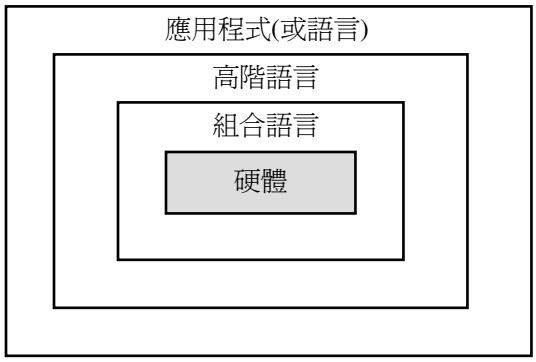
基本程式設計觀念

人 ↑

自然語言（中文、英文、...）
高階語言（C、C++、Java）
組合語言（8051或是80x86）
機器語言（8051或是80x86）

↓ 機器

計算機的階層式結構



儲存程式計算機

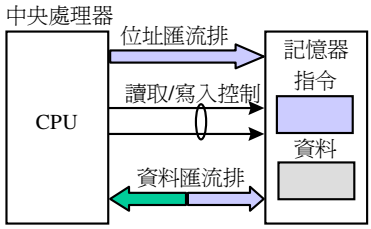


圖2.1-3 計算機的邏輯結構

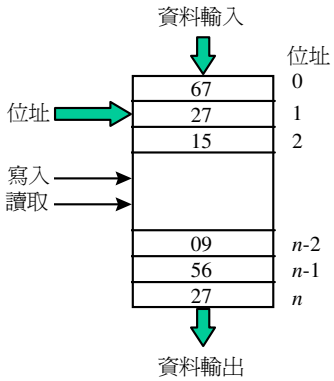


圖2.1-4 記憶器的邏輯結構

CPU的動作

CPU 的動作

CPU 模組

PC ← 0 ；

重複執行下列動作

自記憶器位址爲 PC 的位置中摘取指令；

執行該指令；

PC ← PC + 1

END CPU 模組

CPU基本結構與動作

CPU 的動作

CPU　模組

PC ← 0 ；

重複執行下列動作

自記憶器位址爲 PC 的位置中摘取指令；

執行指令解碼；

若該指令執行時需要資料，則自記憶器中讀取運算元；

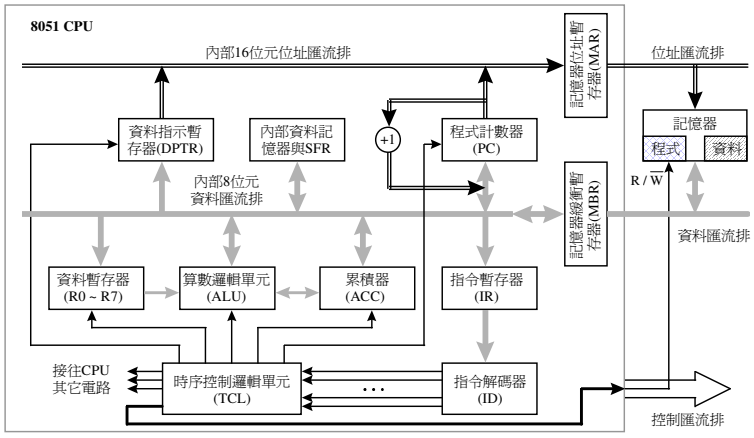
執行指令的動作；

若該指令需要儲存結果，則存回結果於記憶器中；

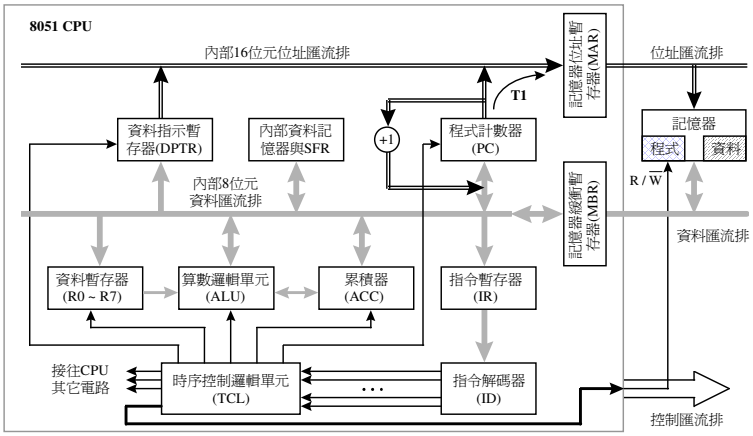
PC ← PC + 1

END　CPU　模組

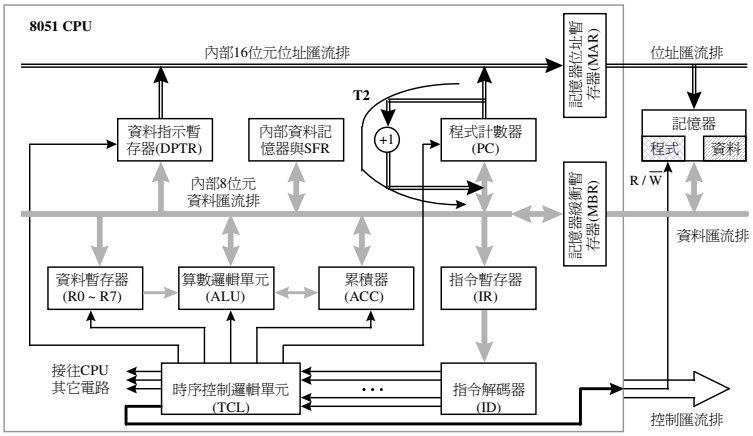
一個簡化的MCS-51 CPU RTL模型



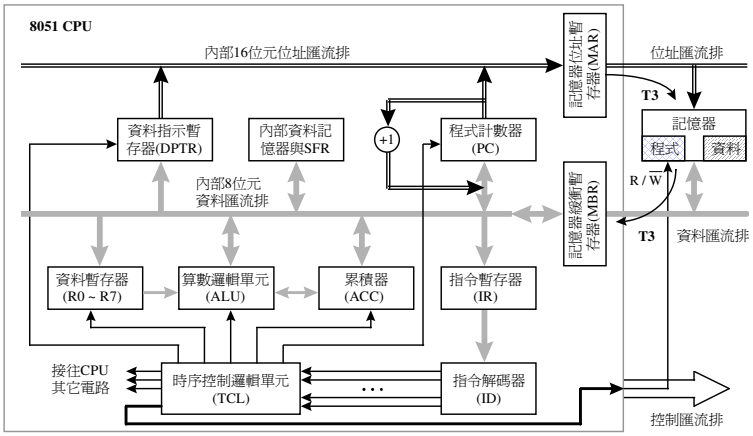
指令讀取的第一個步驟



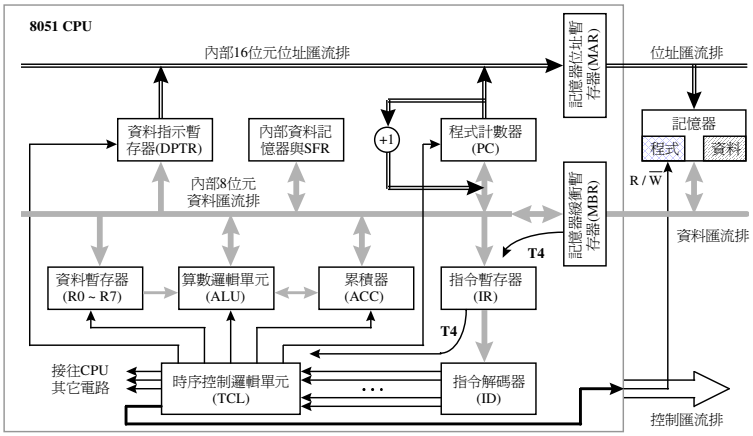
指令讀取的第二個步驟



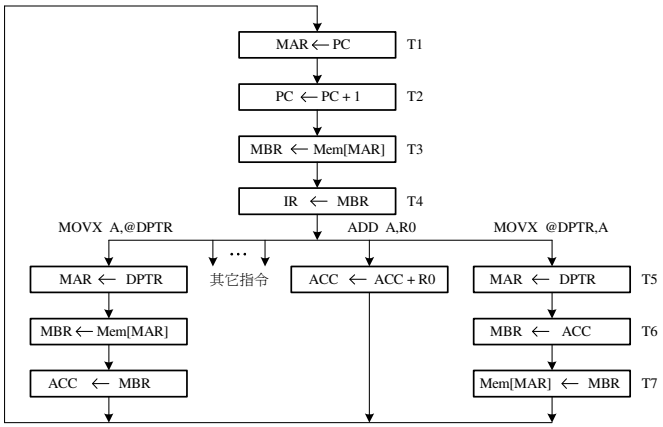
指令讀取的第三個步驟



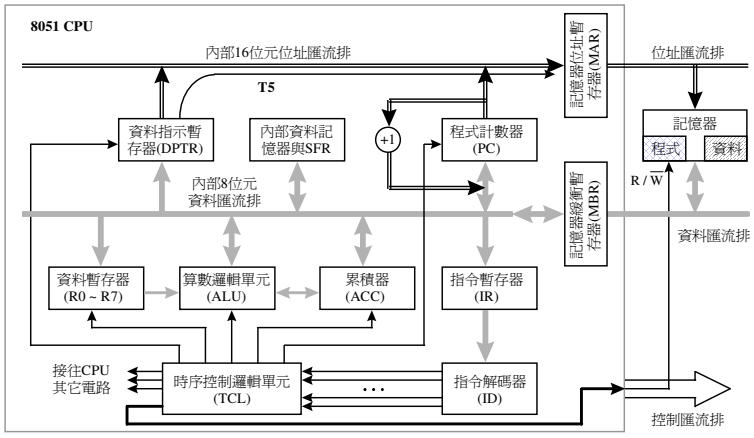
指令讀取的第四個步驟



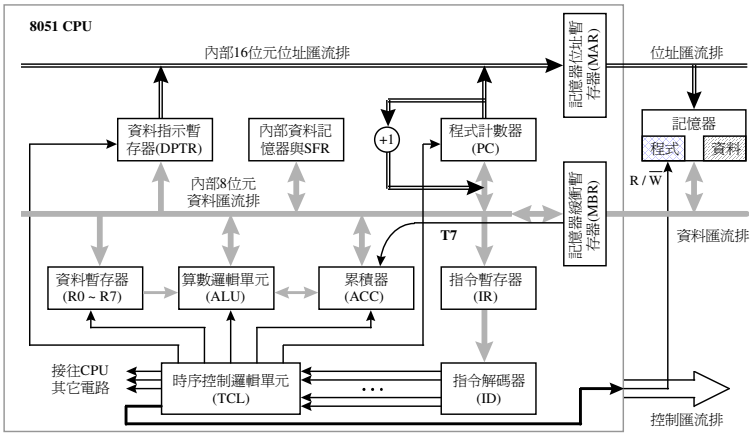
指令的讀取與執行動作時序圖



指令MOVX A,@DPTR執行的第一個步驟



指令MOVX A,@DPTR執行的最後一個步驟



資料轉移指令

指令	RTL 描述	說明
MOV A,Rn	ACC ← Rn	轉移暫存器 Rn 的內容到累積器 ACC
MOV A,#data	ACC ← data	轉移 8 位元的立即資料到累積器 ACC
MOV Rn,A	Rn ← A	轉移累積器 ACC 的內容到暫存器 Rn
MOV Rn,#data	Rn ← data	轉移 8 位元的立即資料到暫存器 Rn
MOV DPTR,#data16	DPTR ← data16	轉移 16 位元的立即資料到 DPTR 中
MOVX A,@DPTR	ACC ← Mem[DPTR]	讀取記憶器中由 DPTR 指定的位置內容後，儲存於累積器 ACC 中

算術運算指令

指令	RTL 描述	說明
ADD A,Rn	ACC ← ACC + Rn	累積器 ACC 與 Rn 相加後，存回 ACC
ADD A,#data	ACC ← ACC + data	累積器 ACC 與 data 相加後，存回 ACC
ADDC A,Rn	ACC ← ACC + Rn+C	累積器 ACC 與 Rn 及進位相加後，存回 ACC
ADDC A,#data	ACC ← ACC + data+C	累積器 ACC 與 data 及進位相加後，存回 ACC
SUBB A,Rn	ACC ← ACC – Rn - C	累積器 ACC 減去 Rn 與 C 後，存回 ACC
SUBB A,#data	ACC ← ACC – data – C	累積器 ACC 減去 data 與 C 後，存回 ACC

邏輯運算指令

指令	RTL 描述	說明
ANL A,Rn	ACC ← ACC ∧ Rn	累積器 ACC 與 Rn AND 後，存回 ACC
ANL A,#data	ACC ← ACC ∧ data	累積器 ACC 與 data AND 後，存回 ACC
ORL A,Rn	ACC ← ACC ∨ Rn	累積器 ACC 與 Rn OR 後，存回 ACC
ORL A,#data	ACC ← ACC ∨ data	累積器 ACC 與 data OR 後，存回 ACC
XRL A,Rn	ACC ← ACC ⊕ Rn	累積器 ACC 與 Rn XOR 後，存回 ACC
XRL A,#data	ACC ← ACC ⊕ data	累積器 ACC 與 data XOR 後，存回 ACC
CPL A	ACC ← $\overline{ACC}$	累積器 ACC 內容取 1 補數

分歧與跳躍指令

指令	RTL 描述	說明
JC disp	C ： PC ← PC + disp(2 補數)	當進位旗號爲 1 時，分歧到標的位址
JNC disp	$\overline{C}$ ： PC ← PC + disp(2 補數)	當進位旗號爲 0 時，分歧到標的位址
JZ disp	ACC=0 ： PC ← PC + disp(2 補數)	當 ACC 爲 0 時，分歧到標的位址
JNZ disp	ACC≠0 ： PC ← PC + disp(2 補數)	當 ACC 爲 1 時，分歧到標的位址
AJMP addr11	PC ← addr11	載入 11 位元的絕對位址於 PC 中

基本組合語言程式例

程式 2.2-1 典型的組合語言程式列表

```
                        1       ;ex4.2-2.a51
----                    2               DSEG  AT 30H
0030                    3       OPR1:   DS    1
0031                    4       OPR2:   DS    1
                        5       ;Exchange two words in memory
                        6       ;using DIRECT addressing mode
----                    7               CSEG  AT 0000H
0000 A830               8       SWAPBYTE: MOV   R0,LOW OPR1 ;get opr1
0002 A931               9               MOV   R1,LOW OPR2 ;get opr2
0004 8831              10               MOV   LOW OPR2,R0 ;save opr1
0006 8930              11               MOV   LOW OPR1,R1 ;save opr2
0008 22                12               RET
                       13               END
```

基本組譯程式假指令

假指令	意義	例子
BSEG AT exp CSEG AT exp DSEG AT exp	定義絕對的位元節區 定義絕對的程式節區 定義絕對的資料節區	BSEG AT 20H CSEG AT 0000H DSEG AT 30H
[標記:] DB <exp>[,<exp>,...] [標記:] DW <exp>[,<exp>,...]	定義位元組資料 定義語句(2 位元組)資料	MESSAGE: DB 0EFH DW 07,0E23FH
[標記:] DBIT <exp> [標記:] DS <exp>	定義位元資料 保留位元組儲存空間	KBFLAG: DBIT 1 DS 50
ORG <exp>	定義機器碼起始位址	ORG 0100H
<name> EQU <exp>	指定 name 的值爲 exp	THREE EQU 3
END	表程式到此結束	END

組合語言程式的建立