A. **Appendix for asz80 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	SETEQU
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 the 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
64180	180
z80	z80 Z80
8085	85
8080	80

A.2 **Instructions**

A.2.1 ***Instruction_List***

Opcode	Syntax	Selection Criteria

Opcode	Syntax	Selection Criteria
ADC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRBC
ADC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRDE
ADC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRHL
ADC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRSP
ADC	reg8 ',' '(' DREGHL ')'	REGISA
ADC	reg8 ',' index	TSZ80PLUS|DRIX|REGISA
ADC	reg8 ',' index	TSZ80PLUS|DRIY|REGISA
ADC	reg8 ',' reg8	0|REGISA
ADC	reg8 ',' topexpr	REGISA
ADD	dreg ',' dreg	DRDESTHL|DRBC
ADD	dreg ',' dreg	DRDESTHL|DRDE
ADD	dreg ',' dreg	DRDESTHL|DRHL
ADD	dreg ',' dreg	DRDESTHL|DRSP
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIX|DRBC
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIX|DRDE
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIX|DRIX
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIX|DRSP
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIY|DRBC
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIY|DRDE
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIY|DRIY
ADD	dreg ',' dreg	TSZ80PLUS|DRDESTIY|DRSP
ADD	reg8 ',' '(' DREGHL ')'	REGISA
ADD	reg8 ',' index	TSZ80PLUS|DRIX|REGISA
ADD	reg8 ',' index	TSZ80PLUS|DRIY|REGISA
ADD	reg8 ',' reg8	0|REGISA
ADD	reg8 ',' topexpr	REGISA
AND	'(' DREGHL ')'	
AND	index	TSZ80PLUS|DRIX
AND	index	TSZ80PLUS|DRIY
AND	reg8	
AND	topexpr	
BIT	expr ',' '(' DREGHL ')'	TSZ80PLUS
BIT	expr ',' index	TSZ80PLUS|DRIX
BIT	expr ',' index	TSZ80PLUS|DRIY
BIT	expr ',' reg8	TSZ80PLUS
CALL	condition ',' expr	CCSELC
CALL	condition ',' expr	CCSELM
CALL	condition ',' expr	CCSELNC
CALL	condition ',' expr	CCSELNZ
CALL	condition ',' expr	CCSELP
CALL	condition ',' expr	CCSELPE
CALL	condition ',' expr	CCSELPO
CALL	condition ',' expr	CCSELZ
CALL	expr	

Opcode	Syntax	Selection Criteria
CCF		
CP	'(' DREGHL ')'	
CP	index	TSZ80PLUS|DRIX
CP	index	TSZ80PLUS|DRIY
CP	reg8	
CP	topexpr	
CPD		TSZ80PLUS
CPDR		TSZ80PLUS
CPI		TSZ80PLUS
CPIR		TSZ80PLUS
CPL		
DAA		
DEC	'(' DREGHL ')'	
DEC	dreg	DRBC
DEC	dreg	DRDE
DEC	dreg	DRHL
DEC	dreg	DRSP
DEC	dreg	TSZ80PLUS|DRIX
DEC	dreg	TSZ80PLUS|DRIY
DEC	index	TSZ80PLUS|DRIX
DEC	index	TSZ80PLUS|DRIY
DEC	reg8	
DI		
DJNZ	topexpr	
EI		
EX	'(' DREGSP ')' ',' dreg	DRHL
EX	'(' DREGSP ')' ',' dreg	TSZ80PLUS|DRIX
EX	'(' DREGSP ')' ',' dreg	TSZ80PLUS|DRIY
EX	dreg ',' dreg	EX1DE|EX2HL
EX	dreg ',' dreg	TSZ80PLUS|EX1AF|EX2AF
EXX		TSZ80PLUS
HALT		
IM	expr	TSZ80PLUS|INTSETMODE0
IM	expr	TSZ80PLUS|INTSETMODE1

Opcode	Syntax	Selection Criteria
IM	expr	TSZ80PLUS|INTSETMODE2
IN0	reg8 ',' '(' topexpr ')'	TS64180
IN	reg8 ',' '(' REGC ')'	TSZ80PLUS
IN	reg8 ',' '(' topexpr ')'	REGISA
INC	'(' DREGHL ')'	
INC	dreg	DRBC
INC	dreg	DRDE
INC	dreg	DRHL
INC	dreg	DRSP
INC	dreg	TSZ80PLUS|DRIX
INC	dreg	TSZ80PLUS|DRIY
INC	index	TSZ80PLUS|DRIX
INC	index	TSZ80PLUS|DRIY
INC	reg8	
IND		TSZ80PLUS
INDR		TSZ80PLUS
INI		TSZ80PLUS
INIR		TSZ80PLUS
JP	'(' dreg ')'	DRHL
JP	'(' dreg ')'	TSZ80PLUS|DRIX
JP	'(' dreg ')'	TSZ80PLUS|DRIY
JP	condition ',' expr	CCSELC
JP	condition ',' expr	CCSELM
JP	condition ',' expr	CCSELNC
JP	condition ',' expr	CCSELNZ
JP	condition ',' expr	CCSELP
JP	condition ',' expr	CCSELPE
JP	condition ',' expr	CCSELPO
JP	condition ',' expr	CCSELZ
JP	expr	
JR	condition ',' expr	CCSELC|TSZ80PLUS
JR	condition ',' expr	CCSELNC|TSZ80PLUS
JR	condition ',' expr	CCSELNZ|TSZ80PLUS
JR	condition ',' expr	CCSELZ|TSZ80PLUS
JR	expr	TSZ80PLUS
LD	'(' dreg ')' ',' reg8	DRBC|REGISA
LD	'(' dreg ')' ',' reg8	DRDE|REGISA
LD	'(' dreg ')' ',' reg8	DRHL
LD	'(' dreg ')' ',' topexpr	DRHL

Opcode	Syntax	Selection Criteria
LD	'(' topexpr ')' ',' REGA	
LD	'(' topexpr ')' ',' dreg	DRHL
LD	'(' topexpr ')' ',' dreg	TSZ80PLUS|DRBC
LD	'(' topexpr ')' ',' dreg	TSZ80PLUS|DRDE
LD	'(' topexpr ')' ',' dreg	TSZ80PLUS|DRIX
LD	'(' topexpr ')' ',' dreg	TSZ80PLUS|DRIY
LD	'(' topexpr ')' ',' dreg	TSZ80PLUS|DRSP
LD	dreg ',' '(' topexpr ')'	DRHL
LD	dreg ',' '(' topexpr ')'	TSZ80PLUS|DRBC
LD	dreg ',' '(' topexpr ')'	TSZ80PLUS|DRDE
LD	dreg ',' '(' topexpr ')'	TSZ80PLUS|DRIX
LD	dreg ',' '(' topexpr ')'	TSZ80PLUS|DRIY
LD	dreg ',' '(' topexpr ')'	TSZ80PLUS|DRSP
LD	dreg ',' dreg	TSZ80PLUS|DRHL|DRDESTSP
LD	dreg ',' dreg	TSZ80PLUS|DRIX|DRDESTSP
LD	dreg ',' dreg	TSZ80PLUS|DRIY|DRDESTSP
LD	dreg ',' topexpr	DRBC
LD	dreg ',' topexpr	DRDE
LD	dreg ',' topexpr	DRHL
LD	dreg ',' topexpr	DRSP
LD	dreg ',' topexpr	TSZ80PLUS|DRIX
LD	dreg ',' topexpr	TSZ80PLUS|DRIY
LD	index ',' expr	TSZ80PLUS|DRIX
LD	index ',' expr	TSZ80PLUS|DRIY
LD	index ',' reg8	TSZ80PLUS|DRIX
LD	index ',' reg8	TSZ80PLUS|DRIY
LD	reg8 ',' '(' dreg ')'	DRBC|REGISA
LD	reg8 ',' '(' dreg ')'	DRDE|REGISA
LD	reg8 ',' '(' dreg ')'	DRHL
LD	reg8 ',' '(' topexpr ')'	REGISA
LD	reg8 ',' index	TSZ80PLUS|DRIX
LD	reg8 ',' index	TSZ80PLUS|DRIY
LD	reg8 ',' reg8	
LD	reg8 ',' specialr	TSZ80PLUS|SPECIALIR|REGISA
LD	reg8 ',' specialr	TSZ80PLUS|SPECIALRR|REGISA
LD	reg8 ',' topexpr	
LD	specialr ',' REGA	TSZ80PLUS|SPECIALIR
LD	specialr ',' REGA	TSZ80PLUS|SPECIALRR
LDD		TSZ80PLUS
LDDR		TSZ80PLUS
LDI		TSZ80PLUS
LDIR		TSZ80PLUS
MULT	dreg	TS64180|DRBC
MULT	dreg	TS64180|DRDE

Opcode	Syntax	Selection Criteria
MULT	dreg	TS64180|DRHL
MULT	dreg	TS64180|DRSP
NEG		TSZ80PLUS
NOP		
OR	'(' DREGHL ')'	
OR	index	TSZ80PLUS|DRIX
OR	index	TSZ80PLUS|DRIY
OR	reg8	
OR	topexpr	
OTDM		TS64180
OTDMR		TS64180
OTDR		TSZ80PLUS
OTIM		TS64180
OTIMR		TS64180
OTIR		TSZ80PLUS
OUT0	'(' topexpr ')' ',' reg8	TS64180
OUT	'(' REGC ')' ',' reg8	TSZ80PLUS
OUT	'(' topexpr ')' ',' reg8	REGISA
OUTD		TSZ80PLUS
OUTI		TSZ80PLUS
POP	dreg	DRAF
POP	dreg	DRBC
POP	dreg	DRDE
POP	dreg	DRHL
POP	dreg	TSZ80PLUS|DRIX
POP	dreg	TSZ80PLUS|DRIY
PUSH	dreg	DRAF
PUSH	dreg	DRBC
PUSH	dreg	DRDE
PUSH	dreg	DRHL
PUSH	dreg	TSZ80PLUS|DRIX
PUSH	dreg	TSZ80PLUS|DRIY
RES	expr ',' '(' DREGHL ')'	TSZ80PLUS

Opcode	Syntax	Selection Criteria
RES	expr ',' index	TSZ80PLUS|DRIX
RES	expr ',' index	TSZ80PLUS|DRIY
RES	expr ',' reg8	TSZ80PLUS
RET		
RET	condition	CCSELC
RET	condition	CCSELM
RET	condition	CCSELNC
RET	condition	CCSELNZ
RET	condition	CCSELP
RET	condition	CCSELPE
RET	condition	CCSELPO
RET	condition	CCSELZ
RETI		TSZ80PLUS
RETN		TSZ80PLUS
RIM		CPU8085
RL	'(' DREGHL ')'	TSZ80PLUS
RL	index	TSZ80PLUS|DRIX
RL	index	TSZ80PLUS|DRIY
RL	reg8	TSZ80PLUS
RLA		
RLC	'(' DREGHL ')'	TSZ80PLUS
RLC	index	TSZ80PLUS|DRIX
RLC	index	TSZ80PLUS|DRIY
RLC	reg8	TSZ80PLUS
RLCA		
RLD		TSZ80PLUS
RR	'(' DREGHL ')'	TSZ80PLUS
RR	index	TSZ80PLUS|DRIX
RR	index	TSZ80PLUS|DRIY
RR	reg8	TSZ80PLUS
RRA		
RRC	'(' DREGHL ')'	TSZ80PLUS
RRC	index	TSZ80PLUS|DRIX
RRC	index	TSZ80PLUS|DRIY
RRC	reg8	TSZ80PLUS
RRCA		

Opcode	Syntax	Selection Criteria
RRD		TSZ80PLUS
RST	expr	
SBC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRBC
SBC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRDE
SBC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRHL
SBC	dreg ',' dreg	TSZ80PLUS|DRDESTHL|DRSP
SBC	reg8 ',' '(' DREGHL ')'	REGISA
SBC	reg8 ',' index	TSZ80PLUS|DRIX|REGISA
SBC	reg8 ',' index	TSZ80PLUS|DRIY|REGISA
SBC	reg8 ',' reg8	0|REGISA
SBC	reg8 ',' topexpr	REGISA
SCF		
SET	expr ',' '(' DREGHL ')'	TSZ80PLUS
SET	expr ',' index	TSZ80PLUS|DRIX
SET	expr ',' index	TSZ80PLUS|DRIY
SET	expr ',' reg8	TSZ80PLUS
SIM		CPU8085
SLA	'(' DREGHL ')'	TSZ80PLUS
SLA	index	TSZ80PLUS|DRIX
SLA	index	TSZ80PLUS|DRIY
SLA	reg8	TSZ80PLUS
SLP		TS64180
SRA	'(' DREGHL ')'	TSZ80PLUS
SRA	index	TSZ80PLUS|DRIX
SRA	index	TSZ80PLUS|DRIY
SRA	reg8	TSZ80PLUS
SRL	'(' DREGHL ')'	TSZ80PLUS
SRL	index	TSZ80PLUS|DRIX
SRL	index	TSZ80PLUS|DRIY
SRL	reg8	TSZ80PLUS
SUB	'(' DREGHL ')'	
SUB	index	TSZ80PLUS|DRIX
SUB	index	TSZ80PLUS|DRIY
SUB	reg8	
SUB	topexpr	
TST	'(' DREGHL ')'	TS64180
TST	reg8	TS64180

Opcode	Syntax	Selection Criteria
TST	topexpr	TS64180
TSTIO	topexpr	TS64180
XOR	'(' DREGHL ')'	
XOR	index	TSZ80PLUS|DRIX
XOR	index	TSZ80PLUS|DRIY
XOR	reg8	
XOR	topexpr	

A.2.2 ***Selection_Criteria_Keywords***

CPU8085	Instruction is only implemented for the 8085.
TSZ80PLUS	Instruction is implemented in the z80 and 64180 instruction sets.
TS64180	Instruction is only implemented in the 64180
DRIX	Instruction refers to the IX index register
DRIY	Instruction refers to the IY index register
DRSP	Instruction refers to the Stack Pointer
DRHL	Instruction refers to the HL register
DRDE	Instruction refers to the DE register
DRBC	Instruction refers to the BC register
DRAF	Instruction refers to the AF concatenated register
DRDESTSP	Instruction which refers to two double register operands, uses the Stack Pointer as the destination.
DRDESTHL	Instruction which refers to two double register operands, uses the HL double register as the

	destination.
DRDESTIX	Instruction which refers to two double register operands, uses the IX index register as the destination.
DRDESTIY	Instruction which refers to two double register operands, uses the IY index register as the destination.
REGISA	Instruction is restricted to using the A register for an 8 bit register operand.
CCSELNZ	Instruction uses NonZero condition.
CCSELZ	Instruction uses Zero condition.
CCSELNC	Instruction uses No Carry condition.
CCSELC	Instruction uses Carry condition.
CCSELPO	Instruction uses Parity Odd condition.
CCSELPE	Instruction uses Parity Even condition.
CCSELP	Instruction uses Plus condition.
CCSELM	Instruction uses Minus condition.
EX1AF	First operand of an Ex instruction is the AF register.
EX1DE	First operand of an Ex instruction is the DE register.
EX2AF	Second operand of an Ex instruction is the AF register.
EX2HL	Second operand of an Ex instruction is the HL register.
SPECIALIR	Instruction uses the I special register.

SPECIALRR	Instruction uses the R special register.

A.2.3 ***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 ***Conditions*** Conditions are represented by the reserved symbols z, nz, nc, pe, po, p, m, and c, and their uppercase versions.

A.3.2 ***Indexed_Addressing*** Index addressing uses the format "(index register + expression)" where index register is IX or IY.

A.3.3 ***Top_Expressions*** The syntax of some of the instructions combined with the standard expression syntax resulted in confusion whether an operand was an expression surrounded by parenthesis, or a memory reference. To get around this, the expressions in these ambiguous cases were restricted to those forms of expression that don't have surrounding parenthesis at the top level. Subexpressions may be parenthesized, but only if an operator seperates or precedes the subexpression.

Example

```
ld a, (47h)     ; load from memory address 0x47
ld a, +(47h)    ; load immediate value 0x47
ld a, ((47h))   ; error
```

A.3.4 ***dreg,_reg8,_specialr*** Double registers (dreg) are the set of symbols af, bc, de, hl, ix, iy, and sp and their uppercase versions.

Eight bit registers (reg8) are the set of symbols a, b, c, d, e, h, l, and their uppercase versions.

Special registers are i, r, I, R.

A.3.5 ***Bit_Numbers*** The bit number expression in the BIT, RES, and SET operations has to have value defined when the instruction is read in the first pass. The value must be between 0 and 7.

A.3.6 ***Reserved_Symbols***

A.3.6.1 ***Machine_Dependent_Reserved_Symbols*** A AF B BC C D DE E H HL I IX IY L M NC NZ P PE PO R SP Z a af b bc c d de e h hl i ix iy l m nc nz p pe po r sp z

A.3.6.2 ***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