

TẬP LỆNH 89C51

Dạng lệnh		Mô tả	Ví dụ		Số MC	Độ dài
LỆNH LUẬN LÝ TÁC ĐỘNG TRÊN BIT						
CLR	C	Gán cờ C=0	CLR C		1	1
CLR	BIT	Gán BIT =0	CLR P1.0		1	2
SETB	C	Gán cờ C=1	SETB C		1	1
SETB	BIT	Gán BIT =1	SETB P1.0		1	2
CPL	C	Đảo giá trị cờ C. Nếu C=1 thì sau lệnh này C=0 và ngược lại.	CPL C		1	1
CPL	BIT	Đảo giá trị BIT. Nếu BIT=1 thì sau lệnh này BIT=0 và ngược lại.	CPL P1.0		1	2
ANL	C, BIT	Phép AND giữa cờ C và BIT. Kết quả cất vào C.	ANL C,P1.0		2	2
ANL	C, BIT	Phép AND giữa cờ C và bù của BIT. Kết quả cất vào C.	ANL C, /P1.0		2	2
ORL	C, BIT	Phép OR giữa cờ C và BIT. Kết quả cất vào C.	ORL C,P1.7		2	2
ORL	C,/BIT	Phép OR giữa cờ C và bù của BIT. Kết quả cất vào C.	ORL C,/P1.7		2	2
MOV	C,BIT	Gán cờ C=BIT	MOV C,ACC.7		1	2
MOV	BIT,C	Gán BIT bằng giá trị cờ C	MOV ACC.7,C		2	2

JC	nhãn	Nhảy đến nhãn nếu cờ C =1	JC	nhohon	2	2
JNC	nhãn	Nhảy đến nhãn nếu cờ C=0	JNC	nhohon	2	2
JB	BIT, nhãn	Nhảy đến nhãn nếu BIT = 1	JB	P1.0,start	2	3
JNB	BIT, nhãn	Nhảy đến nhãn nếu BIT = 0	JNB	P1.0,start	2	3
JBC	BIT, nhãn	Nhảy đến nhãn nếu BIT = 1 và sau đó gán BIT = 0	JBC	P1.0,main	2	3
LỆNH RẼ NHÁNH CHƯƠNG TRÌNH						
ACALL	nhãn	Gọi thực thi chương trình con có tên “nhãn”. Sau khi thực thi chương trình con, VXL sẽ thực hiện câu lệnh ngay phía sau câu lệnh gọi này. Cần cất các giá trị thanh ghi được sử dụng trong chương trình con bằng các lệnh PUSH/POP	ACALL	DELAY	2	2
LCALL	nhãn		LCALL	DELAY	2	3
CALL	nhãn		CALL	DELAY	X	X
RET		Thoát khỏi chương trình con.	RET		2	1
RETI		Thoát khỏi chương trình phục vụ ngắt	RETI		2	1
AJMP	nhãn	Nhảy đến nhãn. Số chu kỳ máy của lệnh JMP được quyết định tại thời điểm dịch phụ thuộc vào khoảng cách từ điểm gọi đến nhãn. Lệnh nhảy này không cần điều kiện.	AJMP	Main	2	2
LJMP	nhãn		LJMP	Main	2	3

SJMP	nhãn		SJMP Main	2	2
JMP	nhãn		JMP Main	X	X
JMP	@A+DPTR	Nhảy đến vị trí cách vị trí được trỏ bởi DPTR một khoảng là A.	JMP @A+DPTR	2	1
JZ	nhãn	Nhảy đến nhãn nếu thanh ghi A=0	JZ Abang0	2	2
JNZ	nhãn	Nhảy đến nhãn nếu thanh ghi A khác 0	JNZ Akhac0	2	2
CJNE	A,direct,nhãn	So sánh và nhảy đến nhãn nếu giá trị thanh ghi A không bằng giá trị ô nhớ direct	CJNE A,B,AkhacB	2	3
CJNE	A,#data,nhãn	So sánh và nhảy đến nhãn nếu giá trị thanh ghi A không bằng giá trị data	CJNE A,#20,Akhac20	2	3
CJNE	Rn,#data,nhãn	So sánh và nhảy đến nhãn nếu giá trị thanh ghi Rn không bằng giá trị data. Trong đó, Rn là một trong các thanh ghi R0, R1, R2, ..., R7	CJNE R5,#20,R5khac20	2	3
CJNE	@Ri,#data,nhãn	So sánh và nhảy đến nhãn nếu giá trị ô nhớ được trỏ bởi Ri (R0 hay R1) không bằng giá trị data.vd: R0=40h, so sánh giá trị ô nhớ 40h với 20 ta dùng câu lệnh bên	CJNE @R0,#20,me mkhac20	2	3
DJNZ	Rn,nhãn	giảm giá trị thanh ghi Rn (R0 -> R7) đi một đơn vị, sau đó nhảy đến nhãn nếu Rn khác 0.	DJNZ R7,khackhong	2	2
DJNZ	direct, nhãn	giảm giá trị của ô nhớ direct đi 1 rồi nhảy đến nhãn nếu giá trị ô nhớ direct khác 0. nếu giá trị ô nhớ direct bằng 0 thì thực hiện câu lệnh ngay sau câu lệnh này.	DJNZ 40h,khackhong	2	3
NOP		tạm dừng VXL trong 1 chu kỳ máy	NOP	1	1
CÂU LỆNH CHUYỂN DỮ LIỆU					

MOV	A, Rn	chép dữ liệu từ thanh ghi Rn vào thanh ghi A	MOV A, R7	1	1
MOV	A, direct	chép dữ liệu từ ô nhớ vào thanh ghi A	MOV A, 40h	1	2
MOV	A, @Ri	chép dữ liệu từ ô nhớ được trỏ bởi thanh ghi Ri vào thanh ghi A	MOV A, @R0	1	1
MOV	A, #data	gán A bằng con số data	MOV A, #10	1	2
MOV	Rn, A	chép dữ liệu từ thanh ghi A vào thanh ghi Rn	MOV R1, A	1	1
MOV	Rn, direct	chép dữ liệu từ ô nhớ vào thanh ghi Rn	MOV R2, 40h	2	2
MOV	Rn, #data	gán Rn bằng con số data	MOV R2, #10	1	2
MOV	direct, A	chép dữ liệu của A vào ô nhớ	MOV 40h, A	1	2
MOV	direct, Rn	chép dữ liệu của Rn vào ô nhớ	MOV 50h, R7	2	2
MOV	direct1, direct2	chép dữ liệu từ ô nhớ direct2 vào ô nhớ direct1	MOV 40h, 50h	2	3
MOV	direct, @Ri	chép dữ liệu từ ô nhớ được trỏ bởi Ri vào ô nhớ direct	MOV 40h, @R0	2	2
MOV	direct, #data	gán nội dung ô nhớ direct bằng giá trị data	MOV 50h, #10	2	3
MOV	@Ri, A	chép dữ liệu trong thanh ghi A vào ô nhớ được trỏ bởi Ri	MOV @R0, A	1	1
MOV	@Ri, direct	chép dữ liệu từ ô nhớ trực tiếp vào ô nhớ được trỏ bởi Ri	MOV @R1, 40h	2	2

MOV	@Ri, #data	gán nội dung ô nhớ được trỏ bởi Ri bằng giá trị data	MOV @R0, #10	1	2
MOV	DPTR, #data	gán giá trị thanh ghi DPTR bằng data	MOV DPTR, #2000h	2	3
MOVC	A, @A+DPTR	chép vào A một byte từ bộ nhớ chương trình tại vị trí cách con trỏ gốc DPTR một khoảng là A	MOVC A, @A+DPTR	2	1
MOVC	A, @A+PC	chép vào A một byte từ bộ nhớ chương trình tại vị trí cách vị trí lệnh hiện hành một khoảng là A	MOVC A, @A+PC	2	1
MOVX	A, @Ri	chép vào A một byte từ ô nhớ ngoài có địa chỉ được trỏ bởi Ri	MOVX A, @R0	2	1
MOVX	A, @DPTR	chép vào A một byte từ ô nhớ ngoài có địa chỉ được trỏ bởi DPTR	MOVX A, @DPTR	2	1
MOVX	@Ri, A	chép nội dung thanh ghi A vào ô nhớ ngoài được trỏ bởi Ri	MOVX @Ri, A	2	1
MOVX	@DPTR, A	chép nội dung thanh ghi A vào ô nhớ ngoài được trỏ bởi DPTR	MOVX @DPTR, A	2	1
PUSH	direct	cất giá trị ô nhớ direct vào vùng nhớ stack	PUSH ACC	2	2
POP	direct	lấy lại giá trị đã cất ở vùng stack vào ô nhớ direct	POP ACC	2	2
XCH	A, Rn	hoán đổi nội dung hai thanh ghi A và Rn	XCH A, R7	1	1
XCH	A, direct	hoán đổi nội dung thanh ghi A và nội dung ô nhớ direct	XCH A, 60h	1	2
XCH	A, @Ri	hoán đổi nội dung thanh ghi A và nội dung ô nhớ được trỏ bởi Ri	XCH A, @R0	1	1

XCHD	A, @Ri	hoán đổi nibble thấp thanh ghi A và nibble thấp của nội dung ô nhớ được trỏ bởi Ri	XCHD A, @R1	1	1
LỆNH SỐ HỌC					
ADD	A, Rn	cộng nội dung thanh ghi A và Rn, kết quả cất vào A	ADD A, R5	1	1
ADD	A, direct	cộng nội dung thanh ghi A với ô nhớ direct, kết quả cất vào A	ADD A, 70h	1	2
ADD	A, @Ri	cộng nội dung thanh ghi A với ô nhớ được trỏ bởi Ri, kết quả cất vào A	ADD A, @R0	1	1
ADD	A, #data	cộng nội dung thanh ghi A và con số data, kết quả cất vào A	ADD A, #10	1	2
ADDC	A, Rn	cộng nội dung thanh ghi A và Rn và cộng thêm cờ nhớ C, kết quả cất vào A	ADDC A, R7	1	1
ADDC	A, direct	cộng nội dung thanh ghi A với ô nhớ direct và cộng thêm cờ nhớ C, kết quả cất vào A	ADDC A, 41h	1	2
ADDC	A, @Ri	cộng nội dung thanh ghi A với ô nhớ được trỏ bởi Ri và cộng thêm cờ nhớ C, kết quả cất vào A	ADDC A, @R0	1	1
ADDC	A, #data	cộng nội dung thanh ghi A và con số data và cộng thêm cờ nhớ C, kết quả cất vào A	ADDC A, #10	1	2
SUBB	A, Rn	trừ nội dung thanh ghi A và Rn và trừ thêm cờ nhớ C, kết quả cất vào A	SUBB A, R7	1	1
SUBB	A, direct	trừ nội dung thanh ghi A với ô nhớ direct và trừ thêm cờ nhớ C, kết quả cất vào A	SUBB A, 41h	1	2
SUBB	A, @Ri	trừ nội dung thanh ghi A với ô nhớ được trỏ bởi Ri và trừ thêm cờ nhớ C, kết quả cất vào A	SUBB A, @R0	1	1

SUBB	A, #data	trừ nội dung thanh ghi A và con số data và trừ thêm cờ nhớ C, kết quả cất vào A	SUB B A, #10	1	2
INC	A	tăng nội dung thanh ghi A lên 1 đơn vị	INC A	1	1
INC	Rn	tăng nội dung thanh ghi Rn lên 1 đơn vị	INC R6	1	1
INC	direct	tăng nội dung ô nhớ direct lên 1 đơn vị	INC 42h	1	2
INC	@Ri	tăng nội dung ô nhớ được trỏ bởi Ri lên 1 đơn vị	INC @R0	1	1
DEC	A	giảm nội dung thanh ghi A lên 1 đơn vị	DEC A	1	1
DEC	Rn	giảm nội dung thanh ghi Rn lên 1 đơn vị	DEC R6	1	1
DEC	direct	giảm nội dung ô nhớ direct lên 1 đơn vị	DEC 43h	1	2
DEC	@Ri	giảm nội dung ô nhớ được trỏ bởi Ri lên 1 đơn vị	DEC @R0	1	1
INC	DPTR	tăng giá trị thanh ghi DPTR lên 1 đơn vị	INC DPTR	2	1
MUL	AB	nhân nội dung thanh ghi A và thanh ghi B, kết quả: A chứa byte thấp, B chứa byte cao	MUL AB	4	1
DIV	AB	chia A cho B, kết quả: A chứa thương số, B chứa số dư	DIV AB	4	1
DA	A	hiệu chỉnh BCD cho nội dung thanh ghi A. Nếu digit nào của A (dạng HEX) lớn hơn 9 sẽ được cộng thêm 6 vào digit đó, ngược lại thì cộng 0. lệnh này chỉ hoạt động đúng cho phép cộng.	DA A	1	1
LỆNH LUẬN LÝ TÁC ĐỘNG TRÊN BYTE					
ANL	A, Rn	thực hiện AND nội dung thanh ghi A với Rn, kết quả cất vào A	ANL A, R7	1	1
ANL	A, direct	thực hiện AND nội dung thanh ghi A với ô nhớ direct, kết quả cất vào A	ANL A, 45h	1	2

ANL	A, @Ri	thực hiện AND nội dung thanh ghi A với ô nhớ có địa chỉ trỏ bởi Ri, kết quả cất vào A	ANL A, @R0	1	1
ANL	A, #data	thực hiện AND nội dung thanh ghi A với con số data, kết quả cất vào A	ANL A, #45h	1	2
ANL	direct, A	thực hiện AND nội dung ô nhớ direct với thanh ghi A, kết quả cất vào ô nhớ direct	ANL 45h, A	1	2
ANL	direct, #data	thực hiện AND nội dung ô nhớ direct với con số data, kết quả cất vào ô nhớ direct	ANL 45h, #45h	2	3
ORL	A, Rn	thực hiện OR nội dung thanh ghi A với Rn, kết quả cất vào A	ORL A, R7	1	1
ORL	A, direct	thực hiện OR nội dung thanh ghi A với ô nhớ direct, kết quả cất vào A	ORL A, 45h	1	2
ORL	A, @Ri	thực hiện OR nội dung thanh ghi A với ô nhớ có địa chỉ trỏ bởi Ri, kết quả cất vào A	ORL A, @R0	1	1
ORL	A, #data	thực hiện OR nội dung thanh ghi A với con số data, kết quả cất vào A	ORL A, #45h	1	2
ORL	direct, A	thực hiện OR nội dung ô nhớ direct với thanh ghi A, kết quả cất vào ô nhớ direct	ORL 45h, A	1	2
ORL	direct, #data	thực hiện OR nội dung ô nhớ direct với con số data, kết quả cất vào ô nhớ direct	ORL 45h, #45h	2	3
XRL	A, Rn	thực hiện XOR nội dung thanh ghi A với Rn, kết quả cất vào A	XRL A, R7	1	1
XRL	A, direct	thực hiện XOR nội dung thanh ghi A với ô nhớ direct, kết quả cất vào A	XRL A, 45h	1	2

XRL	A, @Ri	thực hiện XOR nội dung thanh ghi A với ô nhớ có địa chỉ trỏ bởi Ri, kết quả cất vào A	XRL A, @R0	1	1
XRL	A, #data	thực hiện XOR nội dung thanh ghi A với con số data, kết quả cất vào A	XRL A, #45h	1	2
XRL	direct, A	thực hiện XOR nội dung ô nhớ direct với thanh ghi A, kết quả cất vào ô nhớ direct	XRL 45h, A	1	2
XRL	direct, #data	thực hiện XOR nội dung ô nhớ direct với con số data, kết quả cất vào ô nhớ direct	XRL 45h, #45h	2	3
CLR	A	gán A=0	CLR A	1	1
CPL	A	lấy bù 1 giá trị của A. thực sự là gán A= 255 - A	CPL A	1	1
RL	A	quay trái các bit trong thanh ghi A	RL A	1	1
RLC	A	quay trái các bit trong thanh ghi A có dùng thêm cờ nhớ C	RLC A	1	1
RR	A	quay phải các bit trong thanh ghi A.	RR A	1	1
RLC	A	quay phải các bit trong thanh ghi A có dùng thêm cờ nhớ C.	RLC A	1	1
SWAP	A	hoán đổi nibble cao và nibble thấp của thanh ghi A.	SWAP A	1	1