



**F<sup>2</sup>MC-16L/16LX/16/16H/16F Family  
Binary Comparison Procedure**

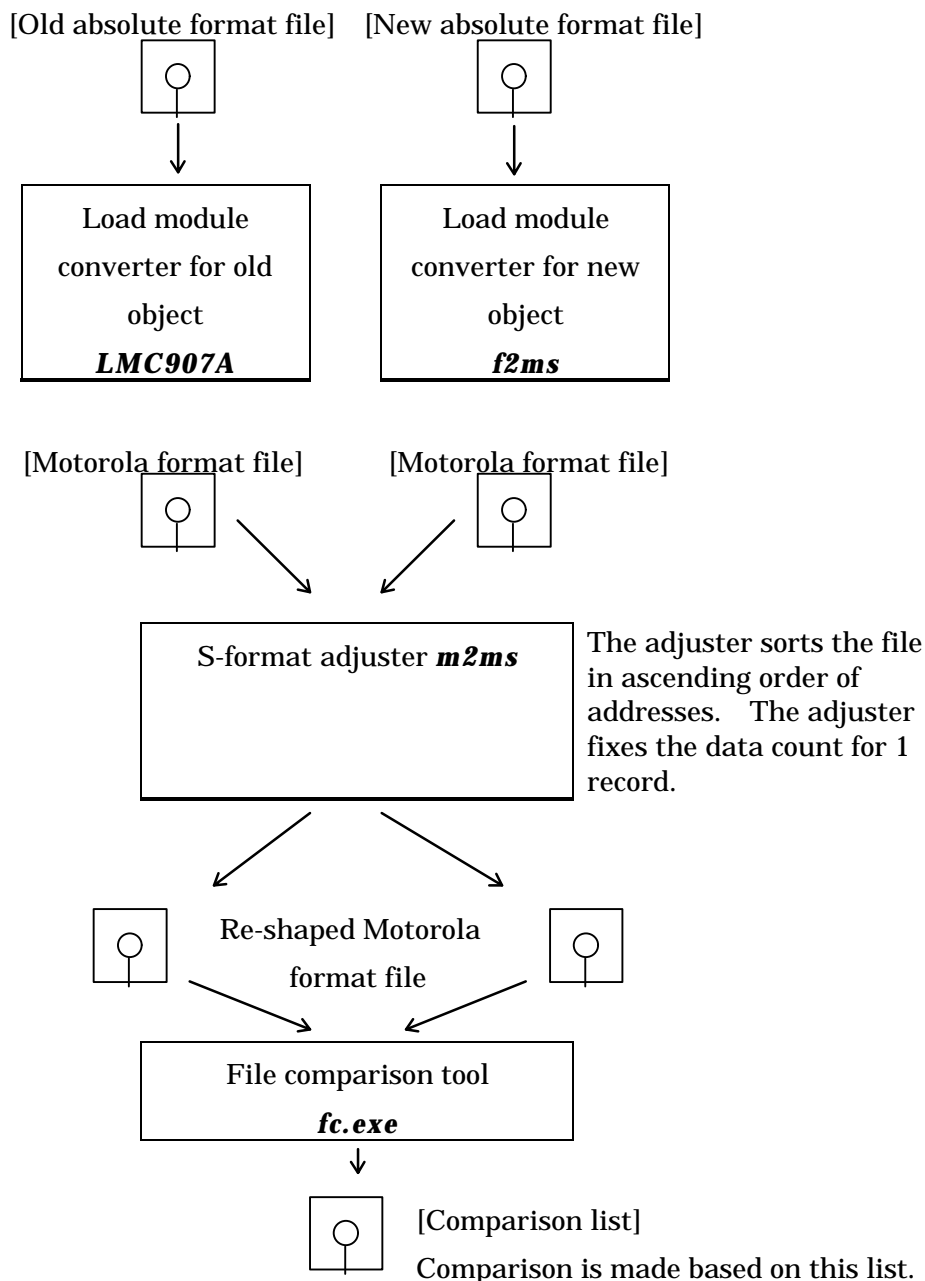
## **Object Information Comparison Document**

This document describes the procedure to examine whether the new and old absolute format files are exactly the same at the binary level when the source program for the old 16L assembler (***asm907a***) is ported to the new assembler (***fasm907s***).

1. Outline of Comparison Procedure

The figure below shows the comparison procedure.

The absolute format object data created by the new and old tools are converted to the Motorola format.



Ignore the differences, if any, between the S0 record at the first line and the S7 record at the last line.

## 2 Comparison Example

The absolute format file created by the old assembler is "**old.abs**". The absolute format file created by the new assembler is "**new.abs**".

```
>lmc907a old.abs -o old.mhx ; Converts old absolute format file to Motorola format
>m2ms old.mhx -o old.ahx -ran 0xff00,0xffff -S3
                                ; Sorts old Motorola format
>f2ms new.abs -o new.mhx ; Converts new absolute format file to Motorola
format
>m2ms new.mhx -o new.ahx -ran 0xff00,0xffff -S3
                                ; Sorts new Motorola format
>fc old.ahx new.ahx ; Compares files
```

The comparison results may indicate the following difference, but data with a record beginning with "S0" or "S7" should be ignored.

This record indicates a comment, so a match is not required.

### [Output example]

```
>fc old.ahx new.ahx
***** old.ahx
S0080000746573743106
***** new.ahx
S0080000544553543285
*****
```

### Notes:

- (1) When an attempt is made to process data in the entire 16-MB memory space of the F2MC-16 family when using **m2ms**, the execution speed may slow down. In such a case, use the **-ran** option to split the file.
- (2) The file should also be split when data to be compared is large.

### 3. When Data Mismatched

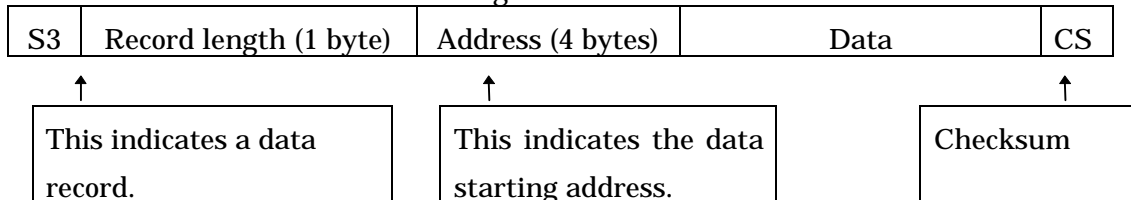
The following shows an FC output example. This information is used to locate mismatched locations.

[Output example]

```
fc o ld.ahx new.ahx
**** old.ahx
S31500FF0000000102032C37060708090A0B0C0D0E0F19
**** new.ahx
S31500FF0000000102030405060708090A0B0C0D0E0F73
****
```

#### 3.1 Obtaining Addresses from Motorola Format

The Motorola format has the following format:



\*Note: Ignore records other than "S3" records.

The address of data with a mismatch can be obtained from the above output.

S3150000FF00000102030405060708090A0B0C0D0E0F73

This record data begins at address 0x00ff00. From this, it is determined that there is mismatched data at addresses 0xff05, and 0xff06.

### 3.2 Locating Module Using Link Map

The module where the address obtained in **Section 3.1** is written is located using the link map.

At linkage, the **-mlf** option and the **-als** option must be specified.

The **-mlf** option creates detailed map information and the **-als** option creates the absolute assemble list.

[Output example]

```
>flnk907s main subl -AL 2 -mlf main.mpm -als -ro rom=0xff00/0xffff -ra rom=0/0xff
```

#### Detailed map information (main.mpm)

Section Mapping List							
Module(s)							
1. main							
2. subl							
S_Addr.	-E_Addr.	Size	Section	Type	Al	M.No	Sec.(Top 80)
00FF0000-00FF000F		00000010	CONST	P R-I	02	1	data
00FF0010-00FF001F		00000010	CONST	P R-I	02	2	data

From the detailed map information, it is determined that the appropriate addresses (0xff05, 0xff06) are defined by the **"main"** module (module No. 1).

### 3.3 Locating Mismatched Locations

Mismatched locations are located from the absolute assemble list created in **Section 3.2**.

#### Absolute assemble list (main.als)

SN	LOC	OBJ	LLINE	SOURCE
			=====	
			1	
			2	.TITLE main
		MODULE NAME = test	3	.PROGRAM main
			4	
			5	
	da 00FF00	-----<data>-----	6	.SECTION data
				, CONST, ALIGN=2
	da 00FF00 00 01 02 03	<u>2C 37</u> 06	7	.DATA.B 0,1,2,3, <u>44,55</u>
	da 00FF07 07 08 09 0A 0B 0C 0D			,6,7,8,9,10,11,12,13,14,15
	da 00FF0E 0E 0F			
			8	;data ends
		==	9	.END