

## A. Appendix for as6301 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.2 Instructions

#### A.2.1 *Instruction\_List*

Opcode	Syntax	Selection Criteria
ABA		
ABX		
ADC	ACCUM '#' expr	REGA
ADC	ACCUM '#' expr	REGB
ADC	ACCUM expr	REGA+DIRECT
ADC	ACCUM expr	REGA+EXTENDED
ADC	ACCUM expr	REGB+DIRECT
ADC	ACCUM expr	REGB+EXTENDED
ADC	ACCUM indexed	REGA
ADC	ACCUM indexed	REGB
ADCA	'#' expr	
ADCA	expr	DIRECT
ADCA	expr	EXTENDED
ADCA	indexed	
ADCB	'#' expr	
ADCB	expr	DIRECT
ADCB	expr	EXTENDED
ADCB	indexed	

Opcode	Syntax	Selection Criteria
ADD	ACCUM '#' expr	REGA
ADD	ACCUM '#' expr	REGB
ADD	ACCUM expr	REGA+DIRECT
ADD	ACCUM expr	REGA+EXTENDED
ADD	ACCUM expr	REGB+DIRECT
ADD	ACCUM expr	REGB+EXTENDED
ADD	ACCUM indexed	REGA
ADD	ACCUM indexed	REGB
ADDA	'#' expr	
ADDA	expr	DIRECT
ADDA	expr	EXTENDED
ADDA	indexed	
ADDB	'#' expr	
ADDB	expr	DIRECT
ADDB	expr	EXTENDED
ADDB	indexed	
ADDD	'#' expr	
ADDD	expr	DIRECT
ADDD	expr	EXTENDED
ADDD	indexed	
AIM	'#' expr ',' expr	
AIM	'#' expr ',' indexed	
AND	ACCUM '#' expr	REGA
AND	ACCUM '#' expr	REGB
AND	ACCUM expr	REGA+DIRECT
AND	ACCUM expr	REGA+EXTENDED
AND	ACCUM expr	REGB+DIRECT
AND	ACCUM expr	REGB+EXTENDED
AND	ACCUM indexed	REGA
AND	ACCUM indexed	REGB
ANDA	'#' expr	
ANDA	expr	DIRECT
ANDA	expr	EXTENDED
ANDA	indexed	
ANDB	'#' expr	
ANDB	expr	DIRECT
ANDB	expr	EXTENDED
ANDB	indexed	
ASL	ACCUM	REGA
ASL	ACCUM	REGB

Opcode	Syntax	Selection Criteria
ASL	expr	
ASL	indexed	
ASLA		
ASLB		
ASLD		
ASR	ACCUM	REGA
ASR	ACCUM	REGB
ASR	expr	
ASR	indexed	
ASRA		
ASRB		
BCC	expr	
BCLR	expr ',' expr	BIT0
BCLR	expr ',' expr	BIT1
BCLR	expr ',' expr	BIT2
BCLR	expr ',' expr	BIT3
BCLR	expr ',' expr	BIT4
BCLR	expr ',' expr	BIT5
BCLR	expr ',' expr	BIT6
BCLR	expr ',' expr	BIT7
BCLR	expr ',' indexed	BIT0
BCLR	expr ',' indexed	BIT1
BCLR	expr ',' indexed	BIT2
BCLR	expr ',' indexed	BIT3
BCLR	expr ',' indexed	BIT4
BCLR	expr ',' indexed	BIT5
BCLR	expr ',' indexed	BIT6
BCLR	expr ',' indexed	BIT7
BCS	expr	
BEQ	expr	
BGE	expr	
BGT	expr	
BHI	expr	
BHS	expr	

Opcode	Syntax	Selection Criteria
BIT	ACCUM '#' expr	REGA
BIT	ACCUM '#' expr	REGB
BIT	ACCUM expr	REGA+DIRECT
BIT	ACCUM expr	REGA+EXTENDED
BIT	ACCUM expr	REGB+DIRECT
BIT	ACCUM expr	REGB+EXTENDED
BIT	ACCUM indexed	REGA
BIT	ACCUM indexed	REGB
BITA	'#' expr	
BITA	expr	DIRECT
BITA	expr	EXTENDED
BITA	indexed	
BITB	'#' expr	
BITB	expr	DIRECT
BITB	expr	EXTENDED
BITB	indexed	
BLE	expr	
BLO	expr	
BLS	expr	
BLT	expr	
BMI	expr	
BNE	expr	
BPL	expr	
BRA	expr	
BRN	expr	
BSET	expr ',' expr	BIT0
BSET	expr ',' expr	BIT1
BSET	expr ',' expr	BIT2
BSET	expr ',' expr	BIT3
BSET	expr ',' expr	BIT4
BSET	expr ',' expr	BIT5
BSET	expr ',' expr	BIT6
BSET	expr ',' expr	BIT7
BSET	expr ',' indexed	BIT0
BSET	expr ',' indexed	BIT1
BSET	expr ',' indexed	BIT2
BSET	expr ',' indexed	BIT3

Opcode	Syntax	Selection Criteria
BSET	expr ', ' indexed	BIT4
BSET	expr ', ' indexed	BIT5
BSET	expr ', ' indexed	BIT6
BSET	expr ', ' indexed	BIT7
BSR	expr	
BTGL	expr ', ' expr	BIT0
BTGL	expr ', ' expr	BIT1
BTGL	expr ', ' expr	BIT2
BTGL	expr ', ' expr	BIT3
BTGL	expr ', ' expr	BIT4
BTGL	expr ', ' expr	BIT5
BTGL	expr ', ' expr	BIT6
BTGL	expr ', ' expr	BIT7
BTGL	expr ', ' indexed	BIT0
BTGL	expr ', ' indexed	BIT1
BTGL	expr ', ' indexed	BIT2
BTGL	expr ', ' indexed	BIT3
BTGL	expr ', ' indexed	BIT4
BTGL	expr ', ' indexed	BIT5
BTGL	expr ', ' indexed	BIT6
BTGL	expr ', ' indexed	BIT7
BTST	expr ', ' expr	BIT0
BTST	expr ', ' expr	BIT1
BTST	expr ', ' expr	BIT2
BTST	expr ', ' expr	BIT3
BTST	expr ', ' expr	BIT4
BTST	expr ', ' expr	BIT5
BTST	expr ', ' expr	BIT6
BTST	expr ', ' expr	BIT7
BTST	expr ', ' indexed	BIT0
BTST	expr ', ' indexed	BIT1
BTST	expr ', ' indexed	BIT2
BTST	expr ', ' indexed	BIT3
BTST	expr ', ' indexed	BIT4
BTST	expr ', ' indexed	BIT5
BTST	expr ', ' indexed	BIT6
BTST	expr ', ' indexed	BIT7
BVC	expr	
BVS	expr	
CBA		
CLC		

Opcode	Syntax	Selection Criteria
CLI		
CLR	ACCUM	REGA
CLR	ACCUM	REGB
CLR	expr	
CLR	indexed	
CLRA		
CLRB		
CLV		
CMP	ACCUM '#' expr	REGA
CMP	ACCUM '#' expr	REGB
CMP	ACCUM expr	REGA+DIRECT
CMP	ACCUM expr	REGA+EXTENDED
CMP	ACCUM expr	REGB+DIRECT
CMP	ACCUM expr	REGB+EXTENDED
CMP	ACCUM indexed	REGA
CMP	ACCUM indexed	REGB
CMPA	'#' expr	
CMPA	expr	DIRECT
CMPA	expr	EXTENDED
CMPA	indexed	
CMPB	'#' expr	
CMPB	expr	DIRECT
CMPB	expr	EXTENDED
CMPB	indexed	
COM	ACCUM	REGA
COM	ACCUM	REGB
COM	expr	
COM	indexed	
COMA		
COMB		
CPX	'#' expr	
CPX	expr	DIRECT
CPX	expr	EXTENDED
CPX	indexed	
DAA		
DEC	ACCUM	REGA

Opcode	Syntax	Selection Criteria
DEC	ACCUM	REGB
DEC	expr	
DEC	indexed	
DECA		
DECB		
DES		
DEX		
EIM	'#' expr ',' expr	
EIM	'#' expr ',' indexed	
EOR	ACCUM '#' expr	REGA
EOR	ACCUM '#' expr	REGB
EOR	ACCUM expr	REGA+DIRECT
EOR	ACCUM expr	REGA+EXTENDED
EOR	ACCUM expr	REGB+DIRECT
EOR	ACCUM expr	REGB+EXTENDED
EOR	ACCUM indexed	REGA
EOR	ACCUM indexed	REGB
EORA	'#' expr	
EORA	expr	DIRECT
EORA	expr	EXTENDED
EORA	indexed	
EORB	'#' expr	
EORB	expr	DIRECT
EORB	expr	EXTENDED
EORB	indexed	
INC	ACCUM	REGA
INC	ACCUM	REGB
INC	expr	
INC	indexed	
INCA		
INCB		
INS		
INX		
JMP	expr	
JMP	indexed	

Opcode	Syntax	Selection Criteria
JSR	expr	DIRECT
JSR	expr	EXTENDED
JSR	indexed	
LDA	ACCUM '#' expr	REGA
LDA	ACCUM '#' expr	REGB
LDA	ACCUM expr	REGA+DIRECT
LDA	ACCUM expr	REGA+EXTENDED
LDA	ACCUM expr	REGB+DIRECT
LDA	ACCUM expr	REGB+EXTENDED
LDA	ACCUM indexed	REGA
LDA	ACCUM indexed	REGB
LDAA	'#' expr	
LDAA	expr	DIRECT
LDAA	expr	EXTENDED
LDAA	indexed	
LDAB	'#' expr	
LDAB	expr	DIRECT
LDAB	expr	EXTENDED
LDAB	indexed	
LDD	'#' expr	
LDD	expr	DIRECT
LDD	expr	EXTENDED
LDD	indexed	
LDS	'#' expr	
LDS	expr	DIRECT
LDS	expr	EXTENDED
LDS	indexed	
LDX	'#' expr	
LDX	expr	DIRECT
LDX	expr	EXTENDED
LDX	indexed	
LSL	ACCUM	REGA
LSL	ACCUM	REGB
LSL	expr	
LSL	indexed	
LSLA		
LSLB		
LSLD		



Opcode	Syntax	Selection Criteria
LSR	ACCUM	REGA
LSR	ACCUM	REGB
LSR	expr	
LSR	indexed	
LSRA		
LSRB		
LSRD		
MUL		
NEG	ACCUM	REGA
NEG	ACCUM	REGB
NEG	expr	
NEG	indexed	
NEGA		
NEGB		
NOP		
OIM	'#' expr ',' expr	
OIM	'#' expr ',' indexed	
ORA	ACCUM '#' expr	REGA
ORA	ACCUM '#' expr	REGB
ORA	ACCUM expr	REGA+DIRECT
ORA	ACCUM expr	REGA+EXTENDED
ORA	ACCUM expr	REGB+DIRECT
ORA	ACCUM expr	REGB+EXTENDED
ORA	ACCUM indexed	REGA
ORA	ACCUM indexed	REGB
ORAA	'#' expr	
ORAA	expr	DIRECT
ORAA	expr	EXTENDED
ORAA	indexed	
ORAB	'#' expr	
ORAB	expr	DIRECT
ORAB	expr	EXTENDED
ORAB	indexed	
PSH	ACCUM	REGA
PSH	ACCUM	REGB

Opcode	Syntax	Selection Criteria
PSH	INDEX	
PSHA		
PSHB		
PSHX		
PUL	ACCUM	REGA
PUL	ACCUM	REGB
PUL	INDEX	
PULA		
PULB		
PULX		
ROL	ACCUM	REGA
ROL	ACCUM	REGB
ROL	expr	
ROL	indexed	
ROLA		
ROLB		
ROR	ACCUM	REGA
ROR	ACCUM	REGB
ROR	expr	
ROR	indexed	
RORA		
RORB		
RTI		
RTS		
SBA		
SBC	ACCUM '#' expr	REGA
SBC	ACCUM '#' expr	REGB
SBC	ACCUM expr	REGA+DIRECT
SBC	ACCUM expr	REGA+EXTENDED
SBC	ACCUM expr	REGB+DIRECT
SBC	ACCUM expr	REGB+EXTENDED
SBC	ACCUM indexed	REGA

Opcode	Syntax	Selection Criteria
SBC	ACCUM indexed	REGB
SBCA	'#' expr	
SBCA	expr	DIRECT
SBCA	expr	EXTENDED
SBCA	indexed	
SBCB	'#' expr	
SBCB	expr	DIRECT
SBCB	expr	EXTENDED
SBCB	indexed	
SEC		
SEI		
SEV		
SLP		
STA	ACCUM expr	REGA+DIRECT
STA	ACCUM expr	REGA+EXTENDED
STA	ACCUM expr	REGB+DIRECT
STA	ACCUM expr	REGB+EXTENDED
STA	ACCUM indexed	REGA
STA	ACCUM indexed	REGB
STAA	expr	DIRECT
STAA	expr	EXTENDED
STAA	indexed	
STAB	expr	DIRECT
STAB	expr	EXTENDED
STAB	indexed	
STD	expr	DIRECT
STD	expr	EXTENDED
STD	indexed	
STS	expr	DIRECT
STS	expr	EXTENDED
STS	indexed	
STX	expr	DIRECT
STX	expr	EXTENDED
STX	indexed	
SUB	ACCUM '#' expr	REGA
SUB	ACCUM '#' expr	REGB

Opcode	Syntax	Selection Criteria
SUB	ACCUM expr	REGA+DIRECT
SUB	ACCUM expr	REGA+EXTENDED
SUB	ACCUM expr	REGB+DIRECT
SUB	ACCUM expr	REGB+EXTENDED
SUB	ACCUM indexed	REGA
SUB	ACCUM indexed	REGB
SUBA	'#' expr	
SUBA	expr	DIRECT
SUBA	expr	EXTENDED
SUBA	indexed	
SUBB	'#' expr	
SUBB	expr	DIRECT
SUBB	expr	EXTENDED
SUBB	indexed	
SUBD	'#' expr	
SUBD	expr	DIRECT
SUBD	expr	EXTENDED
SUBD	indexed	
SWI		
TAB		
TAP		
TBA		
TIM	'#' expr ',' expr	
TIM	'#' expr ',' indexed	
TPA		
TST	ACCUM	REGA
TST	ACCUM	REGB
TST	expr	
TST	indexed	
TSTA		
TSTB		
TSX		
TXS		
WAI		

Opcode	Syntax	Selection Criteria
--------	--------	--------------------

XGDX

#### A.2.2 **Selection\_Criteria\_Keywords**

REGA	The A accumulator can be used for the instruction.
REGB	The B accumulator can be used for the instruction.
DIRECT	The instruction can use the direct addressing mode. The expression must have a value between 0 and 255 that is defined when the instruction is processed in the input pass.
EXTENDED	The instruction can use the extended addressing mode.

#### A.2.3 **Notes**

A.2.3.1 **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.2.3.2 **Bit\_Numbers** The bit number expression in the BSET, BCLR, BTGL, BTST operations has to have value defined when the instruction is read in the first pass. The value must be between 0 and 7.

A.2.3.3 **Indexed Addressing** The indexed addressing is represented in two ways. Either "expression, X" or "X, expression" is accepted by the assembler.

#### A.3 **Reserved Symbols**

A.3.1 **Machine\_Dependent\_Reserved\_Symbols** A B X a b x

A.3.2 **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

A.	Appendix for as6301 Frankenstein Assembler.....	1
A.1	Pseudo Operations.....	1
A.2	Instructions.....	1
A.3	Reserved Symbols.....	13