

51汇编语言指令集

符号定义表

符号	含义		
Rn	R0~R7寄存器n=0~7		
Direct	直接地址，内部数据区的地址RAM(00H~7FH)		
	SFR(80H~FFH) B, ACC, PSW, IP, P3, IE, P2, SCON, P1, TCON, P0		
@Ri	间接地址Ri=R0或R1 8051/31RAM地址(00H~7FH) 8052/32RAM地址(00H~FFH)		
#data	8位常数		
#data16	16位常数		
Addr16	16位的目标地址		
Addr11	11位的目标地址		
Rel	相关地址		
bit	内部数据RAM(20H~2FH)，特殊功能寄存器的直接地址的位		
指令介绍			
指令	字节	周期	动作说明
算数运算指令			
1. ADD A, Rn	1	1	将累加器与寄存器的内容相加，结果存回累加器
2. ADD A, direct	2	1	将累加器与直接地址的内容相加，结果存回累加器
3. ADD A, @Ri	1	1	将累加器与间接地址的内容相加，结果存回累加器
4. ADD A, #data	2	1	将累加器与常数相加，结果存回累加器
5. ADDC A, Rn	1	1	将累加器与寄存器的内容及进位C相加，结果存回累加器
6. ADDC A, direct	2	1	将累加器与直接地址的内容及进位C相加，结果存回累加器
7. ADDC A, @Ri	1	1	将累加器与间接地址的内容及进位C相加，结果存回累加器
8. ADDC A, #data	2	1	将累加器与常数及进位C相加，结果存回累加器
9. SUBB A, Rn	1	1	将累加器的值减去寄存器的值减借位C，结果存回累加器
10. SUBB A, direct	2	1	将累加器的值减直接地址的值减借位C，结果存回累加器
11. SUBB A, @Ri	1	1	将累加器的值减间接地址的值减借位C，结果存回累加器
12. SUBB A, 0data	2	1	将累加器的值减常数减借位C，结果存回累加器
13. INC A	1	1	将累加器的值加1
14. INC Rn	1	1	将寄存器的值加1

15. INC direct	2	1	将直接地址的内容加1
16. INC @Ri	1	1	将间接地址的内容加1
17. INC DPTR	1	1	数据指针寄存器值加1
说明：将16位的DPTR加1，当DPTR的低字节(DPL)从FFH溢出至00H时，会使高字节(DPH)加1，不影响任何标志位			
18. DEC A	1	1	将累加器的值减1
19. DEC Rn	1	1	将寄存器的值减1
20. DEC direct	2	1	将直接地址的内容减1
21. DEC @Ri	1	1	将间接地址的内容减1
22. MUL AB	1	4	将累加器的值与B寄存器的值相乘，乘积的低位字节存回累加器，高位字节存回B寄存器
说明：将累加器A和寄存器B内的无符号整数相乘，产生16位的积，低位字节存入A，高位字节存入B寄存器。如果积大于FFH，则溢出标志位(OV)被设定为1，而进位标志位为0			
23. DIV AB	1	4	将累加器的值除以B寄存器的值，结果的商存回累加器，余数存回B寄存器
说明：无符号的除法运算，将累加器A除以B寄存器的值，商存入A，余数存入B。执行本指令后，进位位(C)及溢出位(OV)被清除为0			
24. DA A	1	1	将累加器A作十进制调整，
若(A) 3-0>9或(AC)=1，则(A) 3-0←(A) 3-0+6			
若(A) 7-4>9或 (C)=1，则(A) 7-4←(A) 7-4+6			
逻辑运算指令			
25. ANL A, Rn	1	1	将累加器的值与寄存器的值做AND的逻辑判断，结果存回累加器
26. ANL A, direct	2	1	将累加器的值与直接地址的内容做AND的逻辑判断，结果存回累加器
27. ANL A, @Ri	1	1	将累加器的值与间接地址的内容做AND的逻辑判断，结果存回累加器
28. ANL A, #data	2	1	将累加器的值与常数做AND的逻辑判断，结果存回累加器
29. ANL direct, A	2	1	将直接地址的内容与累加器的值做AND的逻辑判断，结果存回该直接地址
30. ANL direct, #data	3	2	将直接地址的内容与常数值做AND的逻辑判断，结果存回该直接地址
31. ORL A, Rn	1	1	将累加器的值与寄存器的值做OR的逻辑判断，结果存回累加器
32. ORL A, direct	2	1	将累加器的值与直接地址的内容做OR的逻辑判断，结果存回累加器
33. ORL A, @Ri	1	1	将累加器的值与间接地址的内容做OR的逻辑判断，结果存回累加器
34. ORL A, #data	2	1	将累加器的值与常数做OR的逻辑判断，结果存回累加器
35. ORL direct, A	2	1	将直接地址的内容与累加器的值做OR的逻辑判断，结果存回该直接地址
36. ORL direct, #data	3	2	将直接地址的内容与常数值做OR的逻辑判断，结果存回该直接地址
37. XRL A, Rn	1	1	将累加器的值与寄存器的值做XOR的逻辑判断，结果存回累加器

38. XRL A, direct	2	1	将累加器的值与直接地址的内容做XOR的逻辑判断，结果存回累加器
39. XRL A, @Ri	1	1	将累加器的值与间接地址的内容做XOR的逻辑判断，结果存回累加器
40. XRL A, #data	2	1	将累加器的值与常数作XOR的逻辑判断，结果存回累加器
41. XRL direct, A	2	1	将直接地址的内容与累加器的值做XOR的逻辑判断，结果存回该直接地址
42. XRL direct, #data	3	2	将直接地址的内容与常数的值做XOR的逻辑判断，结果存回该直接地址
43. CLR A	1	1	清除累加器的值为0
44. CPL A	1	1	将累加器的值反相
45. RL A	1	1	将累加器的值左移一位
46. RLC A	1	1	将累加器含进位C左移一位
47. RR A	1	1	将累加器的值右移一位
48. RRC A	1	1	将累加器含进位C右移一位
49. SWAP A	1	1	将累加器的高4位与低4位的内容交换。(A) 3-0 \leftarrow (A) 7-4
数据转移指令			
50. MOV A, Rn	1	1	将寄存器的内容载入累加器
51. MOV A, direct	2	1	将直接地址的内容载入累加器
52. MOV A, @Ri	1	1	将间接地址的内容载入累加器
53. MOV A, #data	2	1	将常数载入累加器
54. MOV Rn, A	1	1	将累加器的内容载入寄存器
55. MOV Rn, direct	2	2	将直接地址的内容载入寄存器
56. MOV Rn, gdata	2	1	将常数载入寄存器
57. MOV direct, A	2	1	将累加器的内容存入直接地址
58. MOV direct, Rn	2	2	将寄存器的内容存入直接地址
59. MOV direct1, direct2	3	2	将直接地址2的内容存入直接地址1
60. MOV direct, @Ri	2	2	将间接地址的内容存入直接地址
61. MOV direct, #data	3	2	将常数存入直接地址
62. MOV @Ri, A	1	1	将累加器的内容存入某间接地址
63. MOV @Ri, direct	2	2	将直接地址的内容存入某间接地址
64. MOV @Ri, #data	2	1	将常数存入某间接地址
65. MOV DPTR, #data16	3	2	将16位的常数存入数据指针寄存器

66. MOVC A, @A+DPTR	1	2	$(A) \leftarrow ((A) + (DPTR))$
累加器的值再加数据指针寄存器的值为其所指定地址，将该地址的内容读入累加器			
67. MOVC A, @A+PC	1	2	$(PC) \leftarrow (PC) + 1; (A) \leftarrow ((A) + (PC))$ 累加器的值加程序计数器的值作为其所指定地址，将该地址的内容读入累加器
68. MOVX A, @Ri	1	2	将间接地址所指定外部存储器的内容读入累加器(8位地址)
69. MOVX A, @DPTR	1	2	将数据指针所指定外部存储器的内容读入累加器(16位地址)
70. MOVX @Ri, A	1	2	将累加器的内容写入间接地址所指定的外部存储器(8位地址)
71. MOVX @DPTR, A	1	2	将累加器的内容写入数据指针所指定的外部存储器(16位地址)
72. PUSH direct	2	2	将直接地址的内容压入堆栈区
73. POP direct	2	2	从堆栈弹出该直接地址的内容
74. XCH A, Rn	1	1	将累加器的内容与寄存器的内容互换
75. XCH A, direct	2	1	将累加器的值与直接地址的内容互换
76. XCH A, @Ri	1	1	将累加器的值与间接地址的内容互换
77. XCHD A, @Ri	1	1	将累加器的低4位与间接地址的低4位互换
布尔代数运算			
78. CLR C	1	1	清除进位C为0
79. CLR bit	2	1	清除直接地址的某位为0
80. SETB C	1	1	设定进位C为1
81. SETB bit	2	1	设定直接地址的某位为1
82. CPL C	1	1	将进位C的值反相
83. CPL bit	2	1	将直接地址的某位值反相
84. ANL C, bit	2	2	将进位C与直接地址的某位做AND的逻辑判断，结果存回进位C
85. ANL C, /bit	2	2	将进位C与直接地址的某位的反相值做AND的逻辑判断，结果存回进位C
86. ORL C, bit	2	2	将进位C与直接地址的某位做OR的逻辑判断，结果存回进位C
87. ORL C, /bit	2	2	将进位C与直接地址的某位的反相值做OR的逻辑判断，结果存回进位C
88. MOV C, bit	2	1	将直接地址的某位值存入进位C
89. MOV bit, C	2	2	将进位C的值存入直接地址的某位
90. JC rel	2	2	若进位C=1则跳至rel的相关地址
91. JNC rel	2	2	若进位C=0则跳至rel的相关地址
92. JB bit, rel	3	2	若直接地址的某位为1，则跳至rel的相关地址
93. JNB bit, rel	3	2	若直接地址的某位为0，则跳至rel的相关地址
94. JBC bit, rel	3	2	若直接地址的某位为1，则跳至rel的相关地址，并将该位值清除为0
程序跳跃			

95. ACALL addr11	2	2	调用2K程序存储器范围内的子程序
96. LCALL addr16	3	2	调用64K程序存储器范围内的子程序
97. RET	1	2	从子程序返回
98. RETI	1	2	从中断子程序返回
99. AJMP addr11	2	2	绝对跳跃(2K内)
100. LJMP addr16	3	2	长跳跃(64K内)
101. SJMP rel	2	2	短跳跃(2K内)-128~+127字节
102. JMP @A+DPTR	1	2	跳至累加器的内容加数据指针所指的相关地址
103. JZ rel	2	2	累加器的内容为0, 则跳至rel所指相关地址
104. JNZ rel	2	2	累加器的内容不为0, 则跳至rel所指相关地址
105. CJNE A, direct, rel	3	2	将累加器的内容与直接地址的内容比较, 不相等则跳至rel所指的相关地址
106. CJNE	3	2	将累加器的内容与常数比较, 若不相等则跳至rel所指的相关地址
107. CJNE @Rn, #data, rel	3	2	将寄存器的内容与常数比较, 若不相等则跳至rel所指的相关地址
108. CJNE @Ri, #data, rel	3	2	将间接地址的内容与常数比较, 若不相等则跳至rel所指的相关地址
109. DJNZ Rn, rel	2	2	将寄存器的内容减1, 不等于0则跳至rel所指的相关地址
110. DJNZ direct, rel	3	2	将直接地址的内容减1, 不等于0则跳至rel所指的相关地址
111. NOP	1	1	无动作