A. **Appendix for as2650 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	RES RESERVE RMB
Define Byte Data	BYTE DATA 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_Address_Constant***

```
[Label] ACON expression [, expression] ...
```

The acon statement generates a two byte integer for each expression in the expression list. Each expression is limited in value from 0 to 32767, values outside this range will result in an "expression exceeds available field width" error. There can be up to 128 expressions on the line, within the line length limit. The optional label is set to the address of the first expression.

A.2 **Instructions**

A.2.1 ***Instruction_List***

Opcode	Syntax	Selection Criteria
ADDA	',' REG expr	
ADDA	',' REG '*' expr ',' REG ',' '+'	
ADDA	',' REG '*' expr ',' REG ',' '-'	
ADDA	',' REG '*' expr ',' REG	
ADDA	',' REG '*' expr	
ADDA	',' REG expr ',' REG ',' '+'	
ADDA	',' REG expr ',' REG ',' '-'	
ADDA	',' REG expr ',' REG	

Opcode	Syntax	Selection Criteria
ADDI	',' REG expr	
ADDR	',' REG '*' expr	
ADDR	',' REG expr	
ADDZ	REG	
ANDA	',' REG expr	
ANDA	',' REG '*' expr ',' REG ',' '+'	
ANDA	',' REG '*' expr ',' REG ',' '-'	
ANDA	',' REG '*' expr ',' REG	
ANDA	',' REG '*' expr	
ANDA	',' REG expr ',' REG ',' '+'	
ANDA	',' REG expr ',' REG ',' '-'	
ANDA	',' REG expr ',' REG	
ANDI	',' REG expr	
ANDR	',' REG '*' expr	
ANDR	',' REG expr	
ANDZ	REG	REG1
ANDZ	REG	REG2
ANDZ	REG	REG3
BCFA	',' CONDITION '*' expr	COND0
BCFA	',' CONDITION '*' expr	COND1
BCFA	',' CONDITION '*' expr	COND2
BCFA	',' CONDITION expr	COND0
BCFA	',' CONDITION expr	COND1
BCFA	',' CONDITION expr	COND2
BCFR	',' CONDITION '*' expr	COND0
BCFR	',' CONDITION '*' expr	COND1
BCFR	',' CONDITION '*' expr	COND2
BCFR	',' CONDITION expr	COND0
BCFR	',' CONDITION expr	COND1
BCFR	',' CONDITION expr	COND2
BCTA	',' CONDITION '*' expr	
BCTA	',' CONDITION expr	
BCTR	',' CONDITION '*' expr	
BCTR	',' CONDITION expr	
BDRA	',' REG '*' expr	
BDRA	',' REG expr	

Opcode	Syntax	Selection Criteria
BDRR	',' REG '*' expr	
BDRR	',' REG expr	
BIRA	',' REG '*' expr	
BIRA	',' REG expr	
BIRR	',' REG '*' expr	
BIRR	',' REG expr	
BRNA	',' REG '*' expr	
BRNA	',' REG expr	
BRNR	',' REG '*' expr	
BRNR	',' REG expr	
BSFA	',' CONDITION '*' expr	COND0
BSFA	',' CONDITION '*' expr	COND1
BSFA	',' CONDITION '*' expr	COND2
BSFA	',' CONDITION expr	COND0
BSFA	',' CONDITION expr	COND1
BSFA	',' CONDITION expr	COND2
BSFR	',' CONDITION '*' expr	COND0
BSFR	',' CONDITION '*' expr	COND1
BSFR	',' CONDITION '*' expr	COND2
BSFR	',' CONDITION expr	COND0
BSFR	',' CONDITION expr	COND1
BSFR	',' CONDITION expr	COND2
BSNA	',' REG '*' expr	
BSNA	',' REG expr	
BSNR	',' REG '*' expr	
BSNR	',' REG expr	
BSTA	',' CONDITION '*' expr	
BSTA	',' CONDITION expr	
BSTR	',' CONDITION '*' expr	
BSTR	',' CONDITION expr	
BSXA	'*' expr ',' REG	REG3
BSXA	expr ',' REG	REG3
BXA	'*' expr ',' REG	REG3
BXA	expr ',' REG	REG3
COMA	',' REG expr	
COMA	',' REG '*' expr ',' REG ',' '+'	

Opcode	Syntax	Selection Criteria
COMA	',' REG '*' expr ',' REG ',' '-'	
COMA	',' REG '*' expr ',' REG	
COMA	',' REG '*' expr	
COMA	',' REG expr ',' REG ',' '+'	
COMA	',' REG expr ',' REG ',' '-'	
COMA	',' REG expr ',' REG	
COMI	',' REG expr	
COMR	',' REG '*' expr	
COMR	',' REG expr	
COMZ	REG	
CPSL	expr	
CPSU	expr	
DAR	',' REG	
EORA	',' REG expr	
EORA	',' REG '*' expr ',' REG ',' '+'	
EORA	',' REG '*' expr ',' REG ',' '-'	
EORA	',' REG '*' expr ',' REG	
EORA	',' REG '*' expr	
EORA	',' REG expr ',' REG ',' '+'	
EORA	',' REG expr ',' REG ',' '-'	
EORA	',' REG expr ',' REG	
EORI	',' REG expr	
EORR	',' REG '*' expr	
EORR	',' REG expr	
EORZ	REG	
HALT		
IORA	',' REG expr	
IORA	',' REG '*' expr ',' REG ',' '+'	
IORA	',' REG '*' expr ',' REG ',' '-'	
IORA	',' REG '*' expr ',' REG	
IORA	',' REG '*' expr	
IORA	',' REG expr ',' REG ',' '+'	
IORA	',' REG expr ',' REG ',' '-'	
IORA	',' REG expr ',' REG	
IORI	',' REG expr	

Opcode	Syntax	Selection Criteria
IORR	',' REG '*' expr	
IORR	',' REG expr	
IORZ	REG	
LODA	',' REG expr	
LODA	',' REG '*' expr ',' REG ',' '+'	
LODA	',' REG '*' expr ',' REG ',' '-'	
LODA	',' REG '*' expr ',' REG	
LODA	',' REG '*' expr	
LODA	',' REG expr ',' REG ',' '+'	
LODA	',' REG expr ',' REG ',' '-'	
LODA	',' REG expr ',' REG	
LODI	',' REG expr	
LODR	',' REG '*' expr	
LODR	',' REG expr	
LODZ	REG	REG0
LODZ	REG	REG1
LODZ	REG	REG2
LODZ	REG	REG3
LPSL		
LPSU		
NOP		
PPSL	expr	
PPSU	expr	
REDC	',' REG	
REDD	',' REG	
REDE	',' REG expr	
RETC	',' CONDITION	
RETE	',' CONDITION	
RRL	',' REG	
RRR	',' REG	
SPSL		

Opcode	Syntax	Selection Criteria
SPSU		
STRA	',' REG expr	
STRA	',' REG '*' expr ',' REG ',' '+'	
STRA	',' REG '*' expr ',' REG ',' '-'	
STRA	',' REG '*' expr ',' REG	
STRA	',' REG '*' expr	
STRA	',' REG expr ',' REG ',' '+'	
STRA	',' REG expr ',' REG ',' '-'	
STRA	',' REG expr ',' REG	
STRR	',' REG '*' expr	
STRR	',' REG expr	
STRZ	REG	REG1
STRZ	REG	REG2
STRZ	REG	REG3
SUBA	',' REG expr	
SUBA	',' REG '*' expr ',' REG ',' '+'	
SUBA	',' REG '*' expr ',' REG ',' '-'	
SUBA	',' REG '*' expr ',' REG	
SUBA	',' REG '*' expr	
SUBA	',' REG expr ',' REG ',' '+'	
SUBA	',' REG expr ',' REG ',' '-'	
SUBA	',' REG expr ',' REG	
SUBI	',' REG expr	
SUBR	',' REG '*' expr	
SUBR	',' REG expr	
SUBZ	REG	
TMI	',' REG expr	
TPSL	expr	
TPSU	expr	
WRTC	',' REG	
WRTD	',' REG	
WRTE	',' REG expr	
ZBRR	'*' expr	
ZBRR	expr	

Opcode	Syntax	Selection Criteria
ZBSR	'*' expr	
ZBSR	expr	

A.2.2 ***Register_and_Condition_Reserved_Symbols*** The REG and CONDITION fields in instructions are restricted to only the values available as reserved symbols. These are r0 - r3, or R0 - R3 for registers, and plus, PLUS, minus, MINUS, zero, ZERO, gt, GT, lt, LT, eq, EQ, un, UN, always, and ALWAYS for conditions.

A.2.3 ***Destination_Register_for_Indexed_Addressing*** If the destination register, the one adjacent to the opcode, is not R0 when one of the absolute indexed addressing modes is used, an error is issued.

A.2.4 ***Selection_Criteria_Keywords***

REG0 REG1 REG2 REG3

Some instructions are restricted to a subset of the registers. Only those instructions with the specified registers are valid.

COND0 COND1 COND2 COND3

Some instructions are restricted to a subset of the conditions. Only those instructions with the specified conditions are valid.

COND0	EQ, or ZERO
COND1	GT, or PLUS
COND2	LT, or MINUS
COND3	ALWAYS, or UN

A.2.5 ***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 ***Location_Counter*** The Dollar Sign is used as the location counter symbol in this assembler.

A.3.2 ***Memory_Addressing_Error_Messages***

ERROR - expression exceeds available field width

This message occurs when :

- The absolute branch address is to an invalid address, one greater than 32767.

- The absolute or relative address was not in the present memory page.

- The relative offset was outside the range that can be represented in 7 bits.

ERROR - instruction crosses page boundary

The first and last bytes of an instruction are on different sides of an 8k page.

WARNING - Page Boundary

The first byte of an instruction is on a 8k page boundry.

A.3.3 ***Page_Wraparound_for_Relative_Addressing*** The wraparound of an effective address, where a relative address from an instruction near a page boundary accesses memory at the other side of the page, is not supported in this assembler and will result in one or more error messages.

This does not apply to the ZBRR and ZBSR instructions. Memory destinations for these can range from 0 to $3f and $1fc0 to $1fff.

A.3.4 ***Relational_Operators*** The relational operator keywords GT, LT, and EQ are not available. The '<', '>', and '==' special character representations must be used.

A.3.5 ***Reserved_Symbols***

A.3.5.1 ***Machine_Dependent_Reserved_Symbols*** ALWAYS AND DEFINED EQ GE GT HIGH LE LOW LT MINUS MOD NE NOT OR PLUS R0 R1 R2 R3 SHL SHR UN XOR ZERO always and defined eq ge gt high le low lt minus mod ne not or plus r0 r1 r2 r3 shl shr un xor zero

CONTENTS