bcc	title		prefix/class–number.revision
	PHASE ONE UNPREPROCESSOR INTERFACE		UPREP/M–11

checked *L. Peter Deutsch*	authors	approval date 10/8/69	revision	date
checked *Roger Sturgeon*		classification Manual		
approved *Mel*	Rick Dove *R.K. Dove*	distribution Company Private		pages 5

ABSTRACT and CONTENTS

This documents the interface between the phase one unpre-
processor and both M1CS and the M1CS user. The system
interface includes all calls and parameters and the user
interface includes all error messages.

The phase one unpreprocessor interfaces primarily with
the model one compiler system. However, a temporary
SPL user interface is present which manifests itself as
a cryptic message whenever an unpreprocessor error occurs.
This user interface is temporary for the SPL debugging
phase only since unpreprocessor errors represent impos-
sible conditions generated by bugs rather than users.
Thus, phase two will see the same system interface with
minor changes and no user interface.

SYSTEM INTERFACE

Calls:

UNPP(LINE,BLOCK,MODE);

This call will take the line whose symbol
index is LINE in the block whose token number is
BLOCK and unpreprocess it into string UPS. See
below for MODE interpretation.

UNPK(BLOCK);

This call will unpreprocess the line which is
in the PPTB area according to its interpreta-
tion in the block whose token number is BLOCK.
The mode is set to 2.

UNPSI(LINE,BLOCK,TOKEN);

This call functions like UNPP except the mode
is set to 2 and TOKEN specifies some informa-
tion to be obtained, specifically, the value

is the length of UPS before the Nth token is
unpreprocessed into it. This value will be in
the A register and global parameter UNCHN
when control is relinquished.

Formal Parameters:

LINE: Line symbol index of line to be unprepro-
cessed.

BLOCK: Global name table token number of block
containing line to be unpreprocessed.

TOKEN: An integer, N, where $N \geq \emptyset$. This signifies
the Nth token of the line to be unprepro-
cessed and is used when requesting posi-
tional information about the token in
question.

MODE: This parameter selects amongst four for-
matting conventions concerned with spacing
and automatic indentation. The conven-
tions are selected by integers 1, 2, 3,
and 4:

1) Standard spacing is enforced regardless of
space tokens embedded in the line. Consider-
ing space tokens invisible, there remain three
classes of tokens for spacing purposes:
keyword (KEY), symbol/constant (S/C), and
all others (OTH). The following table gives
the standard spacing between token classes:

	KEY	S/C	OTH
KEY	1	1	1
S/C	1	1	Ø
OTH	1	Ø	Ø

Standard spacing will put a space in column
1 if the first non-blank token is not a keyword
or symbol/constant.

2) Duplicate the original text as encoded in tokens.

3) This mode is the same as (1) with the addition
of standard indentation. Standard indentation
will make the first label of a line, if it is
the first non-blank token, left justified but
start all other text at a column prescribed by
the compiler. The level of indentation is in-
dicated in the PTIND field of a line cell, where
Ø means the left indentation margin and 1, 2
...etc. mean that many additional indentation
increments. The left indentation margin is set
for column 8 and the indentation increment is
set to 3 spaces.

4) This mode is the same as (2) with the addition
of standard indentation.

Global Parameters:

UMCONT - Controls the unpreprocessing of "L^C".

$\emptyset$ = "&L", 1 = "L^C" (page eject not implemented until phase 2).

PPTB - A single dimension array which must contain the line cell to be unpreprocessed if call UNPK(BLOCK) is used.

UPS - The string which will contain the unpre-processed line when the unpreprocessor relinquishes control.

UNERR - Contains the error code if an error occurs.

UNCHN - First character number in UPS of token number TOKEN if call UNPSI(LINE,BLOCK,TOKEN) is used. If UNCHN is not set it will equal -1.

Termination:

The normal termination is an SRETURN. RETURN occurs only if an error has occured.

USER INTERFACE

The user interface is temporary and used for debugging purposes only. It manifests itself in the form of an error message when an unpreprocessor error occurs:

XXXⱢUER₵

XXX is the error number and may be interpreted according to the list given below. This number should be given to someone on the SPL project as it indicates a system bug rather than a user error.

UNPREPROCESSOR ERROR NUMBERS:

1: unrecognizable token

2: UPS string overflow

3: keyword token not present or UPS overflow when inserting keyword

4: special operator token not present or UPS overflow when inserting keyword

5: language type not implemented

6: negative decimal integer

7: unimplemented integer type (H,O)

10: undefined integer type (6 or 7)

11: impossible error

12: constant type not yet implemented

13: undefined token

14: unimplemented string constant type