

微處理與計算機詳解

Microprocessor and Digital Computer Technology

Oleksy Rutkowski

1			NAM	FILLER	REV 0.1
2			OPT	NOP	
3			ORG	\$FC00	
4	07	007F	AI	\$007F	
5	007F	007F	PRESET	EQU	\$007F
6	2004	2004	PIAOUT	EQU	\$2004
7	007F	007F	LD	EQU	\$007F
8	007F	007F	LD	EQU	\$007F
9	2007	2007	CRB	EQU	\$2007
10			** INITIALIZATION		
11	FC00	0F	INIT	SETI	DISABLE INTERRUPTS DURING INIT
12	FC01	80	LD	#0	INIT STACK POINTER
13	FC04	86 05	LDAA	#05	LOAD ACCA WITH DATA DIR BYTE
14	FC06	B7 2004	STAA	PIAOUT	STORE IN DDRA
15	FC09	86 07	LDAA	#07	LOAD ACCA WITH CONTROL BYTE
16	FC0B	B7 2005	STAA	CRA	STORE IT IN CRA
17	FC0E	86 06	LDAA	#06	LOAD ACCA WITH CONTROL BYTE
18	FC10	B7 2007	STAA	CRB	STORE IT IN CRB
19					
20	FC11	86 2007	LDAA	CRB	POLL CRB
21	FC12	84 80	AND	#180	KEY PRESSED? (BIT 7 -
22	FC13	27 F9	BEO	GETN	NO-POLL AGAIN
23	FC14	86 2006	LDAA	GETN	NO-POLL AGAIN
24	FC15	48	ASLA		SHIFT LEFT 4 TIMES
25	FC16	48	ASLA		TO FORM MSD
26	FC17	48	ASLA		
27	FC18	48	ASLA		
28	FC19	B7 007F	STAA	PRESET	STORE MSD IN PRESET
29	FC1A	B6 2007	LDAA	CRB	POLL CRB
30	FC1B	84 80	AND	#180	KEY PRESSED? (BIT 7 -
31	FC1C	27 F9	BEO	GET2	NO-POLL AGAIN
32	FC1D	86 2005	LDAA	PIAIN	YES-GET IT
33	FC1E	8A 007F	ORAA	PRESET	OR IT WITH PRESET TO
34	FC1F	B7 007F	STAA	PRESET	STORE IT BACK IN PRE
35			** MAIN TASK - START CONV		
36	FC20	0F	CLI		ENABLE INTERRUPTS
37	FC21	86 01	LDAA	#01	SET RUN, RESET FILL
38	FC22	B7 2004	STAA	PIAOUT	
39	FC23	B6 2004	LDAA	PIAOUT	POLL PIAOUT
40	FC24	84 02	AND	#02	BOX PRESENT? (BIT 1 -
41	FC25	27 F9	BEO	NEXBOX	NO-POLL AGAIN
42	FC26	86 04	LDAA	#04	YES-RESET RUN, SET I
43	FC27	B7 2004	STAA	PIAOUT	
44	FC28	7F 007E	CLR	PARTS	CLEAR TEMP STORAGE
45			** LOOP WHILE WAITING FOR INTERRUPT		
46	FC29				
47	FC4C	B1 007F	CPMA	PRESET	PARTS = PRESET?
48	FC4F	26 FB	BNE	TEST	NO-LOOP AGAIN

版權所有・翻印必究

中華民國七十二年一月初版

微處理與計算機詳解

定價：新台幣 90 元

原著者：Oleksy • Rutkowski

譯著者：林 振 發

發行人：黃 旭 政

發行所：曉 國 出 版 社

臺北市永康街12巷2-3號

電話：(02) 394-9931 三線

郵撥：一九四五三號

門市部：台北市新生南路三段96號之三

電話：3917012 • 3947375

台北市重慶南路一段61號地下樓

電話：三一四九五八〇

印刷所：遠 大 印 刷 廠

臺北市武成街36巷16弄15號

出版登記：局版合業字第 1244 號

著作執照：台內著字第 號

前 言

研習理工的同學，都有一種認識，那就是：一本書的習題往往是該書的精華所在，藉着習題的印證，才能對書中的原理原則澈底的吸收與瞭解。

有鑒於此，曉園出版社特地聘請了許多在本科上具有相當研究與成就的人士，精心出版了一系列的題解叢書，為各該科目的研習，作一番介紹與鋪路的工作。

一個問題的解答方法，常因思惟的角度而異。曉園題解叢書，毫無疑問的都是經過一番精微的思考與分析而得。其目的在提供對各該科目研讀時的參考與比較；而對於一般的自修者，則有啓發與提示的作用。希望讀者能藉着這一系列題解叢書的幫助，而在本身的學問進程上有更上層樓的成就。

微處理與計算機詳解

(目 錄)

第一章	電子計算機發展史.....	1
第二章	計算機與微處理機的數碼.....	3
第三章	軟體程式.....	15
第四章	計算機系統的分離元件與邏輯閘.....	29
第五章	搖擺器、解碼器、計數器與其他MSI 配備.....	65
第六章	記憶體.....	91
第七章	計算機之結構.....	105
第八章	微處理器.....	123
第九章	界面裝置 	133
第十章	程式語言、格式及輸出/入裝置.....	147
第十一章	微處理機之應用.....	151
第十二章	8085 A 與 Z-80 微處理機.....	169

贈閱

北京农业工程大学图书馆

年 月 日

第一章 電子計算機發展史

複習問題

1. 電子計算機 (computer) 與計算器 (calculator) 有何不同 ?

圖：電子計算機與計算器最大之不同在於電子計算機為可程式化 (programmable)，其可儲存程式、資料，並執行程式，而計算器無法執行程式。

2. 誰最先發現將卡片 (punch-card) 用於可程式化之計算機 ?

圖：Charles Babbage 首先將計算機程式登入在卡片上。

3. 列出現代電子計算機優於第一代電子計算機之點。

圖：(1)重量減輕，體積縮小。

(2)執行程式速度提高。

(3)機器溫度降低，價格降低，第一代計算機使用真空管而現代計算機使用積體電路 (IC)。

4. 何謂積體電路 (IC) ?

圖：所謂積體電路乃將一些 (數百或數千個) 電晶體濃縮於一個小晶片上 (chip) 以達到某種特定之功能，如加、減、乘、除、和閘 (AND gate)，等功能。

5. 積體電路 (IC) 之基本材料是什麼 ?

圖：IC之基本材料是由矽與鍺構成。

6. 在第二代電子計算機所使用的電子交換電路稱為什麼 ?

圖：在第二代電子計算機使用 IC 以達到電子交換 (electronic switch) 之功能。

7. 什麼是硬體 ? 什麼是軟體 ?

圖：一般而言，硬體乃指計算機的電子電路與機器本身，而軟體是指，命令計算機作工之程式。

8. 變動硬體與變動軟體何者較易？

答：變動軟體較容易，因硬體設備已固定，而軟體是由程式組成，而程式較好變動。

9. 何謂（MPU）微處理機單元？

答：將電子計算機部份或大部份之電路與功能濃縮在一片大型積體電路上（LSI；large scale IC），以達到電子計算機之部份或大部份功能。

10. 為何MPU在應用上成長迅速？

答：因MPU已具有計算機之部份功能故可用於各種用途，如控制系統，自動化系統等，故發展迅速。

第二章 計算機與微處理機的數碼

複習問題

1. 為何數位計算機使用二進位系統？

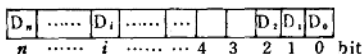
解：因數位計算機硬體電路乃採用邏輯電路（logical circuit）所構成，而二進位系統中的每一比特（bit）可代表邏輯電路中的開與關（ON, or OFF），因每一 bit 可表示 0 或 1。

2. 在數位系統中，HIGH 與 LOW 的信號具有何種意義？

解：在數位系統裡，HIGH 與 LOW 之信號表示 1 或 0，即 HIGH 表示 1，LOW 表示 0。

3. 何為二進位系統被稱為加權重量（weight）？

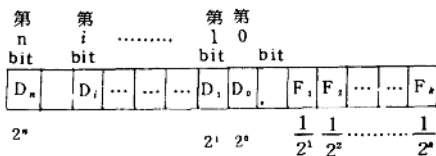
解：當一個二進位數實際上的值在十進位系統中是以加權重量來表示，換句話說二進位的加權重量是以二進位中任何一 bit 為 1 時所在之位址而定，如第 0 bit 其加權重量為 1，第 1 bit 加權重量為 2 等；以下圖作說明：



共有 $n+1$ bits 每一 bit 之加權重量為 2^i （ $i=0 \sim n$ ）即第 i bit 的加權重量 = 2^i 而二進位所表示的值化為十進位時即為每一 bit 的加權重量之和，就是 $\sum_{i=0}^n D_i \times 2^i$ （ $D_i = 0$ or 1 ）

4. 說明二進位數（包括小數部份）之加權重量的分佈。

解：根據問答(3)我們可知二進位所代表的值化為十進位時，是以其加權重量來計算，故我們可得到下列之加權重量之公式：



4. 微處理與計算機詳解

整數部份加權重量 = 2^i ($i = 0 \sim n$)

小數部份加權重量 = $\frac{1}{2^j}$ ($j = 1 \sim k$)

5. 在二進位數位中混合使用小數點之目的為何？

圖：以上題為例，其為帶有小數點之二進位，其目的可使其所表示的值更加精確。

6. 列出四種使用於計算機與微處理機的數位系統。

圖：(1)二進位系統

(2)八進位系統

(3)十六進位系統

(4)BCD 系統 (binary coded decimal)

7. 什麼是二進位的一的補數 (one's complement)？

圖：將二進位中每一 bit 作一種轉換，即把 bit 為 1 的變換成 0，而 bit 為 0 的變換成 1。

$$D_i = 0 \text{ (when } D_i = 1 \text{)}$$

$$\text{and } D_i = 1 \text{ (when } D_i = 0 \text{)} \quad i = 0 \sim n$$

8. 什麼是二進位的二的補數 (two's complement)？

圖：將二進位取一的補數後再加 1，而進位不予考慮，

$$\begin{aligned} 2'S \text{ complement} &= 1'S \text{ complement} + 1 \\ &\text{if carry don't care} \end{aligned}$$

9. 正數與負數如何以邏輯符號表示？(sign bit)

圖：當正數最左邊的一個 bit 為 0，而負數時最左邊的一個 bit 為 1。

10. 那些 ASCII code (美國標準編碼) 表示字母？

圖：見圖 2-21，ASCII 從 41 ~ 5A 表示 A ~ Z。

11. 使用配對比特 (parity bit) 的目的為何？

圖：配對比特 (parity bit) 分兩種型式一種為奇配對 (odd parity) 一種為偶配對 (even parity) 其最主要功用為幫助檢查資料之正確性。

習題

- 2-1 轉換下列各二進位值至對應的十進位值(a) 0111, (b) 110100(c) 01001001(d) 10000001 (e) 11111111(f) 10101010(g) 01010101 (g) 100000000

解：因二進位要變換為十進位乃以加權重量計算，其公式如下：

$$D_n = \sum_{i=0}^n B_i \cdot 2^i \quad (i=0 \sim n, \text{ 即二進位之 bit 的數目 })$$

($B_i = 1$ or 0 視其 bit 的位置所在)

(2^i 為加權重量)

$$\therefore (a) D_3 = \sum_{i=0}^3 B_i \cdot 2^i = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 = 1 + 2 + 4 = 7$$

$\therefore$ 根據公式得出：(a) $0111_2 = 7_{10}$

$$(b) 110100_2 = 52_{10}$$

$$(c) 01001001_2 = 73_{10}$$

$$(d) 10000001_2 = 129_{10}$$

$$(e) 11111111_2 = 255_{10}$$

$$(f) 10101010_2 = 170_{10}$$

$$(g) 01010101_2 = 85_{10}$$

$$(h) 100000000_2 = 256_{10}$$

- 2-2 轉換下列二進位數為對應的十進位值，(a) 1110 (b) 100110, (c) 01110111 (d) 11111110 (e) 01111111 (f) 11100110 (g) 10000000 (h) 110011101

解：根據問題 2-1 解之公式得出：(a) 14_{10} , (b) 38_{10} , (c) 119_{10} , (d) 254_{10} , (e) 127_{10} , (f) 230_{10} , (g) 128_{10} , (h) 413_{10}

- 2-3 將下列十進位轉換為二進位(a) 12 (b) 31 (c) 32 (d) 105 (e) 254 (f) 300 (g) 2047 (h) 2048 (i) 5280 (j) 16384.

解：十進位要換為 2 進位時，將十進位一直除 2 所得出之餘數與最後一個餘數與商（非 1 即 0）重新併在一起即得出對應之二進位，方法以下例作說明：（十進位之 132）

6 微處理與計算機詳解

$$\begin{array}{r}
 2 \overline{) 132} \quad 0 \leftarrow \text{第一次除2之餘數}=0 \text{ (因商尚比2大}\therefore\text{必須除至比2小)} \\
 2 \overline{) 66} \quad 0 \leftarrow \text{第二次除2之餘數}=0 \\
 2 \overline{) 33} \quad 1 \leftarrow \text{第三次除2之餘數}=1 \\
 2 \overline{) 16} \quad 0 \quad \quad \quad " \\
 2 \overline{) 8} \quad 0 \quad \quad \quad \text{以此類推} \\
 2 \overline{) 4} \quad 0 \\
 2 \overline{) 2} \quad 0 \\
 1 \quad \leftarrow \text{最後得出之商}
 \end{array}$$

再將每次所得之餘數加上最後之商排列起來得出 $132 = 10000100$ ，
根據上述方法得出(a) 1100 (b) 11111 (c) 100000 (d) 1101001 (e)
11111110 (f) 100101100 (g) 1111111111 (h) 100000000000 (i)
101001100000 (j) 100000000000000000

- 2-4 將下列十進位數目換成二進位數目：(a) 15 (b) 62 (c) 64 (d) 152 (e)
511 (f) 512 (g) 756 (h) 1023 (i) 1024 (j) 65536

解：根據 2-3 解之方法可得出(a) 1111 (b) 111110 (c) 1000000 (d)
10011000 (e) 111111111 (f) 1000000000 (g) 1011110100 (h)
1111111111 (i) 10000000000 (j) 1000000000000000

- 2-5 將問題 2-1 中的(a)~(h)之二進位換成對應的八進位。

解：二進位，八進位，十六進位事實上是一體的三面，二進位化為八進位之作法是將二進位從最右邊開始，每三個bits為一組（最左邊一組可能不足 3 bits）後再將每組分別換成十進位對應之值即可。

$$\begin{array}{r}
 \text{例：} \quad 1100101_2 = 1 \quad | \quad 100 \quad | \quad 101 = 145_8 \\
 \begin{array}{|c|c|c|} \hline \text{第} & \text{第} & \text{第} \\ \text{三} & \text{二} & \text{一} \\ \text{組} & \text{組} & \text{組} \\ \hline \end{array}
 \end{array}$$

所以根據此法得出(a) 07, (b) 64, (c) 111, (d) 201, (e) 377, (f) 252,
(g) 125, (h) 400。

- 2-6 將問題 2-2 (a)~(h)之二進位換成八進位：

解：(a) 16, (b) 46, (c) 167, (d) 376, (e) 177, (f) 346, (g) 200, (h) 635。

- 2-7 將問題 2-3 所列(a)~(j)之十進位換為十六進位值

解：十進位換為十六進位時先將十進位換成二進位，再將二進位每四 bits 分成一組（從最右邊開始分起），再將每組換成十六進位數，即得

(a) C (b) 1F (c) 20 (d) 69 (e) FE (f) 12C (g) 7FF (h) 800 (i) A60 (j) 4000

2-8 將問題 2-4 之 (a)~(j) 之十進位值換成八進位，與十六進位。

解：十進位換成八進位與十六進位時先將十進位換成二進位，再換成八進位或十六進位。∴得出

八進位：(a) 17 (b) 74 (c) 100 (d) 230 (e) 777 (f) 1000 (g) 1364 (h) 1777 (i) 2000 (j) 200000

十六進位：(a) F (b) 3E (c) 40 (d) 98 (e) 1FF (f) 200 (g) 2F4 (h) 3FF (i) 400 (j) 10000

2-9 將下列八進位換成 8-bit 的二進位：(a) 10 (b) 25 (c) 37 (d) 77 (e) 100 (f) 233 (g) 356 (h) 377

解：將八進位換成二進位時只將每一八進位的位數從右至左依次換成三個 bits 的二進位即可。∴得出（不滿八 bit 補零）

(a) 00001000 (b) 00010101 (c) 00011111 (d) 00111111 (e) 01000000 (f) 10011011 (g) 11101110 (h) 11111111。

2-10 將下列八進位換成 8-bit 的二進位：(a) 45 (b) 64 (c) 177 (d) 200 (e) 277 (f) 300 (g) 333 (h) 366

解：(a) 00100101 (b) 00110100 (c) 01111111 (d) 10000000 (e) 10111111 (f) 11000000 (g) 11011011 (h) 11110110

2-11 在問題 2-9 中的二進位數值那些將第 8 bit 設定為 1 (HIGH)？

解：由問題 2-9 解中得知，(f)，(g)，(h) 之二進位數值皆將第 8 bit 設定為 HIGH

2-12 在問題 2-10 中的二進位數值那些將第 8 bit 設定為 1 (HIGH)？

解：由問題 2-10 解得知 (d)，(e)，(f)，(g)，(h) 將第 8 bit 設定為 1 (HIGH)

2-13 將下列十六進位換成 8-bit 的二進位 (a) 0A (b) C1 (c) EB (d) 7F (e) 8D (f) 49

解：將十六進位換成二進位時只須將每一位數依次換成 4-bit 的二進位即可， $\therefore$ 得出

(a) 00001010 (b) 11000001 (c) 11101011 (d) 01111111 (e)
10001101 (f) 01001001

2-14 將下列十六進位換成 8-bit 的二進位：(a) 50 (b) A6 (c) BF (d) 77 (e)
1D (f) 35

解：(a) 01010000 (b) 10100110 (c) 10111111 (d) 01110111 (e)
00011101 (f) 00110101

2-15 問題 2-13 中那些二進位的值將第 8 bit 設為 HIGH (set to HIGH) ?

解：(b), (c), (e)

2-16 問題 2-14 中那些二進位值將第 8 bit 設為 1 ?

解：(b), (c)

2-17 如最左邊的 bit (第 8 bit) 第一個 8-bit 的字元中為正負號 bit (sign bit)，sign bit 為 HIGH 時表示負數，LOW 時表示正數，則問題 2-13 中那些為負數？並將其換為十進位。

解：由問題 2-13 解得知為負數的有：

(b) $C1_{16} = 11000001_2 = -63_{10}$ 。(負數之第 8 bit = 1，而將此字
元取 2 的補數後即為負值)

(c) $EB_{16} = 11101011_2 = -21_{10}$ 。

(e) $8D_{16} = 10001101_2 = -115_{10}$ 。

2-18 如在 8-bit 字元中最左邊一個 bit 為 sign bit，sign bit 為 HIGH 時表示負數，而為 LOW 時表示正數，則(a)在問題 2-10 中那些八進位的數值為負？(b)問題 2-14 中那些十六進位的數為負數？(c)將這些數換成十進位：

解：(a)問題 2-10 中之八進位數為負數有：(d) $200_8 = 10000000_2 = -128_{10}$
(e) $277_8 = 10111111_2 = -65_{10}$
(f) $300_8 = 11000000_2 = -64_{10}$
(g) $333_8 = 11011011_2 = -37_{10}$
(h) $366_8 = 11110110_2 = -10_{10}$

(b)問題 2-14 中之十六進位數為負數有：

$$(b) A6_{16} = 10100110_2 = -90_{10}$$

$$(c) BF_{16} = 10111111_2 = -65_{10}$$

2-19 完成下列工作：(a) $1000 + 0111$ (b) $1010 + 0101$ (c) $0011 + 0111$
(d) $1100 + 0100$ (e) $1111 + 1111$ (f) $01001110 + 00110111$

解：(a) $1000 + 0111 = 1111$ (b) $1010 + 0101 = 1111$ (c) $0011 + 0111 = 1010$ (d) $1100 + 0100 = 10000$ (e) $1111 + 1111 = 11110$ (f)
 $01001110 + 00110111 = 10000101$

(十進位：(a) $1111_2 = 15_{10}$ (b) $1111_2 = 15_{10}$ (c) $1010_2 = 10_{10}$

(d) $10000_2 = 16_{10}$ (e) $11110_2 = 30_{10}$ (f) $10000101_2 = 133_{10}$)

2-20 完成下列工作：(a) $1010 + 0110$ (b) $1101 + 0100$ (c) $1110 + 0111$
(d) $0011111 + 00011101$ (e) $00010000 + 00010001$

解：(a) $1010 + 0110 = 10000_2 = 16_{10}$ (b) $1101 + 0100 = 10001_2 = 17_{10}$

(c) $1110 + 0111 = 10101_2 = 21_{10}$ (d) $0011111 + 00011101 = 00111100_2 = 60_{10}$ (e) $00010000 + 00010001 = 00100001_2 = 33_{10}$ (一的補數乃將字元中每一 bit 反相，1 變 0，0 變 1)

2-21 列出問題 2-1 各數的一的補數：

解：(a) $0111 \rightarrow 1000$ (b) $110100 \rightarrow 001011$ (c) $01001001 \rightarrow 10110110$
(d) $10000001 \rightarrow 01111110$ (e) $11111111 \rightarrow 00000000$ (f)
 $10101010 \rightarrow 01010101$ (g) $01010101 \rightarrow 10101010$ (h) $100000000 \rightarrow 011111111$

2-22 列出問題 2-2 中各數的一的補數：

解：(a) $1110 \rightarrow 0001$ (b) $100110 \rightarrow 011001$ (c) $01110111 \rightarrow 10001000$
(d) $11111110 \rightarrow 00000001$ (e) $01111111 \rightarrow 10000000$ (f) $11100110 \rightarrow 00011001$ (g) $10000000 \rightarrow 01111111$ (h) $110011101 \rightarrow 001100010$

2-23 將下列十六進位換成二進位後取 2 的補數之後，再換成對應的八進位與十六進位之值：(a) 2B (b) D4 (c) 00 (d) FF。

解：二的補數乃為一的補數加一，(進位不考慮) ∴ 得出下列各項之值

十六進位	二進位	二的補數	八進位	十六進位
2B	00101011	11010101	325	D5
D4	11010100	00101100	054	2C
00	00000000	00000000	000	00
FF	11111111	00000001	001	01

2-24 將下列各八進位之數換成對應 8-bit 的二進位，後取 2 的補數，再將其換成八進位與十六進位：(a) 001 (b) 377 (c) 137 (d) 200

解：

八進位	二進位	二的補數	十六進位	八進位
(a) 001	00000001	11111111	FF	377
(b) 377	11111111	00000001	01	001
(c) 137	01011111	10100001	A1	241
(d) 200	10000000	10000000	80	200

2-25 將問題 2-3 中各數換成對應的 BCD 碼之值。

圖：將十進位換成 BCD 碼乃將十進位從最左邊的位數依次將每位數換成對應的二進位即為對應的 BCD 碼：故

碼	(a)	(b)	(c)	(d)
十進位數	12	31	32	105
BCD 碼	00010010	00110001	00110010	000100000101

(e)	(f)	(g)	(h)
254	300	2047	2048
001001010100	001100000000	001000000100111	0010000001001000

(i) 5280₁₀ → 0101001010000000

(j) 16384 → 00010110001110000100

2-26 將問題 2-4 各項之值換成對應的 BCD 碼

圖：參照問題 2-4 與問題 2-25 解得：

(a) 15 → 00010101 (b) 62 → 01100010 (c) 64 → 01100100

(d) 152 → 000101010010 (e) 511 → 010100010001

(f) $512 \rightarrow 010100010010$ (g) $756 \rightarrow 011101010110$ (h) $1023 \rightarrow 0001000000100011$ (i) $1024 \rightarrow 0001000000100100$ (j) $65536 \rightarrow 01100101010100110110$

- 2-27** 完成下列工作：將減數取二的補數再與被減數相加，所得結果如有進位至第 9 bit 時可忽略其進位：(a) $00001100 - 00000100$
 (b) $00011111 - 00001011$ (c) $00001000 - 00001010$ (d)
 $00000111 - 00111110$

解：(a) $00001100 - 00000100 = 00001100 + 11111100 \rightarrow$
 第 9 bit 100001000 因第 9 bit 為 1 $\therefore$ 忽略掉 $\therefore = 00001000 = 08_H$
 (b) $00011111 - 00001011 = 00011111 + 11110101 = 00010100$
 $= 14_H$ (忽略第 9 bit)
 (c) $00001000 - 00001010 = 00001000 + 11110110 = 11111110$
 (無進位) $\therefore$ 取其 2'S 補助故得出 (負 0000010) $= -2_{10} = -2_{10}$
 (d) $00000111 - 00111110 = 00000111 + 11000010 = 11001001$
 $= C9_H = -37_H = -55_{10}$

- 2-28** 將下列各數的減數取二的補數後再與被減數相加，如第 9 bit 有進位時將其忽略：(a) $00010111 - 00000111$ (b) $01110100 - 00110000$ (c) $00110000 - 01110100$ (d) $00010111 - 00100000$

解：(a) $00010111 - 00000111 = 00010111 + 11111001 = 00010000$
 $= 10_H$
 (b) $01110100 - 00110000 = 01110100 + 11010000 = 01000100$
 $= 44_H$
 (c) $00110000 - 01110100 = 00110000 + 10001100 = 10111100$
 $= BC_H = -44_H = -68_{10}$
 (d) $00010111 - 00100000 = 00010111 + 11100000 = 11110011$
 $= F3_H = -0D_H = -13_{10}$

- 2-29** 實際上最左邊的 bit 為正負號 bit (sign bit) 其為 1 (HIGH) 時表示負值為 0 時表示正值，則問題 2-27 中 (a) 那些結果為負值？
 (b) 其負值乃以二的補數之形式表示，故將其換為二進位與十進位之值。

解：見問題 2-27 解之 (c), (d),

- 2-30** 事實上最左邊的 bit 為正負號 bit (sign bit) 其為 1 時表示負值，為 0 時表示正值，則問題 2-28 中那些結果為負值？其負值乃以二的補數之形式表示，故將其換為二進位與十進位之值。

解：見問題 2-28 中之其中(c)與(d)。

- 2-31** 將下列各數之十進位值換成 8 bit 二進位或二的補數：(a) +2 (b) -2 (c) +63 (d) -63 (e) +116 (f) -116

解：(a) +2 = 00000010 (b) 負數乃將正數取二的補數 $\therefore -2 = 11111110$
(c) +63 = 00111111 (d) -63 = 11000001 (e) +116 = 01110100 (f) -116 = 10001100。

- 2-32** 將下列各字換成對應的 7-bit ASCII 碼 (美國標準碼)：(a) G (b) J (c) Z (d) ! (e) @ (f) & (g) 4 (h) 9 (i) ；

解：參照 ASCII 表得出：(a) G = 47 (b) J = 4A (c) Z = 5A (d) ! = 21
(e) @ = 40 (f) & = 26 (g) 4 = 34 (h) 9 = 39 (i) ； = 3B

- 2-33** 將下列各字或指令換成其對應的 6-bit ASCII 碼：(a) C (b) L (c) line feed (d) carriage return (e) 0 (f) ϕ (g) 2 (h) + (i) =

解：參照 ASCII 表：(a) C = 02 (b) L = 14 (c) line feed = 001010
(d) carriage return = 001101 (e) 0 = 17 (f) ϕ = 60 (g) 2 = 62 (h) + = 53 (i) = = 75

- 2-34** 圖 2-21 表中將指令 null 至 rubout 填入 7 bit ASCII 碼中 (以十六進位)，與填入 6-bit trimmed ASCII 碼 (以八進位表示)：(圖 2-21 見次頁)

解：Null = 000_h = 00_H (7 bits) = 00_h (6 bits)
Horizontal tab = 011_h = 09_H (7 bits) = 11_h (6 bits)
Line feed = 012_h = 0A_H (7 bits) = 12_h (6 bits)
Vertical tab = 013_h = 0B_H (7 bits) = 13_h (6 bits)
Form feed = 014_h = 0C_H (7 bits) = 14_h (6 bits)
Carriage return = 015_h = 0D_H (7 bits) = 15_h (6 bits)
Rubout = 177_h = 7F_H (7 bits) (no 6 bits ASCII code)

Printing character	7-Bit ASCII (octal)	7-Bit ASCII code (hex)	6-Bit trimmed ASCII (octal)	Printing character	7-Bit ASCII (octal)	7-Bit ASCII code (hex)	6-Bit trimmed ASCII (octal)
@	100	40	00	(Space)	040	20	40
A	101	41	01	!	041	21	41
B	102	42	02	"	042	22	42
C	103	43	03	£	043	23	43
D	104	44	04	\$	044	24	44
E	105	45	05	%	045	25	45
F	106	46	06	&	046	26	46
G	107	47	07	'	047	27	47
H	110	48	10	(	050	28	50
I	111	49	11	)	051	29	51
J	112	4A	12	*	052	2A	52
K	113	4B	13	+	053	2B	53
L	114	4C	14	,	054	2C	54
M	115	4D	15	-	055	2D	55
N	116	4E	16	.	056	2E	56
O	117	4F	17	/	057	2F	57
P	120	50	20	0	060	30	60
Q	121	51	21	1	061	31	61
R	122	52	22	2	062	32	62
S	123	53	23	3	063	33	63
T	124	54	24	4	064	34	64
U	125	55	25	5	065	35	65
V	126	56	26	6	066	36	66
W	127	57	27	7	067	37	67
X	130	58	30	8	070	38	70
Y	131	59	31	9	071	39	71
Z	132	5A	32	:*	072	3A	72
[*	133	5B	33	;	073	3B	73
/	134	5C	34	<	074	3C	74
]*	135	5D	35	=	075	3D	75
!*	136	5E	36	>	076	3E	76
~*	137	5F	37	?	077	3F	77
Null	000						
Horizontal tab	011						
Line feed	012						
Vertical tab	013						
Form feed	014						
Carriage return	015						
Rubout	177						