倍。



五、分析結果 (WiMAX)

- IEEE 802.16e (WiMAX) LDPC Codes
- 19個類型的碼長 (n)、4種不同的碼率 ($R = k / n$)
- 理論上可加快最多5倍

n (bits)	n (bytes)	z factor	R = 1/2		R = 2/3 (A, B)		R = 3/4 (A, B)		R = 5/6	
			k (bytes)	Number of chunks	k (bytes)	Number of chunks	k (bytes)	Number of chunks	k (bytes)	Number of chunks
576	72	24	36	3	48	(4, 4)	54	(3, 4)	60	5
672	84	28	42		56		63		70	
768	96	32	48		64	(3, 4)	72	(3, 3)	80	
864	108	36	54		72	(4, 4)	81		90	
960	120	40	60		80	(3, 4)	90		100	
1056	132	44	66		88		99		110	
1152	144	48	72		96		108		120	
1248	156	52	78		104		117		130	
1344	168	56	84		112		126		140	
1440	180	60	90		120		135		150	
1536	192	64	96		128		144		160	
1632	204	68	102		136		153		170	
1728	216	72	108		144		162		180	
1824	228	76	114		152		171		190	
1920	240	80	120		160		180		200	
2016	252	84	126		168		189		210	
2112	264	88	132		176		198		220	
2208	276	92	138		184		207		230	
2304	288	96	144		192		216	(3, 2)	240	



五、分析結果 (Wi-Fi)

- IEEE 802.11n (Wi-Fi) LDPC Codes
- 3個類型的碼長 (n)、4種不同的碼率 ($R = k / n$)
- 理論上可加快最多6倍

n (bits)	Z (bits)	R = 1/2		R = 2/3		R = 3/4		R = 5/6	
		k (bits)	Number of chunks	k (bits)	Number of chunks	k (bits)	Number of chunks	k (bits)	Number of chunks
648	27	324	6	432	3	486	4	540	5
1296	54	648	3	864		972		1080	3
1944	81	972		1296		1458	3	1620	4



六、結論

- 使用Column by Column平行處理的解碼方式
- 分配多組Column Set:
 - 避免資料存取衝突(偏移值重複問題)
 - 提高平行處理速度
 - 滿足5G高速、低延遲的通訊要求
 - 可應用至WiMAX與Wi-Fi



Thank you for listening.



※MSA演算法解碼步驟

- Step 1. 初始化 (Initialize)

- $$L(q_{ij}) = L(c_i) = 2 \frac{y_i}{\sigma^2} \quad (1)$$

- Step 2.

- $$L(r_{ji}) = (\prod_{i' \in R_j \setminus i} \alpha_{i'j}) \times \min_{i' \in R_j \setminus i} \beta_{i'j} \quad (2)$$

- Step 3.

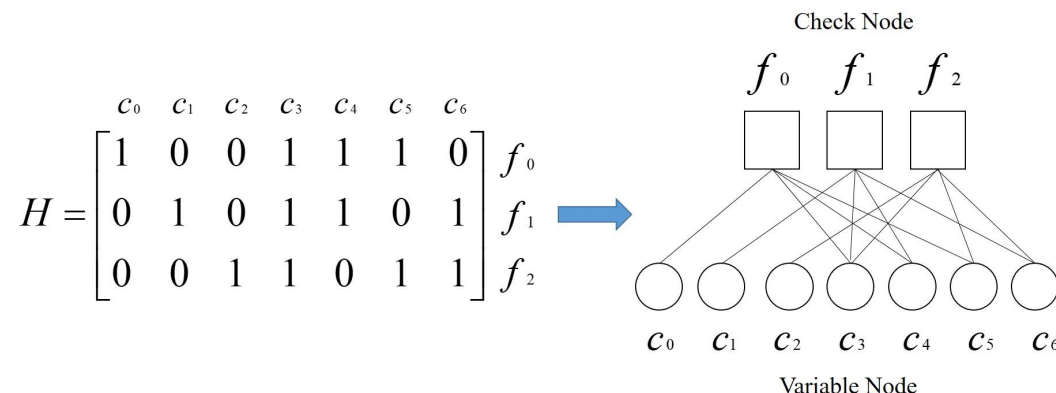
- $$L(q_{ij}) = L(c_i) + \sum_{j \in C_{i \setminus j}} L(r_{ji}) \quad (3)$$

- Step 4. 計算所有的*i*:

- $$L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji}) \quad (4)$$

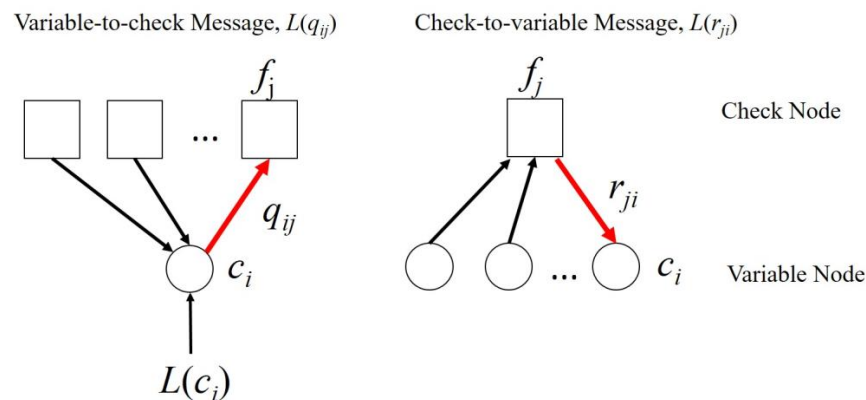
- Step 5. 計算所有的*i*:

- $$\hat{c}_i = \begin{cases} 1, & L(Q_i) < 0 \\ 0, & \text{else} \end{cases} \quad (5)$$



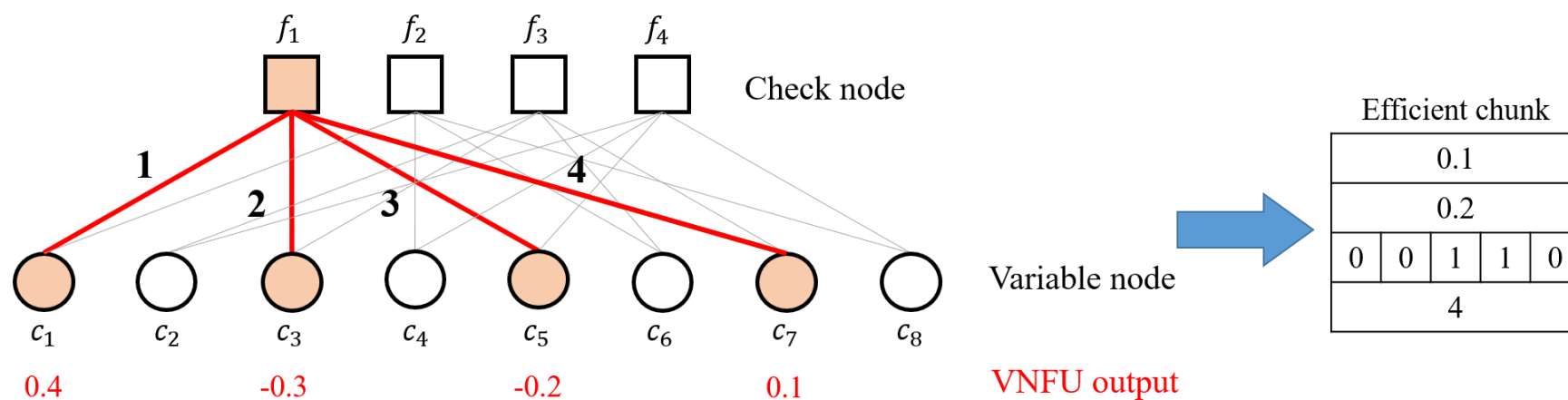
$$\alpha \equiv \text{sign}(L(q_{ij})), \beta \equiv |L(q_{ij})|$$

k : Normalization factor ($0.6 \leq k \leq 0.9$)



※如何還原正確的 $L(r_{ji})$ 訊息

- 假如**不是**在 $L(q_{ij})$ 訊息最小值的索引上，則
 - $L(r_{ji}) = (\prod_{i'} \alpha_{i'j}) \times \phi \left(\sum_{i' \in R_j \setminus i} \phi(\beta_{i'j}) \right) = \left(\prod_{k \in V_j} \alpha_{kj} \right) \times \alpha_{kj} \times \min 1$
- 假如**是**在 $L(q_{ij})$ 訊息最小值的索引上，則
 - $L(r_{ji}) = (\prod_{i'} \alpha_{i'j}) \times \phi \left(\sum_{i' \in R_j \setminus i} \phi(\beta_{i'j}) \right) = \left(\prod_{k \in V_j} \alpha_{kj} \right) \times \alpha_{kj} \times \min 2$



※5G NR LDPC Codes: BG1 & BG2

- NR LDPC Codes的同位檢查矩陣是由Base Graph進行延展來構成。
- **BG1**適用資料區塊大且高碼率；反之**BG2**則用於資料區塊小、低碼率。

Table. 1 BG1和BG2的基本參數

Basic Parameters	BG1	BG2
Matrix size	46 × 68	42 × 52
Number of nonnegative elements	316	197
Minimum code rate (R)	1/3	1/5

Table. 2 BG1和BG2的 Z_C 數值集合 ($Z_C = a \times 2^j$)

Set index	Set of lifting sizes (Z_C)
Set 0 ($a = 2, 0 \leq j \leq 7$)	{2, 4, 8, 16, 32, 64, 128, 256}
Set 1 ($a = 3, 0 \leq j \leq 7$)	{3, 6, 12, 24, 48, 96, 192, 384}
Set 2 ($a = 5, 0 \leq j \leq 6$)	{5, 10, 20, 40, 80, 160, 320}
Set 3 ($a = 7, 0 \leq j \leq 5$)	{7, 14, 28, 56, 112, 224}
Set 4 ($a = 9, 0 \leq j \leq 5$)	{9, 18, 36, 72, 144, 288}
Set 5 ($a = 11, 0 \leq j \leq 5$)	{11, 22, 44, 88, 176, 352}
Set 6 ($a = 13, 0 \leq j \leq 4$)	{13, 26, 52, 104, 208}
Set 7 ($a = 15, 0 \leq j \leq 4$)	{15, 30, 60, 120, 240}

