

助记符		指令说明	字节数	周期数
(数据传递类指令)				
MOV	A, Rn	寄存器传送到累加器	1	1
MOV	A, direct	直接地址传送到累加器	2	1
MOV	A, @Ri	累加器传送到外部RAM(8 地址)	1	1
MOV	A, #data	立即数传送到累加器	2	1
MOV	Rn, A	累加器传送到寄存器	1	1
MOV	Rn, direct	直接地址传送到寄存器	2	2
MOV	Rn, #data	累加器传送到直接地址	2	1
MOV	direct, Rn	寄存器传送到直接地址	2	1
MOV	direct, direct	直接地址传送到直接地址	3	2
MOV	direct, A	累加器传送到直接地址	2	1
MOV	direct, @Ri	间接RAM 传送到直接地址	2	2
MOV	direct, #data	立即数传送到直接地址	3	2
MOV	@Ri, A	直接地址传送到直接地址	1	2
MOV	@Ri, direct	直接地址传送到间接RAM	2	1
MOV	@Ri, #data	立即数传送到间接RAM	2	2
MOV	DPTR, #data16	16 位常数加载到数据指针	3	1
MOVC	A, @A+DPTR	代码字节传送到累加器	1	2
MOVC	A, @A+PC	代码字节传送到累加器	1	2
MOVX	A, @Ri	外部RAM(8 地址)传送到累加器	1	2
MOVX	A, @DPTR	外部RAM(16 地址)传送到累加器	1	2
MOVX	@Ri, A	累加器传送到外部RAM(8 地址)	1	2
MOVX	@DPTR, A	累加器传送到外部RAM(16 地址)	1	2

PUSH	direct	直接地址压入堆栈	2	2
POP	direct	直接地址弹出堆栈	2	2
XCH	A, Rn	寄存器和累加器交换	1	1
XCH	A, direct	直接地址和累加器交换	2	1
XCH	A, @Ri	间接RAM 和累加器交换	1	1
XCHD	A, @Ri	间接RAM 和累加器交换低4 位字节	1	1
(算术运算类指令)				
INC	A	累加器加1	1	1
INC	Rn	寄存器加1	1	1
INC	direct	直接地址加1	2	1
INC	@Ri	间接RAM 加1	1	1
INC	DPTR	数据指针加1	1	2
DEC	A	累加器减1	1	1
DEC	Rn	寄存器减1	1	1
DEC	direct	直接地址减1	2	2
DEC	@Ri	间接RAM 减1	1	1
MUL	AB	累加器和B 寄存器相乘	1	4
DIV	AB	累加器除以B 寄存器	1	4
DA	A	累加器十进制调整	1	1
ADD	A, Rn	寄存器与累加器求和	1	1
ADD	A, direct	直接地址与累加器求和	2	1
ADD	A, @Ri	间接RAM 与累加器求和	1	1
ADD	A, #data	立即数与累加器求和	2	1
ADDC	A, Rn	寄存器与累加器求和(带进位)	1	1

ADDC	A, direct	直接地址与累加器求和(带进位)	2	1
ADDC	A, @Ri	间接RAM 与累加器求和(带进位)	1	1
ADDC	A, #data	立即数与累加器求和(带进位)	2	1
SUBB	A, Rn	累加器减去寄存器(带借位)	1	1
SUBB	A, direct	累加器减去直接地址(带借位)	2	1
SUBB	A, @Ri	累加器减去间接RAM(带借位)	1	1
SUBB	A, #data	累加器减去立即数(带借位)	2	1
(逻辑运算类指令)				
ANL	A, Rn	寄存器“与”到累加器	1	1
ANL	A, direct	直接地址“与”到累加器	2	1
ANL	A, @Ri	间接RAM“与”到累加器	1	1
ANL	A, #data	立即数“与”到累加器	2	1
ANL	direct, A	累加器“与”到直接地址	2	1
ANL	direct, #data	立即数“与”到直接地址	3	2
ORL	A, Rn	寄存器“或”到累加器	1	2
ORL	A, direct	直接地址“或”到累加器	2	1
ORL	A, @Ri	间接RAM“或”到累加器	1	1
ORL	A, #data	立即数“或”到累加器	2	1
ORL	direct, A	累加器“或”到直接地址	2	1
ORL	direct, #data	立即数“或”到直接地址	3	1
XRL	A, Rn	寄存器“异或”到累加器	1	2
XRL	A, direct	直接地址“异或”到累加器	2	1
XRL	A, @Ri	间接RAM“异或”到累加器	1	1
XRL	A, #data	立即数“异或”到累加器	2	1

XRL	direct, A	累加器“异或”到直接地址	2	1
XRL	direct, #data	立即数“异或”到直接地址	3	1
CLR	A	累加器清零	1	2
CPL	A	累加器求反	1	1
RL	A	累加器循环左移	1	1
RLC	A	带进位累加器循环左移	1	1
RR	A	累加器循环右移	1	1
RRC	A	带进位累加器循环右移	1	1
SWAP	A	累加器高、低4 位交换	1	1
(控制转移类指令)				
JMP	@A+DPTR	相对DPTR 的无条件间接转移	1	2
JZ	rel	累加器为0 则转移	2	2
JNZ	rel	累加器为1 则转移	2	2
CJNE	A, direct, rel	比较直接地址和累加器, 不相等转移	3	2
CJNE	A, #data, rel	比较立即数和累加器, 不相等转移	3	2
CJNE	Rn, #data, rel	比较寄存器和立即数, 不相等转移	2	2
CJNE	@Ri, #data, rel	比较立即数和间接RAM, 不相等转移	3	2
DJNZ	Rn, rel	寄存器减1, 不为0 则转移	3	2
DJNZ	direct, rel	直接地址减1, 不为0 则转移	3	2
NOP		空操作, 用于短暂延时	1	1
ACALL	add11	绝对调用子程序	2	2
LCALL	add16	长调用子程序	3	2
RET		从子程序返回	1	2
RETI		从中断服务子程序返回	1	2

AJMP	add11	无条件绝对转移	2	2
LJMP	add16	无条件长转移	3	2
SJMP	rel	无条件相对转移	2	2
(布尔指令)				
CLR	C	清进位位	1	1
CLR	bit	清直接寻址位	2	1
SETB	C	置位进位位	1	1
SETB	bit	置位直接寻址位	2	1
CPL	C	取反进位位	1	1
CPL	bit	取反直接寻址位	2	1
ANL	C, bit	直接寻址位“与”到进位位	2	2
ANL	C, /bit	直接寻址位的反码“与”到进位位	2	2
ORL	C, bit	直接寻址位“或”到进位位	2	2
ORL	C, /bit	直接寻址位的反码“或”到进位位	2	2
MOV	C, bit	直接寻址位传送到进位位	2	1
MOV	bit, C	进位位传送到直接寻址	2	2
JC	rel	如果进位位为1 则转移	2	2
JNC	rel	如果进位位为0 则转移	2	2
JB	bit, rel	如果直接寻址位为1 则转移	3	2
JNB	bit, rel	如果直接寻址位为0 则转移	3	2
JBC	bit, rel	直接寻址位为1 则转移并清除该位	2	2

(伪指令)				

ORG	指明程序的开始位置	
DB	定义数据表	
DW	定义16 位的地址表	
EQU	给一个表达式或一个字符串起名	
DATA	给一个8 位的内部RAM 起名	
XDATA	给一个8 位的外部RAM 起名	
BIT	给一个可位寻址的位单元起名	
END	指出源程序到此为止	

(指令中的符号标识)		
Rn	工作寄存器R0-R7	
Ri	工作寄存器R0 和R1	
@Ri	间接寻址的8 位RAM 单元地址 (00H-FFH)	
#data8	8 位常数	
#data16	16 位常数	
addr16	16 位目标地址，能转移或调用到64KROM 的任何地方	
addr11	11 位目标地址，在下条指令的2K 范围内转移或调用	
Rel	8 位偏移量，用于SJMP 和所有条件转移指令，范围-128~+127	
Bit	片内RAM 中的可寻址位和SFR 的可寻址位	
Direct	直接地址，范围片内RAM 单元 (00H-7FH) 和80H-FFH	
\$	指本条指令的起始位置	