

80C51 汇编指令集

由 jyfjk 于 2007-12-12 整理

按功能排列

1. 数据传送指令				
助记符	说明	字节	周期	代码
MOV A, Rn	寄存器送 A	1	1	E8--EF
MOV A, data	直接字节送 A	2	1	E5
MOV A, @Ri	间接 RAM 送 A	1	1	E6--E7
MOV A, #data	立接数送 A	2	1	74
MOV Rn, A	A 送寄存器	1	1	F8--FF
MOV Rn, data	直接数送寄存器	2	2	A8--AF
MOV Rn, #data	立即数送寄存器	2	1	78--7F
MOV data, A	A 送直接字节	2	1	F5
MOV data, Rn	寄存器送直接字节	2	1	88--8F
MOV data, data	直接字节送直接字节	3	2	85
MOV data, @Ri	间接 Rn 送直接字节	2	2	86;87
MOV data, #data	立即数送直接字节	3	2	75
MOV @Ri, A	A 送间接 Rn	1	2	F6;F7
MOV @Ri, data	直接字节送间接 Rn	1	1	A6;A7
MOV @Ri, #data	立即数送间接 Rn	2	2	76;77
MOV DPTR, #data16	16 位常数送数据指针	3	1	90
MOV C, bit	直接位送进位位	2	1	A2
MOV bit, C	进位位送直接位	2	2	92
MOVC A, @A+DPTR	A+DPTR 寻址程序存贮字节送 A	3	2	93
MOVC A, @A+PC	A+PC 寻址程序存贮字节送 A	1	2	83
MOVX A, @Ri	外部数据送 A (8 位地址)	1	2	E2;E3
MOVX A, @DPTR	外部数据送 A (16 位地址)	1	2	E0
MOVX @Ri, A	A 送外部数据 (8 位地址)	1	2	F2;F3
MOVX @DPTR, A	A 送外部数据 (16 位地址)	1	2	F0
PUSH data	直接字节进栈道, SP 加 1	2	2	C0
POP data	直接字节出栈, SP 减 1	2	2	D0
XCH A, Rn	寄存器与 A 交换	1	1	C8--CF
XCH A, data	直接字节与 A 交换	2	1	C5
XCH A, @Ri	间接 Rn 与 A 交换	1	1	C6;C7
XCHD A, @Ri	间接 Rn 与 A 低半字节交换	1	1	D6;D7

2. 逻辑运算指令				
助记符	说明	字节	周期	代码
ANL A, Rn	寄存器 <u>与</u> 到A	1	1	58—5F
ANL A, data	直接字节 <u>与</u> 到A	2	1	55
ANL A, @Ri	间接RAM <u>与</u> 到A	1	1	56;57
ANL A, #data	立即数 <u>与</u> 到A	2	1	54
ANL data, A	A <u>与</u> 到直接字节	2	1	52
ANL data, #data	立即数 <u>与</u> 到直接字节	3	2	53
ANL C, bit	直接位 <u>与</u> 到进位位	2	2	82
ANL C, /bit	直接位的反码 <u>与</u> 到进位位	2	2	B0
ORL A, Rn	寄存器 <u>或</u> 到A	3	2	48—4F
ORL A, data	直接字节 <u>或</u> 到A	2	1	45
ORL A, @Ri	间接RAM <u>或</u> 到A	2	1	46;47
ORL A, #data	立即数 <u>或</u> 到A	2	1	44
ORL data, A	A <u>或</u> 到直接字节	2	1	42
ORL data, #data	立即数 <u>或</u> 到直接字节	2	1	43
ORL C, bit	直接位 <u>或</u> 到进位位	2	2	72
ORL C, /bit	直接位的反码 <u>或</u> 到进位位	2	2	A0
XRL A, Rn	寄存器 <u>异或</u> 到A	3	2	68—6F
XRL A, data	直接字节 <u>异或</u> 到A	1	1	65
XRL A, @Ri	间接RAM <u>异或</u> 到A	2	1	66;67
XRL A, #data	立即数 <u>异或</u> 到A	1	1	64
XRL data, A	A <u>异或</u> 到直接字节	2	1	62
XRL data, #data	立即数 <u>异或</u> 到直接字节	2	1	63
SETB C	进位位置 1	1	1	D3
SETB bit	直接位置 1	2	1	D2
CLR A	A 清 0	3	2	E4
CLR C	进位位清 0	1	1	C3
CLR bit	直接位清 0	2	1	C2
CPL A	A 求反码	1	1	F4
CPL C	进位位取反	1	1	B3
CPL bit	直接位取反	2	1	B2
RL A	A 循环左移一位	1	1	23
RLC A	A 带进位左移一位	1	1	33

RR A	A 右移一位	1	1	03
RRC A	A 带进位右移一位	1	1	13
SWAP A	A 半字节交换	1	1	C4
3. 算术运算指令				
助记符	说明	字节	周期	代码
ADD A, Rn	寄存器加到 A	1	1	28—2F
ADD A, data	直接字节加到 A	2	1	25
ADD A, @Ri	间接 RAM 加到 A	1	1	26;27
ADD A, #data	立即数加到 A	2	1	24
ADDC A, Rn	寄存器带进位加到 A	1	1	38—3F
ADDC A, data	直接字节带进位加到 A	2	1	35
ADDC A, @Ri	间接 RAM 带进位加到 A	1	1	36;37
ADDC A, #data	立即数带进位加到 A	2	1	34
SUBB A, Rn	从 A 中减去寄存器和进位	1	1	98—9F
SUBB A, data	从 A 中减去直接字节和进位	2	1	95
SUBB A, @Ri	从 A 中减去间接 RAM 和进位	1	1	96;97
SUBB A, #data	从 A 中减去立即数和进位	2	1	94
INC A	A 加 1	1	1	04
INC Rn	寄存器加 1	1	1	08—0F
INC data	直接字节加 1	2	1	05
INC @Ri	间接 RAM 加 1	1	1	06;07
INC DPTR	数据指针加 1	1	2	A3
DEC A	A 减 1	1	1	14
DEC Rn	寄存器减 1	1	1	18—1F
DEC data	直接字节减 1	2	1	15
DEC @Ri	间接 RAM 减 1	1	1	16;17
MUL AB	A 乘 B	1	4	A4
DIV AB	A 被 B 除	1	4	84
DA A	A 十进制调整	1	1	D4
3. 转移指令				
助记符	说明	字节	周期	代码
AJMP addr 11	绝对转移	2	2	*1
LJMP addr 16	长转移	3	2	02
SJMP rel	短转移	2	2	80

JMP @A+DPTR	相对于 DPTR 间接转移	1	2	73
JZ rel	若 A=0 则转移	2	2	60
JNZ rel	若 A≠0 则转移	2	2	70
JC rel	若 C=1 则转移	2	2	40
JNC rel	若 C≠1 则转移	2	2	50
JB bit, rel	若直接位=1 则转移	3	2	20
JNB bit, rel	若直接位=0 则转移	3	2	30
JBC bit, rel	若直接位=1 则转移且清除	3	2	10
CJNE A, data, rel	直接数与 A 比较, 不等转移	3	2	B5
CJNE A, #data, rel	立即数与 A 比较, 不等转移	3	2	B4
CJNE @Ri, #data, rel	立即数与间接 RAM 比较, 不等转移	3	2	B6;B7
CJNE Rn, #data, rel	立即数与寄存器比较不等转移	3	2	B8—BF
DJNZ Rn, rel	寄存器减 1 不为 0 转移	2	2	D8—DF
DJNZ data, rel	直接字节减 1 不为 0 转移	3	2	D5
ACALL addr 11	绝对子程序调用	2	2	*1
LCALL addr 16	子程序调用	3	2	12
RET	子程序调用返回	1	2	22
RETI	中断程序调用返回	1	2	32
NOP	空操作	1	1	00

按字母排列

1. 数据传送指令				
助记符	说明	字节	周期	代码
ACALL addr 11	绝对子程序调用	2	2	*1
ADD A, Rn	寄存器加到 A	1	1	28—2F
ADD A, data	直接字节加到 A	2	1	25
ADD A, @Ri	间接 RAM 加到 A	1	1	26;27
ADD A, #data	立即数加到 A	2	1	24
ADDC A, Rn	寄存器带进位加到 A	1	1	38—3F
ADDC A, data	直接字节带进位加到 A	2	1	35
ADDC A, @Ri	间接 RAM 带进位加到 A	1	1	36;37
ADDC A, #data	立即数带进位加到 A	2	1	34
AJMP addr 11	绝对转移	2	2	*1
ANL A, Rn	寄存器与到 A	1	1	58—5F
ANL A, data	直接字节与到 A	2	1	55

ANL A, @Ri	间接RAM与到A	1	1	56;57
ANL A, #data	立即数与到A	2	1	54
ANL data, A	A与到直接字节	2	1	52
ANL data, #data	立即数与到直接字节	3	2	53
ANL C, bit	直接位与到进位位	2	2	82
ANL C, /bit	直接位的反码与到进位位	2	2	B0
CJNE A, data, rel	直接数与 A 比较, 不等转移	3	2	B5
CJNE A, #data, rel	立即数与 A 比较, 不等转移	3	2	B4
CJNE @Ri, #data, rel	立即数与间接 RAM 比较, 不等转移	3	2	B6;B7
CJNE Rn, #data, rel	立即数与寄存器比较不等转移	3	2	B8—BF
CLR A	A 清 0	3	2	E4
CLR C	进位位清 0	1	1	C3
CLR bit	直接位清 0	2	1	C2
CPL A	A 求反码	1	1	F4
CPL C	进位位取反	1	1	B3
CPL bit	直接位取反	2	1	B2
DA A	A 十进制调整	1	1	D4
DEC A	A 减 1	1	1	14
DEC Rn	寄存器减 1	1	1	18—1F
DEC data	直接字节减 1	2	1	15
DEC @Ri	间接 RAM 减 1	1	1	16;17
DIV AB	A 被 B 除	1	4	84
DJNZ Rn, rel	寄存器减 1 不为 0 转移	2	2	D8—DF
DJNZ data, rel	直接字节减 1 不为 0 转移	3	2	D5
INC A	A 加 1	1	1	04
INC Rn	寄存器加 1	1	1	08—0F
INC data	直接字节加 1	2	1	05
INC @Ri	间接 RAM 加 1	1	1	06;07
INC DPTR	数据指针加 1	1	2	A3
JB bit, rel	若直接位 = 1 则转移	3	2	20
JBC bit, rel	若直接位 = 1 则转移且清除	3	2	10
JC rel	若 C = 1 则转移	2	2	40
JMP @A+DPTR	相对于 DPTR 间接转移	1	2	73
JNB bit, rel	若直接位 = 0 则转移	3	2	30

JNC rel	若 C≠1 则转移	2	2	50
JNZ rel	若 A≠0 则转移	2	2	70
JZ rel	若 A=0 则转移	2	2	60
LCALL addr 16	子程序调用	3	2	12
LJMP addr 16	长转移	3	2	02
MOV A, Rn	寄存器送 A	1	1	E8--EF
MOV A, data	直接字节送 A	2	1	E5
MOV A, @Ri	间接 RAM 送 A	1	1	E6--E7
MOV A, #data	立接数送 A	2	1	74
MOV Rn, A	A 送寄存器	1	1	F8--FF
MOV Rn, data	直接数送寄存器	2	2	A8--AF
MOV Rn, #data	立即数送寄存器	2	1	78--7F
MOV data, A	A 送直接字节	2	1	F5
MOV data, Rn	寄存器送直接字节	2	1	88--8F
MOV data, data	直接字节送直接字节	3	2	85
MOV data, @Ri	间接 Rn 送直接字节	2	2	86;87
MOV data, #data	立即数送直接字节	3	2	75
MOV @Ri, A	A 送间接 Rn	1	2	F6;F7
MOV @Ri, data	直接字节送间接 Rn	1	1	A6;A7
MOV @Ri, #data	立即数送间接 Rn	2	2	76;77
MOV DPTR, #data	16 位常数送数据指针	3	1	90
MOV C, bit	直接位送进位位	2	1	A2
MOV bit, C	进位位送直接位	2	2	92
MOVC A, @A+DPTR	A+DPTR 寻址程序存贮字节送 A	3	2	93
MOVC A, @A+PC	A+PC 寻址程序存贮字节送 A	1	2	83
MOVX A, @Ri	外部数据送 A (8 位地址)	1	2	E2;E3
MOVX A, @DPTR	外部数据送 A (16 位地址)	1	2	E0
MOVX @Ri, A	A 送外部数据 (8 位地址)	1	2	F2;F3
MOVX @DPTR, A	A 送外部数据 (16 位地址)	1	2	F0
MUL AB	A 乘 B	1	4	A4
NOP	空操作	1	1	00
ORL A, Rn	寄存器 <u>或到</u> A	3	2	48--4F
ORL A, data	直接字节 <u>或到</u> A	2	1	45
ORL A, @Ri	间接RAM <u>或到</u> A	2	1	46;47

ORL A, #data	立即数 <u>或到</u> A	2	1	44
ORL data, A	A <u>或到</u> 直接字节	2	1	42
ORL data, #data	立即数 <u>或到</u> 直接字节	2	1	43
ORL C, bit	直接位 <u>或到</u> 进位位	2	2	72
ORL C, /bit	直接位的反码 <u>或到</u> 进位位	2	2	A0
POP data	直接字节出栈, SP 减 1	2	2	D0
PUSH data	直接字节进栈道, SP 加 1	2	2	C0
RET	子程序调用返回	1	2	22
RETI	中断程序调用返回	1	2	32
RL A	A 循环左移一位	1	1	23
RLC A	A 带进位左移一位	1	1	33
RR A	A 右移一位	1	1	03
RRC A	A 带进位右移一位	1	1	13
SETB C	进位位置 1	1	1	D3
SETB bit	直接位置 1	2	1	D2
SJMP rel	短转移	2	2	80
SUBB A, Rn	从 A 中减去寄存器和进位	1	1	98—9F
SUBB A, data	从 A 中减去直接字节和进位	2	1	95
SUBB A, @Ri	从 A 中减去间接 RAM 和进位	1	1	96;97
SUBB A, #data	从 A 中减去立即数和进位	2	1	94
SWAP A	A 半字节交换	1	1	C4
XCH A, Rn	寄存器与 A 交换	1	1	C8—CF
XCH A, data	直接字节与 A 交换	2	1	C5
XCH A, @Ri	间接 Rn 与 A 交换	1	1	C6;C7
XCHD A, @Ri	间接 Rn 与 A 低半字节交换	1	1	D6;D7
XRL A, Rn	寄存器 <u>异或到</u> A	3	2	68—6F
XRL A, data	直接字节 <u>异或到</u> A	1	1	65
XRL A, @Ri	间接RAM <u>异或到</u> A	2	1	66;67
XRL A, #data	立即数 <u>异或到</u> A	1	1	64
XRL data, A	A <u>异或到</u> 直接字节	2	1	62
XRL data, #data	立即数 <u>异或到</u> 直接字节	2	1	63
助记符	说明	字节	周期	代码

注: Rn 表示寄存器 R0—R7。

@Ri 表示 R0、R1。RS1、RS0=00 时为 1 区寄存器,

RS1、RS0=01 时为 2 区寄存器，
 RS1、RS0=10 时为 3 区寄存器，
 RS1、RS0=11 时为 4 区寄存器。

data 表示直接字节位。

#data 表示立即常数。

addr 11 表示 11 位地址。addr 16 表示 16 位地。

bit 表示直接位，rel 表示转移步数。

RAM 中的储存单元：

7FH	7F	7E	7D	7C	7B	7A	79	78	78H
	一般数据存放区或堆栈区								
37H	37	36	35	34	33	32	31	30	30H
2FH	7F	7E	7D	7C	7B	7A	79	78	2FH
2EH	77	76	75	74	73	72	71	70	2EH
2DH	1R7	1R6	1R5	1R4	1R3	1R2	1R1	1R0	2DH
2CH	67	66	65	64	63	62	61	60	2CH
2BH	5F	5E	5D	5C	5B	5A	59	58	2BH
2AH	57	56	55	54	53	52	51	50	2AH
29H	4F	4E	4D	4C	4B	4A	49	48	29H
28H	47	46	45	44	43	42	41	40	28H
27H	3F	3E	3D	3C	3B	3A	39	38	27H
26H	37	36	35	34	33	32	31	30	26H
25H	2F	2E	2D	2C	2B	2A	29	28	25H
24H	27	26	25	24	23	22	21	20	24H
23H	1F	1E	1D	1C	1B	1A	19	18	23H
22H	17	16	15	14	13	12	11	10	22H
21H	0F	0E	0D	0C	0B	0A	09	08	21H
20H	07	06	05	04	03	02	01	00	20H
1FH	3R7	3R6	3R5	3R4	3R3	3R2	3R1	3R0	18H
10H	2R7	2R6	2R5	2R4	2R3	2R2	2R1	2R0	10H
0FH	1R7	1R6	1R5	1R4	1R3	1R2	1R1	1R0	08H
07H	0R7	0R6	0R5	0R4	0R3	0R2	0R1	0R0	00H

注：表示可位寻址，表示寄存器单元。

SFR 存储器分配图：

F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
E0H	E7	E6	E5	E4	E3	E2	E1	E0	Acc
PWS	CY	AC	F0	RS1	RS0	OV	-	P	
D0H	D7	D6	D5	D4	D3	D2	D1	D0	PWS
CDH									TH2*
CCH									TL2*
CBH									RCAP2H*
CAH									RCAP2L*
C8H									T2CON*
IP	-	-	-	PS	PT1	PX1	PT0	PX0	
B8H	-	-	-	BC	BB	BA	B9	B8	IP
B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
IE	EA	-	ET2*	ES	ET1	EX1	ET0	EX0	
A8H	AF	-	-	AC	AB	AA	A9	A8	IE
A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
99H									SBUF
SCON	SM0	SM1	SM2	REN	TB8	RB8	T1	R1	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
8DH									TH1
8CH									TH0
8BH									TL1
8AH									TL0
89H									TMOD
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
87H	SMOD	-	-	-	GF1	GF0	PD	IDL	PCON
83H									DPH
82H									DPL
81H									SP
80H	87	86	85	84	83	82	81	80	P0

注：表示可位寻址。*为 8052 才有的功能。

PSW 程序状态字 (DOH)

PWS	CY	AC	F0	RS1	RS0	OV	-	P	
DOH	D7	D6	D5	D4	D3	D2	D1	D0	PWS

CY: 进位标志。

AC: 辅助进位标志。

F0: 一般用途。

RS1、RS0: 寄存器库选区位。

OV: 溢出标志位。

P: 奇偶校验标志。

IE 中断使能寄存器 (A8H) (=1 时允许中断, =0 时关中断)

IE	EA	-	ET2*	ES	ET1	EX1	ET0	EX0	
A8H	AF	-	-	AC	AB	AA	A9	A8	IE

EA: EA=0 禁止所有中断, EA=1 开中断。

ET2: ET2 溢出或选取中断的使能。

ES: 串口 1 的中断使能。

ET1: 定时中断 1 的使能。

EX1: 外部中断 1 的使能。

ET0: 定时中断 0 的使能。

EX0: 外部中断 0 的使能。

IP 中断优先级寄存器 (B8H) (=1 时优先级高, =0 时优先级低)

IP	-	-	-	PS	PT1	PX1	PT0	PX0	
B8H	-	-	-	BC	BB	BA	B9	B8	IP

PT2: 定义定时器 2 的中断优先层次。

PS: 定义串口 1 的中断优先层次。

PT1: 定义定时/计数器 1 的中断优先层次。

PX1: 定义外部中断 1 的中断优先层次。

PT0: 定义定时/计数器 0 的中断优先层次。

PX0: 定义外部中断 0 的中断优先层次。

TMOD 定时器模式控制寄存器 (89H)

89H	GATE1	C1/T1	M11	M10	GATE0	C0/T0	M01	M00	TMOD
-----	-------	-------	-----	-----	-------	-------	-----	-----	------

GATE1、GATE0: 当 GATE1、GATE0=1 时, 定时/计数器的运行受外部电平的控制, P3.2 控制 T0, P3.3 控制 T1。

当 GATE0=0 时, 定时/计数器的运行不受外部电平的控制, 允许软件控制位 TR0、TR1 启动定时/计数器的运行。

C1/T1、C0/T0：计时或计数选择位，当 C1/T1、C0/T0=0 时为计时，当 C1/T1、C0/T0=1 时为计数。对 P3.4、P3.5 的外部下降沿信号计数。

M1、M0：工作模式选择

M1	M0	工作模式	功 能
0	0	0	13 位计时器
0	1	1	16 位计时、计数器
1	0	2	8 位自动装入
1	1	3	TL0 为 8 位计时/计数器，由 T0 的控制位控制 TH0 为另一个 8 位计时/计数器，由 T1 的控制位控制 定时/计数 1 停止工作

TCON 计时控制寄存器（88H）

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
88H	8F	8E	8D	8C	8B	8A	89	88	TCON

TF1：计时器 1 溢出标志位，当计时器 1 溢出，时此位=1，发出中断请求，然后硬件自动清 0。

TR1：计时器 1 的控制位，TR1=1 时，计时开始，TR1=1 时则停止。

TF0：计时器 0 溢出标志位，当计时器 0 溢出，时此位=1，发出中断请求，然后硬件自动清 0。

TR0：计时器 0 的控制位，TR0=1 时，计时开始，TR0=1 时则停止。

IE1：外部中断 1 的请求标志位，当外部中断有请求时，此位=1，发出中断请求，然后硬件自动清 0。

IT1：外部中断 1 的中断类型控制，当 IT1=1 时，边沿触发方式，下降沿触发有效，当 IT1=0 时，电平触发方式，低电平触发有效。P3.3 信号触发。

IE0：外部中断 0 的请求标志位，当外部中断有请求时，此位=1，发出中断请求，然后硬件自动清 0。

IT0：外部中断 0 的中断类型控制，当 IT0=1 时，边沿触发方式，下降沿触发有效，当 IT0=0 时，电平触发方式，低电平触发有效。P3.2 信号触发

中断入口地址：	中断源	矢量地址
	IE0	0003H
	TF0	000BH
	IE1	0013H
	TF1	001BH
	RI&TI	0023H
	TF2&EXF2	002BH

SCON 串行控制寄存器 (98H)

SCON	SM0	SM1	SM2	REN	TB8	RB8	T1	R1	
98H	9F	9E	9D	9C	9B	9A	99	98	SCON

SM0: 串行模式选择。

SM1: 串行口选择。

SM2: 模式 0 时 SM2 必需为 0, 模式 1 时, 当接收到的停止位不正确时 RI 不操作, 在模式 2 与 3 时, 当 SM2=1 时, 接收到的第 9 个数据为 0 时, RI 不操作。

REN: 由软件设置, REN=1 时, 接收串行数据输入。

TB8: 在模式 2 与 3 时, 传送的第 9 位数据由软件控制。

RB8: 在模式 0 时此位无效。

在模式 1 时, 如 SM2=0, 则 RB8 接收停止位数据。

在模式 2 与 3 时, 传送的第 9 位数据放在这里。

TI: 传送中断标志位, 在模式 0 时第 8 位数据结束时, 硬件自动将此位置 1。其它模式时, 在停止位开始时设置此位为 1, 需由软件清除。

RI: 接收中断标志位, 在模式 0 时, 第 8 位结束时, 硬件会将此位置 1, 在其它模式时, 在停止位的一半时间内由硬件会将此位置 (见 SM2), 此位需由软件清除。

PCON 电源控制寄存器 (87H)

87H	SMOD	-	-	-	GF1	GF0	PD	IDL	PCON
-----	------	---	---	---	-----	-----	----	-----	------

SMOD: 双倍波特率控制位, 当 SMOD=1 时为双波特率发送。

GF1、GF0: 一般用途。

PD: 电源下降控制位, PD=1 时为电源下降模式。

IDL: IDLE 模式控制位, IDL=1 时为 IDLE 模式。

本人 QQ 号: 372130518

单片机与protel交流QQ群:

http://group.qq.com/group_index.shtml?groupid=6792948

MCU电子论坛网址: <http://chinamcu.70bb.net/>