4.4 Instruction Set Summary Tables

The following two tables give a survey about the instruction set of the C500 family microcontrollers. In **table 4-3** the instructions are ordered in functional groups. In **table 4-4** the instructions are ordered in the hexadecimal order of their opcode.

4.4.1 Functional Groups of Instructions

Table 4-3 :
Instruction Set Summary

Mnemonic		Description	Byte	Cycle
Arithmetic Operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A @Ri	Add indirect RAM to accumulator	1	1
ADD	A,#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry flag	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A, #data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1	1
SUBB	A,#data	Subtract immediate data from A with borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Multiply A and B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1

Table 4-3 :
Instruction Set Summary (cont'd)

Mnemonic		Description	Byte	Cycle
Logic Operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive OR register to accumulator	1	1
XRL	A direct	Exclusive OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive OR indirect RAM to accumulator	1	1
XRL	A,#data	Exclusive OR immediate data to accumulator	2	1
XRL	direct,A	Exclusive OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive OR immediate data to direct byte	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate accumulator left through carry	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate accumulator right through carry	1	1
SWAP	A	Swap nibbles within the accumulator	1	1

Table 4-3 :
Instruction Set Summary (cont'd)

Mnemonic		Description	Byte	Cycle
Data Transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri, #data	Move immediate data to indirect RAM	2	1
MOV	DPTR, #data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to accumulator	1	2
MOVC	A,@A + PC	Move code byte relative to PC to accumulator	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr.) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr.) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr.)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr.)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with accumulator	1	1
XCHD	A,@Ri	Exchange low-order nibble indir. RAM with A	1	1

1) MOV A,ACC is not a valid instruction

Table 4-3 :
Instruction Set Summary (cont'd)

Mnemonic		Description	Byte	Cycle
Boolean Variable Manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and Machine Control				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long iump	3	2
SJMP	rel	Short jump (relative addr.)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if carry flag is not set	2	2
JB	bit,rel	Jump if direct bit is set	3	2
JNB	bit,rel	Jump if direct bit is not set	3	2
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2
CJNE	A,direct,rel	Compare direct byte to A and jump if not equal	3	2

Table 4-3 :
Instruction Set Summary (cont'd)

Mnemonic	Description	Byte	Cycle
Program and Machine Control (cont'd)			
CJNE A,#data,rel	Compare immediate to A and jump if not equal	3	2
CJNE Rn,#data rel	Compare immed. to reg. and jump if not equal	3	2
CJNE @Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2
DJNZ Rn,rel	Decrement register and jump if not zero	2	2
DJNZ direct,rel	Decrement direct byte and jump if not zero	3	2
NOP	No operation	1	1

4.4.2 Hexadecimal Ordered Instructions

Table 4-4 :
Instruction List in Hexadecimal Order

Op-Code	Mnemonic		Op-Code	Mnemonic		Op-Code	Mnemonic	
00_H	NOP		20_H	JB	bit.rel	40_H	JC	rel
01_H	AJMP	addr11	21_H	AJMP	addr11	41_H	AJMP	addr11
02_H	LJMP	addr16	22_H	RET		42_H	ORL	direct,A
03_H	RR	A	23_H	RL	A	43_H	ORL	direct,#data
04_H	INC	A	24_H	ADD	A,#data	44_H	ORL	A,#data
05_H	INC	direct	25_H	ADD	A,direct	45_H	ORL	A,direct
06_H	INC	@R0	26_H	ADD	A,@R0	46_H	ORL	A,@R0
07_H	INC	@R1	27_H	ADD	A,@R1	47_H	ORL	A,@R1
08_H	INC	R0	28_H	ADD	A,R0	48_H	ORL	A,R0
09_H	INC	R1	29_H	ADD	A,R1	49_H	ORL	A,R1
$0A_H$	INC	R2	$2A_H$	ADD	A,R2	$4A_H$	ORL	A,R2
$0B_H$	INC	R3	$2B_H$	ADD	A,R3	$4B_H$	ORL	A,R3
$0C_H$	INC	R4	$2C_H$	ADD	A,R4	$4C_H$	ORL	A,R4
$0D_H$	INC	R5	$2D_H$	ADD	A,R5	$4D_H$	ORL	A,R5
$0E_H$	INC	R6	$2E_H$	ADD	A,R6	$4E_H$	ORL	A,R6
$0F_H$	INC	R7	$2F_H$	ADD	A,R7	$4F_H$	ORL	A,R7
10_H	JBC	bit,rel	30_H	JNB	bit.rel	50_H	JNC	rel
11_H	ACALL	addr11	31_H	ACALL	addr11	51_H	ACALL	addr11
12_H	LCALL	addr16	32_H	RETI		52_H	ANL	direct,A
13_H	RRC	A	33_H	RLC	A	53_H	ANL	direct,#data
14_H	DEC	A	34_H	ADDC	A,#data	54_H	ANL	A,#data
15_H	DEC	direct	35_H	ADDC	A,direct	55_H	ANL	A,direct
16_H	DEC	@R0	36_H	ADDC	A,@R0	56_H	ANL	A,@R0
17_H	DEC	@R1	37_H	ADDC	A,@R1	57_H	ANL	A,@R1
18_H	DEC	R0	38_H	ADDC	A,R0	58_H	ANL	A,R0
19_H	DEC	R1	39_H	ADDC	A,R1	59_H	ANL	A,R1
$1A_H$	DEC	R2	$3A_H$	ADDC	A,R2	$5A_H$	ANL	A,R2
$1B_H$	DEC	R3	$3B_H$	ADDC	A,R3	$5B_H$	ANL	A,R3
$1C_H$	DEC	R4	$3C_H$	ADDC	A,R4	$5C_H$	ANL	A,R4
$1D_H$	DEC	R5	$3D_H$	ADDC	A,R5	$5D_H$	ANL	A,R5
$1E_H$	DEC	R6	$3E_H$	ADDC	A,R6	$5E_H$	ANL	A,R6
$1F_H$	DEC	R7	$3F_H$	ADDC	A,R7	$5F_H$	ANL	A,R7

Table 4-4 :
Instruction List in Hexadecimal Order (cont'd)

Op-Code	Mnemonic		Op-Code	Mnemonic		Op-Code	Mnemonic	
60_H	JZ	rel	80_H	SJMP	rel	$A0_H$	ORL	C,/bit
61_H	AJMP	addr11	81_H	AJMP	addr11	$A1_H$	AJMP	addr11
62_H	XRL	direct,A	82_H	ANL	C,bit	$A2_H$	MOV	C,bit
63_H	XRL	direct,#data	83_H	MOVC	A,@A+PC	$A3_H$	INC	DPTR
64_H	XRL	A,#data	84_H	DIV	AB	$A4_H$	MUL	AB
65_H	XRL	A,direct	85_H	MOV	direct,direct	$A5_H$	-	
66_H	XRL	A,@R0	86_H	MOV	direct,@R0	$A6_H$	MOV	@R0,direct
67_H	XRL	A,@R1	87_H	MOV	direct,@R1	$A7_H$	MOV	@R1,direct
68_H	XRL	A,R0	88_H	MOV	direct,R0	$A8_H$	MOV	R0,direct
69_H	XRL	A,R1	89_H	MOV	direct,R1	$A9_H$	MOV	R1,direct
$6A_H$	XRL	A,R2	$8A_H$	MOV	direct,R2	AA_H	MOV	R2,direct
$6B_H$	XRL	A,R3	$8B_H$	MOV	direct,R3	AB_H	MOV	R3,direct
$6C_H$	XRL	A,R4	$8C_H$	MOV	direct,R4	AC_H	MOV	R4,direct
$6D_H$	XRL	A,R5	$8D_H$	MOV	direct,R5	AD_H	MOV	R5,direct
$6E_H$	XRL	A,R6	$8E_H$	MOV	direct,R6	AE_H	MOV	R6,direct
$6F_H$	XRL	A,R7	$8F_H$	MOV	direct,R7	AF_H	MOV	R7,direct
70_H	JNZ	rel	90_H	MOV	DPTR,#data16	$B0_H$	ANL	C,/bit
71_H	ACALL	addr11	91_H	ACALL	addr11	$B1_H$	ACALL	addr11
72_H	ORL	C,direct	92_H	MOV	bit,C	$B2_H$	CPL	bit
73_H	JMP	@A+DPTR	93_H	MOVC	A,@A+DPTR	$B3_H$	CPL	C
74_H	MOV	A,#data	94_H	SUBB	A,#data	$B4_H$	CJNE	A,#data,rel
75_H	MOV	direct,#data	95_H	SUBB	A,direct	$B5_H$	CJNE	A,direct,rel
76_H	MOV	@R0,#data	96_H	SUBB	A,@R0	$B6_H$	CJNE	@R0,#data,rel
77_H	MOV	@R1,#data	97_H	SUBB	A,@R1	$B7_H$	CJNE	@R1,#data,rel
78_H	MOV	R0.#data	98_H	SUBB	A,R0	$B8_H$	CJNE	R0,#data,rel
79_H	MOV	R1.#data	99_H	SUBB	A,R1	$B9_H$	CJNE	R1,#data,rel
$7A_H$	MOV	R2.#data	$9A_H$	SUBB	A,R2	BA_H	CJNE	R2,#data,rel
$7B_H$	MOV	R3.#data	$9B_H$	SUBB	A,R3	BB_H	CJNE	R3,#data,rel
$7C_H$	MOV	R4.#data	$9C_H$	SUBB	A,R4	BC_H	CJNE	R4,#data,rel
$7D_H$	MOV	R5.#data	$9D_H$	SUBB	A,R5	BD_H	CJNE	R5,#data,rel
$7E_H$	MOV	R6.#data	$9E_H$	SUBB	A,R6	BE_H	CJNE	R6,#data,rel
$7F_H$	MOV	R7.#data	$9F_H$	SUBB	A,R7	BF_H	CJNE	R7,#data,rel

Table 4-4 :
Instruction List in Hexadecimal Order (cont'd)

Op-Code	Mnemonic		Op-Code	Mnemonic	
$C0_H$	PUSH	direct	$E0_H$	MOVX	A,@DPTR
$C1_H$	AJMP	addr11	$E1_H$	AJMP	addr11
$C2_H$	CLR	bit	$E2_H$	MOVX	A,@R0
$C3_H$	CLR	C	$E3_H$	MOVX	A,@R1
$C4_H$	SWAP	A	$E4_H$	CLR	A
$C5_H$	XCH	A,direct	$E5_H$	MOV	A,direct
$C6_H$	XCH	A,@R0	$E6_H$	MOV	A,@R0
$C7_H$	XCH	A,@R1	$E7_H$	MOV	A,@R1
$C8_H$	XCH	A,R0	$E8_H$	MOV	A,R0
$C9_H$	XCH	A,R1	$E9_H$	MOV	A,R1
CA_H	XCH	A,R2	EA_H	MOV	A,R2
CB_H	XCH	A,R3	EB_H	MOV	A,R3
CC_H	XCH	A,R4	EC_H	MOV	A,R4
CD_H	XCH	A,R5	ED_H	MOV	A,R5
CE_H	XCH	A,R6	EE_H	MOV	A,R6
CF_H	XCH	A,R7	EF_H	MOV	A,R7
$D0_H$	POP	direct	$F0_H$	MOVX	@DPTR,A
$D1_H$	ACALL	addr11	$F1_H$	ACALL	addr11
$D2_H$	SETB	bit	$F2_H$	MOVX	@R0,A
$D3_H$	SETB	C	$F3_H$	MOVX	@R1,A
$D4_H$	DA	A	$F4_H$	CPL	A
$D5_H$	DJNZ	direct,rel	$F5_H$	MOV	direct,A
$D6_H$	XCHD	A,@R0	$F6_H$	MOV	@R0,A
$D7_H$	XCHD	A,@R1	$F7_H$	MOV	@R1,A
$D8_H$	DJNZ	R0,rel	$F8_H$	MOV	R0,A
$D9_H$	DJNZ	R1,rel	$F9_H$	MOV	R1,A
DA_H	DJNZ	R2,rel	FA_H	MOV	R2,A
DB_H	DJNZ	R3,rel	FB_H	MOV	R3,A
DC_H	DJNZ	R4,rel	FC_H	MOV	R4,A
DD_H	DJNZ	R5,rel	FD_H	MOV	R5,A
DE_H	DJNZ	R6,rel	FE_H	MOV	R6,A
DF_H	DJNZ	R7,rel	FF_H	MOV	R7,A