A. Appendix for as8048 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 ***Instruction_Set_Selection***

CPU string

The instruction set can be specified in the source file with the CPU pseudooperation. The string, delimited by quotes or apostrophes, is scanned for a substring which selects which instruction set is used. When the program is invoked, this operation is performed on the name of the program, then the -p optional arguement, if any, and then any CPU statements. The last one selects which subset of the instructions the assembler will accept. The instruction set can be changed at any place in the source file.

Instruction Set	Substrings
80C48	C48 c48 C35 c35
80C49	C49 c49 C39 c39
80C50	C50 c50 C40 c40
8048	48 35
8049	49 39
8050	50 40
8041	41
8042	42

A.1.2.2 ***Register_Set***

```
     Label REGISTER expr
```

The REGISTER statement is a version of the SET statement that checks that the value of its expression is a valid data register location for the current CPU selection. The REGISTER statement sets the symbol specified in the label field with the numeric value of the expression. The REGISTER statement can change the value of a symbol, but only if the symbol is originally defined in a previous SET or REGISTER statement.

Example

```
     dtabl   register        32

             mov     r1, #dtabl
             mov     a, @r1
```

A.2 **Instructions**

A.2.1 ***Instruction_List***

Opcode	Syntax	Selection Criteria
ADD	A ',' '#' expr	
ADD	A ',' '@' REG	
ADD	A ',' REG	
ADDC	A ',' '#' expr	
ADDC	A ',' '@' REG	
ADDC	A ',' REG	
ANL	A ',' '#' expr	
ANL	A ',' '@' REG	
ANL	A ',' REG	
ANL	BUS ',' '#' expr	INSTNOT41
ANL	P1 ',' '#' expr	
ANL	P2 ',' '#' expr	
ANLD	P47 ',' A	
CALL	expr	
CLR	A	
CLR	C	
CLR	F0	
CLR	F1	
CPL	A	
CPL	C	

Opcode	Syntax	Selection Criteria
CPL	F0	
CPL	F1	
DA	A	
DEC	A	
DEC	REG	
DIS	I	
DIS	TCNTI	
DJNZ	REG ',' expr	
EN	DMA	INST41
EN	FLAGS	INST41
EN	I	
EN	TCNTI	
ENT0	CLK	INSTNOT41
HALT		INSTIDL
IDL		INSTIDL
IN	A ',' DBB	INST41
IN	A ',' P1	
IN	A ',' P2	
INC	'@' REG	
INC	A	
INC	REG	
INS	A ',' BUS	INSTNOT41
JB0	expr	
JB1	expr	
JB2	expr	
JB3	expr	
JB4	expr	
JB5	expr	
JB6	expr	
JB7	expr	

Opcode	Syntax	Selection Criteria
JC	expr	
JF0	expr	
JF1	expr	
JMP	expr	
JMPP	'@' A	
JNC	expr	
JNI	expr	INSTNOT41
JNIBF	expr	INST41
JNT0	expr	
JNT1	expr	
JNZ	expr	
JOBF	expr	INST41
JT0	expr	
JT1	expr	
JTF	expr	
JZ	expr	
MOV	'@' REG ',' '#' expr	
MOV	'@' REG ',' A	
MOV	A ',' '#' expr	
MOV	A ',' '@' REG	
MOV	A ',' PSW	
MOV	A ',' T	
MOV	A ',' REG	
MOV	PSW ',' A	
MOV	STS ',' A	
MOV	T ',' A	
MOV	REG ',' '#' expr	
MOV	REG ',' A	
MOVD	A ',' P47	
MOVD	P47 ',' A	

Opcode	Syntax	Selection Criteria
MOVP3	A ',' '@' A	
MOVP	A ',' '@' A	
MOVX	'@' REG ',' A	INSTNOT41
MOVX	A ',' '@' REG	INSTNOT41
NOP		
ORL	A ',' '#' expr	
ORL	A ',' '@' REG	
ORL	A ',' REG	
ORL	BUS ',' '#' expr	INSTNOT41
ORL	P1 ',' '#' expr	
ORL	P2 ',' '#' expr	
ORLD	P47 ',' A	
OUT	DBB ',' A	INST41
OUTL	BUS ',' A	INSTNOT41
OUTL	P1 ',' A	
OUTL	P2 ',' A	
RET		
RETR		
RL	A	
RLC	A	
RR	A	
RRC	A	
SEL	MSELC	INSTNOT41
SEL	RSELC	
STOP	TCNT	
STRT	CNT	
STRT	T	
SWAP	A	
XCH	A ',' '@' REG	
XCH	A ',' REG	

Opcode	Syntax	Selection Criteria
XCHD	A ',' '@' REG	
XRL	A ',' '#' expr	
XRL	A ',' '@' REG	
XRL	A ',' REG	

A.2.2 ***Operands***

REG — REG can be any of r0, r1, r2, r3, r4, r5, r6, r7, unless proceeded by a

MSELC — MSELC represents the symbols mb0 and mb1.

RSELC — RSELC represents the symbols rb0 and rb1.

P47 — P47 represents the symbols p4, p5, p6, p7.

There are uppercase versions of all the reserved symbols.

A.2.3 ***Selection_Criteria_Keywords***

INSTIDL — The instruction is only available on CMOS implementations.

INSTNOT41 — The instruction is not available in the 8041.

INST41 — The instruction is only available in the 8041.

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 ***Reserved_Symbols***

A.3.1.1 ***Machine_Dependent_Reserved_Symbols*** A BUS C CLK CNT DBB DMA F0 F1 FLAGS I MB0 MB1 P1 P2 P4 P5 P6 P7 PSW R0 R1 R2 R3 R4 R5 R6 R7 RB0 RB1 STS T TCNT TCNTI a bus c clk cnt dbb dma f0 f1 flags i mb0 mb1 p1 p2 p4 p5 p6 p7 psw r0 r1 r2 r3 r4 r5 r6 r7 rb0 rb1 sts t tcnt tcnti

A.3.1.2 ***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