QUICK REFERENCE CHART

MNEMONIC		DESCRIPTION
Arithmetic Operations		
ADD	A,source	add source to A
ADD	A,#data	
ADDC	A,#source	add with carry
ADDC	A,#data	
SUBB	A,source	subtract from A
SUBB	A,#data	with borrow
INC	A	increment
INC	source	
DEC	A	decrement
DEC	source	
INC	DPTR	increment DPTR
MUL	AB	multiply A & B
DIV	AB	divide A by B
DA	A	decimal adjust A
Logical Operations		
ANL	A,source	logical AND
ANL	A,#data	
ANL	direct,A	
ANL	direct,#data	
ORL	A,source	logical OR
ORL	A,#data	
ORL	direct,A	
ORL	direct,#data	
XRL	A,source	logical XOR
XRL	A,#data	
XRL	direct,A	
XRL	direct,#data	
CLR	A	clear A
CPL	A	complement A
RL	A	rotate A left
RLC	A	(through C)
RR	A	rotate A right
RRC	A	(through C)
SWAP	A	swap nibbles

LEGEND

Rn	register addressing using R0–R7
direct	8-bit internal address (00H-FFH)
@Ri	indirect addressing using R0 or R1
source	any of [Rn,direct,@ri]
dest	any of [Rn,direct,@ri]
#data	8-bit constant included in instr.
#data 16	16-bit constant
bit	8-bit direct address of bit
rel	signed 8-bit offset
addr11	11-bit address in current 2k page
addr16	16-bit address

Data Transfer Operations		
MOV	A,source	move source
MOV	A,#data	to destination
MOV	dest,A	
MOV	dest,source	
MOV	dest,#data	
MOV	DPTR,#data16	
MOVC	A,@A+DPTR	move from code
MOVC	A,@A+PC	memory
MOVX	A,@Ri	move from data
MOVX	A,@DPTR	memory
MOVX	@Ri,A	
MOVX	@DPTR,A	
PUSH	direct	push onto stack
POP	direct	pop from stack
XCH	A,source	exchange bytes
XCHD	A,@Ri	exchange low order digits
Boolean Variable Manipulation		
CLR	C	clear bit
CLR	bit	
SETB	C	set bit
SETB	bit	
CPL	C	complement bit
CPL	bit	
ANL	C,bit	AND bit with C
ANL	C,/bit	AND NOT bit with C
ORL	C,bit	OR bit with C
ORL	C,/bit	OR NOT bit with C
MOV	C,bit	move bit to bit
MOV	bit,C	
JC	rel	jump if C set
JNC	rel	if C not set
JB	bit,rel	jump if bit set
JNB	bit,rel	if bit not set
JBC	bit,rel	if set then clear
Program Branching		
ACALL	addr11	call subroutine
LCALL	addr16	
RET		return from sub.
RETI		from interrupt
AJMP	addr11	jump
LJMP	addr16	
SJMP	rel	
JMP	@A+DPTR	
JZ	rel	jump if A = 0
JNZ	rel	if A not = 0
CJNE	A,direct,rel	compare and jump
CJNE	A,#data,rel	if not equal
CJNE	Rn,#data,rel	
CJNE	@Ri,#data,rel	
DJNZ	Rn,rel	decrement and jump
DJNZ	direct,rel	if not zero
NOP		no operation

Instruction Code Summary

H / L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	JBC bit, rel	JB bit, rel	JNB bit, rel	JC rel	JNC rel	JZ rel	JNZ rel	SJMP rel	MOV DPTR, # data 16	ORL C, /bit	ANL C, /bit	PUSH dir	POP dir	MOVX A, @DPTR	MOVX @DPTR, A
1	AJMP (P0)	ACALL (P0)	AJMP (P1)	ACALL (P1)	AJMP (P2)	ACALL (P2)	AJMP (P3)	ACALL (P3)	AJMP (P4)	ACALL (P4)	AJMP (P5)	ACALL (P5)	AJMP (P6)	ACALL (P6)	AJMP (P7)	ACALL (P7)
2	LJMP addr16	LCALL addr16	RET	RETI	ORL dir, A	ANL dir, A	XRL dir, A	ORL C, bit	ANL C, bit	MOV C, bit	MOV C, bit	CPL bit	CLR bit	SETB bit	MOVX A, @R0	MOVX @R0, A
3	RR A	RRC A	RL A	RLC A	ORL dir, # data	ANL dir, # data	XRL dir, # data	JMP @A+DPTR	MOVC A, @A+PC	MOVC A, @A+DPTR	INC DPTR	CPL C	CLR C	SETB C	MOVX A, @R1	MOVX @R1, A
4	INC A	DEC A	ADD A, # data	ADDC A, # data	ORL A, # data	ANL A, # data	XRL A, # data	MOV A, # data	DIV AB	SUBB A, # data	MUL AB	CJNE A, # data, rel	SWAP A	DA A	CLR A	CPL A
5	INC dir	DEC dir	ADD A, dir	ADDC A, dir	ORL A, dir	ANL A, dir	XRL A, dir	MOV dir, # data	MOV dir, dir	SUBB A, dir		CJNE A, dir, rel	XCH A, dir	DJNZ dir, rel	MOV A, dir	MOV dir, A
6	INC @R0	DEC @R0	ADD A, @R0	ADDC A, @R0	ORL A, @R0	ANL A, @R0	XRL A, @R0	MOV @R0, # data	MOV dir, @R0	SUBB A, @R0	MOV @R0, dir	CJNE @R0, # data, rel	XCH A, @R0	XCHD A, @R0	MOV A, @R0	MOV @ R0, A
7	INC @R1	DEC @R1	ADD A, @R1	ADDC A, @R1	ORL A, @R1	ANL A, @R1	XRL A, @R1	MOV @R1, # data	MOV dir, @R1	SUBB A, @R1	MOV @R1, dir	CJNE @R1, # data, rel	XCH A, @R1	XCHD A, @R1	MOV A, @R1	MOV @ R1, A
8	INC R0	DEC R0	ADD A, R0	ADDC A, R0	ORL A, R0	ANL A, R0	XRL A, R0	MOV R0, # data	MOV dir, R0	SUBB A, R0	MOV R0, dir	CJNE R0, # data, rel	XCH A, R0	DJNZ R0, rel	MOV A, R0	MOV R0, A
9	INC R1	DEC R1	ADD A, R1	ADDC A, R1	ORL A, R1	ANL A, R1	XRL A, R1	MOV R1, # data	MOV dir, R1	SUBB A, R1	MOV R1, dir	CJNE R1, # data, rel	XCH A, R1	DJNZ R1, rel	MOV A, R1	MOV R1, A
A	INC R2	DEC R2	ADD A, R2	ADDC A, R2	ORL A, R2	ANL A, R2	XRL A, R2	MOV R2, # data	MOV dir, R2	SUBB A, R2	MOV R2, dir	CJNE R2, # data, rel	XCH A, R2	DJNZ R2, rel	MOV A, R2	MOV R2, A
B	INC R3	DEC R3	ADD A, R3	ADDC A, R3	ORL A, R3	ANL A, R3	XRL A, R3	MOV R3, # data	MOV dir, R3	SUBB A, R3	MOV R3, dir	CJNE R3, # data, rel	XCH A, R3	DJNZ R3, rel	MOV A, R3	MOV R3, A
C	INC R4	DEC R4	ADD A, R4	ADDC A, R4	ORL A, R4	ANL A, R4	XRL A, R4	MOV R4, # data	MOV dir, R4	SUBB A, R4	MOV R4, dir	CJNE R4, # data, rel	XCH A, R4	DJNZ R4, rel	MOV A, R4	MOV R4, A
D	INC R5	DEC R5	ADD A, R5	ADDC A, R5	ORL A, R5	ANL A, R5	XRL A, R5	MOV R5, # data	MOV dir, R5	SUBB A, R5	MOV R5, dir	CJNE R5, # data, rel	XCH A, R5	DJNZ R5, rel	MOV A, R5	MOV R5, A
E	INC R6	DEC R6	ADD A, R6	ADDC A, R6	ORL A, R6	ANL A, R6	XRL A, R6	MOV R6, # data	MOV dir, R6	SUBB A, R6	MOV R6, dir	CJNE R6, # data, rel	XCH A, R6	DJNZ R6, rel	MOV A, R6	MOV R6, A
F	INC R7	DEC R7	ADD A, R7	ADDC A, R7	ORL A, R7	ANL A, R7	XRL A, R7	MOV R7, # data	MOV dir, R7	SUBB A, R7	MOV R7, dir	CJNE R7, # data, rel	XCH A, R7	DJNZ R7, rel	MOV A, R7	MOV R7, A

2Byte	3Byte
2Cycle	4Cycle

FIGURE B–1
Opcode map

8051 Opcode Map.