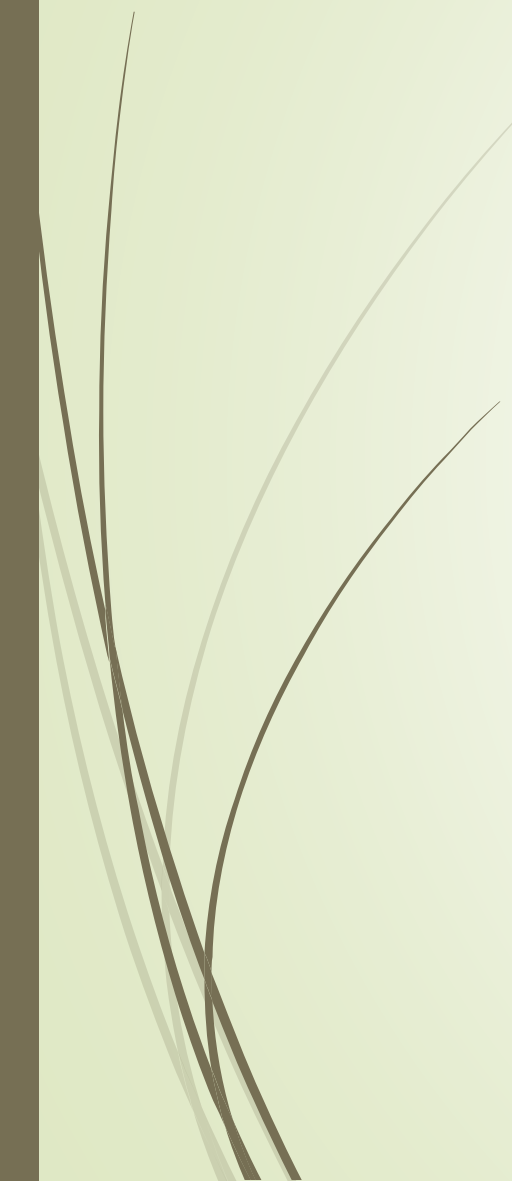




# 8051 實習

2017/10/16



# 觀看program counter與opcode

在debug模式下

The screenshot shows a debugger interface with two main panes: Registers and Disassembly. The Registers pane on the left shows various registers, with PC \$ set to C:0x0000. The Disassembly pane on the right shows a list of instructions. A yellow highlight is on the instruction at address C:0x0000, which is MOV R0, #0xFF. Red arrows point from labels to the PC value, the opcode (78FF), and the instruction text (MOV R0, #0xFF).

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0000
states	0
sec	0.00000000
psw	0x00

```
2: MOV R0, #0xff
C:0x0000 78FF MOV R0, #0xFF
3: MOV A, R0
C:0x0002 E8 MOV A, R0
4: ADD A, #0x01
C:0x0003 2401 ADD A, #0x01
5: CLR A
C:0x0004 00 CLR A
6: JZ HERE
C:0x0005 00 JZ HERE
7: MOV R1, #0xff
C:0x0008 79FF MOV R1, #0xFF
8: MOV R2, #0xff
C:0x000A 7AFF MOV R2, #0xFF
9: HERE: CPL A
C:0x000C F4 CPL A
C:0x000D 00 NOP
C:0x000E 00 NOP
C:0x000F 00 NOP
C:0x0010 00 NOP
C:0x0011 00 NOP
```

lab2example.a51

```
01 ORG 0000
02 MOV R0, #0xff
03 MOV A, R0
04 ADD A, #0x01
05 CLR A
06 JZ HERE
07 MOV R1, #0xff
08 MOV R2, #0xff
09 HERE: CPL A
10 END
```

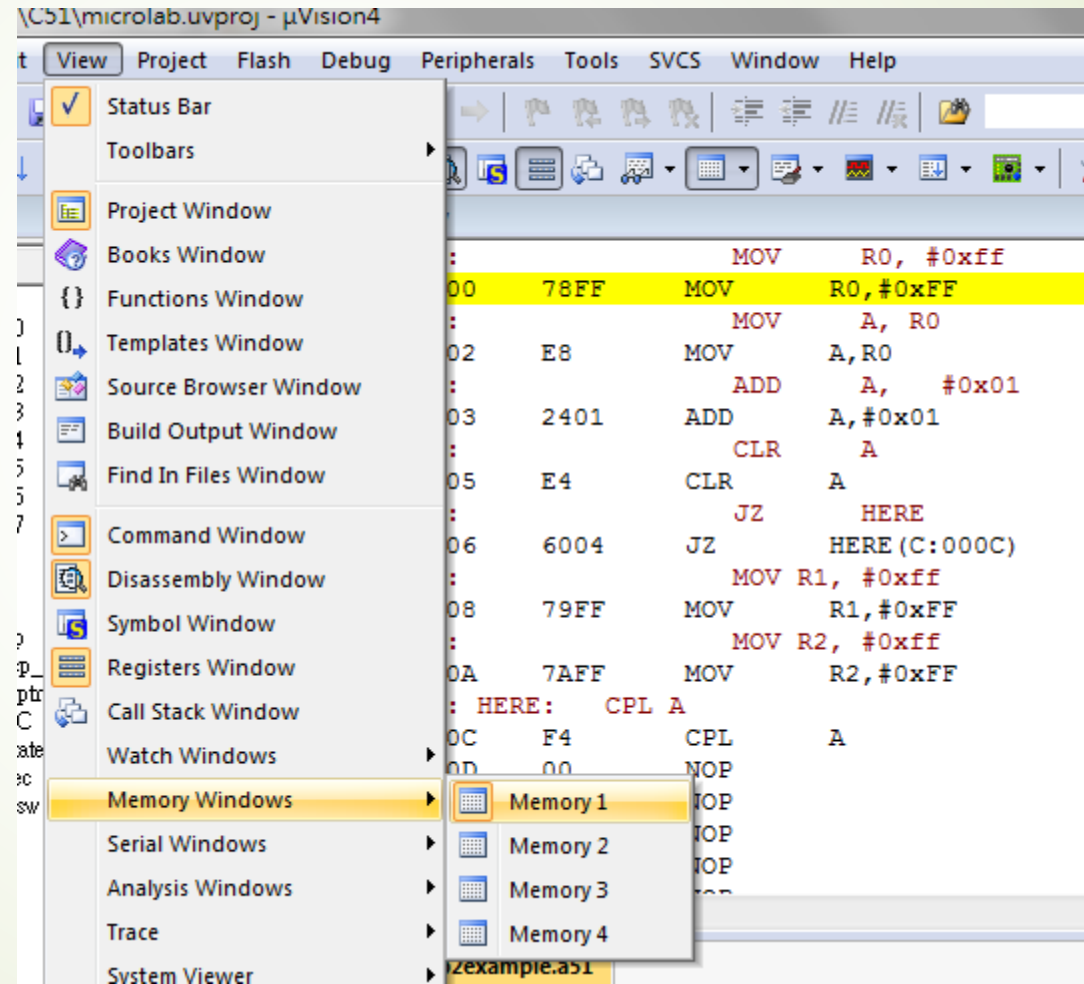
Program counter

Opcode

程式碼

# 觀看記憶體位置的內容

在debug模式下

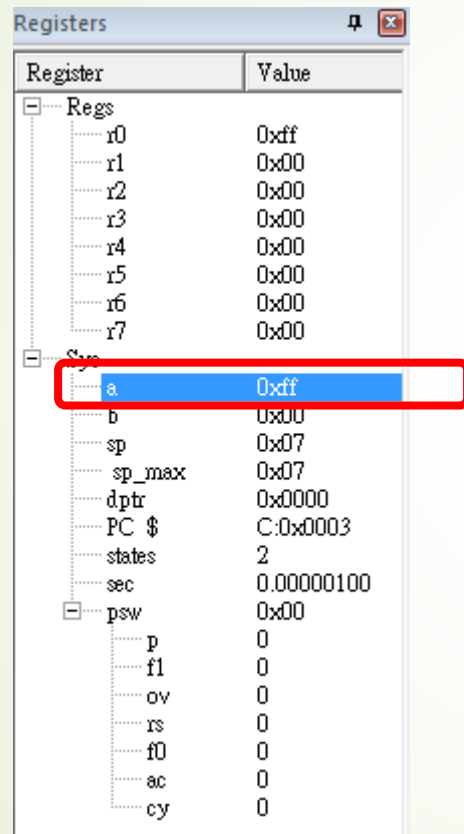




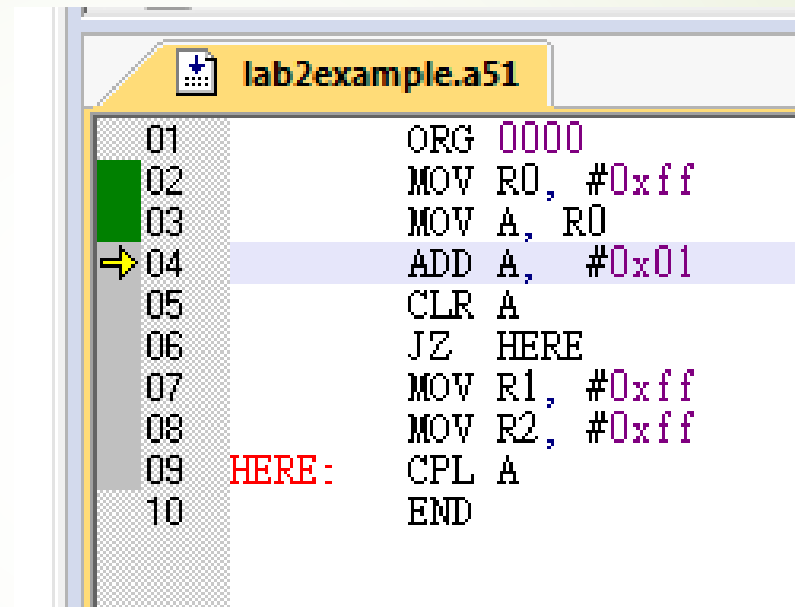


# 觀看CY的變化

在debug模式下



Register	Value
Regs	
r0	0xff
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sp	
a	0xff
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0003
states	2
sec	0.00000100
psw	0x00
p	0
fl	0
ov	0
rs	0
f0	0
ac	0
cy	0

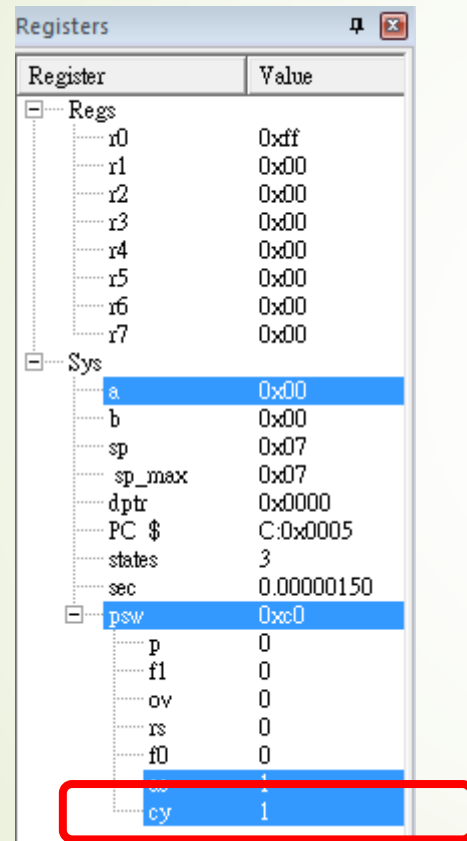


```
lab2example.a51
01      ORG 0000
02      MOV R0, #0xff
03      MOV A, R0
04      ADD A, #0x01
05      CLR A
06      JZ  HERE
07      MOV R1, #0xff
08      MOV R2, #0xff
09  HERE: CPL A
10      END
```

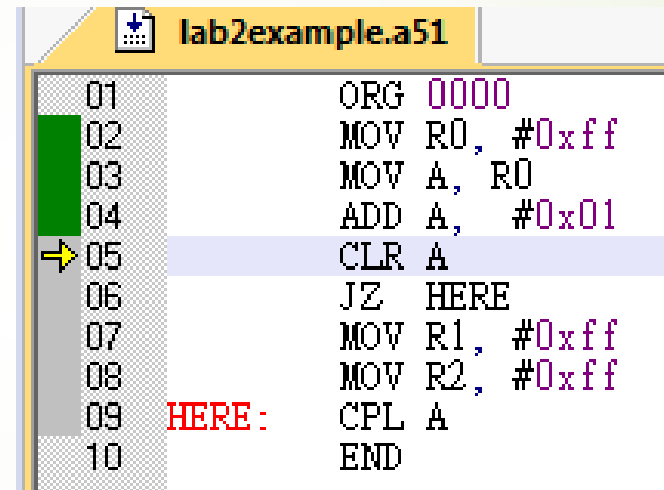
在這個範例程式中，程式執行完第三行，此時A的內容為0xff。

# 觀看CY的變化

在debug模式下



Register	Value
Regs	
r0	0xff
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0005
states	3
sec	0.00000150
psw	0xc0
p	0
fl	0
ov	0
rs	0
fo	0
cy	1



Address	Code
01	ORG 0000
02	MOV R0, #0xff
03	MOV A, R0
04	ADD A, #0x01
05	CLR A
06	JZ HERE
07	MOV R1, #0xff
08	MOV R2, #0xff
09	HERE: CPL A
10	END

接者程式執行完第四行時，將A加上0x01，而導致最高位元有進位的情況產生，所以在執行完第四行後，可以觀察到CY由0變為1。

# Calculating Short Jump Address

在debug模式下

└─ Sys	
└─ a	0x00
└─ b	0x00
└─ sp	0x07
└─ sp_max	0x07
└─ dptr	0x0000
└─ PC \$	C:0x0006
└─ states	4
└─ sec	0.00000200
└─ psw	0xc0

2:		MOV	R0, #0xff
C:0x0000	78FF	MOV	R0, #0xFF
3:		MOV	A, R0
C:0x0002	E8	MOV	A, R0
4:		ADD	A, #0x01
C:0x0003	2401	ADD	A, #0x01
5:		CLR	A
C:0x0005	E4	CLR	A
6:		JZ	HERE
⇒ C:0x0006	6004	JZ	HERE (C:000C)
7:		MOV R1, #0xff	
C:0x0008	79FF	MOV	R1, #0xFF
8:		MOV R2, #0xff	
C:0x000A	7AFF	MOV	R2, #0xFF
9: HERE:		CPL	A
C:0x000C	F4	CPL	A
C:0x000D	00	NOP	

在著個範例程式中，程式執行完第五行，此時A的內容0x00，所以在0x0006的位置執行JZ HERE後會JUMP到0x000C的位置執行CPL A。

HINT:

JZ HERE：如果A的暫存器為0，則跳到HERE，反之執行下一步。

01	ORG	0000
02	MOV	R0, #0xff
03	MOV	A, R0
04	ADD	A, #0x01
05	CLR	A
⇒ 06	JZ	HERE
07	MOV	R1, #0xff
08	MOV	R2, #0xff
09	HERE:	CPL A
10		END



# Calculating Short Jump Address

在debug模式下

Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0006
states	4
sec	0.00000200
psw	0xc0

2:		MOV	R0, #0xff
C:0x0000	78FF	MOV	R0, #0xFF
3:		MOV	A, R0
C:0x0002	E8	MOV	A, R0
4:		ADD	A, #0x01
C:0x0003	2401	ADD	A, #0x01
5:		CLR	A
C:0x0005	E4	CLR	A
6:		JZ	HERE
C:0x0006	6004	JZ	HERE (C:000C)
7:		MOV	R1, #0xff
C:0x0008	79FF	MOV	R1, #0xFF
8:		MOV	R2, #0xff
C:0x000A	7AFF	MOV	R2, #0xFF
9: HERE:		CPL	A
C:0x000C	F4	CPL	A
C:0x000D	00	NOP	

計算方式：位置0x0006內容的運算元，也就是位置0x0007的內容04h，加上下一行程式碼的位置0x0008，其結果為0x00c。

01	ORG	0000
02	MOV	R0, #0xff
03	MOV	A, R0
04	ADD	A, #0x01
05	CLR	A
06	JZ	HERE
07	MOV	R1, #0xff
08	MOV	R2, #0xff
→ 09	HERE:	CPL A
10		END

# 練習

輸入此程式碼，在執行過程中觀察 R0~R3 記憶體內容以及 CY 的變化，並回答出 JZ NEXT, JNC OVER 以及 JNC AGAIN 的 JUMP Address。

HINT:

INC : 暫存器值加一。

JNC : CY=0 時跳。

Line	PC	Opcode	Mnemonic	Operand
01	0000		ORG	0000
02	0000	7800	MOV	R0, #0
03	0002	7455	MOV	A, #55H
04	0004	6003	JZ	NEXT
05	0006	08	INC	R0
06	0007	04	AGAIN:	INC A
07	0008	04		INC A
08	0009	2477	NEXT:	ADD A, #77H
09	000B	5005	JNC	OVER
10	000D	E4	CLR	A
11	000E	F8	MOV	R0, A
12	000F	F9	MOV	R1, A
13	0010	FA	MOV	R2, A
14	0011	FB	MOV	R3, A
15	0012	2B	OVER:	ADD A, R3
16	0013	50F2	JNC	AGAIN
17	0015	80FE	HERE:	SJMP HERE
18	0017			END