Partial Listing of 8051 Mnemonics

when there are 2 operands the format is always: *operation destination, source*

Rn	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's **direct** address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@Ri	8-bit internal data RAM location (0-255) addressed **indirectly through register R1 or R0.**
#data	8-bit constant included in instruction.
#data 16	16-bit constant included in instruction.
port	Ports 0, 1, 2, and 3 (mapped as 080h, 090h, 0A0h, and 0B0h). They are set to output ports on start-up and must be initialized for input by sending the data 0FFh to them, as appripriate.
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64K byte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2K byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and *all conditional jumps*. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit	**Direct** Addressed bit, in Internal Data RAM or Special Function Register (see note at end)

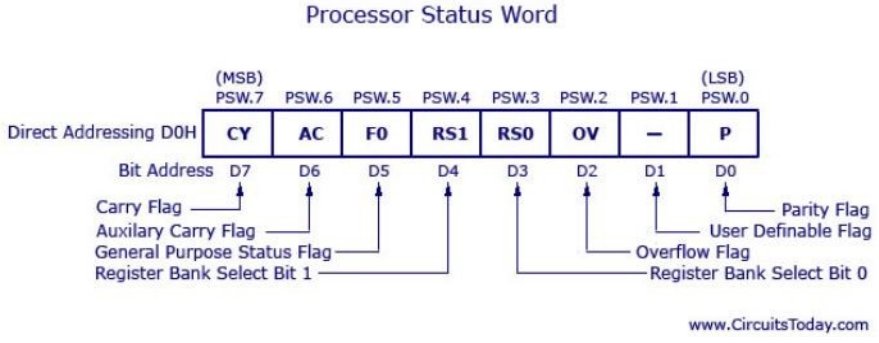
DATA TRANSFER		(MOVC. MOVX. and DPTR instructions not included)		
operation	**Mnemonic**	**Operand 1**	**, Operand 2**	**flags**
copy data to A or B	MOV	A or B	Rn – direct - @Ri - #data – port	
copy data to Reg.	MOV	Rn	A – direct - #data – port	
copy to direct memory	MOV	direct	A – direct – Rn - @Ri - #data – port	
copy to indirect mem	MOV	@Ri	A - direct - #data – port	
copy data to a port (P1 – P3)	MOV	port	A – direct – Rn - @Ri - #data – port	
push byte onto stack	PUSH	direct		
pop byte from stack	POP	direct		
exchange data with A	XCH	A	direct – Rn - @Ri	
copy direct bit to carry	MOV	C	bit	C
copy carry to direct bit	MOV	bit	C	

ARITHMETIC OPERATIONS				
operation	**Mnemonic**	**Operand 1**	**, Operand 2**	**flags**
add to A	ADD	A	Rn – direct - @Ri - #data – port	C, OV, AC
add w/ carry	ADDC	A	Rn – direct - @Ri - #data - port	C, OV, AC
sub w/ burrow	SUBB	A	Rn – direct - @Ri - #data - port	C, OV, AC
increment	INC	A–Rn–direct-@Ri-port		
decrement	DEC	A–Rn–direct-@Ri-port		
multiply A & B	MUL	AB		OV, C=0
divide A by B	DIV	AB		OV, C=0
decimal adj	DA	A		C

LOGICAL OPERATIONS				
operation	**Mnemonic**	**Operand 1**	**, Operand 2**	**flags**
AND to A	ANL	A	Rn – direct - @Ri - #data - port	
AND to direct or port	ANL	Direct - port	A - #data	
OR to A	ORL	A	Rn – direct - @Ri - #data - port	
OR to direct	ORL	Direct - port	A - #data	
XOR to A	XOR	A	Rn – direct - @Ri - #data - port	
XOR to direct	XOR	Direct - port	A - #data	
clear A	CLR	A		
compliment A	CPL	A		
clear C flag	CLR	C		C
compliment C flag	CPL	C		C
rotate A left	RL	A		
rotate A left thru Carry	RLC	A		C
rotate A right	RR	A		
rotate A right thru Carry	RRC	A		C

(continued next page)

LOGICAL OPERATIONS (continued)				
operation	**Mnemonic**	**Operand 1**	**, Operand 2**	**flags**
swap nibbles within A	SWAP	A		
AND direct bit to Carry	ANL	C, bit		
AND compliment of bit to Carry	ANL	C, /bit		
OR direct bit to Carry	ORL	C, bit		
OR compliment of bit to Carry	ORL	C, /bit		
BOOLEAN VARIABLE MANIPULATION				
operation	**Mnemonic**	**Operand 1**	**, Operand 2**	**flags**
clear carry flag	CLR	C		C=0
clear direct bit	CLR	bit		
set carry flag	SETB	C		C=1
set direct bit	SETB	bit		
compliment carry flag	CPL	C		C
compliment direct bit	CPL	bit		
PROGRAM BRANCHING				
operation	**Mnemonic**	**Operand 1**	**, Operand 2**	**flags**
subroutine call	ACALL	addr11		
long subroutine call	LCALL	addr16		
return from subroutine	RET			
Jump	AJMP	addr11		
Long Jump	LJMP	addr16		
Short Jump (relative addr)	SJMP	rel		
Jump if A=0	JZ	rel		
Jump if A<>0	JNZ	rel		
decrement Byte & jump if <>0	DJNZ	Rn – direct	rel	
compare Byte; jump if not equal	CJNE	A	direct, rel	C
compare data; jump if not equal	CJNE	A – Rn - @Ri	#data, rel	C
Jump if carry is set	JC	rel		
Jump if Carry not set	JNC	rel		
Jump if bit is set	JB	bit, rel		
Jump if bit not set	JNB	bit, rel		
Jump if bit set then clear bit	JBC	bit, rel		(bit=0)
no operation	NOP			
NOTES:				
General Purpose RAM available to the user is 020h through 070h (all byte addressible). The Following are **both** byte and bit addressable:				
RAM locations 20h through 2Fh, ACC, B, ports, PSW (0D0h), IP, IE, TCON				



4/22/20-dw