

算數運算 (除法二)

經過上一篇文章的解說，本文先以簡單的 8 bits 運算來推導除法的運作，我們先來看看下面這個程式範例，這段程式的動作為 131 除以 7 的 8 bits 運算。

```

DIV_1BYTE
;
MOV    04H,#131    ;被除數暫存器 (83H)
MOV    05H,#0      ;清空運算暫存器
MOV    06H,#7      ;除數暫存器 (07H)
;
MOV    R3,#8       ;8 bits 運算
CLR    C           ;清除 CY 旗標
;
$1 MOV    R1,#04H   ;被除數暫存器的位址
MOV    R2,#2       ;運算時需搬移的 Byte 總數
                        ;(被除數 + 運算暫存)
;
$2 MOV    A,@R1     ;將被除數的最高位 bit
RLC    A           ;搬至運算暫數存區
MOV    @R1,A
INC    R1
DJNZ  R2,$2
;
MOV    R0,#05H     ;比較被除數的最高位 bit
MOV    R1,#06H     ;與除數的大小
MOV    A,@R0
MOV    B,@R1
CJNE  A,B,$N
;
$N JC    $3
MOV    A,@R0       ;如果較大
MOV    B,@R1       ;則減去除數
SUBB  A,B         ;並以 CY 旗標記錄
MOV    @R0,A
;
$3 CPL    C        ;如果較小
DJNE  R3,$1       ;則再將下一個 bit 搬進來判斷
;
MOV    R1,#04H     ;全部比較完之後
MOV    A,@R1       ;要將最後一個記錄在 CY 的動作
RLC    A           ;搬進被除數的暫存器才算完成
MOV    @R1,A
;
RET

```

接下來是每一步動作的推導：

第一與第二位元的動作	R6	R5	R4	C	R3	R2	R1	R0
MOV R4,#83H MOV R5,#00H MOV R6,#07H MOV R3,#8 CLR C	07H	00H	83H	0	8			
\$1 MOV R1,#04H MOV R2,#2	07H	00H	83H	0	8	2	04H	
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	00H	06H	1	8	2	04H	
INC R1 DJNZ R2,\$2	07H	00H	06H	1	8	1	05H	
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	01H	06H	0	8	1	05H	
INC R1 DJNZ R2,\$2	07H	01H	06H	0	8	0	06H	
MOV R0,#05H MOV R1,#06H	07H	01H	06H	0	8	0	06H	05H
MOV A,@R0 MOV B,@R1 CJNE A,B,\$N	07H	01H	06H	1	8	0	06H	05H
\$N JC \$3 \$3 CPL C DJNZ R3,\$1	07H	01H	06H	0	7	0	06H	05H
\$1 MOV R1,#04H MOV R2,#2	07H	01H	06H	0	7	2	04H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	01H	0CH	0	7	2	04H	05H
INC R1 DJNZ R2,\$2	07H	01H	0CH	0	7	1	05H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	02H	0CH	0	7	1	05H	05H
INC R1 DJNZ R2,\$2	07H	02H	0CH	0	7	0	06H	05H
MOV R0,#05H MOV R1,#06H	07H	02H	0CH	0	7	0	06H	05H
MOV A,@R0 MOV B,@R1 CJNE A,B,\$N	07H	02H	0CH	1	7	0	06H	05H
\$N JC \$3 \$3 CPL C DJNZ R3,\$1	07H	02H	0CH	0	6	0	06H	05H

接續上頁，當被除數所搬進的位數不足讓除數來減，程式會自動將 CY 旗標清除，反之若被除數的所搬進的值大於除數（本頁灰色），便進行一次減法，並利用 CY 旗標記錄下來。

第三與第四位元的動作	R6	R5	R4	C	R3	R2	R1	R0
\$1 MOV R1,#04H MOV R2,#2	07H	02H	0CH	0	6	2	04H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	02H	18H	0	6	2	04H	05H
INC R1 DJNZ R2,\$2	07H	02H	18H	0	6	1	05H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	04H	18H	0	6	1	05H	05H
INC R1 DJNZ R2,\$2	07H	04H	18H	0	6	0	06H	05H
MOV R0,#05H MOV R1,#06H	07H	04H	18H	0	6	0	06H	05H
MOV A,@R0 MOV B,@R1 CJNE A,B,\$N	07H	04H	18H	1	6	0	06H	05H
\$N JC \$3 \$3 CPL C DJNZ R3,\$1	07H	04H	18H	0	5	0	06H	05H
\$1 MOV R1,#04H MOV R2,#2	07H	04H	18H	0	5	2	04H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	04H	30H	0	5	2	04H	05H
INC R1 DJNZ R2,\$2	07H	04H	30H	0	5	1	05H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	08H	30H	0	5	1	05H	05H
INC R1 DJNZ R2,\$2	07H	08H	30H	0	5	0	06H	05H
MOV R0,#05H MOV R1,#06H	07H	08H	30H	0	5	0	06H	05H
MOV A,@R0 MOV B,@R1 CJNE A,B,\$N	07H	08H	30H	0	5	0	06H	05H
\$N JC \$3 MOV A,@R0 MOV B,@R1 SUBB A,B MOV @R0,A	07H	01H	30H	0	5	0	06H	05H
\$3 CPL C DJNZ R3,\$1	07H	01H	30H	1	4	0	06H	05H

發生減法運算後，程式會將該記錄搬進被除數的暫存器中，做為商數的暫存，如本頁第五個 bit 的動作一般。

第五與第六位元的動作	R6	R5	R4	C	R3	R2	R1	R0
\$1 MOV R1,#04H MOV R2,#2	07H	01H	30H	1	4	2	04H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	01H	61H	0	4	2	04H	05H
INC R1 DJNZ R2,\$2	07H	01H	61H	0	4	1	05H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	02H	61H	0	4	1	05H	05H
INC R1 DJNZ R2,\$2	07H	02H	61H	0	4	0	06H	05H
MOV R0,#05H MOV R1,#06H	07H	02H	61H	0	4	0	06H	05H
MOV A,@R0 MOV B,@R1 CJNE A,B,\$N	07H	02H	61H	1	4	0	06H	05H
\$N JC \$3 \$3 CPL C DJNZ R3,\$1	07H	02H	61H	0	3	0	06H	05H
\$1 MOV R1,#04H MOV R2,#2	07H	02H	61H	0	3	2	04H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	02H	C2H	0	3	2	04H	05H
INC R1 DJNZ R2,\$2	07H	02H	C2H	0	3	1	05H	05H
\$2 MOV A,@R1 RLC A MOV @R1,A	07H	04H	C2H	0	3	1	05H	05H
INC R1 DJNZ R2,\$2	07H	04H	C2H	0	3	0	06H	05H
MOV R0,#05H MOV R1,#06H	07H	04H	C2H	0	3	0	06H	05H
MOV A,@R0 MOV B,@R1 CJNE A,B,\$N	07H	04H	C2H	1	3	0	06H	05H
\$N JC \$3 \$3 CPL C DJNZ R3,\$1	07H	04H	C2H	0	2	0	06H	05H

第七與第八位元的動作			R6	R5	R4	C	R3	R2	R1	R0
\$1	MOV R1,#04H MOV R2,#2		07H	04H	C2H	0	2	2	04H	05H
\$2	MOV A,@R1 RLC A MOV @R1,A		07H	04H	84H	1	2	2	04H	05H
	INC R1 DJNZ R2,\$2		07H	04H	84H	1	2	1	05H	05H
\$2	MOV A,@R1 RLC A MOV @R1,A		07H	09H	84H	0	2	1	05H	05H
	INC R1 DJNZ R2,\$2		07H	09H	84H	0	2	0	06H	05H
	MOV R0,#05H MOV R1,#06H		07H	09H	84H	0	2	0	06H	05H
	MOV A,@R0 MOV B,@R1 CJNE A,B,\$N		07H	09H	84H	0	2	0	06H	05H
\$N	JC \$3 MOV A,@R0 MOV B,@R1 SUBB A,B MOV @R0,A		07H	02H	84H	0	2	0	06H	05H
\$3	CPL C DJNZ R3,\$1		07H	02H	84H	1	1	0	06H	05H
\$1	MOV R1,#04H MOV R2,#2		07H	02H	84H	1	1	2	04H	05H
\$2	MOV A,@R1 RLC A MOV @R1,A		07H	02H	09H	1	1	2	04H	05H
	INC R1 DJNZ R2,\$2		07H	02H	09H	1	1	1	05H	05H
\$2	MOV A,@R1 RLC A MOV @R1,A		07H	05H	09H	0	1	1	05H	05H
	INC R1 DJNZ R2,\$2		07H	05H	09H	0	1	0	06H	05H
	MOV R0,#05H MOV R1,#06H		07H	05H	09H	0	1	0	06H	05H
	MOV A,@R0 MOV B,@R1 CJNE A,B,\$N		07H	05H	09H	1	1	0	06H	05H
\$N	JC \$3		07H	05H	09H	0	0	0	06H	05H
\$3	CPL C DJNZ R3,\$1		07H	05H	09H	0	0	0	06H	05H

到此為止已經將八個 bit 的值全部都處理完畢，不過，最後一次記錄在 CY 旗標上的動作並未放進被除數的暫存器，所以我們還要加上最後一個動作。

記錄 CY 旗標的動作	R6	R5	R4	C	R3	R2	R1	R0
MOV R1,#04H MOV A,@R1 RLC A MOV @R1,A	07H	05H	12H	0	0	0	04H	05H

由推導的結果得知：商為 12H (18)，餘數為 05H (5)。

如此一來，你是否對除法運算的運作原理有了更進一步的認識了呢？下一篇文章，我們要探討在『8051 單晶片徹底研究--基礎篇』中所附的 32 bits 除法運算常式，和今天所介紹的 8 bits 運算，在使用上有哪些需要特別注意的地方。

附註：程式動作中紅色文字的部分，為動作時數值會更動的暫存器，表格中背景色的不同，用以區隔每個 bit 的完整動作。