

算數運算 (乘法二)

上一篇文章裡，我們提到了一個 Byte 的相乘，有一些忠實的讀者來函詢問這個動作到底是怎樣運作的呢？又為什麼 CY 需要旋轉九次呢？筆者以下面的表格來介紹，相信一定可以讓您撥雲見日，徹底瞭解其運作的方法。

bit0			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
	MOV	R4,#03H	--	--	0000 0011	-	--	--	--
	MOV	R5,#00H	--	0000 0000	0000 0011	-	--	--	--
	MOV	R6,#10H	0001 0000	0000 0000	0000 0011	-	--	--	--
	MOV	R3,#9	0001 0000	0000 0000	0000 0011	-	9	--	--
	CLR	C	0001 0000	0000 0000	0000 0011	0	9	--	--
\$1	MOV	R1,#05H	0001 0000	0000 0000	0000 0011	0	9	--	05H
	MOV	R2,#02H	0001 0000	0000 0000	0000 0011	0	9	2	05H
\$2	MOV	A,@R1							
	RRC	A	0001 0000	0000 0000	0000 0011	0	9	2	05H
	MOV	@R1,A							
	DEC	R1	0001 0000	0000 0000	0000 0011	0	9	2	04H
	DJNZ	R2,\$2	0001 0000	0000 0000	0000 0011	0	9	1	04H
第二次進入\$2									
\$2	MOV	A,@R1							
	RRC	A	0001 0000	0000 0000	0000 0001	1	9	1	04H
	MOV	@R1,A							
	DEC	R1	0001 0000	0000 0000	0000 0001	1	9	1	03H
	DJNZ	R2,\$2	0001 0000	0000 0000	0000 0001	1	9	0	03H
此時 R2=0，進入下一行									
	JNC	\$3	0001 0000	0000 0000	0000 0001	1	9	0	03H
因為 CY=1，進入下一行									
	CLR	C	0001 0000	0000 0000	0000 0001	0	9	0	03H
	MOV	A,R6							
	ADDC	A,R5	0001 0000	0001 0000	0000 0001	0	9	0	03H
	MOV	R5,A							
\$3	DJNZ	R3,\$1	0001 0000	0001 0000	0000 0001	0	8	0	03H

到此完成 bit0 的乘法，接下來再移到下一個位元。

bit1			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$1	MOV	R1,#05H	0001 0000	0001 0000	0000 0001	0	8	0	05H
	MOV	R2,#02H	0001 0000	0001 0000	0000 0001	0	8	2	05H
\$2	MOV	A,@R1							
	RRC	A	0001 0000	0000 1000	0000 0001	0	8	2	05H
	MOV	@R1,A							
	DEC	R1	0001 0000	0000 1000	0000 0001	0	8	2	04H
	DJNZ	R2,\$2	0001 0000	0000 1000	0000 0001	0	8	1	04H

第二次進入\$2			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 1000	0000 0000	1	8	1	04H
	DEC	R1	0001 0000	0000 1000	0000 0000	1	8	1	03H
	DJNZ	R2,\$2	0001 0000	0000 1000	0000 0000	1	8	0	03H
此時 R2=0, 進入下一行									
	JNC	\$3	0001 0000	0000 1000	0000 0000	1	8	0	03H
因為 CY=1, 進入下一行									
	CLR	C	0001 0000	0000 1000	0000 0000	0	8	0	03H
	MOV ADDC MOV	A,R6 A,R5 R5,A	0001 0000	0001 1000	0000 0000	0	8	0	03H
\$3	DJNZ	R3,\$1	0001 0000	0001 1000	0000 0000	0	7	0	03H

到此完成 bit1 的乘法，您可以特別注意看看 R5 及 CY 的變化，接下來是 bit2 的乘法。

bit2			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$1	MOV	R1,#05H	0001 0000	0001 1000	0000 0000	0	7	0	05H
	MOV	R2,#02H	0001 0000	0001 1000	0000 0000	0	7	2	05H
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 1100	0000 0000	0	7	2	05H
	DEC	R1	0001 0000	0000 1100	0000 0000	0	7	2	04H
	DJNZ	R2,\$2	0001 0000	0000 1100	0000 0000	0	7	1	04H
第二次進入\$2									
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 1100	0000 0000	0	7	1	04H
	DEC	R1	0001 0000	0000 1100	0000 0000	0	7	1	03H
	DJNZ	R2,\$2	0001 0000	0000 1100	0000 0000	0	7	0	03H
此時 R2=0, 進入下一行									
	JNC	\$3	0001 0000	0000 1100	0000 0000	0	7	0	03H
因為 CY=1, 進入\$3									
\$3	DJNZ	R3,\$1	0001 0000	0000 1100	0000 0000	0	6	0	03H

到此完成 bit2 的乘法，由於 CY=0，所以並沒有加法的動作，在這裡我們可以看到一個很重要的關鍵，就是每一個位數進位的時候，加法的動作就在該位數執行，這正是乘法的精髓所在，再往下看就會更清楚囉。

bit3			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$1	MOV	R1,#05H	0001 0000	0000 1100	0000 0000	0	6	2	05H
	MOV	R2,#02H	0001 0000	0000 1100	0000 0000	0	6	2	05H
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 0110	0000 0000	0	6	2	05H
	DEC	R1	0001 0000	0000 0110	0000 0000	0	6	2	04H
	DJNZ	R2,\$2	0001 0000	0000 0110	0000 0000	0	6	1	04H

第二次進入\$2			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 0110	0000 0000	0	6	1	04H
	DEC	R1	0001 0000	0000 0110	0000 0000	0	6	1	03H
	DJNZ	R2,\$2	0001 0000	0000 0110	0000 0000	0	6	0	03H
此時 R2=0, 進入下一行									
	JNC	\$3	0001 0000	0000 0110	0000 0000	0	6	0	03H
因為 CY=1, 進入\$3									
\$3	DJNZ	R3,\$1	0001 0000	0000 0110	0000 0000	0	5	0	03H

到此完成 bit3 的乘法，由於 CY=0，所以還是沒有加法的動作。

bit4			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$1	MOV	R1,#05H	0001 0000	0000 0110	0000 0000	0	5	0	05H
	MOV	R2,#02H	0001 0000	0000 0110	0000 0000	0	5	2	05H
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 0011	0000 0000	0	5	2	05H
	DEC	R1	0001 0000	0000 0011	0000 0000	0	5	2	04H
	DJNZ	R2,\$2	0001 0000	0000 0011	0000 0000	0	5	1	04H
第二次進入\$2			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 0011	0000 0000	0	5	1	04H
	DEC	R1	0001 0000	0000 0011	0000 0000	0	5	1	03H
	DJNZ	R2,\$2	0001 0000	0000 0011	0000 0000	0	5	0	03H
此時 R2=0, 進入下一行									
	JNC	\$3	0001 0000	0000 0011	0000 0000	0	5	0	03H
因為 CY=1, 進入\$3									
\$3	DJNZ	R3,\$1	0001 0000	0000 0011	0000 0000	0	4	0	03H

bit4 的乘法同 bit3 一樣，bit5 的乘法就比較不一樣囉。

bit5			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$1	MOV	R1,#05H	0001 0000	0000 0011	0000 0000	0	4	0	05H
	MOV	R2,#02H	0001 0000	0000 0011	0000 0000	0	4	2	05H
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 0001	0000 0000	1	4	2	05H
	DEC	R1	0001 0000	0000 0001	0000 0000	1	4	2	04H
	DJNZ	R2,\$2	0001 0000	0000 0001	0000 0000	1	4	1	04H
第二次進入\$2									
\$2	MOV RRC MOV	A,@R1 A @R1,A	0001 0000	0000 0001	1000 0000	0	4	1	04H
	DEC	R1	0001 0000	0000 0001	1000 0000	0	4	1	03H
	DJNZ	R2,\$2	0001 0000	0000 0001	1000 0000	0	4	0	03H

此時 R2=0，進入下一行			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
	JNC	\$3	0001 0000	0000 0001	1000 0000	0	4	0	03H
因為 CY=1，進入\$3									
\$3	DJNZ	R3,\$1	0001 0000	0000 0001	1000 0000	0	3	0	03H

看到了嗎？R4 已經將運算值轉回來了，請特別注意到第二次進入\$2 的部份，因為用了 RRC 做旋轉的動作，因此自然而然會將高位元組（R5）的溢位值，再搬到低位元組（R4）的最前端，這樣就可以完整地將 16 個連續 bits 做旋轉的動作而不會遺失任何一個 bit 的資料，這個部份是非常重要的，一定要實際用不同的例子去旋轉看看，接著就輪到 bit6 了。

bit6			R6(06H)	R5(05H)	R4(04H)	CY	R3	R2	R1
\$1	MOV	R1,#05H	0001 0000	0000 0001	1000 0000	0	3	0	05H
	MOV	R2,#02H	0001 0000	0000 0001	1000 0000	0	3	2	05H
\$2	MOV	A,@R1							
	RRC	A	0001 0000	0000 0000	1000 0000	1	3	2	05H
	MOV	@R1,A							
	DEC	R1	0001 0000	0000 0000	1000 0000	1	3	2	04H
	DJNZ	R2,\$2	0001 0000	0000 0000	1000 0000	1	3	1	04H
第二次進入\$2									
\$2	MOV	A,@R1							
	RRC	A	0001 0000	0000 0000	1100 0000	0	3	1	04H
	MOV	@R1,A							
	DEC	R1	0001 0000	0000 0000	1100 0000	0	3	1	03H
	DJNZ	R2,\$2	0001 0000	0000 0000	1100 0000	0	3	0	03H
此時 R2=0，進入下一行									
	JNC	\$3	0001 0000	0000 0000	1100 0000	0	3	0	03H
因為 CY=1，進入\$3									
\$3	DJNZ	R3,\$1	0001 0000	0000 0000	1100 0000	0	2	0	03H

bit6 和 bit5 的運作模式相同，在此就不多做介紹，剩下一個 bit 需要運算，可是 R3 卻還有兩次需要旋轉，這是什麼原因呢？回想一下，當我們一開始旋轉的時候，位於 R5 最前端的值是 R4 最後一個 bit 的值還是 CY 的預設值呢？

答案很明顯，第一個旋轉進來的是 CY 的預設值，因此我們必須再多旋轉一次，將 CY 的值轉回原位才算大功告成，因此 R3 所設定的值是 9 而不是 8。而再旋轉兩次的結果，R4 的值正好是 0011 0000，也就是 30H。