

第二章 功率意識指標

我們將功率意識指標分為未損失品質的功率意識指標以及損失品質的功率意識指標二種，未損失品質的功率意識指標是由[15]所提出，以數學式計算 battery lifetime 的期望值，而我們提出的損失品質的功率意識指標，是以 battery 模擬軟體直接估計 battery lifetime，以下將分別介紹功率意識指標的定義。

2.1 未損失(lossless)品質的功率意識指標

此功率意識指標已經被提出[15]，它代表在未損失品質的情況下將系統的功率意識量化，在本節將介紹[15]所提出的功率意識指標。

假設一個系統 H，改變其操作模式(scenario) S，以達到功率意識之目的。以 16×16 bit 陣列乘法器為例，圖 1 為其能量消耗曲線，X 軸代表 16 種模式(scenario)，亦即代表輸入的精確度 (precision)，也就是輸入的 bit，Y 軸則代表經過 16×16 bit 陣列乘法器運算的能量消耗。當輸入的 bit 較小時，乘法器需要的開關動作 (switching activity) 較少，所以能量消耗也較少。反之，當輸入的 bit 較大時，消耗的能量也較大。

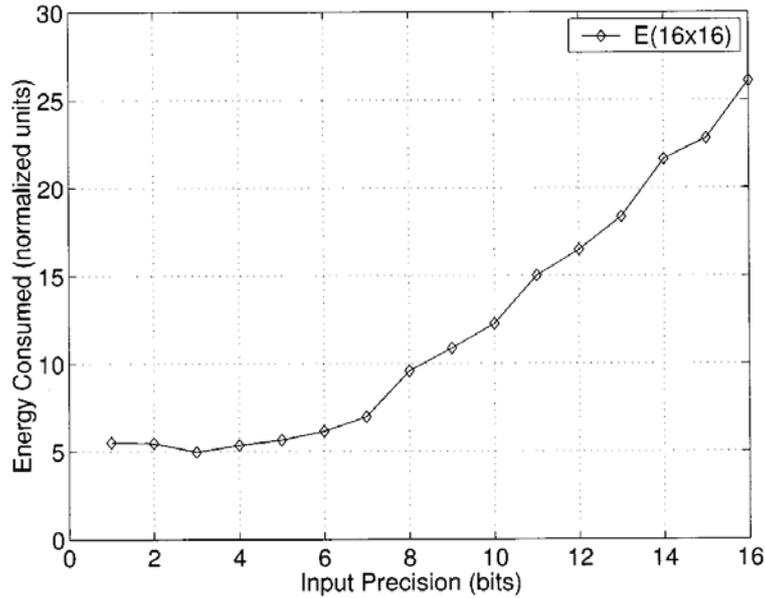


圖 1 16 × 16 bit 陣列乘法器能量消耗曲線

為了將系統的功率意識量化，產生一個最佳系統(perfect system) H_{perfect} ，做為其他系統之比較。顧名思義， H_{perfect} 為功率意識最佳的系統，它所消耗的能量跟模式需要的能量相同。圖 2 為最佳系統的方塊圖，輸入會先經過模式決定單元(scenario determining unit)以決定需使用哪一個點系統(point system) H_{s_i} ，然後由模式決定單元控制 DEMUX 與 MUX 將輸入經由點系統運算之後再做輸出。每一個輸入只對應單一的點系統，因此最佳系統可以視為點系統的所有組合。

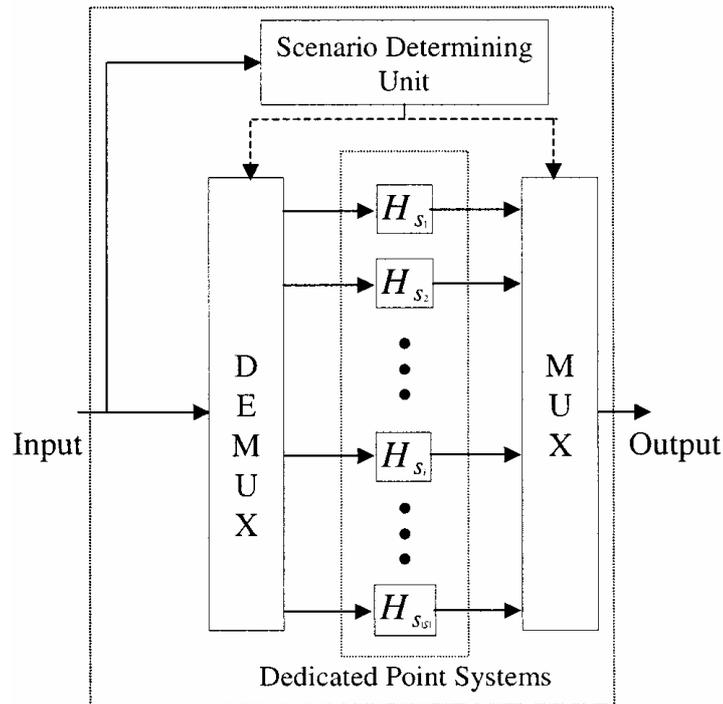


圖 2 最佳系統的方塊圖

16 × 16 bit 乘法器的最佳系統，其點系統共有 16 個：1 × 1、2 × 2、……、16 × 16 bit 乘法器，分別表示為 H_{s_1} 、 H_{s_2} 、……、 $H_{s_{16}}$ 。當輸入精確度(precision)為 i ，即被乘數與乘數為 i bit 時，則經過 H_{s_i} 點系統運算之後再做輸出，所以最佳系統的能量消耗只有 H_{s_i} 點系統所消耗的能量。圖 3 為 16 × 16 bit 乘法器與最佳系統的能量消耗曲線， $E(16 \times 16)$ 為單一 16 × 16 bit 乘法器能量消耗， E_p 為最佳系統的能量消耗，因為 E_p 只有點系統所消耗的能量，所以 E_p 消耗的能量最小。

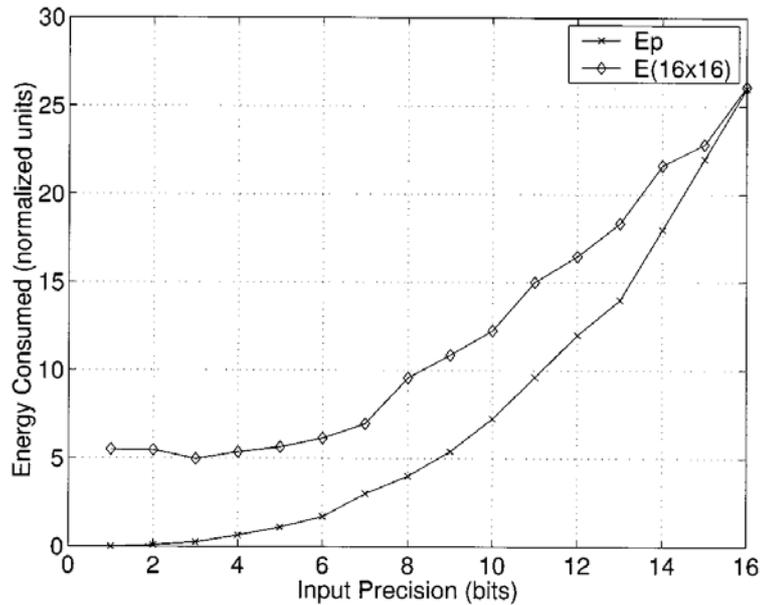


圖 3 16 × 16 bit 乘法器與最佳系統的能量消耗曲線

為了量化功率意識，我們繪出圖 4 的模式效率(scenario efficiency) η_i

$$\eta_i = E(H_{\text{perfect}}, S_i) / E(H, S_i) \quad (1)$$

其中 $E(H_{\text{perfect}}, S_i)$ 為最佳系統在 S_i 模式的能量消耗， $E(H, S_i)$ 為系統在 S_i 模式的能量消耗。 η_i 代表在 S_i 模式的功率意識， η_i 值越大表示系統的功率意識越佳。

由圖 4 得知，在 input precision=16 時 $\eta_i=1$ ，即 $E_p=E(16 \times 16)$ ，最佳系統的能量消耗跟 16 × 16 bit 乘法器相同。當 input precision 越小時， η_i 值越小，表示單一 16 × 16 bit 乘法器比最佳系統能量消耗的差異越大。

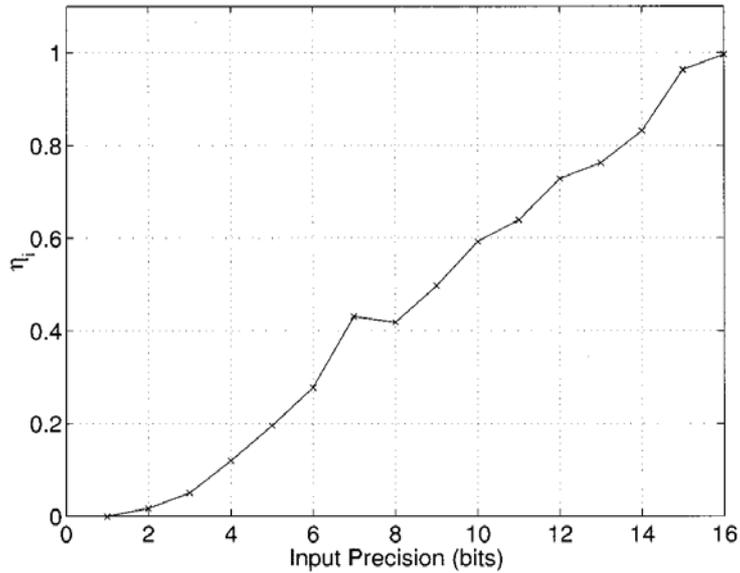


圖 4 16 × 16 bit 乘法器的模式效率

若所有模式出現的機率皆相同，則可以將平均案例 (average-case) 功率意識定義為

$$\phi_1 = E[\eta] = 1/S \cdot \sum_{i=1}^S \eta_i$$

其中S為所有模式的數目， $E[.]$ 為期望值。以 16 × 16 bit 乘法器為例， $\phi_1 = 1/16 \cdot \sum_{i=1}^{16} \eta_i = 0.501$ 。

若模式出現的機率不相同，則可以將預期(expected)功率意識定義為

$$\phi_d = E[\eta] = \sum_{i=1}^S \eta_i d_i \quad (2)$$

其中 d_i 為模式 s_i 發生的機率。圖 5 為 16 × 16 bit 乘法器應用於說話濾波(speech filtering)的模式機率，因此由圖 4 與圖 5 可得知 16 × 16 bit 乘法器的 $\phi_d = \sum_{i=1}^{16} \eta_i d_i = 0.42$ 。

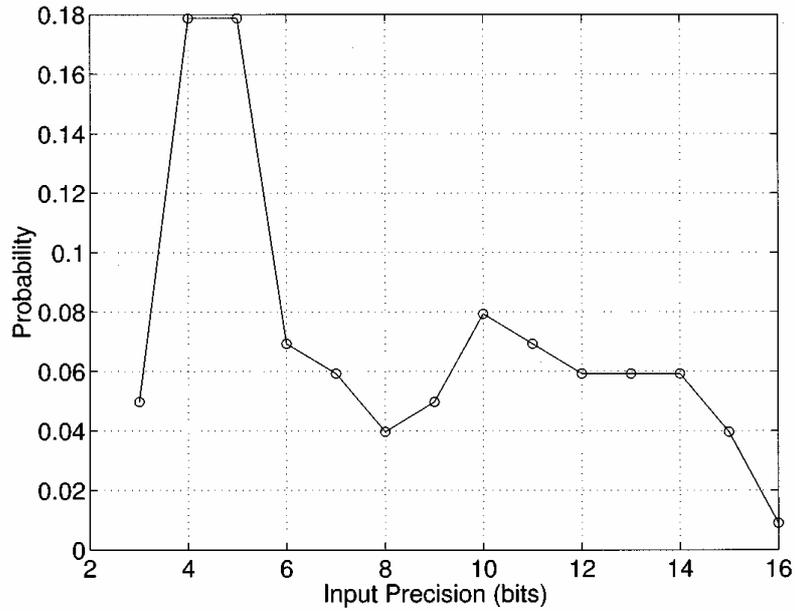


圖 5 16 × 16 bit 乘法器應用於說話濾波的模式機率

模式機率可以顯示出重要性，但是模式機率並非唯一的。舉例來說，在某個模式下發生的機率很小，但是當它一旦發生，設計者可能要追蹤該模式的變化，因此，我們將(2)式插入重要性(importance) m_i ，為了延長電池的壽命， m_i 其實就是該模式的能量消耗，所以得到

$$\phi = \sum_{i=1}^S \eta_i d_i m_i / \sum_{i=1}^S d_i m_i \quad (3)$$

$$\text{where } m_i = E(H, s_i)$$

最後我們將(1)式代入(3)式，可以得到功率意識指標 ϕ 為

$$\phi = \frac{\sum_{i=1}^S E(H_{\text{perfect}}, s_i) d_i}{\sum_{i=1}^S E(H, s_i) d_i} \quad (4)$$

$$= E(\text{Normalized Battery Lifetime})$$

(4)式即未損失品質的功率意識指標，其分母為系統H在模式機率下的能量消耗總和，分子為最佳系統 H_{perfect} 在模式機率下的能量消耗

總和，因此功率意識指標常態化電池的壽命，即代表系統H相對於最佳系統 H_{perfect} 的電池壽命。以 16×16 bit 乘法器為例，應用於圖 5 的說話濾波(speech filtering)的模式機率，則可算出 $\phi = 0.57$ ，亦即該乘法器的電池壽命是最佳系統的 0.57 倍。

2.2 損失(lossy)品質的功率意識指標

本節將介紹損失品質的狀況下，我們所使用的功率意識指標。由 2.1 節的定義，當輸入精確度小於系統精確度時，該系統消耗較多的能量；當輸入精確度等於系統精確度時，該系統為最佳系統，消耗較少的能量，這二種是未損失品質的狀況。第三種則為損失品質的情況，當輸入精確度大於系統精確度時，該系統消耗最少的能量，但是會降低品質。

Lossless: Input precision < System precision

Lossless: Input precision = System precision (Perfect system)

Lossy : Input precision > System precision

舉例來說，若輸入為 8 bit，使用 16 bit 乘法器做運算，則會消耗較多的能量，但是不會損失品質；若輸入為 8 bit，使用 8 bit 乘法器做運算，則此系統為最佳系統，能量消耗較少，且不會損失品質；若輸入同樣為 8 bit，使用 4 bit 乘法器做運算，則能量消耗最

少，但是會損失品質。

我們將損失品質的功率意識指標以直接模擬 battery lifetime 的軟體來實現。在本論文中乘法器的輸入精確度固定為 16 bit，以第三章介紹的功率控制之機制，去改變系統精確度，亦即改變乘法器本身的精確度，來產生功率意識乘法器。

本實驗是用 VHDL 設計 $16 \times 16 = 16$ bit 管線(pipelined)正負號(signed) Dadda 乘法器[16]，以 Synopsys 公司的 Design Analyzer 作合成[17]，Prime Power 做為估計消耗功率的工具。Prime Power 是 Synopsys 公司最近推出的工具，以 gate-level 估算 switching activity 的消耗功率。它可以估計各元件(cell)的全部功率、動態(dynamic)功率、洩漏(leakage)功率、開關(switching)功率、以及內部(internal)功率等等，在本論文將以全部功率做為模擬的結果。

另外，我們使用訊號-雜音比(Signal-Noise Ratio, SNR)來評價系統品質的好壞，SNR 越高代表品質越佳。本論文是以 Matlab 隨機產生 10000 筆 16 bit 的被乘數與乘數，經過我們設計的功率意識乘法器得到 10000 筆的乘積，因此 SNR 為

$$\text{SNR} = \left[\sum_{i=1}^{10000} M_i^2 \right]^{1/2} / \left[\sum_{i=1}^{10000} (Y_i - M_i)^2 \right]^{1/2} \quad (5)$$

我們定義乘法器輸出與輸入皆為小數，且 MSB 為正負號位元 (signed bit)。Y_i 是功率意識乘法器的實驗結果，為 16 bit 的正負

號(signed)小數；而 M_i 是未做任何節制，為最精確的 32 bit 的正負號小數。SNR 我們取對數再乘以 20，亦即取其 dB 值來代表品質。

$n \times n$ bit 乘法的乘積只取 n bit，亦即取 P_n 至 P_{2n-1} 的乘積，則必須做修正(correction)，以降低誤差。我們在這裡比較二種修正的方法，一種是係數修正法，另一種是圍繞至最接近(round-to-nearest) 修正法，我們將比較這二種方法的功率與 SNR。

1. 係數修正法

節制乘法器(truncated multiplier)與係數修正法已經被發展出來[18][19]，但是因為它是直接切掉較小乘積位元的加法器，所以無法做功率意識。本論文則是維持原來的硬體，而非切除較小乘積位元的加法器，如此才得以做功率意識的乘法器。

係數修正法分為常數修正(constant correction)與變數修正(variable correction)，Dadda 乘法器使用常數修正較有效率，並且比陣列乘法器有較少的功率消耗與閘延遲[20]，因此在這裡即以常數修正做為係數修正法。

圖 6 為 $n \times n = n$ bit 的常數修正圖示，其中 $S_i, j = X_i \text{ AND } Y_j$ 。 P_0 至 P_{n-k-1} bit 的部分乘積設為零，並加入修正常數 C 。 C 加在 $n-1$ 至 $n-k$ 列，如圖 6 所示。

$$E_{\text{round}} = -0.5 \cdot \sum_{q=n-k}^{n-1} 2^{-2n+q} = -2^{-n-1} \cdot (1-2^{-k})$$

則全部誤差為

$$\begin{aligned} E_{\text{total}} &= E_{\text{red}} + E_{\text{round}} \\ &= -0.25 \cdot \sum_{q=0}^{n-k-1} (q+1) 2^{-2n+q} - 2^{-n-1} \cdot (1-2^{-k}) \end{aligned} \quad (6)$$

又為了將常數 C 限制在 n+k bit，所以常數為

$$C = -[\text{round}(2^{n+k} \cdot E_{\text{total}})] / 2^{n+k} \quad (7)$$

其中 $\text{round}(x)$ 表示 x 取最接近的整數。

表 1 為 n=16 時，k 從 1 到 16 的常數，其中 $C' = C \cdot 2^{n+k}$ 。n=16 即為 $16 \times 16 = 16$ bit 乘法。k=1 為 P0 至 P14 bit 的部分乘積設為零，k=2 為 P0 至 P13 bit 的部分乘積設為零，……，k=15 為 P0 bit 的部分乘積設為零，k=16 為部分乘積皆不設為零。

k	1	2	3	4	5	6	7	8
C'	4	5	7	10	18	34	66	129
k	9	10	11	12	13	14	15	16
C'	257	513	1025	2048	4096	8192	16384	32768

表 1 n=16 的常數

舉例來說，當 n=16，k=7 時， E_{total} 由 (6) 式得知

$$\begin{aligned} E_{\text{total}} &= -0.25 \cdot \sum_{q=0}^8 (q+1) 2^{-32+q} - 2^{-17} \cdot (1-2^{-7}) \\ &= (-134145) \cdot 2^{-34} \end{aligned}$$

則 C 由(7)式得知

$$C = -\{\text{round}[2^{23} \cdot (-134145) \cdot 2^{-34}]\} / 2^{23}$$

其中 $-\text{round}[2^{23} \cdot (-134145) \cdot 2^{-34}] = \text{round}(65.5)$

65.5 取最接近的整數為 66，因此 $C = 66 \cdot 2^{-23}$ 。

由表 1 可得到表 2 的常數修正位置。P0 至 P17 為乘積代表的 bit 數，空格內的數目為常數的二進位制數目。以 $k=7$ 為例， $C = 66 \cdot 2^{-23}$ ，而 66 的二進位制為 1000010，如表 2 所示，1 需要加在 P10 bit 與 P15 bit 的地方。

	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
k=1	1	0	0															
k=2		1	0	1														
k=3			1	1	1													
k=4			1	0	1	0												
k=5			1	0	0	1	0											
k=6			1	0	0	0	1	0										
k=7			1	0	0	0	0	1	0									
k=8			1	0	0	0	0	0	0	1								
k=9			1	0	0	0	0	0	0	0	1							
k=10			1	0	0	0	0	0	0	0	0	1						
k=11			1	0	0	0	0	0	0	0	0	0	1					
k=12			1	0	0	0	0	0	0	0	0	0	0	0				
k=13			1	0	0	0	0	0	0	0	0	0	0	0	0			
k=14			1	0	0	0	0	0	0	0	0	0	0	0	0	0		
k=15			1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
k=16			1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 2 常數修正位置

為了將常數加入乘法器中，必須把原來乘法器內的半加器更改為全加器，並且將其全加器的一個輸入設為 1。舉例來說，當 $k=7$ 時， $C = 66 \cdot 2^{-23}$ ，而 66 的二進位制為 1000010。圖 7 為 Dadda 乘法器加入常

數修正之圖解，1 需要加在第 10 bit 與第 15 bit 的地方，因此在 Stage 2 將二個半加器改為全加器，以 \otimes 表示之，並將 1 加在這二個全加器之中。

因為 16 × 16 bit Baugh-Wooley 符號(signed)乘法器需要在 P16 bit 的半加器加 1，因此只剩下 P3 至 P15 bit 有半加器，受限於半加器的數量，只有從 P3 至 P15 bit 的半加器加入常數修正，所以最多修正至 $k=3$ ，亦即最多節制到 P12 bit。

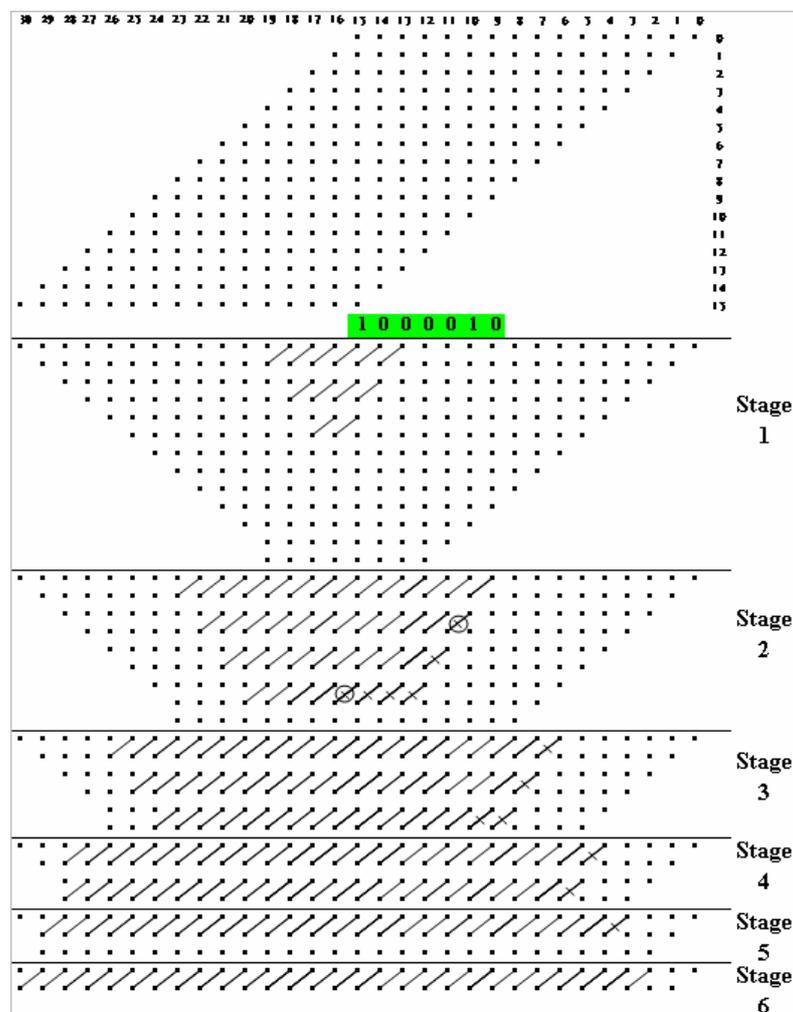


圖 7 Dadda 乘法器加入修正係數之總和圖解($n=16$, $k=7$)

2. 圍繞至最接近(round-to-nearest)修正法

所謂圍繞至最接近(round-to-nearest)修正法，即將 P_{n-1} 加到 P_n ，再取 P_n 至 P_{2n-1} 做為最後的乘積。以 $16 \times 16 = 16 \text{ bit}$ 乘法為例，如圖 8 所示， P_0 到 P_{31} 為 32 bit 的乘積，但是最後的乘積只取 16 bit，若將 P_0 至 P_{15} 捨棄而取 P_{16} 至 P_{31} 做為乘積，則誤差會相當的大。因此我們保留 P_{15} ，將 P_{15} 加到 P_{16} ，再取 P_{16} 至 P_{31} 成為輸出 Q_0 至 Q_{15} ，以做為最後的乘積。

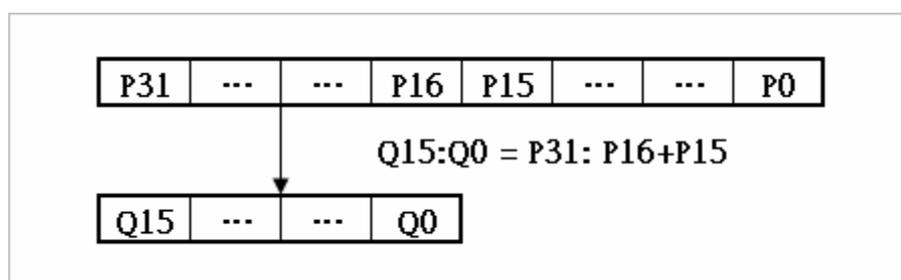


圖 8 $16 \times 16 = 16 \text{ bit}$ 乘法的圍繞至最接近修正

我們將以上這二種修正法，用 $16 \times 16 = 16 \text{ bit signed Dadda}$ 乘法器來實現，圖 9 為二種修正法的功率比較，其中 original 代表沒有任何的部分乘積設為零，P0 代表 P_0 bit 設為零，P1 代表 P_1 至 P_0 bit 設為零，P2 代表 P_2 至 P_0 bit 設為零，……，依此類推。係數修正法因半加器數目的緣故，最多節制到 P_{12} bit，而圍繞至最接近修正法在不影響 signed bit 情況下，最多可以節制到 P_{14} bit。

係數修正法需要額外的電路去存取係數，而圍繞至最接近修正法

只要將輸出暫存器(output register)加入一個加法器即可完成，因此圍繞至最接近修正法所消耗的功率較小。由圖 9 得知，圍繞至最接近修正法的平均功率為 2.682 mW，比係數修正法的 2.777 mW 來得小。

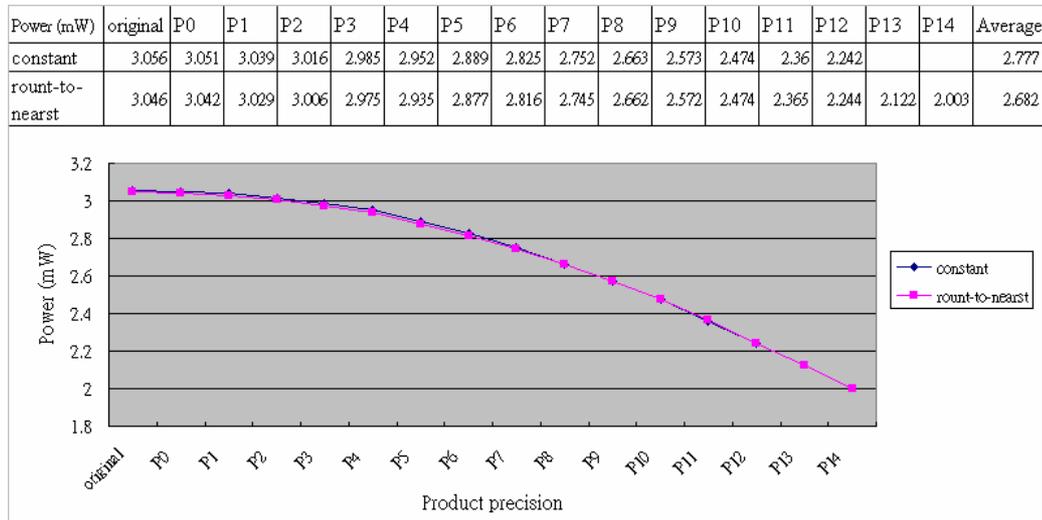


圖 9 二種修正法的功率比較

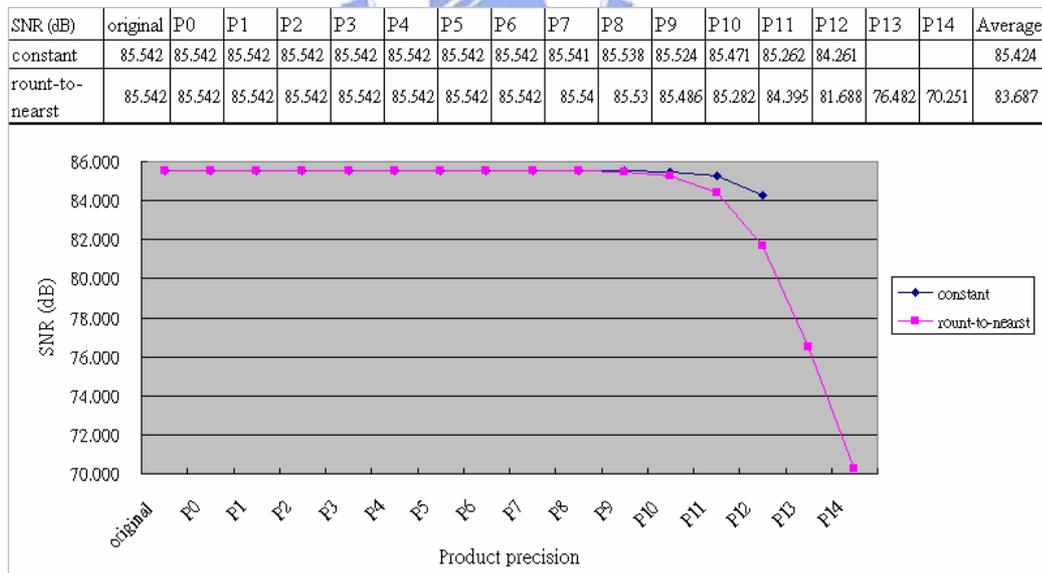


圖 10 二種修正法的 SNR 比較

圖 10 為二種修正法的 SNR 比較，可以發現係數修正法的平均 SNR

較高，為 85.424 dB，而圍繞至最接近修正法的平均 SNR 為 83.687 dB，因此這二種修正法的功率與 SNR 各有優劣，然而係數修正法最多只能節制到 P12 bit，為了讓我們有更多的模式(mode)來做選擇，我們將以圍繞至最接近修正法來做修正。

再來介紹功率意識指標，我們將損失品質的功率意識指標分為 3 個步驟，包括步驟 1: 求取功率意識系統的功率與 SNR，步驟 2: 由 SNR 設定 mode 數、 Q_{ave} 、以及 Q_{th} ，以及步驟 3: 使用 Battery simulation tool 求取 battery lifetime，以下分別敘述其做法。

步驟 1: 求取功率意識系統的功率與 SNR



我們以節制輸入設為零 (Inputs remain Zero, IZ) 為例，說明損失品質的功率意識指標做法。乘法器的輸入為被乘數與乘數，以 I 代表輸入，我們將節制輸入代表同時節制被乘數與乘數。所謂輸入設為零就是同時將被乘數與乘數較小的次要位元(lower bits)設為零，如圖 11 所示。

圖 11 為輸入設為零之 $n \times n$ bit 乘法，其中 X 與 Y 代表被乘數與乘數，S 代表部分乘積， $S_i, j = X_i \text{ AND } Y_j$ ，P 代表乘積。輸入設為零即將被乘數與乘數 I0 至 I_{n-k-1} bit 設為零，因此有些部分乘積 S 變成零，這樣的結果會減少乘法器內部加法器的開關動作

(switching activities)，所以會降低功率消耗，但缺點是 SNR 也會降低。我們將最後的乘積只取 n bits，亦即取 P_n 至 P_{2n-1} 的乘積。

									X_{n-1}	X_{n-2}	X_{n-k}	0	...	0	0
									Y_{n-1}	Y_{n-2}	Y_{n-k}	0	...	0	0
								0	0	0	0	0	0	0	0	0	0
							0	0		
				0	0	0	0	0	0	0	...	0	0				
						$S_{n-1,n-k}$	$S_{n-2,n-k}$	$S_{n-k,n-k}$	0	...	0	0			
						0	...	0	0				
						0	...	0	0				
						$S_{n-1,n-2}$	$S_{n-2,n-2}$	$S_{n-k,n-2}$	0	...	0	0			
						$S_{n-1,n-1}$	$S_{n-2,n-1}$	$S_{n-k,n-1}$	0	...	0	0			
P_{2n-1}	P_{2n-2}	P_{2n-3}	P_n	P_{n-1}	0	0	0

圖 11 輸入設為零之 $n \times n$ bit 乘法

圖 12 為 16×16 bit 乘法輸入設為零的功率，X 軸為輸入精確度 (input precision)，original 代表未節制任何輸入，I0 代表節制輸入 0 bit，亦即將被乘數與乘數第 0 bit 設為零，I1 節制輸入 0 至 1 bit，亦即將被乘數與乘數第 0 至 1 bit 設為零，I2 節制輸入 0 至 2 bit，亦即將被乘數與乘數第 0 至 2 bit 設為零，……，依此類推。因為 16×16 bit 的正負號乘法器，其輸入的 MSB (I15) 為 signed bit，因此輸入最多節制到 I13 bit。

須注意這裡的輸入精確度與 2.1 節未損失品質的輸入精確度代表意義不同，我們固定輸入位元為 16 bit，以節制乘法器的輸入來做損失品質的功率意識，因此輸入的位元會大於乘法器運算的位元。Y 軸為使用 Prime Power 模擬出來的功率，未節制任何輸入的功率最

高，為 3.046 mW，節制輸入 bit 數越多，乘法器的功率越小。

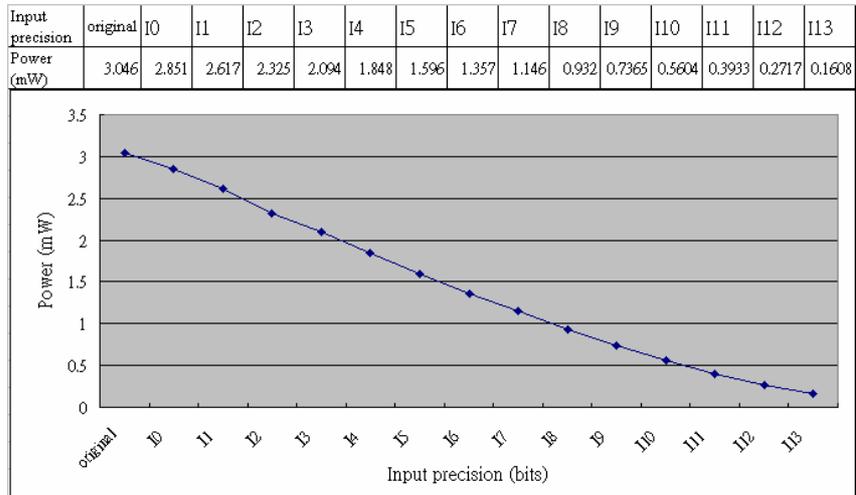


圖 12 16 × 16 bit 乘法輸入設為零的功率

圖 13 為 16 × 16 bit 乘法輸入設為零的 SNR，我們以(5)式計算 SNR，未節制任何輸入的 SNR 最高，為 85.542 dB，節制輸入 bit 數越多，計算出來的結果越不精確，因此 SNR 越小。由圖 12 與圖 13 得知，節制輸入 bit 數越多，消耗功率減少，但是相對地 SNR 也會越小。

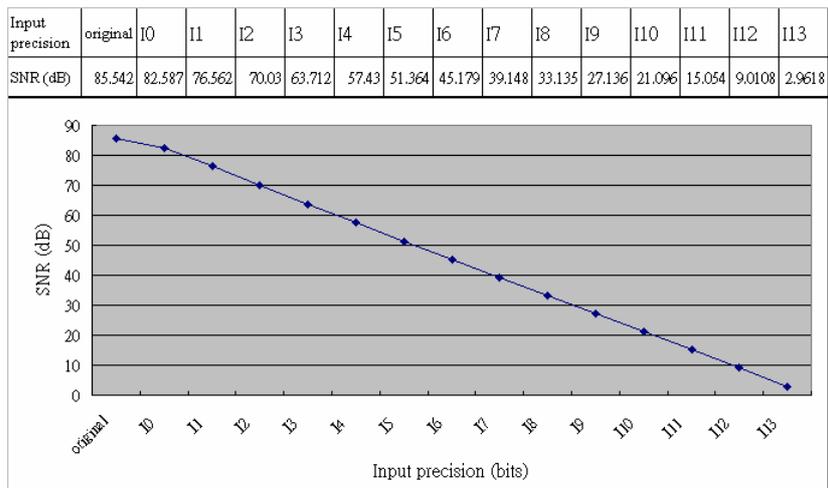


圖 13 16 × 16 bit 乘法輸入設為零的 SNR

步驟 2: 由SNR設定mode數、 Q_{ave} 、以及 Q_{th}

此步驟是使用者由SNR自行設定mode數、 Q_{ave} 、以及 Q_{th} 。mode數為操作模式的數目，所謂操作模式即為功率意識的模式。 Q_{ave} 為選定操作模式的SNR平均值，亦即在固定 Q_{ave} 的情況下求取battery lifetime。 Q_{th} 為品質臨界值(threshold)，SNR小於 Q_{th} 代表品質太差，因此不使用小於 Q_{th} 的操作模式。

以圖 13 為例，我們設定mode數為 4， $Q_{ave}=74$ dB， $Q_{th}=50$ dB。因為 $Q_{th}=50$ dB，Input precision從I6 至I13 的SNR小於 50 dB，所以不使用I6 至I13 的操作模式。mode數為 4，即Input precision從original至I5 之中挑選 4 個功率意識的操作模式，通常我們會選擇圖 13 的original，亦即未做任何節制為第一個操作模式，因為original雖然消耗功率最高，但是SNR最好，一般我們會希望電池剛使用的時候SNR是最好的，而當電池快沒電時又會希望降低SNR以換取battery lifetime，所以會保留original做為第一個操作模式。

當設定mode數為 4， $Q_{ave}=74$ dB，因此操作模式(original, I0, I2, I4)的 $Q_{ave}=(85.542+82.587+70.03+57.43)/4=73.897$ dB，另一個操作模式(original, I1, I2, I3)的 $Q_{ave}=(85.542+76.562+70.03+63.712)/4=73.962$ dB，若以小數點第一位做四捨五入計算，這二種組合的 Q_{ave} 皆可視為 74 dB，因此我們將以

(original, I0, I2, I4)與(original, I1, I2, I3)這二種組合來做 Battery lifetime的比較。

步驟 3:使用 Battery simulation tool 求取 battery lifetime

我們在固定的品質的情況下求取 battery lifetime，以 battery lifetime 最長者為佳。我們以 Battery Design 公司的 Battery Design Player 做為模擬 battery lifetime 的軟體，它可以將系統的電壓或電流模擬出電池的 lifetime。我們使用的 cell 為 HE18650，此為鋰離子電池，其規格與可容許的工作範圍如表 3 所列。

由表 3 得知鋰離子電池的正常工作電壓在 3.7 V，可容許的工作電壓範圍在 2.7 V 至 4.3 V 之間，在可容許的範圍內模擬出來的結果才是正確的。



Voltage, V	(Nominal Values)	3.7
Capacity, mAh		2100
Weight, g		41
Diameter, mm		16
Height, mm		65
Width, mm		0
Allowable ranges	min	max
Current, Amp	-2	6.3
Voltage, V	2.7	4.3
Temperature, °C	-30	70

表 3 HE18650 鋰電池之規格與工作容許範圍

我們先介紹電池的特性[21]，圖 14 為電流為 1 與 0.5 Amp 之 Battery lifetime 比較，X 軸為放電時間，即 battery lifetime，Y 軸為電池電壓。由圖 14 可得知，電流 1 Amp 的 battery lifetime 較 0.5 Amp 為短。

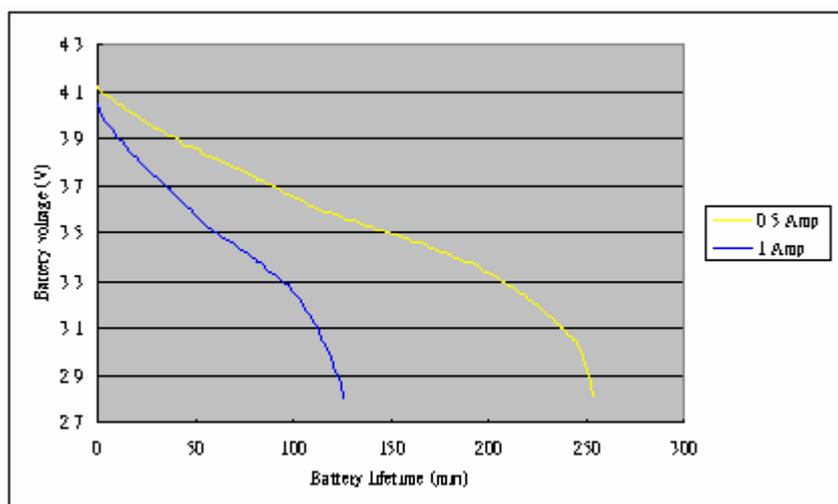


圖 14 電流為 1 與 0.5 Amp 之 Battery lifetime 比較

由於電池容量(Capacity)的定義為

$$C = I \cdot t$$

其中 C 為電池容量，I 為電流，t 為時間。由表 1 得知 HE18650 鋰電池的容量為 2100 mAh，若電流 1 Amp 則可以有 2.1 小時的使用時間，即 126 分鐘的 battery lifetime，若電流 0.5 Amp 則可以有 4.2 小時的使用時間，即 252 分鐘的 battery lifetime，由圖 14 可以驗證，因此電流 1 Amp 的 battery lifetime 較 0.5 Amp 為短。

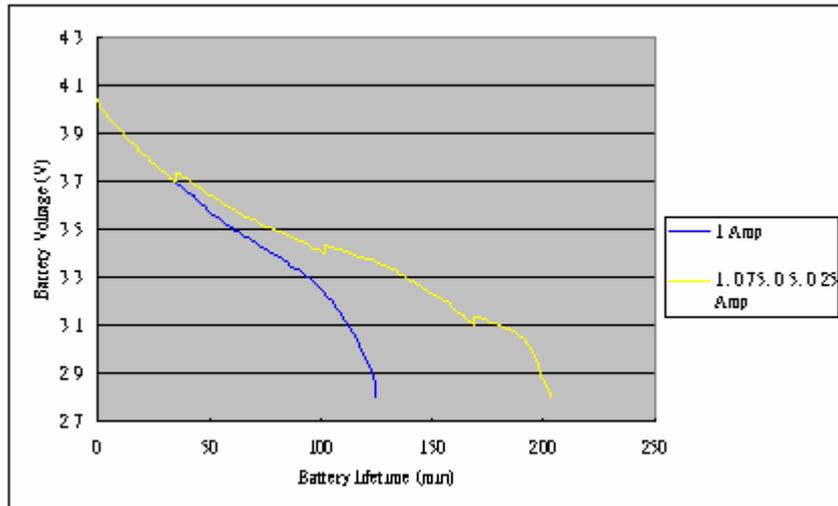


圖 15 電池放電作用

圖 15 為電池放電(discharge)作用，電流固定 1 Amp 稱為連續放電，此時放電時間較短。若將電流分段減少，如圖 15 依序設定為 1, 0.75, 0.5, 0.25 Amp，則稱為間接放電，此時放電時間較長，並且在切換電流的時候會產生電壓回升的情形，如圖所示產生鋸齒狀的電壓，這個結果有助於提高 battery lifetime。

而我們功率意識的操作模式(mode)切換，是將乘法器使用不同的功率模式，由於功率的定義為

$$P = I \cdot V \quad (8)$$

其中 P 為功率(單位 W)，I 為電流(單位 Amp)，V 為電壓(單位 V)。

我們將 mode 數設為 4，亦即乘法器使用 4 種不同的功率，並且由(8)式得到相對應的電流，利用不同電流的切換，產生電池間接放電作

用，以提高 battery lifetime。

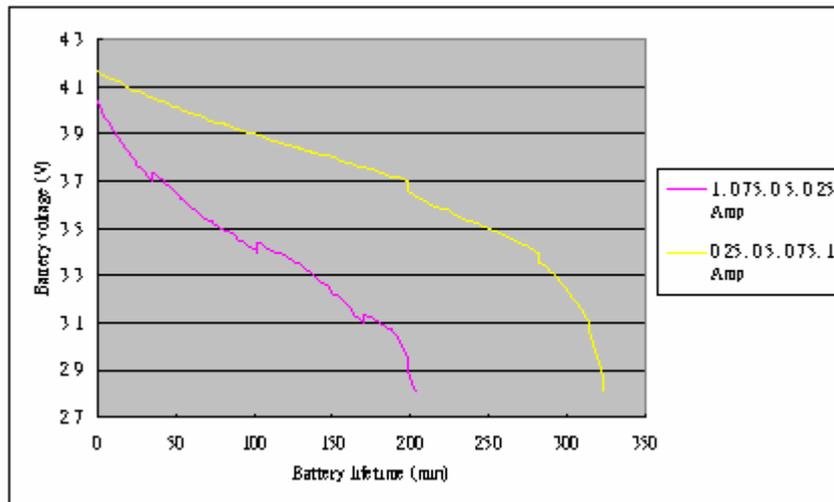


圖 16 相同電流但不同操作順序的電壓曲線

另外，我們需要探討模式的操作順序，圖 16 為相同電流但不同操作順序的電壓曲線，二者的電流皆為 1, 0.75, 0.5, 0.25 Amp 共 4 個模式，一個為電流依序遞減，另一個為電流依序遞增，而 4 個模式的電壓區間皆相同，我們設定為 4.0~3.7 V、3.7~3.4 V、3.4~3.1 V、以及 3.1~2.8 V 四個區間，由圖可得知電流依序遞增操作順序的 battery lifetime 較長。

我們再由電流與 battery lifetime 的關係來觀察此問題，圖 17 為相同電流但不同操作順序的電流曲線，由圖可得知雖然電流依序遞增(0.25, 0.5, 0.75, 1 Amp)的 battery lifetime 較長，在第一個模式 0.25 Amp 的 battery lifetime 最長，大約 200 分鐘，但是在第

四個模式 1 Amp 的 battery lifetime 最短，大約僅有 10 分鐘。應用在損失品質的功率意識上，電流小代表功率小，而功率小相對地 SNR 就小，也就是說，在第一個模式 SNR 最差但是使用時間最久，第四個模式 SNR 最好但是使用時間最短，這並非我們想要得到的，我們希望系統一開始使用的品質最好，而且使用的時間越長越好，當電池沒電時我們希望降低品質以換取 Battery lifetime，而且品質最差使用的時間越短越好。因此基於品質的考量，我們選擇電流依序遞減，亦即功率依序遞減的操作模式順序來產生功率意識乘法器。

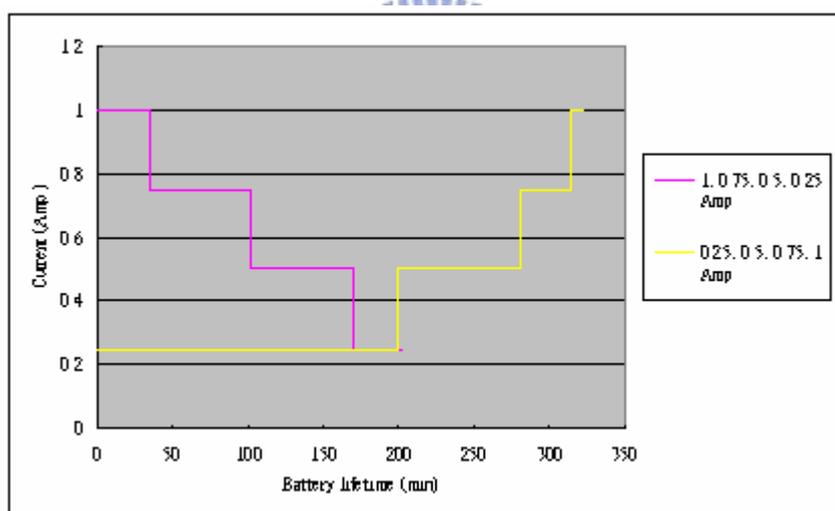


圖 17 相同電流但不同操作順序的電流曲線

電源端電池的電壓 V_{bat} 會因為長期使用而慢慢減少，但是負載端的電壓 V_{dd} 須維持一定，因此電源端與負載端中間需要加入調整器 (Regulator)，如圖 18 所示，電池的電壓 V_{bat} 經過調整器之後供給

固定電壓 V_{dd} 給負載端，所以在負載端功率維持定值，而電壓 V_{dd} 也固定，由(8)式負載電流 I_{out} 亦維持定值。

然而在電源端，電池電壓 V_{bat} 會慢慢減少，若是功率要維持定值，由(8)式電池電流 I_{in} 須慢慢增加，因此我們將電池作用視為離散(discrete)，在每一個操作模式的電池電壓分割成 10 個小區段，由已知的消耗功率求得電流，將每個小區段的電流輸入 Battery simulation tool 以求取 battery lifetime。

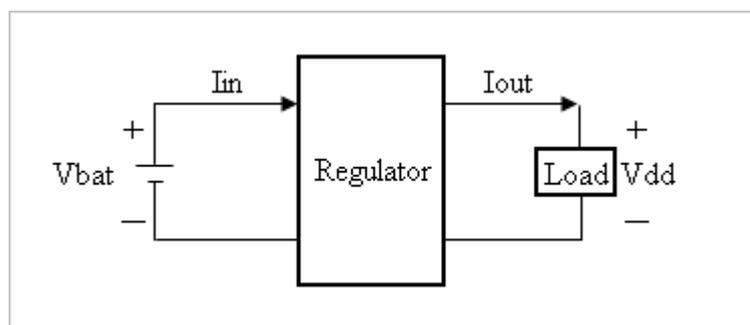


圖 18 電池與負載電路示意圖

介紹完電池特性之後，再來說明使用 Battery simulation tool 的做法。由表 3 得知電壓值最大 4.3 V，最小 2.7 V，因此我們選擇電壓值的工作範圍介於 4.1 V 至 2.9 V 之間。mode 數為 4，必須將 4.1 V 至 2.9 V 的電壓值平均分割成 4 個區段，所以 1 個區段的電壓為 $(4.1-2.9)/4=0.3$ V，為了得到精確的結果，我們將電池作用視為離散(discrete)，把每個電壓區段細分成 10 個小區段，因此每個小區段的電壓為 $0.3/10=0.03$ V。

表 4 為(original, I0, I2, I4)操作模式的條件，其中(mode 1, mode 2, mode 3, mode4)分別表示(original, I0, I2, I4)。由圖 12 可得知 mode 1 的功率為 3.046 mW，mode 2 的功率為 2.851 mW，mode 3 的功率為 2.325 mW，以及 mode 4 的功率為 1.848 mW。使用 Battery Design Player 需輸入以下的條件：

1. Step Type:

我們選擇” Discharge” 做為 Battery 的工作形態。

2. Control Condition:

我們選擇” Current” 做為控制條件，亦即使用電流來模擬 battery lifetime。因為 $P=I \cdot V$ ，已知 4 個 mode 的消耗功率以及 battery 電壓值從 4.1 V 至 2.9 V 分割成 40 個小區域，所以由 $I=P/V$ 可得到電流值。

表 4 的一列即顯示每個小區域的條件，以 mode 1 為例，Power = 0.003046 W，第一個小區域的電壓值為 4.1 V，因此電流值為 $0.003046/4.1$ Amp，第二個小區域的電壓值為 $4.1-0.03=4.07$ V，因此電流值為 $0.003046/4.07$ Amp，……依此類推。當第十一個小區域時，則跳至 mode 2，Power = 0.002851 W，第十一個小區域的電壓值為 $3.83-0.03=3.8$ V，因此電流值為 $0.002851/3.8$ Amp，第十二個小區域的電壓值為 $3.8-0.03=3.77$ V，因此電流值為

0.002851/3.77 Amp，……依此類推。最後第四十個小區域為 mode 4，
 Power=0.001848 W，電壓值為 2.93V，因此電流值為 0.001848/2.93
 Amp。

3. End Condition:

我們選擇” Voltage” 做為截止條件，亦即到達設定的電壓則結束此區段的模擬。第一個小區域的電壓值降至 $4.1-0.03=4.07$ V 則結束模擬，跳至第二個小區域，第二個小區域的電壓值降至 $4.07-0.03=4.04$ V 則結束模擬，跳至第三個小區域，……依此類推，最後第四十個小區域的電壓值降至 2.9 V 則結束模擬。

Mode	Power (W)	Step	Control Condition		End Condition	
		Type	Type	Value	Type	Value
1	0.003046	Discharge	Current	0.003046/4.1	Voltage	4.07
		Discharge	Current	0.003046/4.07	Voltage	4.04
	
		Discharge	Current	0.003046/3.83	Voltage	3.8
2	0.002851	Discharge	Current	0.002851/3.8	Voltage	3.77
		Discharge	Current	0.002851/3.77	Voltage	3.74
	
		Discharge	Current	0.002851/3.53	Voltage	3.5
3	0.002325	Discharge	Current	0.002325/3.5	Voltage	3.47
		Discharge	Current	0.002325/3.47	Voltage	3.44
	
		Discharge	Current	0.002325/3.23	Voltage	3.2
4	0.001848	Discharge	Current	0.001848/3.2	Voltage	3.17
		Discharge	Current	0.001848/3.17	Voltage	3.14
	
		Discharge	Current	0.001848/2.93	Voltage	2.9

表 4 (original, I0, I2, I4) 操作模式的條件

表 5 為(original, I1, I2, I3)操作模式的條件，其中(mode 1, mode 2, mode 3, mode4)分別表示(original, I1, I2, I3)。表 5 跟表 4 的差異在於 mode 2 與 mode 4 的功率不同，因此 Control Condition 的電流值亦不同，但是 End Condition 的電壓值皆一樣。

Mode	Power (W)	Step	Control Condition		End Condition	
		Type	Type	Value	Type	Value
1	0.003046	Discharge	Current	0.003046/4.1	Voltage	4.07
		Discharge	Current	0.003046/4.07	Voltage	4.04
	
		Discharge	Current	0.003046/3.83	Voltage	3.8
2	0.002617	Discharge	Current	0.002617/3.8	Voltage	3.77
		Discharge	Current	0.002617/3.77	Voltage	3.74
	
		Discharge	Current	0.002617/3.53	Voltage	3.5
3	0.002325	Discharge	Current	0.002325/3.5	Voltage	3.47
		Discharge	Current	0.002325/3.47	Voltage	3.44
	
		Discharge	Current	0.002325/3.23	Voltage	3.2
4	0.002094	Discharge	Current	0.002094/3.2	Voltage	3.17
		Discharge	Current	0.002094/3.17	Voltage	3.14
	
		Discharge	Current	0.002094/2.93	Voltage	2.9

表 5 (original, I1, I2, I3) 操作模式的條件

另外我們將溫度設定為室溫 25°C，報告時間(Report Time)設定為 5000 秒，亦即每 5000 秒記錄一筆資料。在輸入表 4 與表 5 之後，Battery Design Player 即可模擬出 battery lifetime。圖 19 為(original, I0, I2, I4)與(original, I1, I2, I3)操作模式的 battery

lifetime，其中(original, I0, I2, I4)的 battery lifetime 為 172465.7 分鐘，(original, I1, I2, I3)的 battery lifetime 為 176016.3 分鐘，由圖可得知(original, I1, I2, I3)的 lifetime 較長。

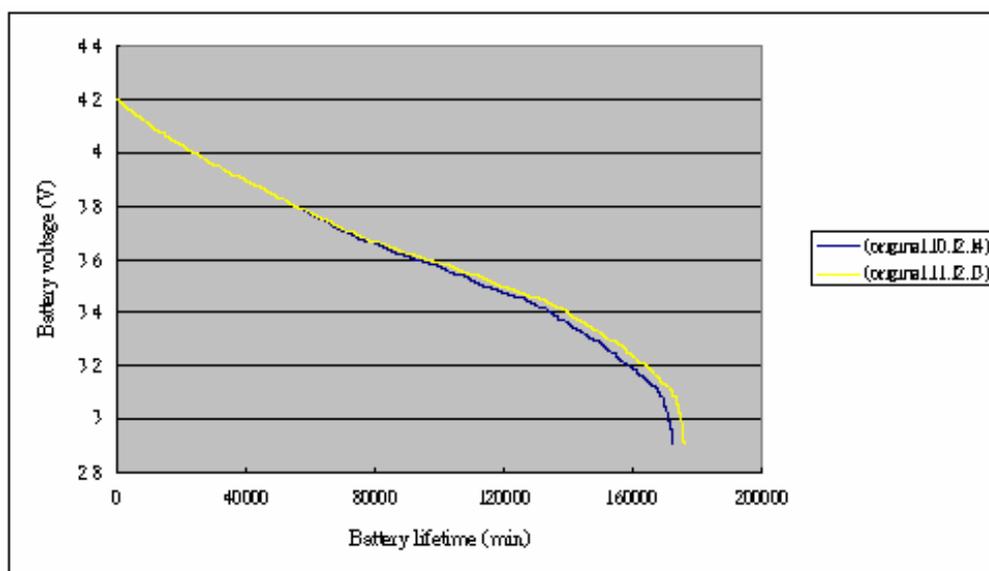


圖 19 (original, I0, I2, I4)與(original, I1, I2, I3)操作模式的電壓曲線

圖 20 為(original, I0, I2, I4)與(original, I1, I2, I3)操作模式的電流曲線，在同一 mode 中電流會呈階梯式的上升，這是因為同一 mode 的功率相同，但是電池電壓會逐漸下降，由(8)式得到電流會逐漸上升，我們將同一 mode 的電壓分割成 10 個小區域，所以電流會呈 10 個小區域階梯式的上升。比較這二種操作模式的 mode 1，因為二者的 mode 1 皆為 original，因此在 mode 1 的電流與 battery lifetime

相同。比較二者的 mode 2，因為 I1 的功率較 I0 小，由(8)式得知 I1 的電流亦較 I0 小，所以在 mode 2 中 I1 的 battery lifetime 較長。比較二者的 mode 3，因為這二種操作模式的 mode 1 皆為 I2，因此在 mode 3 的電流與 battery lifetime 相同。再來比較二者的 mode 4，因為 I3 的功率較 I4 大，其電流亦較大，所以在 mode 4 中 I3 的 battery lifetime 較短，但是因為 mode 4 的電壓已接近截止電壓，其 battery lifetime 都會很短，所以 I3 與 I4 的差距並不會太大。

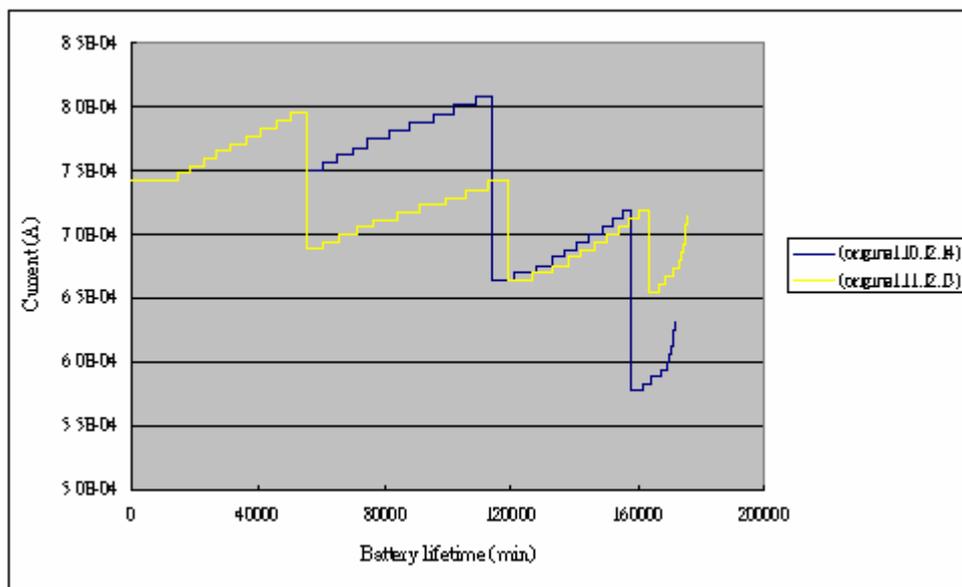


圖 20 (original, I0, I2, I4)與(original, I1, I2, I3)操作模式的電流曲線

圖 12 可算出(original, I0, I2, I4)的消耗功率為 $(3.046 + 2.851 + 2.325 + 1.848) = 10.07 \text{ mW}$ ，(original, I1, I2, I3)的消耗功率為

$(3.046 + 2.617 + 2.325 + 2.094) = 10.082 \text{ mW}$ ，雖然

(original, I1, I2, I3)的消耗功率較高，但是因為HE18650 鋰電池的正常工作電壓為 3.7 V，在此電壓附近的條件影響 battery lifetime 較大，3.7 V 的電壓在 mode 2 範圍內，因為(original, I1, I2, I3)在 mode 2 的消耗功率較小，所以其 battery lifetime 較長。

