A. Appendix for as8051 Frankenstein Assembler

A.1 Pseudo Operations

A.1.1 *Standard_Pseudo_Operation_Mnemonics*

End	END
File Inclusion	INCL INCLUDE
If	IF
Else	ELSE
End If	ENDI
Equate	EQU
Set	SET
Org	ORG
Reserve Memory	RESERVE RMB
Define Byte Data	BYTE DB FCB
Define Word Data	DW FDB WORD
Define String Data	FCC STRING
Define Character Set Translation	CHARSET
Define Character Value	CHARDEF CHD
Use Character Translation	CHARUSE

A.1.2 *Machine_Dependent_Pseudo_Operations*

A.1.2.1 ***Define_Word_Data*** The Define Word Data pseudo-operations generate the byte reversed form for 16 bit constants. The first byte is the low order half, followed by the high order byte.

A.2 Instructions

A.2.1 *Instruction_List*

Opcode	Syntax	Selection Criteria
ACALL	expr	
ADD	A ',' '#' expr	
ADD	A ',' '@' REG	R01
ADD	A ',' REG	R07
ADD	A ',' expr	
ADDC	A ',' '#' expr	
ADDC	A ',' '@' REG	R01
ADDC	A ',' REG	R07
ADDC	A ',' expr	
AJMP	expr	
ANL	A ',' '#' expr	

Opcode	Syntax	Selection Criteria
ANL	A ',' '@' REG	R01
ANL	A ',' REG	R07
ANL	A ',' expr	
ANL	C ',' '/' bit	
ANL	C ',' '/' expr	
ANL	C ',' bit	
ANL	C ',' expr	
ANL	expr ',' '#' expr	
ANL	expr ','A	
CJNE	A ',' '#' expr ',' expr	
CJNE	A ',' expr ',' expr	
CJNE	'@' REG ',' '#' expr ',' expr	R01
CJNE	REG ',' '#' expr ',' expr	R07
CLR	A	
CLR	C	
CLR	bit	
CLR	expr	
CPL	A	
CPL	C	
CPL	bit	
CPL	expr	
DA	A	
DEC	'@' REG	R01
DEC	A	
DEC	REG	R07
DEC	expr	
DIV	AB	
DJNZ	REG ',' expr	R07
DJNZ	expr ',' expr	
INC	'@' REG	R01
INC	A	
INC	DPTR	
INC	REG	R07
INC	expr	
JB	bit ',' expr	
JB	expr ',' expr	
JBC	bit ',' expr	
JBC	expr ',' expr	

Opcode	Syntax	Selection Criteria
JC	expr	
JMP	'@' A '+' DPTR	
JNB	bit ',' expr	
JNB	expr ',' expr	
JNC	expr	
JNZ	expr	
JZ	expr	
LCALL	expr	
LJMP	expr	
MOV	'@' REG ',' '#' expr	R01
MOV	'@' REG ',' A	R01
MOV	'@' REG ',' expr	R01
MOV	A ',' '#' expr	
MOV	A ',' '@' REG	R01
MOV	A ',' REG	R07
MOV	A ',' expr	
MOV	C ',' bit	
MOV	C ',' expr	
MOV	DPTR ',' '#' expr	
MOV	REG ',' A	R07
MOV	REG ',' '#' expr	R07
MOV	REG ',' expr	R07
MOV	bit ',' C	
MOV	expr ',' '#' expr	
MOV	expr ',' '@' REG	R01
MOV	expr ',' A	
MOV	expr ',' C	
MOV	expr ',' REG	R07
MOV	expr ',' expr	
MOVC	A ',' '@' A '+' DPTR	
MOVC	A ',' '@' A '+' PC	
MOVX	'@' DPTR ',' A	
MOVX	'@' REG ',' A	R01
MOVX	A ',' '@' DPTR	
MOVX	A ',' '@' REG	R01
MUL	AB	
NOP		

Opcode	Syntax	Selection Criteria
ORL	A ',' '#' expr	
ORL	A ',' '@' REG	R01
ORL	A ',' REG	R07
ORL	A ',' expr	
ORL	C ',' '/' bit	
ORL	C ',' '/' expr	
ORL	C ',' bit	
ORL	C ',' expr	
ORL	expr ',' '#' expr	
ORL	expr ',' A	
POP	expr	
PUSH	expr	
RET		
RETI		
RL	A	
RLC	A	
RR	A	
RRC	A	
SETB	C	
SETB	bit	
SETB	expr	
SJMP	expr	
SUBB	A ',' '#' expr	
SUBB	A ',' '@' REG	R01
SUBB	A ',' REG	R07
SUBB	A ',' expr	
SWAP	A	
XCH	A ',' '@' REG	R01
XCH	A ',' REG	R07
XCH	A ',' expr	
XCHD	A ',' '@' REG	R01
XRL	A ',' '#' expr	
XRL	A ',' '@' REG	R01

Opcode	Syntax	Selection Criteria
XRL	A ',' REG	R07
XRL	A ',' expr	
XRL	expr ',' '#' expr	
XRL	expr ',' A	

A.2.2 ***Selection_Criteria_Keywords***

R01 Only the reserved symbols R0 and R1 (or r0, r1) are acceptable as the REG operand.

R07 All the reserved symbols R0 thru R7 are acceptable as the REG operand.

A.2.3 ***Bit_Operands*** Bit Addresses can be in two forms.

One form is "expression.constant" where expression gives an address for a bit addressable register and constant (in the range of 0 to 7) gives the bit number in the byte. Expression values must be either in the range of 20h to 2fh, or the multiples of 8 between 80h and f8h, and defined in the first pass.

The second form is an symbolic expression with value between 0 and 255. The value of the expression is placed in the operand field of the instruction without modification.

A.2.4 ***Apostrophes*** The apostrophes in the syntax field are a notation used for the parser generator and are not put in the assembler source statement.

A.3 **Notes**

A.3.1 ***Address_Spaces*** The Four address spaces, Program Memory, Internal Data Memory, Bit Numbers, and External Data Memory are not differentiated in the symbol table. Which address space a symbol refers to is determined by the context of the instruction operands it appears in.

A.3.2 ***Reserved_Symbols***

A.3.2.1 ***Machine_Dependent_Reserved_Symbols*** A AB C DPTR PC R0 R1 R2 R3 R4 R5 R6 R7 a ab c dptr pc r0 r1 r2 r3 r4 r5 r6 r7

A.3.2.2 ***Standard_Reserved_Symbols*** AND DEFINED EQ GE GT HIGH LE LOW LT MOD NE NOT OR SHL SHR XOR and defined eq ge gt high le low lt mod ne not or shl shr xor

CONTENTS