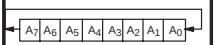
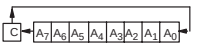
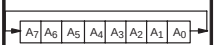
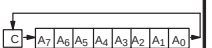
SECTION 16: INSTRUCTION SET DETAILS

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
ARITHMETIC OPERATION	ADD A, Rn	0	0	1	0	1	n_2	n_1	n_0	28-2F	1	1	(A) = (A) + (Rn)
	ADD A, direct	0 a_7	0 a_6	1 a_5	0 a_4	0 a_3	1 a_2	0 a_1	1 a_0	25 Byte 2	2	2	(A) = (A) + (direct)
	ADD A, @Ri	0	0	1	0	0	1	1	i	26-27	1	1	(A) = (A) + ((Ri))
	ADD A, #data	0 d_7	0 d_6	1 d_5	0 d_4	0 d_3	1 d_2	0 d_1	0 d_0	24 Byte 2	2	2	(A) = (A) + #data
	ADDC A, Rn	0	0	1	1	1	n_2	n_1	n_0	38-3F	1	1	(A) = (A)+(C)+(Rn)
	ADDCA,direct	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	0 a_1	1 a_0	35 Byte 2	2	2	(A) = (A)+(C)+(direct)
	ADDC A, @Ri	0	0	1	1	0	1	1	i	36-37	1	1	(A) = (A)+(C)+((Ri))
	ADDC A,#data	0 d_7	0 d_6	1 d_5	1 d_4	0 d_3	1 d_2	0 d_1	0 d_0	34 Byte 2	2	2	(A) = (A)+(C)+#data
	SUBB A, Rn	1	0	0	1	1	n_2	n_1	n_0	98-9F	1	1	(A) = (A)-(C)-(Rn)
	SUBB A,direct	1 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	0 a_1	1 a_0	95 Byte 2	2	2	(A) = (A)-(C)-(direct)
	SUBB A, @Ri	1	0	0	1	0	1	1	i	96-97	1	1	(A) = (A)-(C)-((Ri))
	SUBB A,#data	1 d_7	0 d_6	0 d_5	1 d_4	0 d_3	1 d_2	0 d_1	0 d_0	94 Byte 2	2	2	(A) = (A)-(C)-#data
	INC A	0	0	0	0	0	1	0	0	04	1	1	(A) = (A) + 1
	INC Rn	0	0	0	0	1	n_2	n_1	n_0	08-0F	1	1	(Rn) = (Rn) + 1
	INC direct	0 a_7	0 a_6	0 a_5	0 a_4	0 a_3	1 a_2	0 a_1	1 a_0	05 Byte 2	2	2	(direct) = (direct)+1
	INC @Ri	0	0	0	0	0	1	1	i	06-07	1	1	((Ri)) = ((Ri)) + 1
	INC DPTR	1	0	1	0	0	0	1	1	A3	1	3	(DPTR)=(DPTR)+1
	DEC A	0	0	0	1	0	1	0	0	14	1	1	(A) = (A) - 1
	DEC Rn	0	0	0	1	1	n_2	n_1	n_0	18-1F	1	1	(Rn) = (Rn) - 1
	DEC direct	0 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	0 a_1	1 a_0	15 Byte 2	2	2	(direct) = (direct)-1
	DEC @Ri	0	0	0	1	0	1	1	i	16-17	1	1	((Ri)) = ((Ri)) - 1
	MUL AB	1	0	1	0	0	1	0	0	A4	1	5	(B_{15-8}), (A_{7-0}) = (A) X (B)
	DIV AB	1	0	0	0	0	1	0	0	84	1	5	(A_{15-8}), (A_{7-0}) = (A) ÷ (B)

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
ARITHMETIC OPER.	DA A	1	1	0	1	0	1	0	0	D4	1	1	Contents of Accumulator are BCD, IF [[(A_{3-0}) > 9] OR [(AC) = 1]] THEN (A_{3-0}) = (A_{3-0}) + 6 AND IF [[(A_{7-4}) > 9] OR [(C) = 1]] THEN (A_{7-4}) = (A_{7-4}) + 6
LOGICAL OPERATION	ANL A, Rn	0	1	0	1	1	n_2	n1	n_0	58-5F	1	1	(A) = (A) AND (Rn)
	ANL A, direct	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	1 a_2	0 a_1	1 a_0	55 Byte 2	2	2	(A) = (A) AND (direct)
	ANL A, @Ri	0	1	0	1	0	1	1	i	56-57	1	1	(A) = (A) AND ((Ri))
	ANL A, #data	0 d_7	1 d_6	0 d_5	1 d_4	0 d_3	1 d_2	0 d_1	0 d_0	54 Byte 2	2	2	(A)=(A) AND #data
	ANL direct, A	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	0 a_2	1 a_1	0 a_0	52 Byte 2	2	2	(direct) = (direct) AND A
	ANL direct, #data	0 a_7 d_7	1 a_6 d_6	0 a_5 d_5	1 a_4 d_4	0 a_3 d_3	0 a_2 d_2	1 a_1 d_1	1 a_0 d_0	53 Byte 2 Byte 3	3	3	(direct) = (direct) AND #data
	ORL A, Rn	0	1	0	0	1	n_2	n_1	n_0	48-4F	1	1	(A) = (A) OR (Rn)
	ORL A, direct	0 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	0 a_1	1 a_0	45 Byte 2	2	2	(A) = (A) OR (direct)
	ORL A, @Ri	0	1	0	0	0	1	1	i	46-47	1	1	(A) = (A) OR ((Ri))
	ORL A, #data	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	1 d_2	0 d_1	0 d_0	44 Byte 2	2	2	(A) = (A) OR #data
	ORL direct, A	0 a_7	1 a_6	0 a_5	0 a_4	0 a_3	0 a_2	1 a_1	0 a_0	42 Byte 2	2	2	(direct) = (direct) OR (A)
	ORL direct,#data	0 a_7 d_7	1 a_6 d_6	0 a_5 d_5	0 a_4 d_4	0 a_3 d_3	0 a_2 d_2	1 a_1 d_1	1 a_0 d_0	43 Byte 2 Byte 3	3	3	(direct) = (direct) OR #data
	XRL A, Rn	0	1	1	0	1	n_2	n_1	n_0	68-6F	1	1	(A) = (A) XOR (Rn)
	XRL A, direct	0 a_7	1 a_6	1 a_5	0 a_4	0 a_3	1 a_2	0 a_1	1 a_0	65 Byte 2	2	2	(A) = (A) XOR (direct)
	XRL A, @ Ri	0	1	1	0	0	1	1	i	66-67	1	1	(A) = (A) XOR ((Ri))
	XRL A, #data	0 d_7	1 d6	1 d5	0 d_4	0 d_3	1 d_2	0 d_1	0 d_0	64 Byte 2	2	2	(direct) = (A) XOR #data
	XRL direct, A	0 a_7	1 a_6	1 a_5	0 a_4	0 a_3	0 a_2	1 a_1	0 a_0	62 Byte 2	2	2	(direct) = (direct) XOR (A)
	XRL direct, #data	0 a_7 d_7	1 a_6 d_6	1 a_5 d_5	0 a_4 d_4	0 a_3 d_3	0 a_2 d_2	1 a_1 d_1	1 a_0 d_0	63 Byte 2 Byte 3	3	3	(direct) = (direct) XOR #data
	CLR A	1	1	1	0	0	1	0	0	E4	1	1	(A) = 0
	CPL A	1	1	1	1	0	1	0	0	F4	1	1	(A) = $(\overline{A})$

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
LOGICAL OPERATION	RL A	0	0	1	0	0	0	1	1	23	1	1	 The contents of the accumulator are rotated left by one bit.
	RLC A	0	0	1	1	0	0	1	1	33	1	1	 The contents of the accumulator are rotated left by one bit.
	RR A	0	0	0	0	0	0	1	1	03	1	1	 The contents of the accumulator are rotated right by one bit.
	RRC A	0	0	0	1	0	0	1	1	13	1	1	 The contents of the accumulator are rotated right by one bit.
	SWAP A	1	1	0	0	0	1	0	0	C4	1	1	$(A_{3-0}) \leftrightarrows (A_{7-4})$
DATA TRANSFER	MOV A, Rn	1	1	1	0	1	n_2	n_1	n_0	E8-EF	1	1	(A) = (Rn)
	MOV A, direct	1 a_7	1 a_6	1 a_3	0 a_4	0 a_3	1 a_2	0 a_1	1 a_0	E5 Byte 2	2	2	(A) = (direct)
	MOV A, @Ri	1	1	1	0	0	1	1	i	E6-E7	1	1	(A) = ((Ri))
	MOV A, #data	0 d_7	1 d_6	1 d_5	1 d_4	0 d_3	1 d_2	0 d_1	0 d_0	74 Byte 2	2	2	(A) = #data
	MOV Rn, A	1	1	1	1	1	n_2	n_1	n_0	F8-FF	1	1	(Rn) = (A)
	MOV Rn, direct	1 a_7	0 a_6	1 a_5	0 a_4	1 a_3	n_2 a_2	n_1 a_1	n_0 a_0	A8-AF Byte 2	2	2	(Rn) = (direct)
	MOV Rn, #data	0 d_7	1 d_6	1 d_5	1 d_4	1 d_3	n_2 d_2	n_1 d_1	n_0 d_0	78-7F Byte 2	2	2	(Rn) = #data
	MOV direct, A	1 a_7	1 a_6	1 a_5	1 a_4	0 a_3	1 a_2	0 a_1	1 a_0	F5 Byte 2	2	2	(direct) = (A)
	MOV direct, Rn	1 a_7	0 a_6	0 a_5	0 a_4	1 a_3	n_2 a_2	n_1 a_1	n_0 a_0	88-8F Byte 2	2	2	(direct) = (Rn)
	MOV direct1, direct2	1 a_7 a_7	0 a_6 a_6	0 a_5 a_5	0 a_4 a_4	0 a_3 a_3	1 a_2 a_2	0 a_1 a_1	1 a_0 a_0	85 Byte 2 Byte 3	3	3	(direct1) = (direct2) (source) (destination)
	MOV direct, @Ri	1 a_7	0 a_6	0 a_5	0 a4	0 a_3	1 a_2	1 a_1	i a_0	86-87 Byte 2	2	2	(direct) = ((Ri))

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
DATA TRANSFER	MOV direct, #data	0 a_7 d_7	1 a_6 d_6	1 a_5 d_5	1 a_4 d_4	0 a_3 d_3	1 a_2 d_2	0 a_1 d_1	1 a_0 d_0	75 Byte 2 Byte 3	3	3	(direct) = #data
	MOV @Ri, A	1	1	1	1	0	1	1	i	F6-F7	1	1	((Ri)) = A
	MOV @Ri, direct	1 a_7	0 a_6	1 a_5	0 a_4	0 a_3	1 a_2	1 a_1	i a_0	A6-A7 Byte 2	2	2	((Ri)) = (direct)
	MOV @Ri, #data	0 d_7	1 d_6	1 d_5	1 d_4	0 d_3	1 d_2	1 d1	i d_0	76-77 Byte 2	2	2	((Ri)) = #data
	MOV DPTR, #data16	1 d_7 d_7	0 d_6 d_6	0 d_5 d_5	1 d4 d_4	0 d_3 d_3	0 d_2 d_2	0 d_1 d_1	0 d_0 d_0	90 Byte 2 Byte 3	3	3	(DPTR) = #data$_{15-0}$ (DPH) = #data$_{15-8}$ (DPL) = #data$_{7-0}$
	MOVC A, @A + DPTR	1	0	0	1	0	0	1	1	93	1	3	(A)=((A) + (DPTR))
	MOVC A, @A + PC	1	0	0	0	0	0	1	1	83	1	3	(A) = ((A) + (PC))
	MOVX A, @Ri	1	1	1	0	0	0	1	i	E2-E3	1	2–9	(A) = ((Ri))
	MOVX @DPTR	1	1	1	0	0	0	0	0	E0	1	2–9	(A) = ((DPTR))
	MOVX @Ri, A	1	1	1	1	0	0	1	i	F2-F3	1	2–9	((Ri)) = (A)
	MOVX @DPTR,A	1	1	1	1	0	0	0	0	F0	1	2–9	((DPTR)) = (A)
	PUSH direct	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	0 a_2	0 a_1	0 a_0	C0 Byte 2	2	2	(SP) = (SP) + 1 ((SP)) = (direct)
	POP direct	1 a_7	1 a_6	0 a_5	1 a_4	0 a_3	0 a_2	0 a_1	0 a_0	D0 Byte 2	2	2	(direct) = ((SP)) (SP) = (SP) - 1
	XCH A, Rn	1	1	0	0	1	n_2	n_1	n_0	C8-CF	1	1	(A) = (Rn)
	XCH A, direct	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	0 a_1	1 a_0	C5 Byte 2	2	2	(A) = (direct)
	XCH A, @Ri	1	1	0	0	0	1	1	i	C6-C7	1	1	(A) = ((Ri))
	XCHD A, @Ri	1	1	0	1	0	1	1	i	D6-D7	1	1	(A_{3-0}) = ((Ri_{3-0}))

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
BOOLEAN VARIABLE MANIPULATION	CLR C	1	1	0	0	0	0	1	1	C3	1	1	(C) = 0
	CLR bit	1 b_7	1 b_6	0 b_5	0 b_4	0 b_3	0 b_2	1 b_1	0 b_0	C2 Byte 2	2	2	(bit) = 0
	SETB C	1	1	0	1	0	0	1	1	D3	1	1	(C) = 1
	SETB bit	1 b_7	1 b_6	0 b_5	1 b_4	0 b_3	0 b_2	1 b_1	0 b_0	D2 Byte 2	2	2	(bit) = 1
	CPL C	1	0	1	1	0	0	1	1	B3	1	1	(C) = ($\overline{C}$)
	CPL bit	1 b_7	0 b_6	1 b_5	1 b_4	0 b_3	0 b_2	1 b_1	0 b_0	B2 Byte 2	2	2	(bit) = ($\overline{bit}$)
	ANL C, bit	1 b_7	0 b_6	0 b_5	0 b_4	0 b_3	0 b_2	1 b_1	0 b_0	82 Byte 2	2	2	(C) = (C) AND (bit)
	ANL C, $\overline{bit}$	1 b_7	0 b_6	1 b_5	1 b_4	0 b_3	0 b_2	0 b_1	0 b_0	B0 Byte 2	2	2	(C) = (C) AND ($\overline{bit}$)
	ORL C, bit	0 b_7	1 b6	1 b_5	1 b_4	0 b_3	0 b_2	1 b_1	0 b_0	72 Byte 2	2	2	(C) = (C) OR (bit)
	ORL C, $\overline{bit}$	1 b_7	0 b_6	1 b_5	0 b_4	0 b_3	0 b_2	0 b_1	0 b_0	A0 Byte 2	2	2	(C) = (C) OR ($\overline{bit}$)
	MOV C, bit	1 b_7	0 b_6	1 b_5	0 b_4	0 b_3	0 b_2	1 b_1	0 b_0	A2 Byte 2	2	2	(C) = (bit)
	MOV bit, C	1 b_7	0 b_6	0 b_5	1 b_4	0 b_3	0 b_2	1 b_1	0 b_0	92 Byte 2	2	2	(bit) = (C)

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
PROGRAM BRANCHING	ACALL addr 11	a_{10} a_7	a_9 a_6	a_8 a_5	1 a_4	0 a_3	0 a_2	0 a_1	1 a_0	Byte 1 Byte 2	2	3	(PC) = (PC) + 2 (SP) = (SP) + 1 ((SP)) = (PC_{7-0}) (SP) = (SP) + 1 ((SP)) = (PC_{15-8}) (PC)=page address
	LCALL addr 16	0 a_{15} a_7	0 a_{14} a_6	0 a_{13} a_5	1 a_{12} a_4	0 a_{11} a_3	0 a_{10} a_2	1 a_9 a_1	0 a_8 a_0	12 Byte 2 Byte 3	3	4	(PC) = (PC) + 3 (SP) = (SP) + 1 ((SP)) = (PC_{7-0}) (SP) = (SP) + 1 ((SP)) = (PC_{15-8}) (PC) = $addr_{15-0}$
	RET	0	0	1	0	0	0	1	0	22	1	4	(PC_{15-8}) = ((SP)) (SP) = (SP) - 1 (PC_{7-0}) = ((SP)) (SP) = (SP) - 1
	RETI	0	0	1	1	0	0	1	0	32	1	4	(PC_{15-8}) = ((SP)) (SP) = (SP) - 1 (PC_{7-0}) = ((SP)) (SP) = (SP) - 1
	AJMP addr 11	a_{10} a_7	a_9 a_6	a_8 a_5	0 a_4	0 a_3	0 a_2	0 a_1	1 a_0	Byte 1 Byte 2	2	3	(PC) = (PC) + 2 (PC_{10-0}) = page addr
	LJMP addr 16	0 a_{15} a_7	0 a_{14} a_6	0 a_{13} a_5	0 a_{12} a_4	0 a_{11} a_3	0 a_{10} a_2	1 a_9 a_1	0 a_8 a_0	02 Byte 2 Byte 3	3	4	(PC) = addr15-0
	SJMP rel	1 r_7	0 r_6	0 r_5	0 r_4	0 r_3	0 r_2	0 r_1	0 r_0	80 Byte 2	2	3	(PC) = (PC) + 2 (PC) = (PC) + rel
	JMP @A + DPTR	0	1	1	1	0	0	1	1	73	1	3	(PC) = (A) + (DPTR)
	JZ rel	0 r_7	1 r_6	1 r_5	0 r_4	0 r_3	0 r_2	0 r_1	0 r_0	60 Byte 2	2	3	(PC) = (PC) + 2 IF (A) = 0 THEN (PC) = (PC) + rel
	JNZ rel	0 r_7	1 r_6	1 r_5	1 r_4	0 r_3	0 r_2	0 r_1	0 r_0	70 Byte 2	2	3	(PC) = (PC) + 2 IF (A) $\neq$ 0 THEN (PC) = (PC) + rel

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
PROGRAM BRANCHING	JC rel	0	1	0	0	0	0	0	0	40	2	3	(PC) = (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 2			IF (C) = 1 THEN (PC) = (PC) + rel
	JNC rel	0	1	0	1	0	0	0	0	50	2	3	(PC) = (PC) + 2
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 2			IF (C) $\neq$ 0 THEN (PC) = (PC) + rel
	JB bit, rel	0	0	1	0	0	0	0	0	20	3	4	(PC) = (PC) + 3
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2			IF (bit) = 1 THEN
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			(PC) = (PC) + rel
	JNB bit, rel	0	0	1	1	0	0	0	0	30	3	4	(PC) = (PC) + 3
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2			IF (bit) = 0 THEN
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			(PC) = (PC) + rel
	JBC bit, direct rel	0	0	0	1	0	0	0	0	10	3	4	(PC) = (PC) + 3
		b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	Byte 2			IF (bit) = 1 THEN
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			(bit) = 0 (PC) = (PC) + rel
	CJNE A, direct rel	1	0	1	1	0	1	0	1	B5	3	4	(PC) = (PC) + 3
		a7	a6	a5	ad	a3	a2	a1	a0	Byte 2			IF (direct) < (A) THEN (PC) = (PC) + rel and (C) = 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			OR
													IF (direct) > (A) THEN (PC) = (PC) + rel and (C) = 1
	CJNE A, #data rel	1	0	1	1	0	1	0	0	B4	3	4	(PC) = (PC) + 3
		d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			IF #data < (A) THEN (PC) = (PC) + rel and (C) = 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			OR
													IF #data > (A) THEN (PC) = (PC) + rel and (C) = 1
	CJNE Rn, #data rel	1	0	1	1	1	n_2	n_1	n_0	B8-BF	3	4	(PC) = (PC) + 3
		d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			IF #data < (Rn) THEN (PC) = (PC) + rel and (C) = 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			OR
													IF #data > (Rn) THEN (PC) = (PC) + rel and (C) = 1
	CJNE @Ri, #data rel	1	0	1	1	0	1	1	i	B6-B7	3	4	(PC) = (PC) + 3
		d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			IF #data < ((Ri)) THEN (PC) = (PC) + rel and (C) = 0
		r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0	Byte 3			OR
													IF #data > ((Ri)) THEN (PC) = (PC) + rel and (C) = 1

	MNEMONIC	INSTRUCTION CODE								HEX	BYTE	CYCLE	EXPLANATION
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0				
PROGRAM BRANCHING	DJNZ Rn, rel	1 r_7	1 r_6	0 r_5	1 r_4	1 r_3	n_2 r_2	n_1 r_1	n_0 r_0	D8-Df Byte 2	2	3	(PC) = (PC) + 2 (Rn) = (Rn) - 1 IF (Rn) 0 THEN (PC) = (PC) + rel
	DJNZ direct rel	1 a_7 r_7	1 a_6 r_6	0 a_5 r_5	1 a_4 r_4	0 a_3 r_3	1 a_2 r_2	0 a_1 r_1	1 a_0 r_0	D5 Byte 2 Byte 3	3	4	(PC) = (PC) + 3 (direct) = (direct) - 1 IF (direct)≠0 THEN (PC) = (PC) + rel
	NOP	0	0	0	0	0	0	0	0	00	1	1	(PC) = (PC) + 1