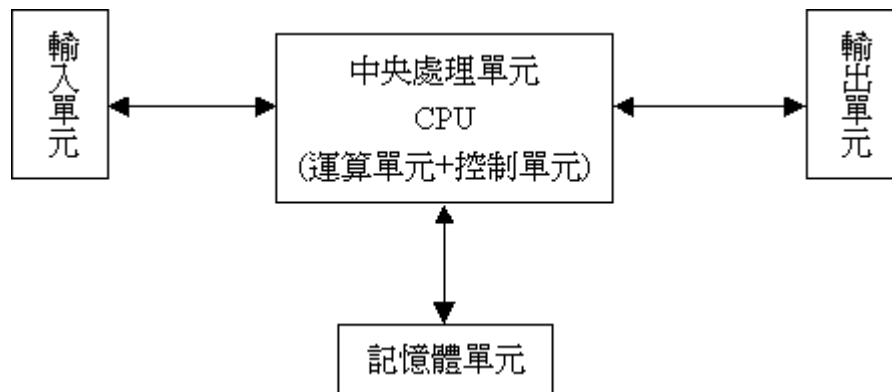


第一章 單晶片微電腦簡介

1.1 微電腦硬體結構

微電腦硬體結構包含中央處理單元、記憶體單元、輸入單元與輸出單元等四個主要單元，其結構關係則如下圖所示。



其中，中央處理單元則是由運算單元與控制單元兩部分所組成的單元，即是一般所通稱的 CPU(Center Processing Unit)，此為微電腦最重要的部分。以下就微電腦中各單元的功能做簡單介紹：

1. 運算單元(Arithmetic Logic Unit，簡稱 ALU)

運算單元又稱為算數邏輯單元，在中央處理單元中可用於執行算數運算，(如：加、減、乘、除等)，以及邏輯運算(如：AND、OR、NOT 等)，能將記憶體單元或輸入單元送至中央處理單元的資料執行各種運算。當運算完成後再由控制單元將結果資料送至記憶體單元或輸出單元。

2. 控制單元(Control Unit，簡稱 CU)

此單元在中央處理單元中，負責協調與指揮各單元間的資料傳送與運作，使得微電腦可依照指令的要求完成工作。在執行一個指令時，控制單元先予以解碼(Decode)，瞭解指令的動作意義後再執行(Execute)該指令，因此控制單元將指令逐一執行，直到做完整個程式的所有指令為止。

3. 輸入單元(Input Unit，簡稱 IU)

此單元是用以將外部的資訊傳送到 CPU 做運算處理或存入記憶體單元，一般在為電腦的輸入單元有鍵盤、磁碟機、光碟機、滑鼠、光筆、掃描器或讀卡機等週邊設備。

4. 輸出單元(Output Unit，簡稱 OU)

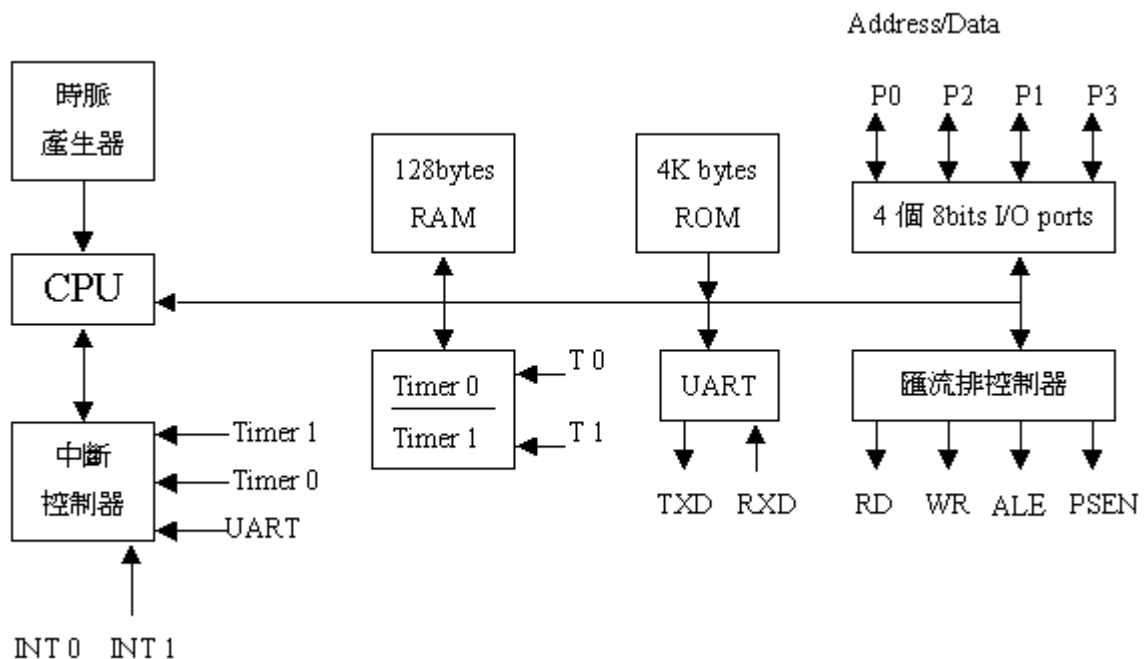
此單元是用以將 CPU 處理過的資料輸出或儲存傳送外部週邊設備，一般在為電腦的輸出單元有顯示器、印表機、繪圖機、燒錄機或磁碟機等週邊設備。

5. 記憶體單元(Memory Unit，簡稱 MU)

記憶體單元是用來儲存輸入單元傳送來的資料，或儲存經過中央處理單元處理完成的資料。記憶體單元之記憶體可分為主記憶體(Main Memory)與輔助記憶體(Auxiliary Memory)兩種，而主記憶體依存取方式不同，又可分為主讀記憶體(Read Only Memory，簡稱 ROM)與隨機存取記憶體(Random Access Memory，簡稱 RAM)。ROM 所儲存的資料，在微電腦中只能被讀出但不能被寫入，也不會因為關機斷電而使資料流失；至於 RAM 在微電腦中，則可被讀出或寫入資料，但在關機斷電後儲存於 RAM 中的資料將會流失。輔助記憶體則是指磁片、硬碟或磁帶等週邊硬體，一般亦為輸出入單元，主要用來彌補主記憶體的不足，其容量可無限制擴充。

1-2 8051 單片的內部結構：

8051 為 Intel 公司所推出的 MCS-51 系列產品之一，其內部結構如下：



8051 單片具有以下之特性：

1. 專為控制使用所設計的 8 位元單晶片。
2. 具有位元邏輯運算能力。
3. 具有 128 位元的 RAM，以及 4K 位元的 ROM。
4. 具有 4 個 8 位元 I/O 埠。
5. 具有 2 個 16 位元的計時/計數器。
6. 具有全雙工的 UART。
7. 具有 5 個中斷源及兩層中斷優先權結構。
8. 具有時脈產生電路。

具有外部電路擴充 64 位元程式記憶體的能力。

1-3 8051 單晶片的接腳：

8051 為 40 支接腳之單晶片，其接腳圖與功能說明如下：

P1.0	1		40	Vcc
P1.1	2		39	P0.0/AD0
P1.2	3		38	P0.1/AD1
P1.3	4		37	P0.2/AD2
P1.4	5		36	P0.3/AD3
P1.5	6	8	35	P0.4/AD4
P1.6	7	0	34	P0.5/AD5
P1.7	8	5	33	P0.6/AD6
R5T	9	1	32	P0.7/AD7
RXD/p3.0	10		31	$\overline{EA}$
TXD/P3.1	11	單	30	ALE
$\overline{INT0}$ /P3.2	12		29	$\overline{PSEN}$
$\overline{INT1}$ /P3.3	13	晶	28	P2.7/A15
T0/P3.4	14		27	P2.6/A14
T1/P3.5	15	片	26	P2.5/A13
$\overline{WR}$ /P3.6	16		25	P2.4/A12
$\overline{RD}$ /P3.7	17		24	P2.3/A11
XTAL2	18		23	P2.2/A10
XTAL1	19		22	P2.1/A9
GND	20		21	P2.0/A8

1. Vcc：+5 電源供應接腳。
2. GND：接地接腳。
3. P0.0~P0.7：埠 0，為開洩極(Open Drain)雙向 I/O 埠。在做為外部擴充記憶體時，可低八位元位址線 (A0~A7 address line)與資料匯流排(data bus)雙重功能。在做為一般 I/O 埠時必須加上如下之外部提升電路。
4. P1.0~P1.7：埠 1，為具有內部提升電路的雙向 I/O 埠。
5. P2.0~P2.7：埠 2，為具有內部提升電路的雙向 I/O 埠。在做為外部擴充記憶體時，可為高八位元位址線(A8~A15 address line)。
6. P3.0~P3.7：埠 3，為具有內部提升電路的雙向 I/O 埠。此外，埠 3 的每支接腳都具有另一特殊功能，其功能如下：
 - RXD(P3.0)：串列傳輸的接收端。
 - TXD(P3.1)：串列傳輸的輸出端。
 - $\overline{INT0}$ (P3.2)：外部中斷輸入端。
 - $\overline{INT1}$ (P3.3)：外部中斷輸入端。
 - T0(P3.4)：計時/計數器外部輸入端。
 - T1(P3.5)：計時/計數器外部輸入端。
 - $\overline{WR}$ (P3.6)：外部資料記憶體寫入激發信號(Strobe)。
 - $\overline{RD}$ (P3.7)：外部資料記憶體讀取激發信號(Strobe)。
7. RST：重置信號(Reset)輸入端。在單晶片工作時，將此腳保持在“Hi”兩個機械週期，CPU 將重置。
8. ALE：位址鎖住致能(Address Latch Enable)，在每個機械週期都會出現，可做為外部電路的時脈源。
9. $\overline{PSEN}$ ：程式激發致能(Program Strobe Enable)，可輸入外部程式記憶體的讀取信號。
10. $\overline{EA}$ ：外部存取致能(External Access Enable)，當 EA 接腳為“L0”時，則讀取外部程式記憶體執行。
11. XTAL1：反相振盪放大器的輸入端。

第二章 程式與編譯

§ 2-1 程式編寫流程

8051 單晶片應用於控制上時，整體系統的設計包括軟體程式及硬體電路兩方面。硬體電路設計是依受控系統之不同而異，雖然有時候系統的某些功能可以採用軟體或硬體來完成，但在考量硬體成本及 8051 單晶片運算能力所及程度，以軟體程式來完成為較佳方法。

對於 8051 單晶片之編譯流程如下：

§ 2-2 8051 組合語言程式的格式

8051 組合語言程式是由一系列一系列的敘述(statement)所組成，而程式的執行則須先經 8051 組譯器編譯後，並經燒錄器燒錄至 8051 單晶片中方可執行。而 8051 組合語言的格式則是由 4 個欄位所構成，其格式如下：

[Label]	[Mnemonic]	[Operand]	[Comment]
標記欄	指令欄	運算元欄	註解欄

1. 標記欄 (Label)

標記的功用是用以替代繁複的記憶體位址計算，以方便程式的編寫、分析與維護。標記的編寫必須從文書編輯軟體的第一格開始，標記可以由英文字母、阿拉伯數字、問號及底線字元組合而成，長度最多可以達 32 個位元，最後必須以冒號來結束。

2. 指令欄 (Mnemonic)

指令可分為兩種，一種是 8051 單片指令，另一種則是編譯程式的虛指令，用以通知編譯器對程式作某些特定的處理。如果一系列指令開頭沒有標記時，則指令前必須保留一個以上的空格。

3. 運算元欄 (Operand)

運算元依指令決定需要與否，且其需要長度亦依指令而異。

4. 註解欄 (Comment)

註解欄是以分號起頭的一段說明文字，直到該行結束。可提供程式設計師註解說明。

第三章 8051 單晶片程式指令

8051 單晶片程式指令依其功能，可分為以下 5 類：

1. 算術指令
2. 邏輯指令
3. 資料轉移指令
4. 布林（Boolean）運算指令
5. 程式跳躍指令

以下就上列指令；分別說明每個指令動作、佔用的位元數及執行時間週期：

■ 指令符號定義

符號	說明
Rn	所選擇暫存器庫中之 R0~R7，n=0~7。
direct	直接定址位址。
@Ri	間接定址位址，i=0 或 1。
#data	8 位元資料常數。
#data16	16 位元資料常數。
addr11	11 位元位址常數，使用於 ACALL 及 AJMP 指令。
addr16	16 位元位址常數，使用於 LCALL 及 LJMP 指令。
rel	8 位元偏移位址常數，使用於 SJMP 及相對跳躍指令中。
bit	位元定址位址。
←	以右方資料取代左方資料。
(X)	將 X 內容取出。
((X))	以 X 內容為位址，以間接定址方式取出資料。
rrr	n 之 2 進制值。如 n=6，rrr=110。

符號定義

■ 算數運算指令

指令	說明	位元組	機械週期
ADD A, Rn	將暫存器內容加入 A 累加器	1	1
ADD A, direct	將直接位址內容加入 A 累加器	2	1
ADD A, @Ri	將間接位址內容加入 A 累加器	1	1
ADD A, #data	將 8 位元常數資料加入 A 累加器	2	1

ADDC A , Rn	將暫存器與進位 C _F 加入 A 累加器	1	1
ADDC A , direct	直接位址內容與進位 C _F 加入累加器	2	1
ADDC A , @Ri	間接位址內容與進位 C _F 加入累加器	1	1
ADDC A , #data	將 8 位元常數資料與進位加入累加器	2	1
SUBB A , Rn	A 累加器內容減暫存器與借位 C _F	1	1
SUBB A , direct	A 累加器內容減直接位址內容與借位	2	1
SUBB A , @Ri	A 累加器內容減間接位址內容與借位	1	1
SUBB A , #data	累加器內容減 8 位元常數資料與借位	2	1
INC A	A 累加器內容加 1	1	1
INC Rn	暫存器內容加 1	1	1
INC direct	直接位址內容加 1	2	1
INC @Ri	間接位址內容加 1	1	1
INC DPTR	資料指標 DPTR 內容加 1	1	2
DEC A	A 累加器內容減 1	1	1
DEC Rn	暫存器內容減 1	1	1
DEC direct	直接位址內容減 1	2	1
DEC @Ri	間接位址內容減 1	1	1
MUL AB	A 累加器乘以暫存器 B，相乘結果之高 8 位元存入 B，低 8 位元存入 A	1	4
DIV AB	A 累加器除以暫存器 B，相除結果之商存入 A，餘數存入 B	1	4
DA A	A 累加器內容調整成 10 進制 BCD 數	1	1

算數運算指令

■ 邏輯運算指令

指令	說明	位元組	機械週期
ANL A , Rn	暫存器 AND 至 A 累加器內	1	1
ANL A , direct	直接位址內容 AND 至 A 累加器內	2	1
ANL A , @Ri	間接位址內容 AND 至 A 累加器內	1	1
ANL A , #data	8 位元資料 AND 至 A 累加器內	2	1
ANL direct , A	A 累加器內容 AND 至直接位址內	2	1
ANL direct , #data	8 位元資料 AND 至直接位址內	3	2
ORL A , Rn	暫存器 OR 至 A 累加器內	1	1
ORL A , direct	直接位址內容 OR 至 A 累加器內	2	1
ORL A , @Ri	間接位址內容 OR 至 A 累加器內	1	1
ORL A , #data	8 位元資料 OR 至 A 累加器內	2	1
ORL direct , A	A 累加器內容 OR 至直接位址內	2	1
ORL direct , #data	8 位元資料 OR 至直接位址內	3	2

XRL A, Rn	暫存器 XOR 至 A 累加器內	1	1
XRL A, direct	直接位址內容 XOR 至 A 累加器內	2	1
XRL A, @Ri	間接位址內容 XOR 至 A 累加器內	1	1
XRL A, #data	8 位元資料 XOR 至 A 累加器內	2	1
XRL direct, A	A 累加器內容 XOR 至直接位址內	2	1
XRL direct, #data	8 位元資料 XOR 至直接位址內	3	2
CLR A	清除 A 累加器	1	1
CPL A	A 累加器內容取補數	1	1
RL A	A 累加器內容向左旋轉 1 位元	1	1
RLC A	A 累加器與進位 C _F 一起左旋 1 位元	1	1
RR A	A 累加器內容向右旋轉 1 位元	1	1
RRC A	A 累加器與進位 C _F 一起右旋 1 位元	1	1
SWAP A	A 累加器的高低 4 位元互相交換	1	1

邏輯運算指令

■ 資料轉移指令

指令	說明	位元組	機械週期
MOV A, Rn	將暫存器內容移入 A 累加器	1	1
MOV A, direct	將直接位址內容移入 A 累加器	2	1
MOV A, @Ri	暫存器間接位址內容移入 A 累加器	1	1
MOV A, #data	將 8 位元常數資料移入 A 累加器	2	1
MOV Rn, A	將 A 累加器內容移入暫存器	1	1
MOV Rn, direct	將直接位址內容移入暫存器	2	2
MOV Rn, #data	將 8 位元常數資料移入暫存器	2	1
MOV direct, A	將 A 累加器內容移入直接位址內	2	1
MOV direct, Rn	將暫存器內容移入直接位址內	2	2
MOV direct, direct	將直接位址內容移入直接位址內	3	2
MOV direct, @Ri	暫存器間接定址內容移入直接位址內	2	2
MOV direct, #data	將 8 位元常數資料移入直接位址內	3	2
MOV @Ri, A	將 A 累加器內容移入間接位址內	1	1
MOV @Ri, direct	將直接位址內容移入間接位址內	2	2
MOV @Ri, #data	將 8 位元常數資料移入間接位址內	2	1
MOV DPTR, #data16	將 16 位元常數資料移入資料指標內	3	2
MOVC A, @A+DPTR	將程式記憶體內容移入 A 累加器內	1	2
MOVC A, @A+PC	將程式記憶體內容移入 A 累加器內	1	2
MOVX A, @Ri	將外部資料記憶體內容移入 A 累加器	1	2
MOVX A, @DPTR	將外部資料記憶體內容移入 A 累加器	1	2
MOVX @Ri, A	將 A 累加器內容移入外部資料記憶體	1	2

MOVX @DPTR, A	將 A 累加器內容移入外部資料記憶體	1	2
PUSH direct	將直接位址內容存入堆疊內	2	2
POP direct	自堆疊頂端取出資料存入直接位址內	2	2
XCH A, Rn	A 累加器內容與暫存器內容互換	1	1
XCH A, direct	A 累加器內容與直接位址內容互換	2	1
XCH A, @Ri	A 累加器內容與間接位址內容互換	1	1
XCHD A, @Ri	累加器與間接位址低 4 位元內容互換	1	1

資料轉移指令

■ 布林運算指令

指令	說明	位元組	機械週期
CLR C	清除進位旗標 $C_F=0$	1	1
CLR bit	清除位元位址內容	2	1
SETB C	設定進位旗標 $C_F=1$	1	1
SETB bit	設定位元位址內容	2	1
CPL C	將進位旗標 C_F 內容取補數	1	1
CPL bit	將位元位址內容取補數	2	1
ANL C, bit	將位元位址內容 AND 至 C_F 內	2	1
ANL C, /bit	將位元位址內容取補數 AND 至 C_F 內	2	2
ORL C, bit	將位元位址內容 OR 至 C_F 內	2	2
ORL C, /bit	將位元位址內容取補數 OR 至 C_F 內	2	2
MOV C, bit	將位元位址內容移入進位旗標 C_F 內	2	1
MOV bit, C	將進位旗標 C_F 移入位元位址內	2	2
JC rel	若 $C_F=1$ ，則跳至相對位址 rel	2	2
JNC rel	若 $C_F=0$ ，則跳至相對位址 rel	2	2
JB bit, rel	若 (bit)=1，則跳至相對位址 rel	3	2
JNB bit, rel	若 (bit)=0，則跳至相對位址 rel	3	2
JBC bit, rel	若 (bit)=1，則跳至相對位址 rel，同時清除位元位址 bit 內容	3	2

布林運算指令

■ 程式分支指令

指令	說明	位元組	機械週期
ACALL addr11	副程式呼叫 (可定址 2KB 範圍)	2	2
ACALL addr16	副程式呼叫 (可定址 64KB 範圍)	3	2
RET	自副程式返回主程式	1	2
RETI	自中斷副程式返回主程式	1	2
AJMP addr11	絕對跳躍 (2KB 範圍)	3	2
LJMP addr16	遠程跳躍 (64KB 範圍)	3	2
SJMP rel	相對跳躍 (-128byte ~ +127byte)	2	2
JMP @A+DPTR	間接跳躍 (64KB 範圍)	1	2
JZ rel	若 $A=0$ ，則跳至 rel 位址 範圍-128byte ~ +127byte	2	2
JNZ rel	若 $A \neq 0$ ，則跳至 rel 位址 範圍-128byte ~ +127byte	2	2
CJNE A, direct, rel	若 A 累加器與直接位址內 容不相等，則跳至 rel 位址 範圍-128byte ~ +127byte	3	2
CJNE A, #data, rel	若 $A \neq data$ ，則跳至 rel 位址，範圍 -128byte ~ +127byte	3	2
CJNE Rn, #data, rel	若暫存器內容 $\neq data$ ，則跳至 rel 位址 範圍-128byte ~ +127byte	3	2
CJNE @Ri, #data, rel	若間接位址內容 $\neq data$ ，則跳至 rel 位 址，範圍-128byte ~ +127byte	3	2
DJNZ Rn, rel	暫存器內容減 1，若不等於 0，則跳 至 rel 位址	2	2
DJNZ direct, rel	直接位址內容減 1，若不等於 0，則 跳至 rel 位址	3	2
NOP	無動作	1	1

程式分支指令

第四章 暫存器與資料記憶體

§4-1 累加器

累加器 (Accumulator) 一般以 A 或 Acc 簡稱，是使用頻率最高的暫存器，許多算數運算、邏輯運算及資料搬移等工作，都需要藉由累加器來完成。

§4-2 工作暫存器

在 8051 中共有 8 個 8 位元 (bits) 工作暫存器，分別為 0、1、2、3、4、5、6 及 7。這些工作暫存器可用以輔助累加器在運算上的不足，如儲存即將被處理的資料，或已完成的結果及迴圈數值控制等。

由於在撰寫較複雜程式尤其是呼叫副程式時，為了避免工作暫存器之內容遭到破壞，在 8051 中提供四個暫存器庫，分別為 RB、RB1、RB 及 RB3，每一個暫存器庫均有個 8 位元工作暫存器，並可經由工作暫存器 0~7 來存取，但程式執行中只能選擇四個暫存器庫中的一個暫存器來使用，而其選擇方法則可透過設定 RS1 與 RS0 此兩位元來選擇：

RS1	RS0	暫存器庫	位址
0	0	RB0	00H~07H
0	1	RB1	08H~0FH
1	0	RB2	10H~17H
1	1	RB3	18H~1FH

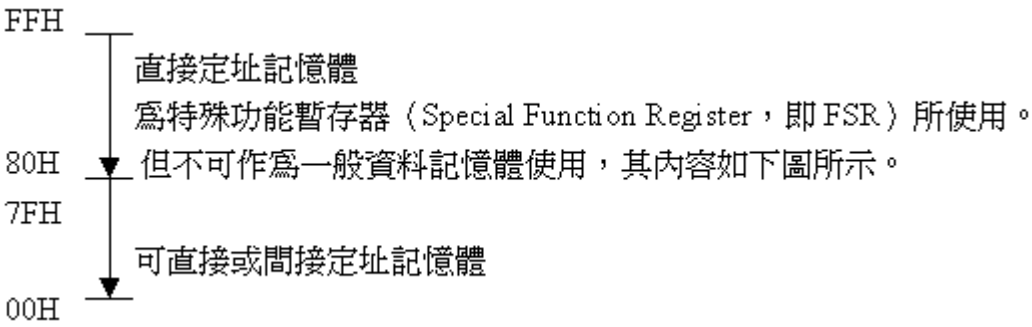
當 8051 選擇使用 RB0 時，程式中存取 R0~R7 暫存器的值，實際上是在存取資料記憶體位址 00H~07H 的內容；而使用 RB1 時，程式中存取 R0~R7 暫存器的值，實際上是在存取資料記憶體位址 08H~0FH 的內容；而使用 RB2 時，程式中存取 R0~R7 暫存器的值，實際上是在存取資料記憶體位址 10H~17H 的內容；而使用 RB3 時，程式中存取 R0~R7 暫存器的值，實際上是在存取資料記憶體位址 18H~1FH 的內容。所以在複雜程式中，主程式與副程式可分配使用不同暫存器庫，即可避免暫存器的值被破壞。

§4-3 輸出/輸入埠暫存器

8051 具有 4 個 8 位元(bits)的輸出輸入埠，經由這四個輸出輸入埠與外界進行資料交換因此 8051 內部用個暫存器來記錄輸出/輸入接腳的狀態，分別為資料記憶體 80H、90H、A0H、B0H 等四個位元組(byte)，並一輸出/輸入埠分別命名為 P0、P1、P2 及 P3。當軟體程式對輸出輸入埠 P0~3 作輸出/輸入的動作，即是對 80H、90H、A0H 及 B0H 等四個位元組作寫入/讀出的動作。

§4-4 資料記憶體

8051 的記憶體可分為兩大部份，一是程式記憶體，即是使用者撰寫軟體程式的存放記憶體區塊;另一是資料記憶體，是用以存放程式執行結果所使用的記憶體。而在 8051 中暫存器與資料記憶體則是結合在一起，均存放在資料記憶體中，及結構如下圖所示：



SFR 的內部結構

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8								CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F

90	P1								97
88	TCON	TMOD	TLO	TL1	TH0	TH1			8F
80	P0	SP	DPL					PCON	87

▲
└─ 此行記憶體位元組可做位元定址。

在 SFR 內部結構圖中可以發現累加器 Acc、四個輸出/輸入暫存器 0、1、2 及 P3。

其他暫存器的功能簡介如下：

B：用於乘、除法中所使用到之暫存器。

PSW (Program Status Word)：程式狀態字元暫存器。

IP (Interrupt Priority Register)：中斷優先暫存器。

IE (Interrupt Enable Register)：中斷致能暫存器。

SCON (Serial Port Control Register)：串列埠控制暫存器。

SBUF (Serial Port Buffer)：串列埠資料緩衝器。

TCON (Timer/Counter Control Register)：計時/計數控制暫存器。

TMOD (Timer/Counter Mode Control Register)：計時/計數模式控制暫存器。

TL0：Timer 0 16 位元計時/計數直之低 8 位元

TL1：Timer 1 16 位元計時/計數直之低 8 位元

TH0：Timer 0 16 位元計時/計數直之高 8 位元

TH1：Timer 1 16 位元計時/計數直之高 8 位元

SP (Stack Pointer)：堆疊指標暫存器。

DPL：DPTR (Data Pointer) 資料指標暫存器 16 位元值之低 8 位元值

DPH：DPTR (Data Pointer) 資料指標暫存器 16 位元值之高 8 位元值

PCON (Power Control Register)：電源控制暫存器。

SFR 內各暫存器的值在 8051 重置（Reset）後，會自動設如下：

暫存器	二進位表示值
*Acc	00000000
*B	00000000
*PSW	00000000
SP	00000111
DPTR	
DPH	00000000
DPL	00000000
*P0	11111111
*P1	11111111
*P2	11111111
*P3	11111111
IP	XXXX0000
IE	0XX00000
TMOD	00000000
*TCOM	00000000
TH0	00000000
TL0	00000000
TH1	00000000
TL1	00000000
*SCON	00000000
SBUF	XXXXXXXX
PCON	HMOS 0XXXXXXX
	CHMOS 0XXX0000

X：未定

*：可位元定址

另外在資料記憶體中，亦可以位元定址 00H~7 共 248 個位元位址，其在記憶中之位址如下圖：

F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
E0H	E7	E6	E5	E4	E3	E2	E1	E0	Acc
D0H	CY	AC	F0	RS1	RS0	OV		P	PSW
	D7	D6	D5	D4	D3	D2	D1	D0	

B8H	—	—	—	PS	PT1	PX1	PT0	PX0	IP
				BC	BB	BA	B9	B8	
B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
				ES	ET1	EX1	ET0	EX0	
A8H	AF	—	—						IE
				AC	AB	AA	A9	A8	
A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
98H									SCON
	9F	9E	9D	9C	9B	9A	99	98	
90H	97	96	95	94	93	92	91	90	P1
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H									TCON
	8F	8E	8D	8C	8B	8A	89	88	
80H	87	86	85	84	83	82	81	80	P0
7FH 30H	一般資料存放區或堆疊區								
2FH	7F	7E	7D	7C	7B	7A	79	78	
2EH									
2DH									
2CH									
2BH									
2AH									
29H									
28H									
27H									
26H									
25H	2F	—	—	—	—	—	—	28	
24H	27	—	—	—	—	—	—	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	—	—	—	—	—	—	10	
21H	0F	—	—	—	—	—	—	08	
20H	07	06	05	04	03	02	01	00	
	RB3								
	RB2								
	RB1								
	RB0								

在上圖表中，最左側為可位元定址記憶體之位址；中間數字代表可畏原定址的位元位址，其中位元位址上之文字即為該位元的名稱；最右側為暫存器的名稱。在位原定址使用上，8051 提供具彈性的使用方法，例如：

1. 將位址 20H 的第 0 Bit 設定為 1，可以寫成下面兩種方式：

SETB 00H ; 位元定址

SETB 20H.0 ; 第 20H 位元組的第 0 位元

2. 將 P0 的第 3 Bit 清除為 0，可以寫成下面三種方式：

CLR 83H ; 位元位址

CLR 80H.3 ; 第 80H 位元組的第 3 位元

CLR P0.3 ; P0 暫存暫存器的第 3 位元

3. 將 IT0 位元的值反相，可以寫成下面四種方式：

CPL 88H ; 位元位址

CPL 88H.0 ; 第 88 位元組的第 0 位元

CPL IT0 ; 位元名稱

CPL TCON.0 ; TCON 暫存器的第 0 位元

第七章 中斷服務與中斷設定

§ 7-1 中斷的功能

8051 單晶片的中斷服務功能，可使中斷服務的需求以中斷的方式通知 8051CPU，以使 CPU 獨立執行主程式，而提升執行效率。在 8051 單晶片中提供 5 個中斷源，分別為：

1. INT0：外部中斷，由 8051 單晶片第 12 接腳輸入。
2. Timer0：計時/計數器中斷。
3. INT1：外部中斷，由 8051 單晶片第 13 接腳輸入。
4. Timer1：計時/計數器中斷。
5. UART：串列埠中斷。

上列中斷源在 8051 中都有相對應的旗標，當中斷條件產生時，中斷源就會使其相對應的旗標值設定為 1。8051 的 CPU 會在每一個機械週期檢查這些旗標的狀態，若系統允許相對的中斷源產生中斷，且該中斷相對應的旗標值亦為 1 時，則 CPU 會在執行完目前正在執行的指令後，將程式在記憶體中的位址存入堆疊中，並產生中斷服務副程式的呼叫，跳到該中斷所對應之中斷向量位址去執行，CPU 執行該中斷服務副程式，直到「RETI」指令後才結束中斷副程式，再從堆疊中取出先前存入的位址值繼續執行被中斷的程式。

§ 7-2 8051 的中斷向量與中斷相關暫存器

8051 單晶片的 5 個中斷源，其中斷向量、旗標名稱與該旗標所屬暫存器如下：

中斷源	中斷向量(位址值)	旗標	所屬暫存器
INT0	0003H	IE0	TCON.1
Timer0	000BH	TF0	TCON.5
INT1	0013H	IE1	TCON.3
Timer1	001BH	TF1	TCON.7
UART(TXD)	0023H	TI	SCON.1
UART(RXD)	0023H	RI	SCON.0

中斷源 INT0 與 INT1 分別位於 8051 單晶片接腳第 12 與 13 支，當此二接腳為低電位(或“0”)時，則 IE0 與 IE1 會設定為“1”；而當對應之中斷服務副程式執行完畢後，則 8051 會自動清除 IE0 與 IE1 旗標。Timer0 與 Timer1 的中斷產生則如第六章所介紹，當計時/計數值產生溢位時，則對應之旗標 TF0 與 TF1 設定為“1”；而當對應之中斷服務副程式執行完畢後，則 8051 會自動清除 TF0 與 TF1 旗標。UART 為串列埠中斷源，當串列埠做為傳送或接收時其對應不同的旗標 TI 與 RI，其使用方式將在下一章介紹，當其對應旗標設定為“1”後，且中斷致能，則中斷服務副程式將會執行。當對應之中斷服務副程式執行完畢後，則 8051 會自動清除 TI 與 RI 旗標。

中斷致能暫存器(Interrupter Enable register，簡稱 IE，可位元定址)用於致能中斷的發生，若被致能，則中斷發生後將執行中斷服務副程式，否則即使中斷發生亦不會執行中斷服務副程式。

以下為中斷致能暫存器結構：

位址：A8H

EA	----	----	ES	ET1	EX1	ET0	EX0
----	------	------	----	-----	-----	-----	-----

EA(IE.7)：若 EA=0，則禁止所有中斷；若 EA=1，則各中斷是否致能可由各自的中斷致能位元來各別設定。

----(IE.6)：未使用。

----(IE.5)：未使用，但於 8052 單晶片中則為 Timer2 致能位元。

ES(IE.4)：致能串列埠的中斷。

ET1(IE.3)：致能 Timer1 的中斷。

EX1(IE.2)：致能 INT1 的中斷。

ET0(IE.1)：致能 Timer0 的中斷。

EX0(IE.0)：致能 INT0 的中斷。

在 8051 單晶片工作時，並不一定要使用全部的中斷源來產生中斷，因此可藉由 IE 來設定致能部分所要用到的中斷源。

此外，每一個中斷源在中斷優先暫存器(Interrupt Priority register，簡稱 IP)中都有一個位元來決定該中斷源之中斷服務副程式被執行的優先順序，設定為“1”表示為高優先權，清除為“0”則表示為低優先權。當一個中斷要求發生時，若中斷是被致能的，則 8051CPU 會執行該中斷服務副程式。然而在執行中若有較高優先權的中斷源要求中斷，則 CPU 會先暫停目前

正在執行的中斷服務副程式，而立即執行這個較高優先權的中斷服務副程式。如果相同優先權或優先權較低的中斷源要求中斷，則 CPU 將會不予理會。另外，若兩中斷同時發生，則高優先權中斷源優先執行；但若優先權相同時，則依 INT0、Timer0、INT1、Timer1、UART 之順序先後執行。

中斷優先暫存器(Interrupt Priority register，可位先定址)，其結構如下：

位址：B8H

----	----	----	PS	PT1	PX1	PT0	PX0
------	------	------	----	-----	-----	-----	-----

----(IP.7)：未使用。

----(IP.6)：未使用。

----(IP.5)：未使用，但在 8052 單晶片，則為 Timer2 的優先權位元。

PS(IP.4)：定義串列埠優先權位元。

PT1(IP.3)：定義 Timer1 優先權位元。

PX1(IP.2)：定義 INT1 優先權位元。

PT0(IP.1)：定義 Timer0 優先權位元。

PX0(IP.0)：定義 INT0 優先權位元。

第八章 串列埠

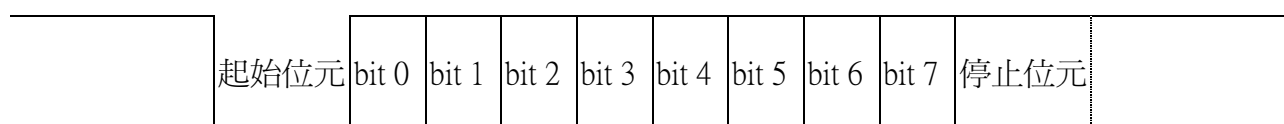
§ 8-1 串列傳輸

串列傳輸為 CPU 與周邊裝置或 CPU 與 CPU 間的資料傳輸方法之一，最簡單的串列傳輸只需兩條傳輸線，使用時的方式每次傳輸一個位元的資料，所以具有傳輸線少的優點，並且容易防止雜訊干擾，適合較遠距離的資料傳輸。然而，由於資料傳輸一次僅送一個位元，因此傳輸資料的速度慢是其缺點。

串列傳輸的結構雖然簡單，但也由於太簡略所以產生許多問題，必須藉由傳輸協定來解決。然而，一個完整的傳輸協定包括從硬體到軟體，相當複雜。其中最基本的一種非同步式串列介面（Universal Asynchronous Receiver Transmitter，簡稱 UART）常被用於一般的串列傳輸應用中。

串列傳輸在傳送一個位元組時，必須要傳送 8 次，而 UART 的串列傳輸方式是在傳送 8 個位元資料之前加上一個起始位元，並在傳送 8 個位元資料之後加上一個停止位元，於是原先傳送一個位元組要傳送 8 次就增為 10 次。以下是 UART 串列傳輸的示意圖，傳輸時間順序由左至右：

等待接收資料
等待接收資料



在 UART 的傳輸結構中，起始位元固定為 0，停止位元固定為 1，所以接收端的動作是一直不斷的檢查傳輸線的狀態。當傳輸線上的信號一直為 1 就表示沒有資料傳送；當傳輸線上的信號由 1 變為 0，即表示有資料將傳送，接收端就會開始準備接收 8 個位元資料，直到傳送完 8 個位元資料，傳送端最後會送出停止位元，並使傳輸線的信號保持為 1，以等待下一次的資料傳輸。經由增加起始位元與停止位元方式，雖然會使串列傳輸效率更降低，但可解決位元資料傳輸的起始與停止之問題。另一串列傳輸協定為傳輸速度，通常以鮑率(Baud Rate)，即每秒傳輸的位元數來衡量，一般 UART 常使用的鮑率有 1200、2400、4800、9600 及 19200 等。兩種裝置在進行串列傳輸時，必須決定以何種鮑率來進行資料傳輸，當兩種裝置使用同一鮑率才能確保資料傳輸正確無誤。

§ 8-2 UART 的結構

8051 單晶片的串列埠是一組全雙工的 UART，即 8051 的 UART 可以在同一時間進行串列資料的傳送與接收。8051 單晶片使用 P3.0 接腳做為串列傳輸的接收端(RXD)，P3.1 接腳做為串列傳輸的輸出端(TXD)，並利用特殊功能暫存器(Special Function Register，簡稱 SFR)中的串

列埠緩衝器(Serial Port Buffer，簡稱 SBUF)執行串列傳輸的工作。當串列傳輸工作設定完成之後，傳送端會存入一筆資料到 SBUF 中，並藉以引發資料傳送的動作；當串列傳輸工作設定完成之後，接收端會將接收資料放入 SBUF 中。但在 8051 單晶片的 UART 結構中，接收資料端與傳送資料端實際使用的暫存器並不是同一個，只不過它們均對應到相同的定址位址，因此在傳送或接收資料時，8051 單晶片會自動選擇使用不同的暫存器，所以 8051 的串列埠可以同時進行資料的傳送與接收。

8051 單晶片進行串列資料傳輸時，串列埠具有輸入緩衝的功能，即當串列埠接收到一筆資料後，會把資料存放至 SBUF 中，然後繼續接收資料，並在接收或等待接收下一筆資料的過程中處理 SBUF 中的資料。因此，串列埠可以持續不斷的接收資料，而不必在接收一筆資料後等待該資料完全處理完畢才進行下一筆資料的接收。但在第二筆資料被 UART 接收完畢前，第一筆資料須被處理完畢由程式讀入，否則會產生資料流失的問題。

§ 8-3 UART 相關暫存器

在 SFR 記憶體中與 UART 相關的暫存器有兩個，分別為串列埠控制暫存器(Serial Port Control register，簡稱為 SCON)及電源控制暫存器(Power Control register，簡稱為 PCON)。以下為此二暫存器的結構圖：

§ 8-4 UART 串列埠的四種工作模式

在 SCON 結構圖中可知 SCON 位元是由模式選擇位元，可規劃資料位元及旗標位元所組成。而 PCON 結構圖中可知只有 SMOD 位元與串列埠傳輸速度有關，其他位元則是用於省電模式的設定。

利用 SCON 的 SM0 及 SM1 可以來選擇四種工作模式：

1. 模式 0：SM1=SM0=1

串列埠設定為模式 0 時，串列資料的傳送與接收都是利用 RXD 接腳進行，而 TXD 接腳則做為輸出移位脈波，此脈波的頻率固定為 8051 單晶片的振盪頻率之 1/12。當要從串列埠傳送資料時，只要執行一個資料寫入 SBUF 指令，則會引發資料傳送的動作；資料傳送完畢後，8051CPU 會將 SCON 中的 TI 位元設定為 1，通知串列中斷產生。當要從串列埠接收資料時，須先以軟體設定 SCON 中的 REN 位元，然後執行清除 RI 位元，串列埠就會依時序進行接收的工作，資料接收完畢後，8051CPU 會將 SCON 中的位元設定為 1，通知串列中斷產生。

模式 0 通常是用於 I/O 的擴充，而非用於串列通訊。只要將 RXD 及 TXD 接腳連接到一個並入串出(PISO)的 IC，就可以擴充一個 8 位元的輸入埠；將 RXD 及 TXD 接腳連接到一個串入並出(SIPO)的 IC，就可以擴充一個 8 位元的輸出埠。

2. 模式 1：SM1=1、SM0=0

串列埠設定為模式 1 時，8051CPU 每次傳送與接收的資料為 10 位元，這 10 位元分成下列 3 部分，分別為：

(1)起始位元：固定為 0，佔用一個位元。

(2)資料位元：佔 8 個位元，依低位元至高位元傳輸順序。

(3)停止位元：固定為 1，佔用一個位元。

模式 1 資料傳輸的鮑率是由 Timer 1 設定，其設定如下表：

常用的鮑率值

Timer 1

鮑率	振盪器頻率	SMOD	C/T	模式	載入值
模式 0 (最大 1M)	12M Hz	×	×	×	×
模式 2 (最大 375K)	12M Hz	1	×	×	×
模式 1、3 (最大 62.5K)	12M Hz	1	0	2	FFH
19200	11.0592M Hz	1	0	2	FDH
9600	11.0592M Hz	0	0	2	FDH
4800	11.0592M Hz	0	0	2	FAH
2400	11.0592M Hz	0	0	2	F4H
1200	11.0592M Hz	0	0	2	E8H
137.5	11.0592M Hz	0	0	2	1DH
110	6M Hz	0	0	2	72H
110	12M Hz	0	0	1	FEEDH

串列埠設定完畢後，8051CPU 執行寫入資料到 SBUF 指令時，就會進行資料傳送的動作。當資料傳送完畢後，CPU 會將 SCON 中的 TI 位元設定，通知串列中斷產生。而在資料接收時，當 RXD 接腳由 1 變為 0 時開始接收資料，CPU 依序接收 10bit 資料；接收資料完畢後，CPU 會測試 RI、SM2 及停止位元是否符合下列條件：

(1)RI 位元清除為 0

(2)SM2 位元清除為 0 或所接收到的停止位元設定為 1

當上列條件都符合時，8051CPU 則將所接收到的 8 位元資料存入 SBUF 中，並將所接收到的停止位元存入 SCON 的 RB8 位元中，再將 RI 位元設定為 1，通知串列中斷產生。若上列條件不符合時，則該次所接收的資料將會流失。

3. 模式 2：SM1=0、SM0=1

串列模式設定為 2 時，8051CPU 每次傳送與接收的資料為 11 位元，這 11 位元是由下列 4 部分所組成，分別為：

- (1)起始位元：固定為 0，佔用一個位元
- (2)資料位元：佔 8 個位元，依低位元至高位元傳輸順序
- (3)可規劃資料位元：佔用一個位元(TB8 或 RB8)
- (4)停止位元：固定為 1，佔用一個位元

模式 2 資料傳輸的鮑率是由 SMOD 決定，當 SMOD=0 時，鮑率為 375K Hz；當 SMOD=1 時，鮑率為 187.5K Hz。當傳送資料時，必須先由軟體設定 SCON 中 TB8 的位元值，然後再執行資料寫入 SBUF 指令，以驅動資料開始傳送的動作，然後串列埠會依序傳送起始位元、資料位元、可規劃資料位元 TB8 及停止位元；傳送完畢後，8051CPU 會設定 SCON 中 TI 位元值，以通知串列中斷產生。

當接收資料時，若 RXD 接腳信號由 1 變為 0 時開始接收資料，8051CPU 會依序接收 11 位元資料；接收資料完畢後，CPU 會測試 RI、SM2 及停止位元是否符合下列條件：

- (1)RI 位元清除為 0
- (2)SM2 位元清除為 0 或所接收之可規劃資料位元為 1

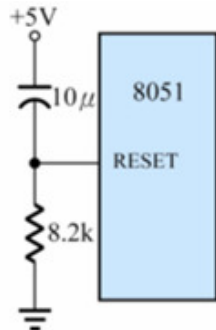
當上列條件都符合時，8051CPU 則將所接收到的 8 位元資料存入 SBUF 中，且將所接收到的可規劃資料位元存入 SCON 的 RB8 位元，再將 RI 位元設定為 1，以通知串列中斷發生。若上列條件不能同時符合時，則該次所接收的資料將會流失。

4. 模式 3：SM1=1、SM0=1

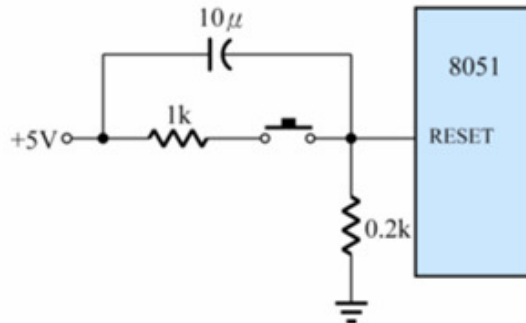
串列埠設定為模式 3 時，其動作與模式 2 相似，其唯一的差別在於模式 3 的傳輸速度之鮑率值設定與模式 1 相同，是由 Timer1 設定。

以上四種串列埠模式，在傳輸資料時，鮑率的準確與否對資料之接收非常的重要，因此因此在使用模式 1 與模式 3 時，要先啟動 Timer 工作。以下列串列埠使用的步驟以檢查串列埠設定是否正確：

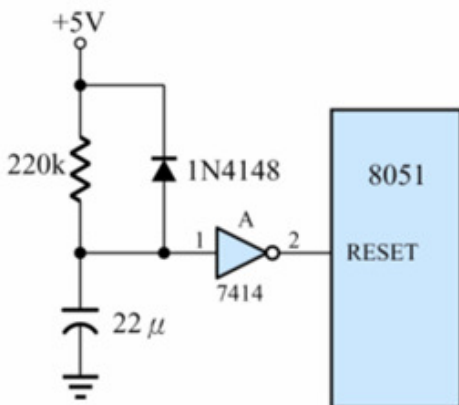
- (1) 設定 Timer1 工作模式並根據傳輸速率設定 TH1 及 TL1(UART 模式 0 與模式 2 不用此項)
 - (2) 決定 SMOD 位元值為 0 或 1
 - (3) 設定串列埠工作模式，並清除 RI、TI 位元為 0，及設定 REN 位元為 1
 - (4) 致能串列埠中斷
 - (5) 啟動 Timer1 開始計時(UART 模式 0 與模式 2 不用)
 - (6) 執行 "MOV SBUF, XX" 指令，來啟動 UART 傳送資料
- 重置電路 (RESET)



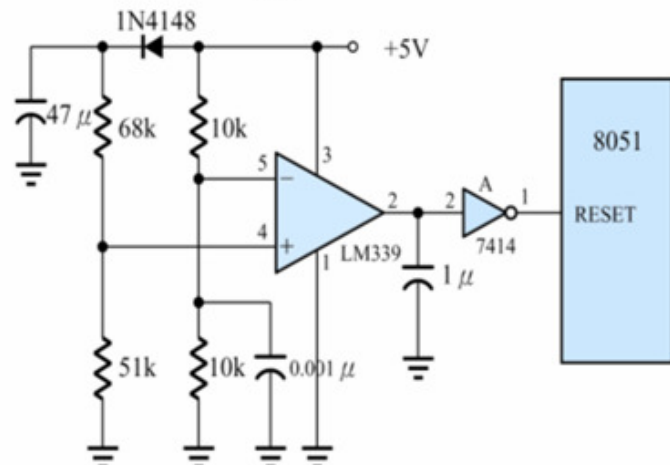
(A)



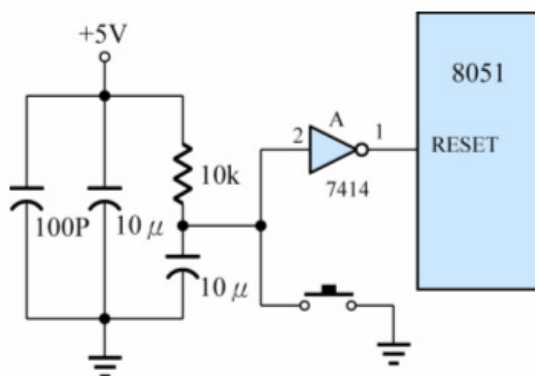
(B)



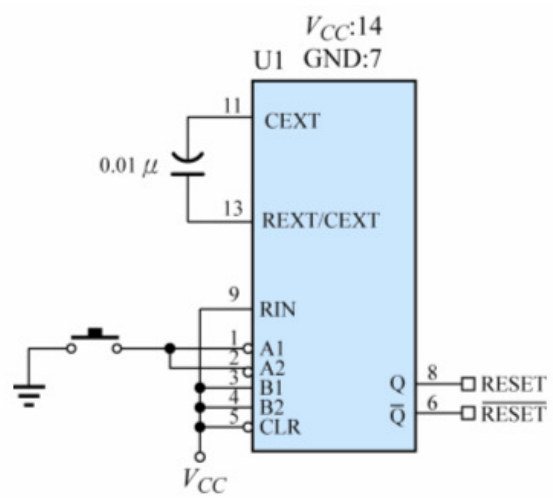
(C)



(D)



(E)



(F)

