算術運算電路(設計)

5-1 簡介

依被運算資料分:

① 二進制資料—二進位加減法器

② BCD資料— BCD加減法器

為了簡化減法的運算，計算機中採用補數的觀念，是以設計算術電路，先來回憶一下補數的觀念。

pb:0之10's complement?

$(0.0110)_2$之1's complement?

$(0.0110)_2$之2's complement?

5-2 補數的取法

補數(complement)常用於數位系統中，以簡化減法運算，在每一個基底為 r 的系統中都有兩種基本的補數型式:

<1>基底補數(r’s complement)
<2>基底減1補數((r-1)’s complement)

1. r 之補數

對於具有 n 位整數，基底為 r 的正數 N，則 N 的 r 之補數可定義為:

As $N \neq 0$ 時為 $r^n - N$

$N = 0$ 時為 0

<例題>:$(52520)_{10}$之10的補數為

$10^5 - 52520 = 47480$

<例題>:$(101100)_2$之2的補數為

$(2^6)_{10} - (101100)_2 = (1000000)_2 - (101100)_2$

$= (010100)_2$

2. r-1之補數

對於含有 n 位整數 m 位小數，基底為 r 的正數 N，其 r-1 之補數可定義為:

$r^n - n^{-m} - N$

<例題>:$(52520)_{10}$之9的補數為

$10^5 - 1 - 52520 = 47479$

<例題>:$(101100)_2$之1的補數為

$(2^6 - 1)_{10} - (101100)_2 = (010011)_2$

3. 比較

一般2補數的取法較1補數麻煩(0→1，1→0)，故可先求取1補數後，將r^{-m}加到最小有效數字上而得2的補數。

即$(r^n-r^{-m}-N)+r^{-m}=r^n-N$

<例題>$(0.0110)_2$的2補數為

$$(\underbrace{0.1001}_{1's}+\underbrace{0.0001}_{r^{-m}})_2=(0.1010)_2$$

5-3 補數的減法

1. (r-1)之補數的減法
   (subtraction with (r-1)'s complement)
   若M，N是同基底為r之兩正數，則M-N可由下法得知:
   (1)將M與N的(r-1)之補數相加。
   (2)觀察是否有末進位(end carry)
   a.有:將末進位加至最末數即為Ans。
   b.無:將1之結果再取(r-1)之補數，並冠上負號即為答案。

<例題>用1之補數法求M-N

(a) M=1010100 ， N=1000100

解: N的1之補數為0111011

```
                 1010100
               + 0111011
              ----------
(end carry) 1    0001111
                       1
              ----------
                   10000   —  Ans
```

(b) M=1000100 ， N=1010100

解: N的1之補數為0101011

$$
\begin{array}{r}
1000100 \\
+\ 0101011 \\
\hline
1101111
\end{array}
\xrightarrow[\text{1's並加負號}]{} -10000 \text{ — Ans}
$$

2. r的補數的減法(subtraction with r’s complement)

若M，N為同基底 r 的正數，則 M-N 可用下法得知:

(1)將 M 和 N 的 r 之補數相加

(2)觀察是否有末進位

有:去掉進位即為答案

無:將(1)之結果再取 r 之補數並冠上負號即得答案。

<例題>用2之補數法求M-N

(a) M=1010100 ， N=1000100

解: N的2之補數為0111100

$$
\begin{array}{r}
1010100 \\
+\ 0111100 \\
\hline
\cancel{1}\ 0010000
\end{array}
$$

(a)drop $\cancel{1}$ 0010000 → Ans為10000

(b) M=1000100 ， N=1010100

解:　　N的2之補數為0101100

　　1000100

+ 0101100

　　1110000 取2’s且冠上負號　-10000→Ans

pb:請分別用1’s及2’s complement

求解$11011_{(2)}-1010_{(2)}$=? (★不要忘了空位視為0)

5-4 數目表示法

在數位系統中，數目表示法可分成兩種:

<1>未帶號數(unsigned number):無正數與複數之區分

<2>帶號數(signed number):數位系統中常用帶號表示法

1. 二進位負數表示法

(a)符號-大小表示法(sign-magnitude representation)

(b)(符號-)1的補數表示法(sign-1’s complement representation)

(c)(符號-) 2的補數表示法(sign-2’s complement representation)

<例題>4位元有號數1011於三種表示法中各代表何值

(a) sign-magnitude representation

$$\underbrace{1}_{\text{sign}}\ \underbrace{011}_{\text{magnitude}} = -3$$

(b) (sign-)1's complement representation

$$\underbrace{1}_{\text{sign}}\ \underbrace{011}_{\text{取1's complement = magnitude}} = -0100 = -4$$

(c) (sign-)2's complement representation

$$\underbrace{1}_{\text{sign}}\ \underbrace{011}_{\text{取2's complement = magnitude}} = -0101 = -5$$

2. 二進位帶號數表示範圍

	Sign-magnitude	Sign-1's	Sign-2's
0000	+0	+0	+0
0001	+1	+1	+1
0010	+2	+2	+2
0011	+3	+3	+3
0100	+4	+4	+4
0101	+5	+5	+5
0110	+6	+6	+6
0111	+7	+7	+7
1000	-0	-7	-8
1001	-1	-6	-7
1010	-2	-5	-6
1011	-3	-4	-5
1100	-4	-3	-4
1101	-5	-2	-3
1110	-6	-1	-2
1111	-7	-0	-1

故對於1個 n 位元的二進制帶號數而言，其所代表的數目範圍為:

sign-magnitude: $-(2^{n-1}-1) \sim (2^{n-1}-1)$

(sign-)1's complement: $-(2^{n-1}-1) \sim (2^{n-1}-1)$

(sign-)2's complement: $-(2^{n-1}) \sim (2^{n-1}-1)$

<補充>補數的減法:

不一定為正數

若M，N是同基底二進位帶號數，則M-N為M+(N的1's or 2's補數)

結果即為1's or 2's

表示法有號數ie Ans

<例題>若下列2進位數已經使用sign 2's的負數表示，請求其結果?

M=0110　N=1111　求M-N

解:
```
   0110
 + 0001
 ------
   0111  — Ans
```

Because: (±A)-(+B)=(±A)+(-B)

(±A)-(-B)=(±A)+(+B)

<注意>當題目為sign 2's時一定要用2's求解

當題目為sign 1's時一定要用1's求解

Pb.利用5bit的sign 2's的二進制求

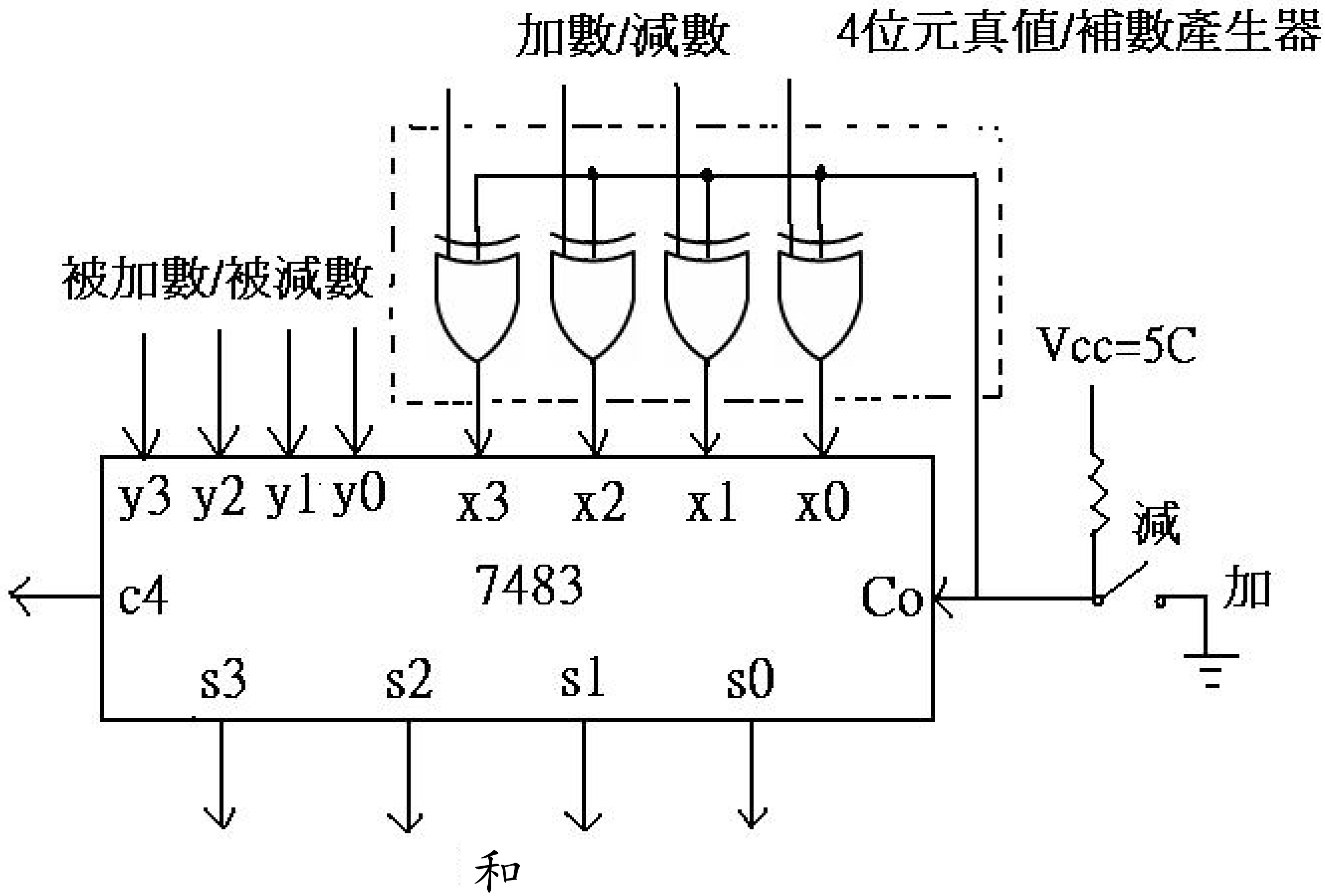
$-8_{(10)}-(-7)_{(10)} = ?$　Ans:11111

3. 比較

2補數的減法規則較1補數簡單，同時它的 +0 和 -0 相同，<u>故廣泛使用於數位系統中</u>。(故之後討論要特別指定，則表以2補數觀念為之)

<例題>利用4位元並聯加法器(7483)，設計一個4位元加減法器電路。

Sol:



5-5 算術溢位

※ 2的補數算術中，當算術運算產生的數目大於暫存器所能表示的最大數時，會產生上限溢位；當結果產生的數目小於暫存器所能表示的最小數目時，則發生下限溢位。

<例題>在8位元暫存器中的能處理數目的範圍為

$$-2^{n-1} \sim 2^{n-1}-1 = -2^{7} \sim 2^{7}-1$$
$$= -128 \sim +127$$

(a)

負	10000001	(-127)
+)	11111100	(-4)
正	01111101	←下限溢位

(b)

	01111111	(+127)
+)	00000001	(+1)
	10000000	←上限溢位

※假若上限溢位與下限溢位可能引起問題的話，就必須製作一個線路來偵測這種情形。

2補數算術於四種情況可能產生溢位:

<1>兩大正數相加 ⎫ 加法
<2>兩大負數相加 ⎭

<3>大負數減大正數 ⎫ 減法
<4>大正數減大負數 ⎭

布林式子為: (A±B)=R → 7表sign bit

$$V_{+}=\overline{A_7}\,\overline{B_7}R_7+A_7B_7\overline{R_7} \qquad \text{(加法)}$$

$$V_{-}= A_7\overline{B_7}\,\overline{R_7}+\overline{A_7}B_7R_7 \qquad \text{(減法)}$$

$$V=V_{+}\overline{S}+V_{-}S$$

5-6 BCD加/減法運算電路

1. BCD加法器:將兩BCD碼數目相加後產生BCD碼結果。
   優:清除轉換需要
   缺:較二進制線路為複雜且昂貴

在BCD碼中若以4位元的並聯加法器執行運算，則因執行是以16進制進行，而其數字則以十進制來代表，故須加以調整。

（表）二進制與BCD碼的關係

必須調整

十進制	二進制和	BCD和	十	二進制和	BCD和
	C_4 S_3 S_2 S_1 S_0	C_4 S_3 S_2 S_1 S_0		C_4 S_3 S_2 S_1 S_0	C_4 S_3 S_2 S_1 S_0
0	0 0 0 0 0	0 0 0 0 0	10	0 1 0 1 0	1 0 0 0 0
1	0 0 0 0 1	0 0 0 0 1	11	0 1 0 1 1	1 0 0 0 1
2	0 0 0 1 0	0 0 0 1 0	12	0 1 1 0 0	1 0 0 1 0
3	0 0 0 1 1	0 0 0 1 1	13	0 1 1 0 1	1 0 0 1 1
4	0 0 1 0 0	0 0 1 0 0	14	0 1 1 1 0	1 0 1 0 0
5	0 0 1 0 1	0 0 1 0 1	15	0 1 1 1 1	1 0 1 0 1
6	0 0 1 1 0	0 0 1 1 0	16	1 0 0 0 0	1 0 1 1 0
7	0 0 1 1 1	0 0 1 1 1	17	1 0 0 0 1	1 0 1 1 1
8	0 1 0 0 0	0 1 0 0 0	18	1 0 0 1 0	1 1 0 0 0
9	0 1 0 0 1	0 1 0 0 1	19	1 0 0 1 1	1 1 0 0 1

當二進制和為下列結果時，必須做十進制調整 (decimal adjust)

1.當C_4為1時

OR 2.當和$S_3S_2S_1S_0$大於1001(9)時。

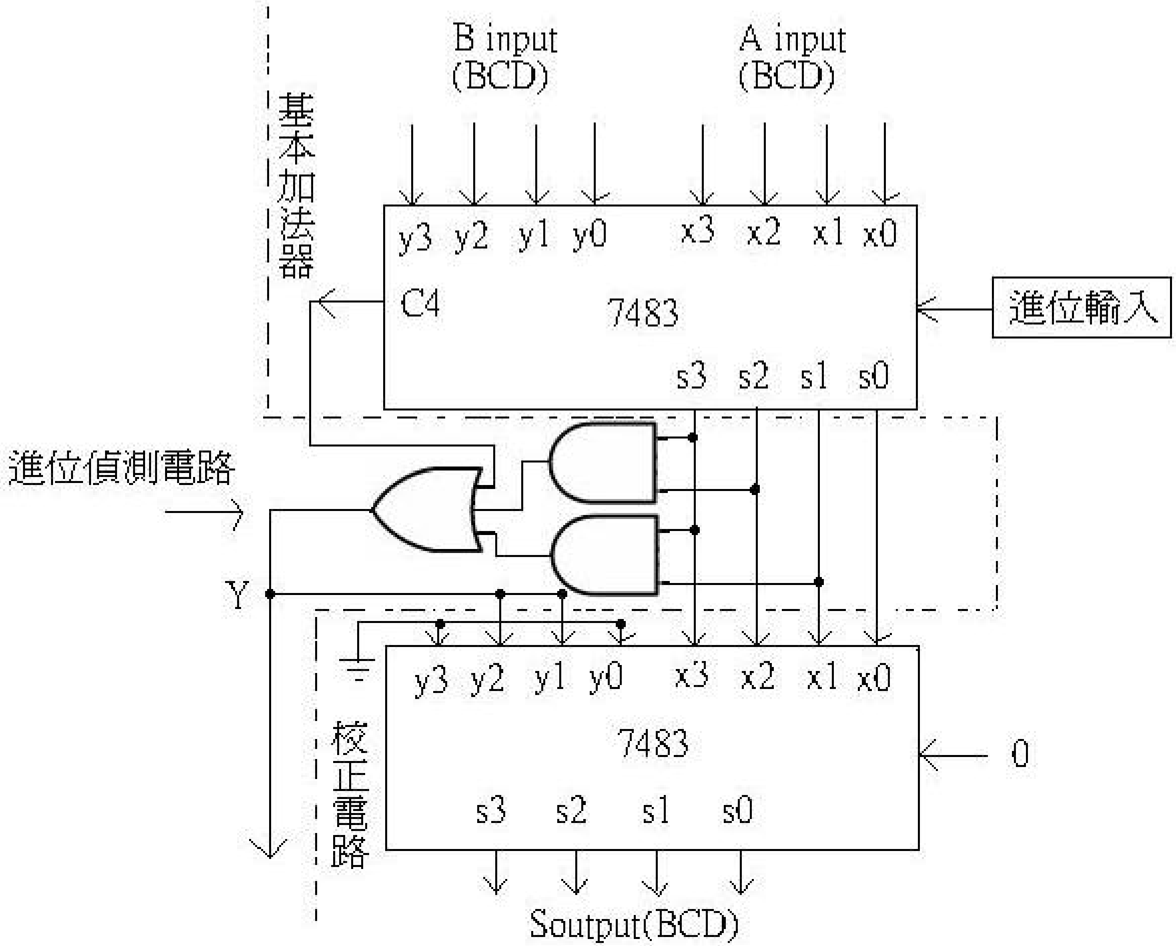
若設Y為十進制調整電路的致能控制，即

當Y為1時，電路將6(0110)加到二進制和$S_3S_2S_1S_0$上

Y為0時，加上0(0000) 則

$$Y=C_4+\underbrace{S_3S_2+S_3S_1}_{\text{可由卡諾圖化簡}} \quad \text{or} \quad Y=C_4+S_3(S_2+S_1)$$

※Y也即是BCD和的進位輸出



4位元(一位數)BCD加法器

☆BCD加法器可以串接起來，以做多位數的加法。

只要將某一級的C_{output}加到下一級C_{input}即可，最低位級不會有C_{input}，故得其接地。

2. BCD減法器

和二進制減法一樣，BCD的減法運算，通常也是將減數取10補數後加到被減數而完成的。然而，一個數的10補數可以由該數的9補數加1得到，故只需設計一個9補數產生器即可。

介紹三種求BCD碼9進補數的方法:

<方法一>先對BCD碼之各數先做補數，然後加上二進位數1010(即10進位的10)以作為修正，結果便是該BCD碼的9進補數值。

此一方法之數學原理是15-N+10=9-N+16，其中15-N便是對BCD碼各數先作補數之意。而由於16在四位元表示法中已被移出，所以上式就變成9-N，即9進補數之意。

<例題>求0111之BCD碼9進補數

解:15-N=1111-0111=1000

$1000+1010=\boxed{1}0010$ 移出

<方法二>先在BCD碼上加入二進位數0110(10進位之6)，然後對其各數先做補數，所得結果便是原來BCD碼的9進補數值。

數學原理:15-(N+6)=9-N

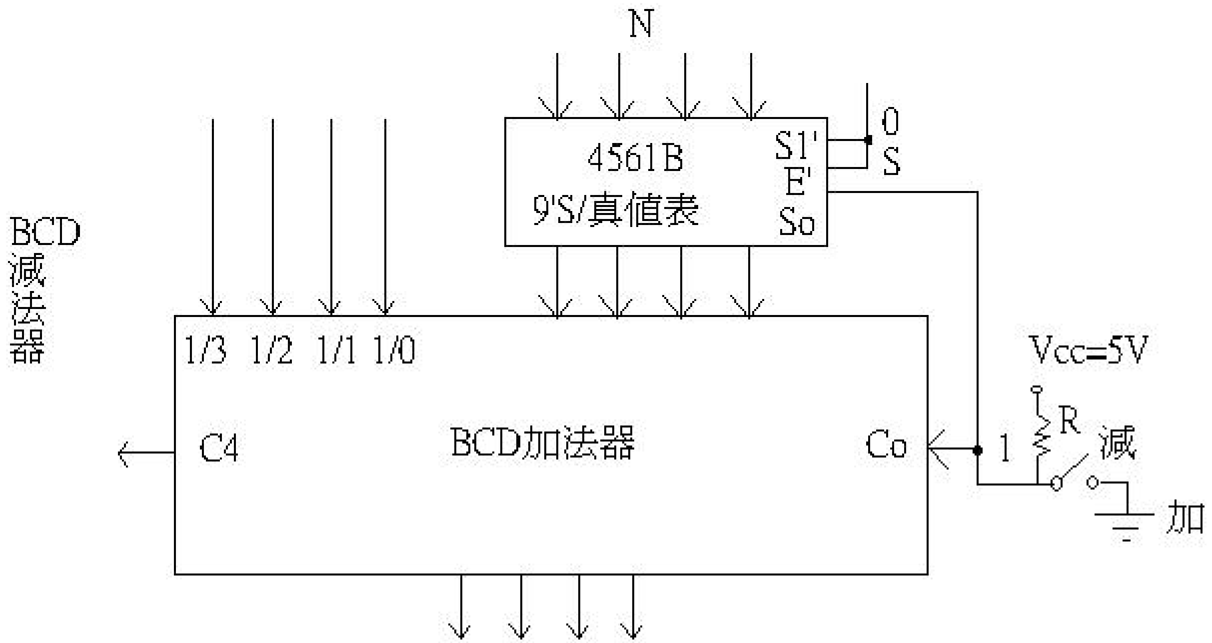
<例題>求BCD碼0111之9進補數

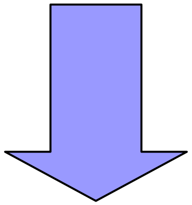
解:0111+0110=1101　　N+6

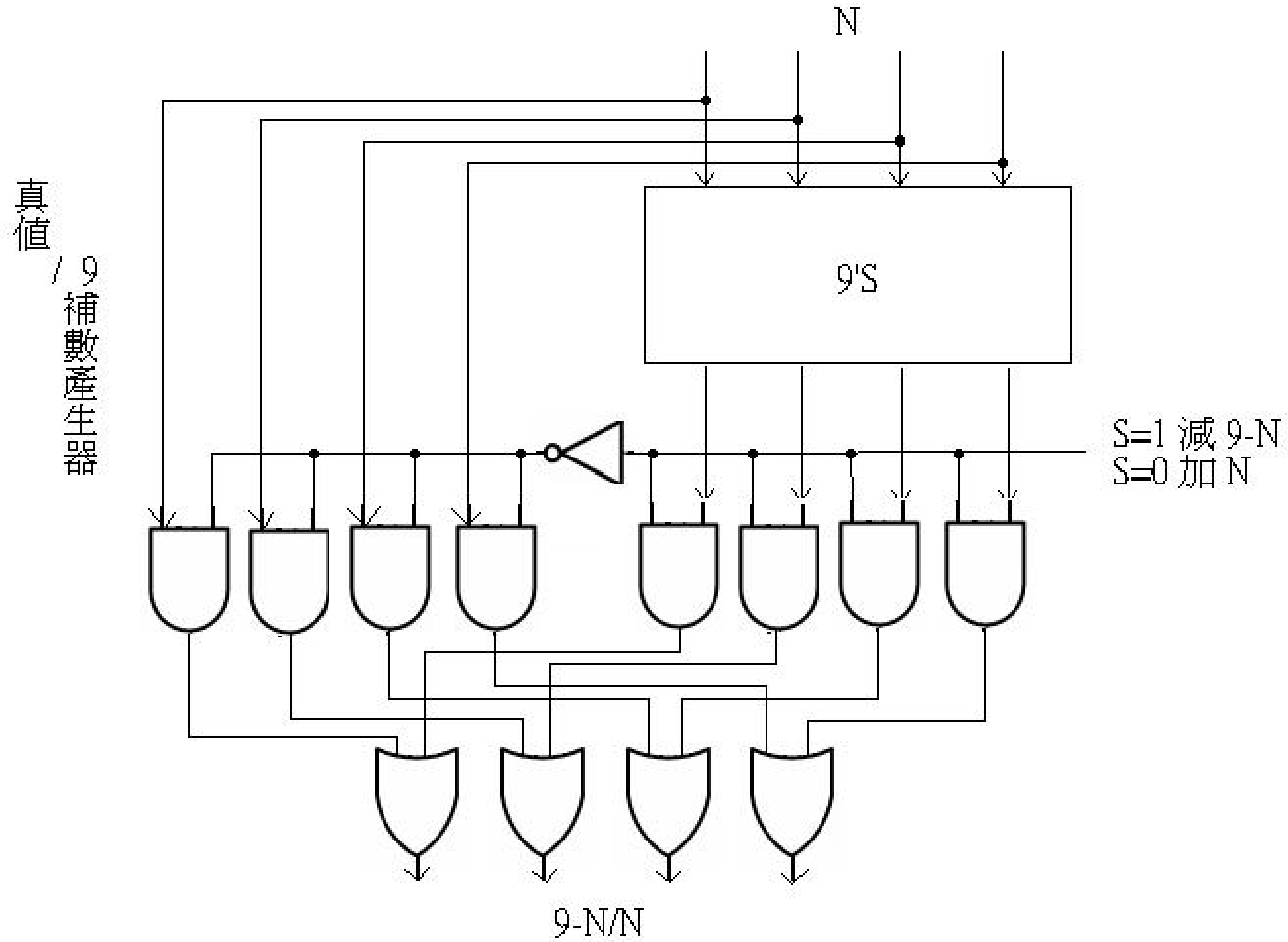
1111-1101=0010　　15-(N+6)

<方法三>直接以組合電路接出BCD數字的9進補數。

3. BCD加/減法器


N
4561B
9'S/真值表
S1'
E'
So
0
S
BCD
減
法
器
1/3
1/2
1/1
1/0
C4
BCD加法器
Co
1
Vcc=5V
R
減
加





☆4561B即為 1 真值/9補數產生器之SSI電路。

pb. 利用方法三設計一真值/9補數產生器電路。

組合電路設計

真值表

BCD碼					9補數			
S	A_3	A_2	A_1	A_0	Z_3	Z_2	Z_1	Z_0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1
		｜				｜		
0	1	0	0	1	1	0	0	1
1	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	1
		｜				｜		
1	1	0	0	1	0	0	0	0