

**Difference in Specification of Softune
Assembler (*FASM907S*) and Old Assembler
(*ASM907A*)**

(F²MC-16 Family Assembler)

1. Preface

This document describes the difference in the specification of the Softune assembler (*FASM907S*) and the old assembler (*ASM907A*).

2. Assembler Language Basic Rules

This section explains the difference in the source line syntax for *FASM907S* and *ASM907A*.

2.1 Statement Format

The assembler 1 line format is as shown below.

- Input for each field omitted when not required
- Each field delimited by one or more spaces or one tab
- Difference in comment field and continuation display field

[FASM907S input]

Symbol field	Operation field	Operand field	Comment field	Continuation display field
--------------	-----------------	---------------	---------------	----------------------------

- The max. length of one line is 4095 characters.

[ASM907A input]

Symbol field	Operation field	Operand field	Comment field
--------------	-----------------	---------------	---------------

- The max. length of one line is 127 characters.

2.1.1 Comment field

FASM907S	ASM907A
A comment begins with a semicolon (;) or 2 slashes (//). The comment must continue until the end of this line.	A comment begins with a semicolon (;). The comment must continue until the end of this line.
Like in C, a comment enclosed by /* */ can be input.	

2.1.2 Continuation display field

FASM907S	ASM907A
It is possible to continue this line by entering a backslash (\) immediately before a line feed.	No continuation lines can be specified.

2.2 Character Set

The following characters can be used at source input.

In addition, all characters including Kanji code can be entered in the comment field.

Type	Character
Alphabetic character (uppercase letter)	A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
Alphabetic character (lowercase letter)	a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z
Numerical character	0,1,2,3,4,5,6,7,8,9
Special character	+ - * / () ; : ' " , _ . @ # \$! % & space, tab < > \

The following four special characters can be used for *FASM907S*, but not for *ASM907A*.

	FASM907S	ASM907A
~	Can be entered	Cannot be entered
=	Can be entered	Cannot be entered
	Can be entered	Cannot be entered
^	Can be entered	Cannot be entered

2.3 Name (Symbol, Label)

When defining/inputting a name, use the following definition.

	FASM907S	ASM907A
Name rule	Input begins with an alphabetic character or an underbar (_). Subsequently, alphabetic characters, numerical characters, and underbars are used.	
Maximum character string length	255 characters	31 characters
Case sensitive enabled/disabled	Enabled	Can be selected using optional specification

2.4 Input of Location Counter Symbol

	FASM907S	ASM907A
Character used	\$	*

2.5 Input of Integer Constant

	FASM907S	ASM907A
Binary input	B'nnnnnnnn 0nnnnnnnnB 0Bnnnnnnnn	B'nnnnnnnn 0nnnnnnnnB
Octal input	Q'nnnnnnnn nnnnnnnnQ 0nnnnnnnn	Q'nnnnnnnn nnnnnnnnQ
Decimal input	D'nnnnnnnn nnnnnnnnD nnnnnnnn	D'nnnnnnnn nnnnnnnnD nnnnnnnn
Hexadecimal input	H'nnnnnnnn nnnnnnnnH 0Xnnnnnnnn	H'nnnnnnnn nnnnnnnnH

2.6 Character Constant

	FASM907S	ASM907A
Input	Characters enclosed in single quotes	
Maximum character length	4	
Control code input	Allowed	Not allowed
Extended notation	Special characters can be specified immediately after a backslash (\) symbol.	To enter a single quote, enter it twice in a row.
	C-type control characters can be specified (\n,\t etc.).	Numerical information cannot be input.

2.7 Character String Constant

	FASM907S	ASM907A
Input	Characters are enclosed in double quotes.	
Control code input	Allowed	Not allowed
Extended notation	Special characters can be specified immediately after a backslash (\) symbol.	To enter a double quote, enter it twice in a row.
Others	C-type control characters can be specified.	Numerical information cannot be input.

2.8 Floating Decimal Number

	FASM907S	ASM907A
Input format	[0r][+ -] { .d d[.d] } [e[+ -]d] [F][+ -] { d d[.d] } [e[+ -]d] H'h d: Decimal 0xh h: Hexadecimal	F'[+ -] { d d[.d] } [S[+ -]d] F'[+ -] { .d d[.d] } [D[+ -]d] H'h d: Decimal h: Hexadecimal
Input example	0r-1.34e-20 F'-2.5 -1.2e10	F'-12.3s10 F'1.2d-3

3. Expression Processing

3.1 Operator List

Function	FASM907S	ASM907A
Priority change		()
Negation of logical value	!	NOT
Bit reversion	~	
Unary plus		+
Unary minus		-
Address extraction	bitadr	
Bit position extraction	bitpos	%
Page number extraction	page	\$
Bank number extraction (paired section)	bnksec	?
Bank number extraction (paired symbol)	bnksym	!
Section size	sizeof	segsiz
Multiplication		*
Division		/
Remainder calculation	%	MOD
Addition		+
Subtraction		-
Left shift	<<	SHL
Right shift	>>	SHR
Less than (comparison)	<	LT
Equal to or less than (comparison)	<=	LE
Larger than (comparison)	>	GT
Equal to or larger than (comparison)	>=	GE
Equal to (comparison)	==	EQ
Unmatch (comparison)	!=	NE
Bit AND	&	AND
Bit XOR	^	XOR
Bit OR		OR
Logical AND	&&	
Logical OR		

3.2 Difference in Operator Priority

Function	FASM907S	ASM907A
Priority change	1	1
Negation of logical value	2	6
Bit reversion	2	
Unary plus	2	2
Unary minus	2	2
Address extraction	2	2
Bit position extraction	2	2
Page number extraction	2	2
Bank number extraction (paired section)	2	2
Bank number extraction (paired symbol)	2	2
Section size	2	2
Multiplication	3	3
Division	3	3
Remainder calculation	3	3
Addition	4	4
Subtraction	4	4
Left shift	5	3
Right shift	5	3
Less than (comparison)	6	5
Equal to or less than (comparison)	6	5
Larger than (comparison)	6	5
Equal to or larger than (comparison)	6	5
Equal to (comparison)	7	5
Unmatch (comparison)	7	5
Bit AND	8	7
Bit XOR	9	8
Bit OR	10	8
Logical AND	11	
Logical OR	12	

3.3 Other Notes

Function	FASM907S	ASM907A
Arithmetic precision	32 bits with/without sign	33 bits with sign
Sign	With/without sign	With sign
Support of data with type	Provided	Not provided

4. Pseudo-Instruction

4.1 List of Differences in Pseudo-Instruction

Function	FASM907S	ASM907A
Object name setting	.PROGRAM	NAME
Source input end declaration	.END	END
External reference symbol declaration	.IMPORT	EXTRN/EXTERN
External definition symbol declaration	.EXPORT	PUBLIC
Instruction area definition	.SECTION CODE (*1)	CSEG
I/O related area definition	.SECTION IO (*1)	IOSEG
Data area definition	.SECTION DATA (*1)	DSEG
Stack area definition	.SECTION STACK (*1)	SSEG
Area definition end	No input required	ENDS
Bit-wise data definition	.DATA.I	DBIT
Byte-wise data definition	.DATA.B or SDATA (*2)	DB
2-byte unit data definition	.DATA.H	DW
3-byte unit data definition	.DATA.E	DSL
4-byte unit data definition	.DATA.L	DL
Bit-wise area reservation	.RES.I	RBIT
Byte-wise area reservation	.RES.B	RB
2-byte unit area reservation	.RES.H	RW
3-byte unit area reservation	.RES.E	RSL
4-byte unit area reservation	.RES.L	RL
Single-precision floating decimal data definition	.FDATA.S	DFS
Double-precision floating decimal data definition	.FDATA.D	DFD
Single-precision floating decimal area reservation	.FRES.S	RFS
Double-precision floating decimal area reservation	.FRES.D	RFD
Structure declaration	.STRUCT~.ENDS	STRCUT~ENDS
Symbol definition	.EQU	EQU
Register symbol definition	.REG	REG
Boundary alignment	.ALIGN	EVEN
Location adjustment	.ORG	ORG
List format control	.LIST	LIST
List output suppression	.LIST	LISTOFF
List output specification	.LIST	LISTON

List page break specification	.PAGE	PAGE
List title specification	.HEADING	TITLE
Subtitle specification		SUBTTL

(*1) The actual correspondence varies depending on the attributes to be added (**ABS**, **WORD**, **COMMON**, etc.).

(*2) One is selected depending on whether character string data or numerical data is defined.

4.2 Details of Difference in Pseudo-Instruction

Only differences requiring explanation are given below.

4.2.1 Definition of section (segment)

The *FASM907S* section is defined according to the following rule:

[Input format]

.SECTION section name [, section type] [, section arrangement format]
:
text
:

Section type	:	{ CODE DATA CONST COMMON STACK DUMMY IO IOCOMMON }
Section arrangement format	:	{ ALIGN= boundary value LOCATE= starting address }
Boundary value	:	Expression (absolute expression)
Starting address	:	Expression (absolute expression)

■ Section type

The following section types are available:

- CODE : Code section
- DATA : Data section
- CONST : Data section with initial value
- COMMON : Shared section
- STACK : Stack section
- DUMMY : Dummy section
- I/O : I/O section
- IOCOMMON : Shared attribute I/O section

■ Section arrangement format

- ALIGN : Relative section
- LOCATE : Absolute section

When the section arrangement format is omitted, ALIGN = 2.

The corresponding patterns are shown below.

	FASM907S input format	ASM907A input format
Code division definition	.SECTION section name CODE ALIGN=1	section name CSEG
	.SECTION section name CODE LOCATE=0 ALIGN=1	section name CSEG ABS
	.SECTION section name CODE LOCATE=h'xx0000	section name CSEG ABS=h'xx
	.SECTION section name CODE LOCATE=h'xyyyy	section name CSEG ABS=h'xh'yyy
	.SECTION section name CODE ALIGN=1	section name CSEG BYTE
	.SECTION section name CODE ALIGN=2	section name CSEG WORD
	.SECTION section name CODE ALIGN=256	section name CSEG PAGE
Data division definition	.SECTION section name DATA	section name DSEG
	.SECTION section name DATA LOCATE=0	section name DSEG ABS
	.SECTION section name DATA LOCATE=h'xx0000	section name DSEG ABS=h'xx
	.SECTION section name DATA LOCATE=h'xyyyy	section name DSEG ABS=h'xh'yyy
	.SECTION section name DATA ALIGN=1	section name DSEG BYTE
	.SECTION section name DATA	section name DSEG WORD
	.SECTION section name DATA ALIGN=256	section name DSEG PAGE
I/O division definition	.SECTION section name CONST	section name DSEG ROM
	.SECTION section name COMMON	section name DSEG COMMON
	.SECTION section name IO	section name IOSEG
	.SECTION section name IO LOCATE=0 ALIGN=1	section name IOSEG ABS
	.SECTION section name IO LOCATE=h'xx0000	section name IOSEG ABS=h'xx
	.SECTION section name IO LOCATE=h'xyyyy	section name IOSEG ABS=h'xh'yyy
	.SECTION section name IO ALIGN=1	section name IOSEG BYTE
Stack division definition	.SECTION section name IO ALIGN=2	section name IOSEG WORD
	.SECTION section name IOCOMMON ALIGN=1	section name IOSEG COMMON
	.SECTION section name STACK ALIGN=1	section name SSEG

4.2.2 Character string data definition

FASM907S	ASM907A
.SDATA "ABC\n"	DB "ABC",h'0d,h'0a
.DATA.B D'00,D'05	DB D'00,D'05,"ABCDE",D'00
.SDATA "ABCDE"	
.DATA.B D'00	

5. Macro Processing

5.1 Difference in Macro Instruction

Function	FASM907S	ASM907A
Macro definition start	#MACRO	&MACRO
Local symbol setting	#LOCAL	&LOCAL
Macro symbol setting	#SET	&SET
Cancels a macro definition	#PURGE	&PURGE
Iterative macro definition	#REPEAT	&REPEAT
Conditional macro	#IF	&IF
True	No input required	&THEN (can be omitted)
False	#ELSE	&ELSE
Conditional macro end	#ENDIF	&ENDIF
Macro definition end	#ENDM	&ENDM
File read	#INCLUDE	&INCLUDE

6. Structured Control Instruction

The structured control instructions that can be used for *FASM907S* and *ASM907A* are the same. *Table 6-1* gives the list of structured control instructions. Input of condition code conditional expressions is also the same for *FASM907S* and *ASM907A*. *Table 6-2* gives the difference in the input of comparison conditional expressions.

Table 6-1 List of Structured Control Instructions

Function	FASM907S	ASM907A
Dual-processing selection syntax	.IF to .ELSE to .ENDIF	
Multiprocessing selection syntax	.SWITCH to .CASE	
Iterative processing syntax	.FOR to .ENDF	
Iterative processing syntax	.REPEAT to .UNTIL	
Iterative processing syntax	.WHILE to .ENDW	
.BREAK statement	.BREAK	
.CONTINUE statement	.CONTINUE	

Table 6-2 Input of Comparison Conditional Expressions

	FASM907S		ASM907A	
	With sign	Without sign	With sign	Without sign
Equal to	==		Z(EQ)	
Not equal to	!=		NZ(NE)	
Less than	<	<.U	LT	LO
Equal to or less than	<=	<=.U	LE	LS
Larger than	>	>.U	GT	HI
Equal to or larger than	>=	>=.U	GE	

[Example input]

```
.if((A == #10)||(A == #20)||(RW0 != #30))
    User-defined block
.endif
```

7. Machine Instruction

Input of machine instructions is specified by the *hardware manual*. Only operand field input is changed.

For the difference in operand input, see "*2. Assembler Language Basic Rule.*"

8. Difference in Command Line

The difference in the command line specification at activation is given in *Tables 8-1* and *8-2*.

Table 8-1 Assembler Activation Format

FASM907S	ASM907A
fasm907s [option]... [file name]	asm907a file name [option]
	asm907a @option file

Table 8-2 Option Specification for Assembler

	FASM907S	ASM907A
Where to specify options	Any position is allowed	After specification of file name
Specification of option file	-f option file	@ option file
Specification of two or more option files	Allowed	Not allowed

9. Environmental Variables

The difference in the environmental variables is given in Table 9.

Table 9 Environmental Variables

	FASM907S	ASM907A
Specification of working directory	TMP	TMP
Specification of include file retrieval path	INC907	INC907(16L/16/16H) INC905(16LX) INC902(16F)
Specification of installation path	FETOOL	
Specification of display character code	FELANG	
Specification of default option file store path	OPT907	

10. Options

The difference in the options is given in *Table 10*.

Table 10 Assembler Options

	FASM907S	ASM907A
Object file name specification	-o	OJ
Object file output suppression	-Xo	NO
Debug information output	-g/-Xg	DBG(DEB)/NDBG
Working directory specification	Environmental variables are used	WD pathname
Case sensitive enabled/disabled	Always enabled	CP/NCP
Warning message output level specification	-W[0-3]	
Address bank check		BCK/NCBK
Module name specification	-name module	
List file output	-l/-Xl -lf filename	L [filename]
List 1 page line count	-pl {0 20-255}	PL {40-128}
List width character count	-pw {80-255}	PW {80-136}
List page break processing suppression	-pl 0	NPC
Information list output	-linf {ON OFF}	
Source list output	-lsrc {ON OFF}	
Section list output	-lsec {ON OFF}	
Cross reference list output	-lcros {ON OFF}	XR/NXR
Include file list output control	-linc {ON OFF}	IP/NIP
Macro <i>IF</i> statement list control		CD/NCD
Macro definition division list control		DF/NDF
List control for macro call and structured control instruction line		CL/NCL
List control for expanded division of structured control instruction		STR/NSTR
List output control for macro expanded division	-lexp {ON OFF OBJ}	EXP/NEXP
Does not start preprocessor	-p	
Start only preprocessor	-P or -Pf filename	
Macro name specification	-D name[=def]	
Macro name cancel	-U name	
Assignment of value to undefined macro symbol		
Display of include path	-H	
Leaving comments in preprocessor output	-C	

Error information file output enabled/suppressed		-[NO]EP
Status code file output		-RC
Option file end	No input required	-E
Help display	-help	
Target CPU specification	-cpu	-CPU
CPU type check suppression		-NCTC
Default option file input suppression	-Xdof	
Activation message display/no display	-V / -XV	
Warning-time return value change allowed/suppressed	-cwno / -Xcwno	
Termination message display/no display	-cmsg / -Xcmsg	

11. Alleviation of Restrictions

The difference in the processing limits is given in *Table 11*.

Table 11 Processing Limits

	FASM907S	ASM907A
1 line character count	4095 characters	128 characters
Source file line count	65535 lines	32000 lines
Symbol count	Unlimited	32000 symbols
Name length	255 characters	31 characters
Number of macro names that can be registered	Unlimited	Macro definition count: 512 Local symbol count: 256
Macro nest	256 levels	8 levels