Appendix A Compatibility

A.1 Differences between LSI C and ANSI C

This section lists up differences in specification between LSI C and ANSI C.

A.1.1 Language specifications

1. Keyword nonrec, and recurive are added to the LSI C. Additionally, far and near are added to the LSI C-86. For details, see section 5.2.1, Keywords.
2. The integer conversion rule varies slightly from that in the ANSI definition. For details, see section 5.6, Type conversion rule.
3. The preprocessor does not support the trigraph sequence (expressed by ?? and a character).
4. `wchar_t, L'…'`, and `L"…"` are not supported.
5. Variable parameter functions cannot return the structure, such as float, double, or long double. Another word, variable parameter functions can return only values which are allocated to registers. For details, see section 6.6 Protocols called up by functions.
6. If an external variable or function declaration, which appears first, has extern, it is possible to specify "`static`" in the same variables, function declarations, and definitions located after that.
7. In the ANSI standard, the type of a constant exceeding INT_MAX is long. However, the type of a constant which equals or is smaller than UNIT_MAX is unsigned int.
8. The LSI C handles all floating point constants as long double type data.

A.1.2 Library

1. The LSI C does not support `<locale.h>`, and macros and functions defined in `<locale.h>`.
2. The LSI C does not support function groups (such as `mblen ( ),` etc.) that handles characters having 2 or more bytes.

A.4 Bug

This section describes bugs and problems which are already found at present.

1. No errors are reported for declaration without `declarator`.
2. `sizeof (type) expr` is not handled as a syntax error, but interpreted to `sizeof ((type)expr)`
3. Member operator “`.`” cannot be used for the structure value (`r-value`). The following example explains this problem.

```
struct foo {
    int i;
    char c;
}   bar ();

int zod ()
{
    return bar () .i;
}
```

“.” returned from the function bar () to the structure may cause an error (l-value required).

4. Operators, [ ] , ., ->, and & cannot be used for the constant expression except for the initialization.
5. When near pointer to the function is cast to far pointer, DS register may be used instead of CS register.
6. If the capacity of the segment in a file exceeds 64K bytes, the assembler r86 makes incorrect object files. At this time, no errors are reported. Therefore, the data size must always be checked carefully.