

51 指令系统

内部 RAM 传送指令

指令名称	指令格式	功能	字节数	周期
以 A 为目的操作数	MOV A,RN	A <- RN	1	1
	MOV A,direct	A <- (direct)	2	1
	MOV A,@Ri	A <- (Ri)	1	1
	MOV A,#data	A <- #data	2	1
以 RN 为止目的操作数	MOV RN,A	RN <- A	1	1
	MOV RN,direct	RN <- (direct)	2	2
	MOV RN,#data	RN <- #data	2	1
以直接地址为目的的操作数	MOV direct,A	(direct) <- A	2	1
	MOV direct,RN	(direct) <- RN	2	2
	MOV direct, direct	(direct) <- (direct)	2	2
	MOV direct, @Ri	(direct) <- (Ri)	3	2
	MOV direct, #data	(direct) <- #data	2	2
以寄存储间接地址为目的的操作数	MOV @Ri, A	(Ri) <- A	2	2
	MOV @Ri, direct	(Ri) <- (direct)	2	2
	MOV @Ri, #data	(Ri) <- #data	1	1
16 位数据传输	MOV DPTR,#data16	DPH <- #data(15-8) DPL <- #data(7-0)	3	2

外部传送指令

指令格式	功能	字节数	周期
MOVX A,@RI	A <- (Ri)	1	2
MOVX A,@DPTR	A <- (DPTR)	2	2
MOVX @RI,A	(Ri) <- A	1	2
MOVX @DPTR,A	(DPTR) <- A	2	2

程序存储器数据传送指令

指令格式	功能	字节数	周期
MOVC A,@A+DPTR	A <- (A+DPTR)	1	2
MOVC A,@A+PC	A <- (A+PC)	1	2
指令格式	功能	字节数	周期
PUSH direct	SP <- SP+1, (SP) <- (direct)	2	2
POP direct	(direct) <- (SP), SP <- SP-1	2	2

字节交换指令

指令格式	功能	字节数	周期
XCH A,RN	A <--> RN	1	1
XCH A, direct	A <--> (direct)	1	1
XCH A,@RI	A <--> (RI)	1	1
SWAP A	A(0-3) <--> A(4-7)	1	1
XCHD A,@RI	A(0-3) <--> (RI)(0-3)	1	1

算术运算指令-加法

指令名称	指令格式	功能	字节数	周期
不带进位加法	ADD A,RN	A <- A+RN	1	1
	ADD A,@RI	A <- A+(RI)	1	1

	ADD A, direct	A <- A+(direct)	2	1
	ADD A, #data	A <- A+data	2	1
带进位加法	ADDC A, RN	A <- A+RN+CY	1	1
	ADDC A, @RI	A <- A+(RI) +CY	1	1
	ADDC A, direct	A <- A+(direct) +CY	2	1
	ADDC A, #data	A <- A+data+CY	2	1

算术运算指令-减法

指令格式	功能	字节数	周期
SUBB A,RN	A <- A-RN-CY	1	1
SUBB A,@RI	A <- A-(RI) -CY	2	1
SUBB A, direct	A <- A-(direct) -CY	1	1
SUBB A, #data	A <- A-data-CY	2	1

指令名称	指令格式	功能	字节数	周期
加 1 指令	INC A	A <- A+1	1	1
	INC RN	RN <- RN +1	1	1
	INC @RI	(RI) <- (RI)+1	1	1
	INC direct	(direct) <- (direct)+1	2	1
	INC DPTR	DPTR <- DPTR +1	1	2
减 1 指令	DEC A	A <- A-1	1	1
	DEC RN	RN <- RN -1	1	1
	DEC @RI	(RI) <- (RI)-1	1	1
	DEC direct	(direct) <- (direct)-1	2	1

算术运算指令-十进制调整指令

指令格式	功能	字节数	周期
DA A	将 A 中内容转换为 BCD 码	1	1

方法：将累加器 A 中的低半字节的值大于 9 或 AC=1 时低半字节加 6，当高半字节大于 9 或 C=1 时，高半字节加 6，

算术运算指令-乘除法指令

指令格式	功能	字节数	周期
MUL AB	B A <- A*B	1	4
DIV AB	A <- A/B，（B 放余数）	1	4

逻辑运算类指令

指令名称	指令格式	功能	字节数	周期
逻辑与指令	ANL A,RN	A <- A^RN	1	1
	ANL A,@RI	A <- A^(RI)	1	1
	ANL A, direct	A <- A^(direct)	2	1
	ANL A,# data	A <- A^ data	2	1
	ANL direct, # data	(direct) <- (direct)^RN	2	1
	ANL direct,A	(direct) <- (direct)^A	3	2
逻辑或指令	ORL A,RN	A <- AVRN	1	1
	ORL A,@RI	A <- AV(RI)	1	1
	ORL A, direct	A <- AV (direct)	2	1
	ORL A,# data	A <- AV data	2	1
	ORL direct, # data	(direct) <- (direct)VRN	2	1
	ORL direct,A	(direct) <- (direct)VA	3	2
逻辑异或指令	XOR A,RN	A <- A✖RN	1	1

	XOR A,@RI	A <- AV※(RI)	1	1
	XOR A, direct	A <- A※(direct)	2	1
	XOR A,# data	A <- A※ data	2	1
	XOR direct, # data	(direct) <- (direct) ※RN	2	1
	XOR direct,A	(direct) <- (direct) ※A	3	2
累加器 A 清 0 与取反	CPL A	A <- ~A	1	1
	CLR A	A <- 0	1	1
空操作	NOP	空操作	1	1
不带进位循环移位指令	RL A	An+1 <- An(n=0--6),A0 <- A7	1	1
	RR A	An <- An+1 (n=0--6), A7 <- A0	1	1
带进位循环移位指令	RLC A	An+1 <- An(n=0--6), CY <- A7, A0 <- CY	1	1
	RRC A	An <- An+1 (n=0--6), A7 <- CY, CY <- A0	1	1

控制转移类指令

指令名称	指令格式	功能	字节数	周期
无条件转移类指令	LJMP addr16	PC <- PC+3,PC <- addr16	3	2
	AJMP addr11	PC <- PC+3,PC <- addr11	2	2
	SJMP rel	PC <- PC+2+ rel	2	2
	JMP @A+DPTR	PC <- (A+DPTR)	1	2
累加器判 0 转发指令	JZ rel	若 A=0 则 PC <- PC+2+ rel 若 A≠0 则 PC <- PC+2	2	2
	JNZ rel	若 A≠0 则 PC <- PC+2+ rel 若 A=0 则 PC <- PC+2	2	2
比较转移指令	CJNE A, direct, rel	若 A ≠ (direct) 则 PC <- PC+3+ rel 若 A=(direct) 则 PC <- PC+3 若 A ≥ (direct), 则 CY <- 0	3	2
	CJNE A, #data, rel	若 A≠data 则 PC <- PC+3+ rel	3	2
		若 A= data 则 PC <- PC+3 若 A≥data,则 CY <- 0		
	CJNE RN, #data, rel	若 RN ≠ data 则 PC <- PC+3+ rel 若 RN = data 则 PC <- PC+3 若 RN≥data,则 CY <- 0	3	2
	CJNE @Ri, #data, rel	若 (Ri) ≠ data 则 PC <- PC+3+ rel 若(Ri)= data 则 PC <- PC+3 若(Ri)≥data,则 CY <- 0	3	2
减 1 不为 0 转移指令	DJNZ RN, rel	若 RN-1 ≠ 0 , 则 PC <- PC+2+ rel 若 RN-1=0, 则 PC <- PC+2	3	2
	DJNZ direct, rel	若 (direct)-1 ≠ 0 , 则 PC <- PC+2+ rel 若 (direct)-1=0 , 则 PC <- PC+2	3	2
子程序调用和返回指令	LCALL addr16	PC <- PC+3,PC 值大压栈, PC <- addr16	3	2
	ACALL addr11	PC <- PC+1,PC 值大压栈, PC10--0 <- addr11	2	2
	RET	程序返回	1	2

	RETI	中断返回	1	2
--	------	------	---	---

布尔操作类指令

指令名称	指令格式	功能	字节数	周期
位数据传送指令	MOV C,bit	CY <- bit	2	1
	MOV bit,c	bit <- CY	2	1
位状态控制指令	CLR C	CY <- 0	1	1
	CLR bit	bit <- 0	2	1
	SETB C	CY <- 1	1	1
	SETB bit	bit <- 1	2	1
位逻辑操作指令	ANL C, bit	CY <- CY^bit	2	2
	ANL C, /bit	CY <- CY^~bit	2	2
	ORL C, bit	CY <- CY^bit	2	2
	ORL C, /bit	CY <- CY^~bit	2	2
	CPL C	CY <- ~CY	1	2
	CPL bit	bit <- ~ bit	2	2
位条件转移指令	JC rel	若 CY=0,则转移 PC <- PC+2+rel 否则 PC <- PC+2	2	2
	JNC rel	若 CY≠0,则转移 PC <- PC+2+rel	2	2
		否则 PC <- PC+2		
	JB bit, rel	若 bit =1,则转移 PC <- PC+2+rel 否则 PC <- PC+2	3	2
	JNB bit, rel	若 bit≠1,则转移 PC <- PC+2+rel 否则 PC <- PC+2	3	2
	JNC bit, rel	若 bit =1,则转移 PC <- PC+2+rel, 且 bit <- 0, 否则 PC <- PC+2	3	2

伪指令

指令格式	功能
ORG XXXXH	指定本条指令下面的程序或表格数据的起始地址
END	指示源程序到此结束
EQU	把“项”赋给“字符名称”
DATA	数据地址赋值命 <-
DB XXH,XXH..	定义字节指令
DW XXXXH,XXXXH,..	定义字指令
DS	预留存储空间指令
BIT	位地址符指令