

實驗一 內部 RAM 與 Flash 空間定址與存取

1. 實驗內容

利用 8051 的六種定址法變更 A 暫存器的內容，並使用 Keil C51 debug mode 觀察 8051 內部暫存器的變化。

附註：本週基礎題課程為純軟體模擬，不用接電路。

2. 學習重點

- 8051 內部記憶體 (Flash 與 RAM) 配置。
- 8051 暫存器庫使用方式。
- 8051 的各種定址法使用，使用的指令包含了 MOV、MOVC、SETB、CLR。

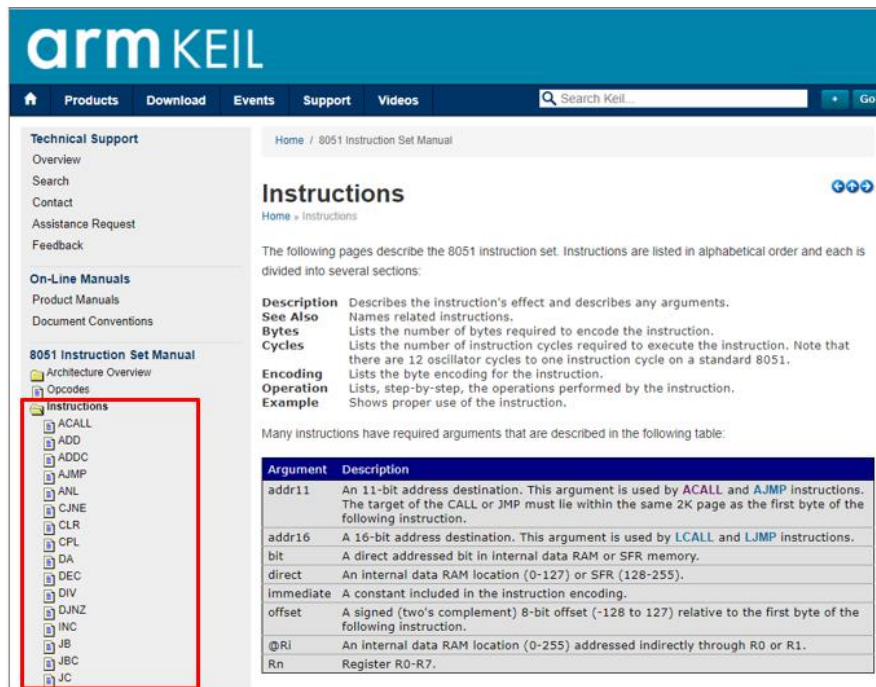
組合語言格式

- 程式經由組譯器組譯後，運算碼及運算元會產生機械碼
- 標籤相當於程式記憶體的位置

	Label 標籤	Mnemonic 運算碼	Operand 運算元	Comment 註解
1		ORG	0	
2	LOOP:	MOV	P1, A	
3		CALL	DELAY	;delay 0.5 sec
4		RL	A	
5		JMP	LOOP	
6	DELAY:	...		

指令 Instruction

- 組合語言指令集查詢 8051 Instruction Set Manual – Keil

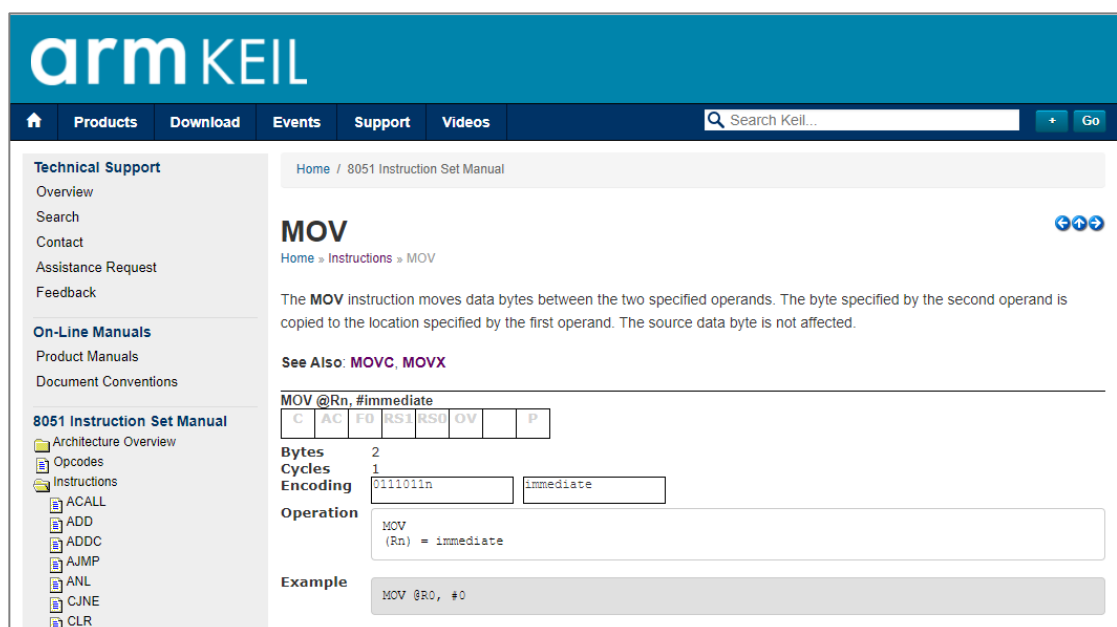


指令 Instruction-MOV

- MOV destination, source; copy source to destination

範例一：MOV A, #55H ; A 暫存器存入 55H 的數值

範例二：MOV R0, A ; 複製 A 暫存器的內容到 R0



- The **MOV** instruction moves data bytes between the two specified operands.
- The byte specified by the second operand is copied to the location specified by the first operand. The source data byte is not affected.

3. 資料記憶體 (RAM)

8051 有 128Bytes (位址 00H ~ 7FH) 的內部 RAM 記憶體，另外還有可以控制 8051 許多行為模式的 SFR 記憶體 (位址 80H ~ FFH)，memory map 如下表。

7F	Scratch Pad Area							
...								
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Register Bank 3							
...								
18	Register Bank 2							
...								
10								
0F	Register Bank 1							
...								
08	Register Bank 0							
07								
...								
00								

...										
F0	F7	F6	F5	F4	F3	F2	F1	F0	B	
...										
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC	
...										
D0	D7	D6	D5	D4	D3	D2	D1	D0	PSW	
...										
B8	BF	BE	BD	BC	BB	BA	B9	B8	IP	
...										
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3	
...										
A8	AF	AE	AD	AC	AB	AA	A9	A8	IE	
...										
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2	
...										
99									SBUF	
98	9F	9E	9D	9C	9B	9A	99	98	SCON	
...										
90	97	96	95	94	93	92	91	90	P1	
...										
8D									TH1	
8C									TH0	
8B									TL1	
8A									TL0	
89									TMOD	
88	8F	8E	8D	8C	8B	8A	89	88	TCON	
87									PCON	
...										
83									DPH	
82									DPL	
81									SP	
80	87	86	85	84	83	82	81	80	P0	

圖 1-1、8051 內部 RAM (左) 及 SFR (右)

在這 00 ~ FF 的空間中，主要可以分為四個區域：可位元定址區 Bit Addressable Area、一般資料存放區 / 堆疊區 Scratch Pad Area、特殊功能暫存器區 Special Function Register (SFR) 以及暫存器庫區 Register Bank。

可位元定址區 Bit Addressable Area

該區域位於 8051 內部 RAM 的 20H ~ 2FH 位址，空間大小僅有 2bytes，可以使用 SETB 與 CLR 指令來控制某個位址 byte 中的一個 bit 數值，也可以使用 MOV 指令來改變整個 bytes 的數值，在 Keil C51 中，可以利用以下幾種指令形式修改此記憶體區域的值：

- **MOV 20H, #01H**
- **SETB 00H**
- **SETB 20H.0**

7F	Scratch Pad Area							
...								
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Register Bank 3							
...								
18								
17	Register Bank 2							
...								
10								
0F	Register Bank 1							
...								
08								
07	Register Bank 0							
...								
00								

圖 1-2、可位元定址區

一般資料存放區/堆疊區 Scratch Pad Area

該區域位於 8051 內部 RAM 的 30H ~ 7FH 位址，空間大小有 80bytes，不可使用 SETB、CLR 指令，僅能使用 MOV 指令一次修改一整個 byte 的值，在 Keil C51 中，可以利用以下指令形式修改此記憶體的值：

- `MOV 30H, #01H`

7F	Scratch Pad Area							
...								
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Register Bank 3							
...								
18								
17	Register Bank 2							
...								
10								
0F	Register Bank 1							
...								
08								
07	Register Bank 0							
...								
00								

圖 1-3、一般資料存放區/堆疊區

特殊功能暫存器區 Special Function Register (SFR)

該區域位於 8051 內部 RAM 的 80H ~ FFH 位址，部分空間無法使用，此處有許多會影響 8051 行為模式的暫存器，另外，**僅可使用直接定址法讀取與修改此位址的值**，在 Keil C51 中，可以利用以下幾種指令修改此記憶體的值：

- **MOV A, #01H**
MOV 0E0H, #01H
SETB 0E0H
SETB 0E0H.0

附註：在 Keil C51 的 16 進制中，如果要定的位址或數值最高位數>=A，則要在前方加上 0。

...									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
...									
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
...									
D0	D7	D6	D5	D4	D3	D2	D1	D0	PSW
...									
B8	BF	BE	BD	BC	BB	BA	B9	B8	IP
...									
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
...									
A8	AF	AE	AD	AC	AB	AA	A9	A8	IE
...									
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
...									
99									SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
...									
90	97	96	95	94	93	92	91	90	P1
...									
8D									TH1
8C									TH0
8B									TL1
8A									TL0
89									TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87									PCON
...									
83									DPH
82									DPL
81									SP
80	87	86	85	84	83	82	81	80	P0

圖 1-4、特殊功能暫存器區 (SFR)

本課程中，有些同學拿到的晶片為 8052，有 256bytes 的內部記憶體空間，比 8051 多出來的 128bytes 空間也是在 80H ~ FFH 的位址，但實際上這個區域與

SFR 是不同的記憶體空間，要使用這個區域時必須使用間接定址法來存取資料，而 8051 沒有這個內部 RAM 空間所以無法使用。

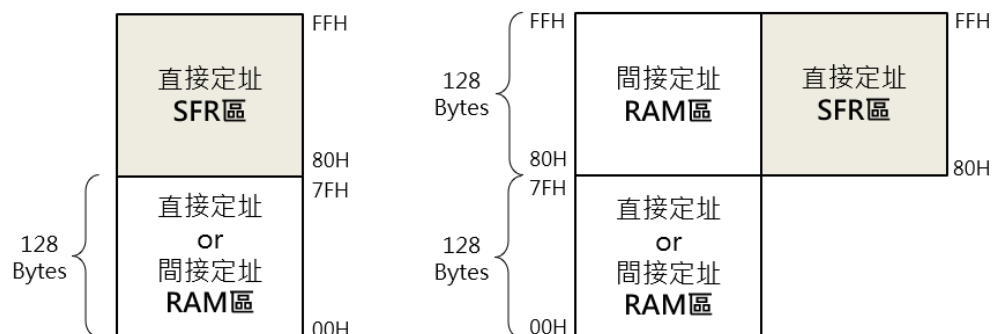


圖 1-5、8051 與 8052 記憶體空間比較

附註：當 8051 剛上電或者是觸發 Reset 時，SFR 的資料將會被初始化成下表中的樣子，除此之外的記憶體空間將無法保證其數值為何。

表 1-1、Contents of the SFRs after reset

Register	Value in Binary
*ACC	00000000
*B	00000000
*PSW	00000000
SP	00000111
DPTR	
DPH	00000000
DPL	00000000
*P0	11111111
*P1	11111111
*P2	11111111
*P3	11111111
*IP	8051 XXX00000
	8052 XX000000
*IE	8051 0XX00000
	8052 0X000000
TMOD	00000000
*TCON	00000000
*T2CON	00000000
TH0	00000000
TL0	00000000
TH1	00000000
TL1	00000000
+TH2	00000000
+TL2	00000000
+RCAP2H	00000000
+RCAP2L	00000000
*SCON	00000000
SBUF	Indeterminate
PCON	HMOS 0XXXXXXX
	CMOS 0XXX0000
X = Undefined	
* = Bitaddressable	
+ = 8052 Only	

暫存器庫區 Register Bank

該區域位於 8051 內部 RAM 的 00H ~ 1FH 位址，總共有四組暫存器庫，每個暫存器庫各包含了 R0、R1...R7 等 8 個暫存器，要使用哪一組的 R0 ~ R7 暫存器可使用 SFR 中 Program Status Word (PSW) 暫存器中的 RS1 和 RS0 做切換。

7F	Scratch Pad Area							
...								
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Register Bank 3							
...								
18	Register Bank 2							
17								
...								
10	Register Bank 1							
0F								
...								
08	Register Bank 0							
07								
...								
00								

...									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
...									
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
...									
D0	D7	D6	D5	D4	D3	D2	D1	D0	PSW
...									
B8	BF	BE	BD	BC	BB	BA	B9	B8	IP
...									
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
...									
A8	AF	AE	AD	AC	AB	AA	A9	A8	IE
...									
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
...									
99									SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
...									
90	97	96	95	94	93	92	91	90	P1
...									
8D									TH1
8C									TH0
8B									TL1
8A									TL0
89									TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87									PCON
...									
83									DPH
82									DPL
81									SP
80	87	86	85	84	83	82	81	80	P0

圖 1-6、暫存器庫 (左) 與 SFR 中的控制暫存器庫的位元 (右)

根據設定的 RS0 以及 RS1，R0 ~ R7 分別有不同的位址，例如當 RS0 與 RS1 皆為 0 時，將會使用暫存器庫 0，R0 的位址會在 00H，而當 RS0 為 1，RS1 為 0 時，將使用暫存器庫 1，R0 的位址會在 08H，不同設定對應的位址如下表。

PSW	7	6	5	4	3	2	1	0
	CY	AC	F0	RS1	RS0	OV		P

圖 1-7、SFR 中的 PSW 暫存器

表 1-2、設定 RS0、RS1 所代表的暫存器庫

RS1	RS0	暫存器庫	位址
0	0	RB0	0000H ~ 0007H
0	1	RB1	0008H ~ 000FH
1	0	RB2	0010H ~ 0017H
1	1	RB3	0018H ~ 001FH

8051 在上電或是 Reset 後，RS0 與 RS1 預設皆為 0，若想要使用暫存器庫 1，可以利用以下指令達成：

- **SETB** RS0
CLR RS1
- **SETB** 0D3H
CLR 0D4H
- **SETB** 0D0H.3
CLR 0D0H.4
- **MOV** PSW, #18H (假設不使用到其他暫存器)
- **MOV** 0D0H, #18H (假設不使用到其他暫存器)

附註：關於 PSW 暫存器將在第 4 章有更詳細的介紹。

4. 六種定址法與程式碼

8051 總共有六種定址法可以存取記憶體空間 (RAM 或者是 Flash 空間) 的資料，包含了直接定址法 Direct addressing、間接定址法 Indirect addressing、暫存器定址法 Register addressing、立即定址法 Immediate addressing、索引定址法 Index addressing 以及位元定址法 Bit addressing，接下來將以各個程式碼逐一做介紹。

在直接定址法、間接定址法、暫存器定址法以及立即定址法中，程式碼的前 5 行我們會先在記憶體的一些位址中放入一些數值，如下表，接下來請同學根據敘述認識這種定址法的形式，以及判斷將會有什麼數值被放入 A 暫存器。

資料記憶體 RAM	
Address	Data
FFH	?
...	?
31H	24H
30H	55H
...	?
01H	10H
00H	30H

Register bank 0 R1

Register bank 0 R0

圖 1-8、下列程式碼前 5 行在資料記憶體中的位址及其存放的數值

直接定址法 Direct addressing

MOV A, direct

直接定址法的運算元為「位址」，以 16 進位的數值表示。直接定址法的指令可以直接存取某個位址內的數值。

```
1  ORG 0           ; code start from 0
2  MOV R0, #30H    ; R0 = 30H
3  MOV R1, #10H    ; R1 = 10H
4  MOV 30H, #55H   ; address 30H = 55H
5  MOV 31H, #24H   ; address 31H = 24H
6  MOV A, 30H      ; A = ?
7  SJMP $          ; infinite loop
8  END
```

間接定址法 Indirect addressing

MOV A, @Ri

間接定址法的運算元為「暫存器」，以「@」作為前置符號。間接定址法的指令可以將存入 R0 或 R1 暫存器的數值當作位址，並且根據這個位址存取資料，使用上類似 C 語言指標的概念。

```
1  ORG 0           ; code start from 0
2  MOV R0, #30H    ; R0 = 30H
3  MOV R1, #10H    ; R1 = 10H
4  MOV 30H, #55H   ; address 30H = 55H
5  MOV 31H, #24H   ; address 31H = 24H
6  MOV A, @R0      ; A = ?
7  SJMP $          ; infinite loop
8  END
```

附註：僅有 R0 與 R1 可以使用間接定址法

立即定址法 Immediate addressing

MOV A, #immediate

立即定址法的運算元為「常數資料」，以「#」號作為前置符號。立即定址法的指令可以直接存取指定的數值。

```
1  ORG 0           ; code start from 0
2  MOV R0, #30H    ; R0 = 30H
3  MOV R1, #10H    ; R1 = 10H
4  MOV 30H, #55H   ; address 30H = 55H
5  MOV 31H, #24H   ; address 31H = 24H
6  MOV A, #31H     ; A = ?
7  SJMP $          ; infinite loop
8  END
```

常數資料表示形式分為三種，二進位、十進位，以及十六進位：

- 二進位

存取範圍為 00000000B~11111111B，數字後加「B」做為標示以表示二進位。

- 十進位

存取範圍為 0~255，十進位數字後可加上「D」，或不加任何標示。

- 十六進位

存取範圍為 00H~FFH，數字後加「H」做為標示，或字首加上「0x」做為前綴，以表示十六進位。

暫存器定址法 Register addressing

MOV A, Rn

暫存器定址法的運算元為「暫存器」，以 R0~R7 表示。暫存器定址法的指令可以直接存取某個暫存器內的數值。

```
1  ORG 0           ; code start from 0
2  MOV R0, #30H    ; R0 = 30H
3  MOV R1, #10H    ; R1 = 10H
4  MOV 30H, #55H   ; address 30H = 55H
5  MOV 31H, #24H   ; address 31H = 24H
6  MOV A, R0       ; A = ?
7  SJMP $          ; infinite loop
8  END
```

索引定址法 Index addressing

MOVC A, @A+DPTR

索引定址法的運算元同時為「索引暫存器+基底暫存器」，以「@」作為前置符號。索引定址法使用 MOVC 指令存取程式記憶體的资料，以基底暫存器 (DPTR 或 PC) 的數值，再加上索引暫存器 (A 暫存器) 的數值，所得的值即是運算元所在的位址，概念類似 C 語言中的陣列。

```
1  ORG 0           ; code start from 0
2  MOV DPTR, #TABLE ; DPTR = 'TABLE' label address
3  MOV A, #01H     ; A = 01H
4  MOVC A, @A+DPTR ; A = ?
5  SJMP $          ; infinite loop
6  TABLE: DB 01H  ; save 01H at 'TABLE' address
7             DB 02H ; save 02H at 'TABLE' label +1 address
8  END
```

附註：DB 指令代表在程式碼接下來的位址填入該數值。

程式碼在 Flash 的位置

在撰寫程式碼中使用到的 ORG 指令為假指令，也就是說這個指令是給 assembler 看的，8051 並不會執行該指令，當 assembler 看到 ORG 指令時，會根據 ORG 指令指定的位址開始填入轉換後的機械碼。

每個指令所需要的程式記憶體空間不同 (8051 為複雜指令集)，可以查詢指令集得知每個指令所佔的長度以及需要消耗的時間，藉此推得 8051 每個 Flash 空間位址儲存的資料，指令集可參考：[intel MCS@51 MICROCONTROLLER](http://www.intel.com/lit/litweb/51801.htm)

[FAMILY USER'S MANUAL p.2-21 Instruction Set](#)。在 Keil C51 中，將程式碼編譯完後可以利用 Disassembly Window 或是 Memory Windows 功能對此輕易地做觀察。接下來以一個簡單的範例來模擬一次程式碼將會被放進 Flash 的位置：

```
1  ORG 0
2  MOV R0, #30H
3  MOV A, R0
4  END
```

首先，程式碼第 1 行的 **ORG 0** 為假指令，告訴 assembler 要將程式碼從 0000H 開始填入。程式碼第 2 行需要 2 bytes 的空間，指令 hex code 為 78 (如表 1-2)，data 為程式輸入的 30H。因此位址 0000H 被 assembler 填入指令 78H，位址 0001H 被填入 data 30H (如圖 1-10)。

表 1-3、Hex code – 78

Hex Code	Number of Bytes	Mnemonic	Operands
78	2	MOV	R0, #data

程式記憶體 Flash

Address	Data
0000H	78
0001H	30
0002H	
0003H	
0004H	
0005H	

圖 1-9、Flash memory 之值及位址

程式碼第 3 行僅需要 1 byte 的空間，hex code 為 E8H (如表 1-3)。Assembler 在接下來的 0002H 位址填入 E8H。圖 1-11 即為該程式碼被燒錄進 8051 Flash 的結果。

表 1-4、Hex code - E8

Hex Code	Number of Bytes	Mnemonic	Operands
E8	1	MOV	A,R0

程式記憶體 Flash

Address	Data
0000H	78
0001H	30
0002H	E8
0003H	
0004H	
0005H	

圖 1-10、Flash memory 之值及位址

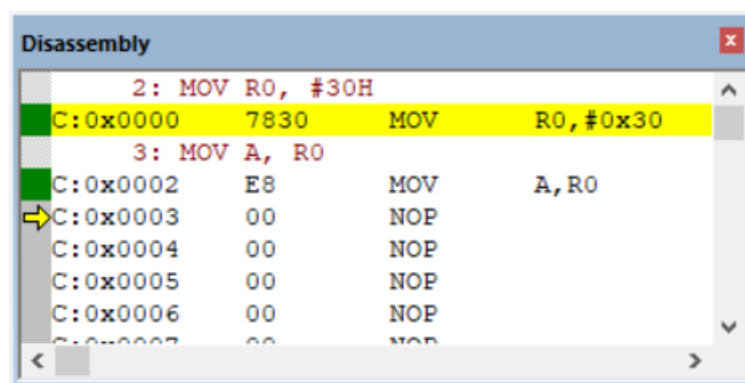


圖 1-11、Keil C51 - Disassembly Window

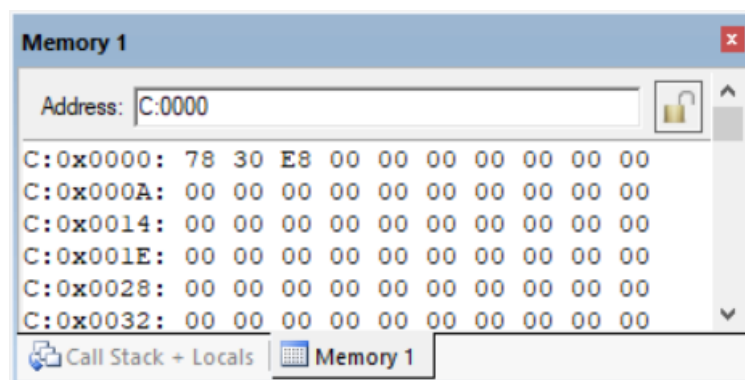


圖 1-12、Keil C51 - Memory Windows

位元定址法 Bit addressing

位元定址法的運算元為「可位元定址」的位址。在可位元定址區以及部份的 SFR 中，可以利用 SETB 或是 CLR 指令來存取某個 byte 位址中一個 bit 的數值。

```

1  ORG 0 ; code start from 0
2  MOV 20H, #00H ; address 20H = 00H
3  SETB 20H.1 ; address 20H bit 1 = 1
4  MOV A, 20H ; A = ?
5  SETB 03H ; bit address 03H = 1
6  MOV A, 20H ; A = ?
7  SJMP $ ; infinite loop
8  END

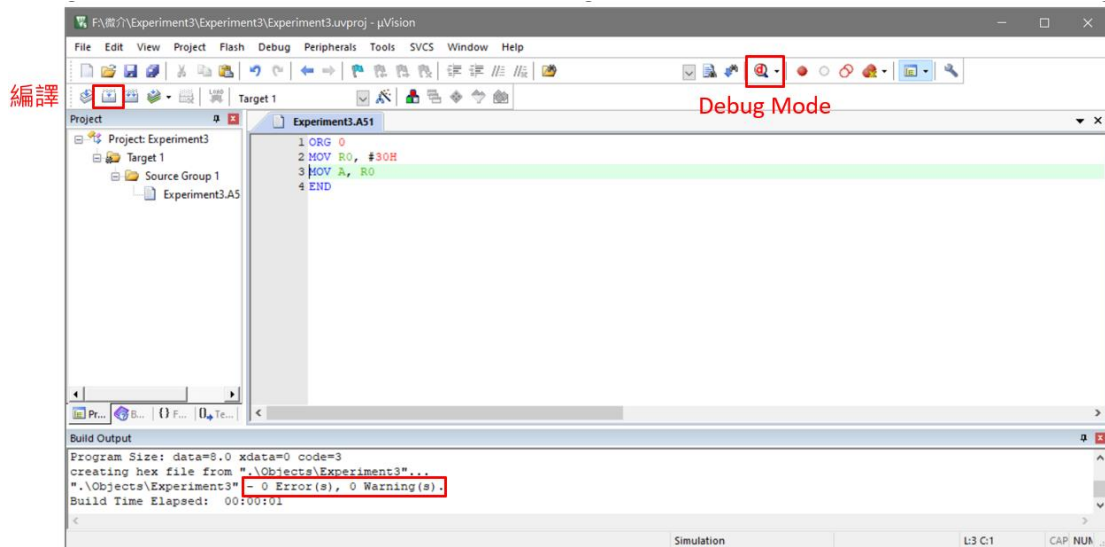
```

7F	Scratch Pad Area							
...								
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3F	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Register Bank 3							
...								
18								
17	Register Bank 2							
...								
10								
0F	Register Bank 1							
...								
08								
07	Register Bank 0							
...								
00								

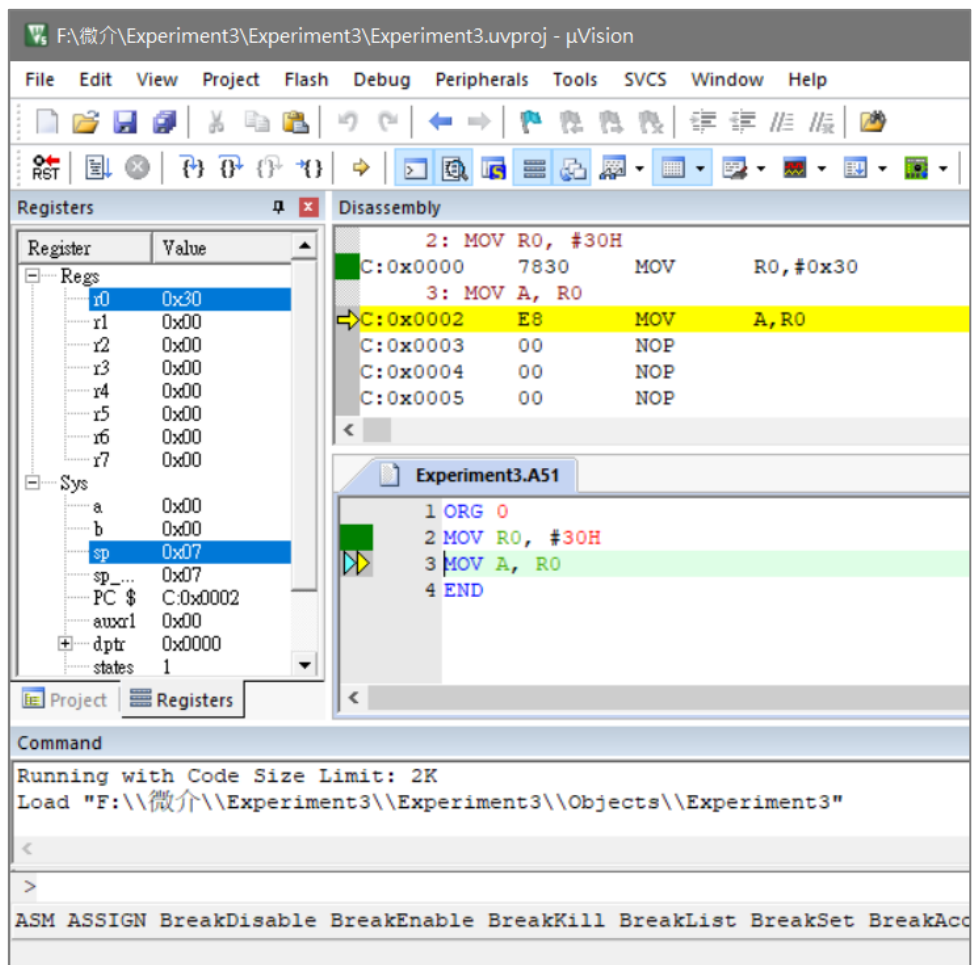
圖 1-13、Address 20H in 8051 RAM

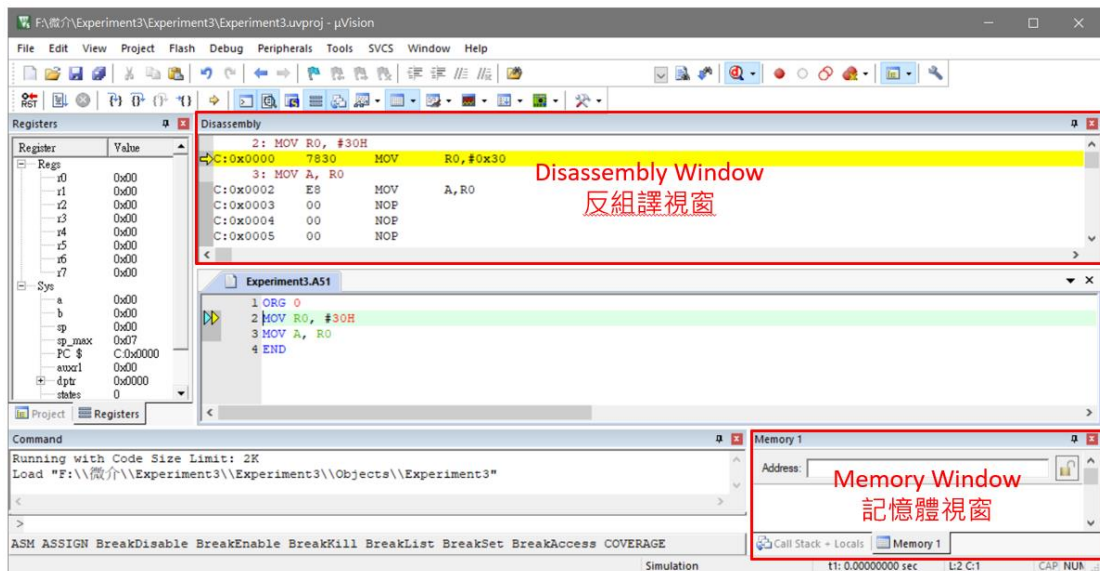
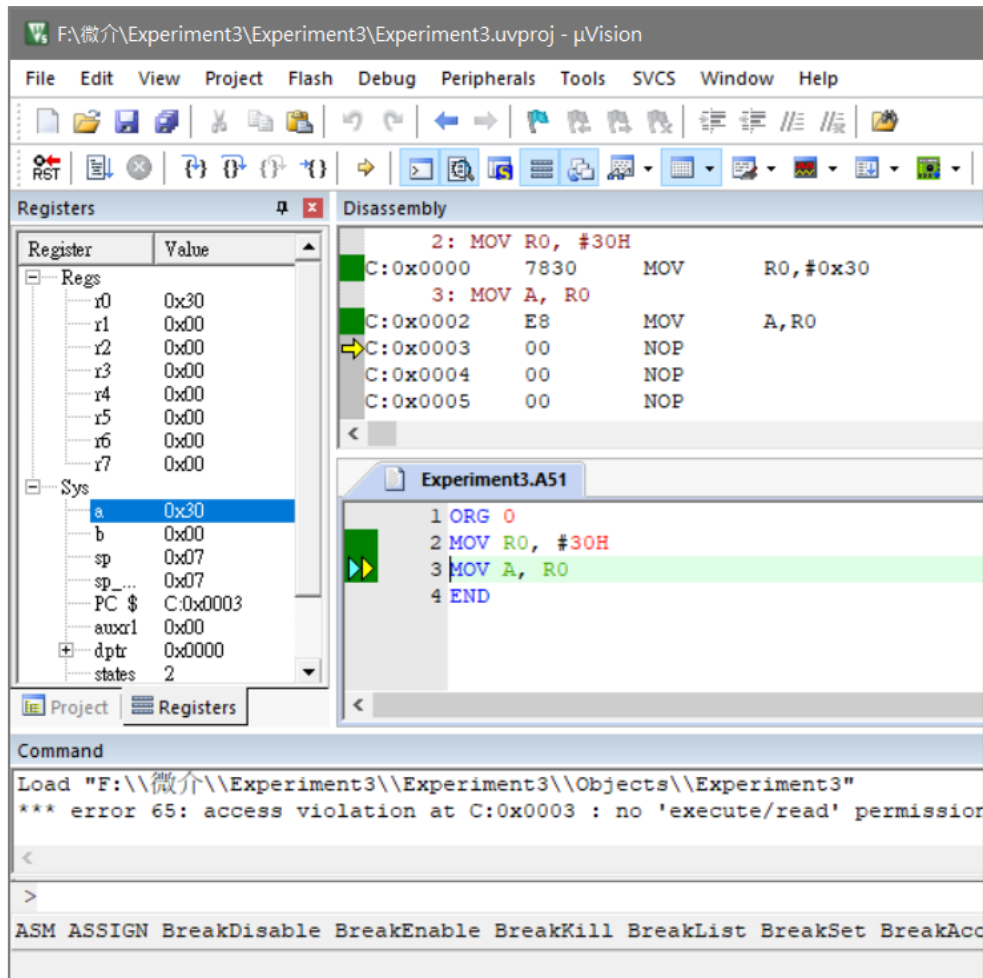
5. Keil C51 實際操作

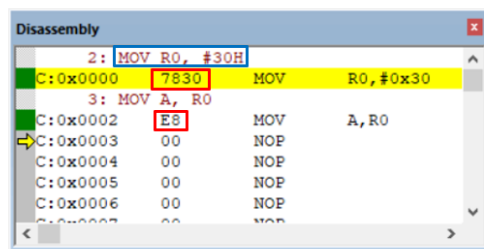
- Keil C51 – Debug Mode



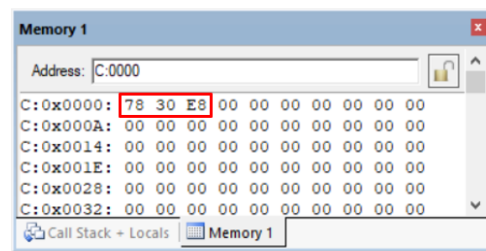
- Keil C51 – 單步執行







C：程式記憶體



D：資料記憶體

