Department of Computing Services

University of Waterloo

V2L7A ASMG Implementation Guide

```
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
****  ********  ***          ***  ************            **************
****  ********  **            **  ************           ***************
****  ********  **  ********  **  ************  ******  ****************
****  ********  ************  **  *******************  *****************
****  ********  ************  **  ******************  ******************
****  ********  ***********  ***  *****************  *******************
****  ********  *********  *****  *****************  *******************
*****  ******  ********  *******  *****************  ******AAA**********
******  ****  *******  *********  *****************  *****A***A*********
*******  **  ******  ***********  *****************  *****AAAAA*********
********    ******            **            *******  *****A***A*********
*********  *******            **            *******  *****A***A*********
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
*******     ********     *****  ***  *****  **********    **  **********
******       ******       ****  **    ***    ********      *  **********
*****  ****   ****  *****  ***  *  **  *  **  ******  ****    **********
****  ****  *  ***  **********    ****   ****  ****  ******   **********
**** ****  **  ***  **********   *****  *****  ***  ********  **********
********  ***  ****     ******  ******  *****  ***  ********  **********
*******  ****  ******     ****  ******  *****  ***  ********  **********
******  *****  **********  ***  ******  *****  ***  ********  **********
*****  ******  **********  ***  ******  *****  ***  *******   **********
****  *******  **********  ***  ******  *****  ***  ******    **********
****  *****    ***  *****  ***  ******  *****  ****  ****  *  **********
*****       *   ***       ****  ******  *****  *****      **  **********
******     ***  ****     *****  ******  *****  ******    ***  **********
************************************************************  **********
************************************************************  **********
************************************************************  **********
******************************************************  ****  **********
*******************************************************  ***  **********
********************************************************      **********
*********************************************************    ***********
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
```

A GUIDE TO IMPLEMENTING THE UNIVERSITY OF WATERLOO
LEVEL G ASSEMBLER FOR THE IBM SYSTEM/360 OR SYSTEM/370

PROJECT MEMBERS

MIKE DOYLE
RENNIE PETERSEN
STEVE SCHROETER
DAVE POTTER
RODNEY COOPER
BRUCE UTTLEY

TENTH EDITION

MARCH 1976

University of Waterloo
Waterloo, Ontario.

PRINTED IN CANADA

DISCLAIMER

Although this program has been tested by its authors, no warranty, expressed or implied, is made by the authors, or the University of Waterloo, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the authors, or the University of Waterloo, in connection therewith.

ACKNOWLEDGEMENTS

ASMG is a modification to IBM'S level (F) Assembler IEUASM. Extensive use was made of the program logic manual for IEUASM and we are indebted to the writers and documentors who provided such a clear description of such a very large program.

The changes to ASMG to permit it to build larger local dictionaries (and thus assemble larger programs than Assembler (F)) are due to Christine Packard and George Sjoberg of the Pennsylvania State University Computation Center.

The changes to ASMG to support named common, and the optional support for the Model 67 RPQ instructions are due to Martin Raim of the University of Michigan Computing Center.

The part of ASMG which determines the day of the week for printing on the heading page was inspired by, and is somewhat modelled after, the program 'WEEKDAY' written by Richard L. Conner.

The alternate root phase for the Assembler called ASMGWYL which supports WYLBUR format input files was adapted from code written by Andrew Koenig of Columbia University.

TABLE OF CONTENTS

ASSEMBLER (G) IMPLEMENTATION GUIDE

A. INTRODUCTION

ASMG is a modification of IBM OS/360 Assembler (F). These modifications are primarily aimed at increasing speed and adding a student program processing capability. These modifications introduce several operational differences (control card and output listing changes), however, ASMG is completely source-language compatible with Assembler (F).

This manual attempts to describe how to add ASMG to an OS system. A rough outline of changes made to Assembler (F) and some hints on possible changes to make to ASMG are also given.

This documentation conforms to V2L7a of ASMG, which is roughly equivalent to Release 21.8 of Assembler (F).

B. MACHINE AND SYSTEM REQUIREMENTS

As well as the minimum OS requirements, ASMG requires the Commercial and Scientific instruction sets. A minimum of 85K of core must be available in the problem program partition for PCP or MFT. The core requirement for MVT is about 15K more. The core requirement is a function of the size of the deck being assembled and the data set blocksizes being used. To assemble an 8000 line program under MVT requires a region size of 116K.

Performance is dependent on available storage and CPU model, among other things. Highest performance will be obtained running in 150K or more on a model 65 or higher.

A 9-track 800 BPI (or 1600 BPI, if requested) tape drive will be required to process the distribution tape.

A UCS 1403 printer with PN or larger character set or a 1443 printer with 62 character bar is needed to properly print the documentation data sets (this manual and the User's Guide). Also, if the printer used to print ASMG's output listings does not have the colon character, then there will be blanks between the hours, minutes and seconds in the TIME= field in the heading.

C. OPERATIONAL DIFFERENCES WITH ASSEMBLER (F)

1. Incompatibilites.

The default options are LOAD,NODECK instead of DECK,NOLOAD. Also, NUM,STMT is default instead of NONUM,NOSTMT.

The default instruction set includes the extended branch conditional register mnemonics. This will result in error messages for programs which contain definitions of these mnemonics as macros.

SYSLIN is the name of the DD card preferred for writing the object deck under the LOAD option. If SYSLIN is missing and SYSGO is present it will be used instead.

2. Major Extensions.

Minimum unique length abbreviations are accepted for the PARM options, plus many new PARMs exist to control the other extensions.

The printing of the ESD and RLD is normally suppressed, but is optionally allowed.

The XREF is normally printed in a squished format. It may optionally be printed in full format or may be suppressed.

A literal cross-reference (LREF) is printed by default. This provides the same information about literals as an XREF provides about symbols. The literals are listed in the EBCDIC collating sequence of the character string that defines them. This listing has been formatted to allow for the extreme and variable lengths of literal strings.

There is a BATCH option which allows multiple source decks to be assembled in one job step.

There is an EXECUTE option which allows simple programs to be loaded and executed by the assembler immediately following the assembly.

There is an INSTSET= option which allows the choosing of different instruction sets. Distributed instruction sets include an OS Assembler (F) compatible one, plus one oriented toward assembling for a System/360 or System/370 with extended branch register instructions and a DOS Assembler (F) compatible one. There are also instruction sets oriented toward assembling programs on or for models 20, 44, and 67.

The size of the unsubsetted local dictionary may exceed 64K, permitting the assembly of extremely large programs, provided there is sufficient core available.

Datasets with unlike characteristics or on unlike devices may be concatenated on SYSIN or SYSUP.

There is an EXTEN option to control the following language extensions:

1) PRINT statements allowed in macros.
2) Attributes of symbols defined in macros available.
3) &SYSDATE, &SYSTIME, &SYSSTYP and &SYSPARM system variable symbols available in open code and in macros.
4) Named COMMON supported.
5) SETC variables may be declared as having maximum lengths less or greater than 8 bytes.
6) The K' (count) operator may be used on any SETA, SETB or SETC variable. This is in addition to the former parameters only restriction.
7) SETC variables containing C, X, or B type self-defining terms may be used in SETA expressions.
8) Macro definitions may be included in source code as programmer macros using COPY.
9) The extended DROP feature is supported.
10) The extended EQU feature is supported.
11) Labelled CNOP, labelled ORG and unlabelled DSECT are valid.
12) Comments may be generated in the operand field of generated statements.
13) In the Assembly Phase, unary + and - are supported with eleven levels of parentheses and twenty-five terms.
14) Eight character TITLE labels are supported.
15) To allow a label on other than the first TITLE statement of an assembly.
16) Expressions allowed for literal duplications and length factors.
17) Current Location Counter '*' allowed in Duplication and Length factor calculations in DCs, DSs and literals.
18) Support of 4 byte self-defining terms in Assembly Phases.
19) Allow SETx variable dimension to be up to 9999.
20) 'END' statement allowed in COPY code. Useful for SMP.
21) Treat an MNOTE with only a quoted string as an operand as a comment when printing.
22) Positional and Keyword parameters may be intermixed in the Macro Prototype and Macro Instruction statements.
23) LCLx, GBLx and ACTR statements can appear anywhere before the variable's use in Open code or within a Macro definition.

There is a FULLLIST option which prints the library macros as they are edited, allowing syntax errors in the library macros to be pinpointed.

Support for some Model 67 RPQ instructions can be enabled by reassembling three of the assembler modules. See the user modifications section.

An UPDATE facility is available which permits ASMG to simultaneously read an update deck and an old master data set, doing the assembly on the resulting (non-existing) new master.

An optional first load of the assembler is available, called ASMGWYL, which allows the support of Wylbur Format input files on the SYSIN, SYSUP and SYSLIB data sets.

D. DISTRIBUTION

Distribution of the current version and level of ASMG will be made to those registered users who submit a reel of tape containing at least 1200 feet. The distribution tape is the result of doing IEHMOVEs from 2314 disk to tape. The tape will be 9-track 800 BPI unless 1600 BPI is specifically requested. It is labelled 'ASMG27', and contains the following datasets.

DSNAME	SEQ #	FORMAT	CONTENTS
ASMG.JCL	1	Sequential	Sample jobs to copy ASMG into an OS system, test it and modify it.
ASMG.LOADMODS	2	Unloaded PDS	Load modules for ASMG
ASMG.GUIDE.USERS	3	Sequential	Users guide
ASMG.GUIDE.IMPL	4	Sequential	Implementation guide
ASMG.SOURCE	5	Unloaded PDS	Source decks for basic V2L7 ASMG.
ASMG.OBJMODS	6	Unloaded PDS	Corresponding object deck
ASMG.MACROLIB	7	Unloaded PDS	Macros needed to assemble
ASMG.LKEDIN	8	Unloaded PDS	Link editor control cards mapping the object decks into loadmods
ASMG27A.UPDATE	9	Unloaded PDS	V2L7a update cards for SOURCE and MACROLIB

To retrieve the ASMG.JCL Dataset run the following job.

```
//ASMGO      JOB   'C0018R.V.PETERSEN,TIME=5,CARDS=1500,PAGES=50',
//                 MSGLEVEL=1
//           EXEC  PGM=IEBUPDTE,REGION=22K
//SYSPRINT   DD    SYSOUT=A
//SYSUT1     DD    UNIT=2400,VOLUME=(PRIVATE,SER=ASMG27),DISP=OLD,
//                 DSNAME=ASMG.JCL
//SYSUT2     DD    SYSOUT=B
//SYSIN      DD    *
./         REPRO      LIST=ALL
./         ENDUP
/*
```

This will both print and punch the ASMG.JCL dataset. The jobs in ASMG.JCL have comments in them which should make their use self-explanatory.

A master copy of the ASMG User's Guide and ASMG Implementation Guide can be generated by using the IEBGENER programs provided in ASMG.JCL, using a TN print chain and full page carriage tape. If you do not have a TN print chain at your installation, ASMG.JCL (job ASMG#13) also contains a program which will convert all lower case letters in files 3 and 4 to upper case and generate master copies of these manuals.

E. INTERNAL DIFFERENCES WITH ASSEMBLER (F)

1. GENERAL:

Deck and module names are changed from IEUXXX to ASMGXXX. ASMG reverses the use of SYSUT1 and SYSUT3 (i.e. ASMG uses SYSUT1 where Assembler (F) uses SYSUT3 and vice-versa). The root segment (ASMGASM) has been aliased 'ASMG' for ease of reference.

The following modules have been deleted:

ERR - The I/O error abort module. Its functions have been included in the ASM deck.

F3E - The F3 substitution module in case of dictionary overflow. This situation now results in an abortion of the assembly with a return code of 20.

MAC - The macro I/O interface module. Phases F1 to F3 now do their I/O directly or through BUFF.

2. DECKS ADDED:

| WYL - WYL is an update to deck ASM and performs all the functions of that driver routine. The difference is that two adcons at LIST1+L1DBLK are defined and pointing to deblock routines for Wylbur Format input files. These addresses are used by routines ASMGF1, ASMGUP and ASMGF2A to correctly process GET/READ requests to input files SYSIN, SYSUP and SYSLIB if those input files are in Wylbur format.

BUFF - This deck handles the I/O buffering of the utilities. It also handles the core management functions for the assembler. The I/O buffering routines try to keep all utility records in core. When it runs out of core it begins to spill records out onto the disk. When the records are read by the assembler, BUFF simply does a move if the record is still in core, if not it does a spillin to retrieve the record from the disk. This deck is in the ASMGASM module.

UP - This deck performs the update input functions for Assembler (G). It is only loaded and used if PARM=UPDATE is specified. Each time the assembler (or the user's program under EXECUTE) attempts to read a card on SYSIN, this program is invoked instead. This is because ASMGF1 has placed the address of ASMGUP into the DCBGET address field of the SYSIN DCB. This routine will read records from SYSIN and/or SYSUP to determine which record should be passed to the assembler next. The record is moved to the buffer specified by the user. All ASMGUP registers are saved each time it returns so it will remember where it was and what it was doing the last time it was executed. When end of file is hit on both SYSIN and SYSUP, ASMGUP terminates itself by taking the EODAD exit of SYSIN. If an internal UPDATE error severity code should exceed the UPCOND= parameter, then a switch is set in LIST1 and the update continues. ASMGUP is loaded by ASMGF1 and remains resident for the entire assembly.

FEX - This deck loads and executes the object deck of the program just assembled. It contains a simple one-pass loader which cannot linkedit two or more decks together. Before executing the loaded program a SPIE and STIMER are issued to regain control in case of program interrupts or infinite loops. FEX also contains a core-dump routine to aid in finding the cause of program interrupts and timer overflows. The FEX deck is in the FPP module.

MACP - This routine is an experimental routine that makes only the macro expansion facilities of the Assembler available to the user. The input files are still SYSIN, SYSUP and SYSLIB containing normal conditional assembly statements, macro calls, COPY code etc. All valid macro calls and COPY requests are changed to comment statements in the output. Programmer macro definitions delimited by MACRO and MEND in the SYSIN stream are flagged in the SYSPRINT listing and removed from the source output. In this way, a partial assembly may be done to expand all or some of the conditional assembly statements of an assembly to produce a source file that contains fewer or no macros. This routine has been used to expand user macros in an assembler source deck, without expanding IBM macros from SYS1.MACLIB. It has been used to preprocess mini computer source files to allow access to the (G) level macro facilities where the assembler for the mini either had little or no macro facilities. It has been used to give access to COPY facilities in Fortran source files.

To use this routine, it must be given a ALIAS name of ASMGF7 and be STEPLIBed to in an assembly. The result is that the first assembly phase after the macro expansion phase is replaced by this routine.

3. RE-STRUCTURING OF DECKS IN MODULES:

FI - This deck is now in the F7 module.

4. MAJOR CHANGES MADE TO DECKS:

ASM - ASM does a BLDL on all the load modules necessary for every assembly. The results are used in LINKs and XCTLs by the assembler. All the assembler DCB's are now in ASM. The SYNAD routines for SYSPRINT and SYSPUNCH attempt to count the number of I/O errors (I.E. the number of times SYNAD was entered) and then ignore them. The SYNAD routine for all the other DCB's prints a diagnostic which includes SYNADAF information, and then returns with a completion code of 20. A routine is added to ASM which monitors the SYSIN dataset to handle EOF situations. In BATCH and EXECUTE it also checks for $JOB cards. This routine also handles concatenation of unlike datasets on SYSIN. ASM contains an I/O monitor for all non-utility output data sets. Its address is patched into the DCBPUT field by F1. By flags in a DCB trailer it accepts no carriage control, numeric carriage control or ASA carriage control done with PUT MOVE conventions. The numeric or ASA is converted to machine carriage control for SYSPRINT and SYSTERM. All datasets are closed and FREEPOOLs issued by ASM.

F1 - PARM scan changed to accept new PARMS and to print out the options line. The TIOT is scanned to determine if SYSLIN or SYSGO should be used for object output. All necessary data sets are opened, and success of each open is checked and a diagnostic printed in case of failure. If no SYSLIB card is found in the TIOT then SYSLIB will not be opened. A SYSPRINT DD card need not be present if NOLIST,TERM is specified in the parameter list and SYSTERM is present. In such a case a dummy I/O routine is provided in ASM for SYSPRINT (BR 14). SYSIN and SYSLIN may be given the same buffers out of the same buffer pool if the options are LOAD,NOEXECUTE,NOBATCH. The SYSLIB directories are read and a macro name vs. TTR (disk address) is set up. If this incore directory should overflow, a flag is set in ASM and F2 will do conventional OS FINDs to attempt to locate macros not contained in the incore table. A count of these FINDs is kept for a diagnostic in FD. Available storage is GETMAINed as specified by the SPACE= parameter and given to BUFF for future allocation. Buffer allocation is revised, and the necessary core retrieved from BUFF. If UPDATE has been specified F1 loads the ASMGUP module and places its address in the DCBGET field of SYSIN after saving the real address in ASMGASM. An ASMGISXX module is loaded, the initial global dictionary moved out of it, and the module deleted (XX comes from the INSTSET=XX parameter), unless BATCH is specified in which case it is loaded and kept resident.

F2 - Dictionary allocation is altered so that unsubsetted blocks must stay in core and be subsetted in core (SYSUT2 is unused). Also more than 64 blocks of unsubsetted dictionary are allowed. MACRO and MEND statements are allowed within COPY code and PRINT statements are permitted in macros unless NOEXTEN is specified in the parm field. SYSLIB is double buffered for everything except copies from system macros. Instead of doing FINDs, F2 looks up the name in the macro directory set up by F1 and does a POINT. If the macro directory is full then a conventional OS FIND is done to search for the macro on disk and a counter is incremented. Utility I/O is done through BUFF and SYSIN and SYSLIB I/O is done directly. The library macros are written on UT3 if FULLLIST is specified. A check to determine whether it is necessary to issue ERROR 78 is made. Labels in macros and the extra system variable symbols are placed in the local dictionary if EXTEN is specified. The length declaration is scanned off for SETC variables, and the subsetting altered accordingly.

F3 - Core management revised to use BUFF. Handling of I/O altered to attempt to reduce the number of records read when expanding macros. This is done by allocating one buffer for SYSUT3 and two buffers for SYSUT1 and checking if the block to be read is not already in one of these buffers. The output is single buffered instead of double buffered. If the abort switch in LIST1 has been set by ASMGUP, then only a generated comment, an ASMG115 diagnostic and an END statement is output. F3 XCTLs directly to RTA. The extra system variable symbols are processed. The ACTR is divided by 2 on each error. Handling of SETC variables is revised to check the maximum length. Support of SET's containing C, X or B self-defining terms is included.

F7 - The ability to iterate is removed, requiring that the entire symbol table fit in core. The symbol table is changed so that it can also be used as the LREF and the XREF definition records. Only reference type XREF records are written out, and the size of these records are reduced. The LBT and ESD are kept in core intermingled with symbol table entries, as are LREF reference records. Core for the symbol table is dynamically obtained from BUFF as the symbol table grows. The symbol table grows by 24 bytes per symbol definition if XREF is specified and by 17 bytes per symbol if NOXREF. If LREF is on, the symbol table grows by 6 bytes per literal definition plus 5 bytes per literal reference. The previous location counter is added to ORG records. The text I/O is single buffered instead of double buffered and the I/O is done through BUFF.

FI - The LBT is converted into the LAT in core where it sits. The ESD is written on SYSUT2 for FEX if EXECUTE. The ESD is not printed unless requested. Utility I/O is done through BUFF.

F8 - I/O changed to use BUFF. Symbol table lookup made compatible with F7. TXT records put on SYSUT2 if EXECUTE. The text input is single buffered instead of double buffered. Each USING, DROP and POP USING operation saves away the current statement number if the UMAP option has been specified. An attempt is made to print lines in error even under PRINT OFF,NOGEN,NODATA. The first location counter generated by a macro call under PRINT NOGEN is printed beside the last statement of the call. The print switch is turned on after the END card in case of FULLLIST. The previous location counter is printed in the ADDR2 field for ORG. USING and EQU arguments print in the ADDR2 field as well. Non-comment MNOTES are flagged.

The data on TXT cards is packed more efficiently.

FPP - The UMAP is processed before the LREF. The entries are processed sequentially in the order of the USING. A DROP statement number of zero is replaced by 'END'.

The LREF is processed before the XREF as follows: 1. Definition records are taken from the symbol table in core, one at a time via linear search (because of core restrictions). A chosen literal is removed from the literal chain. 2. The reference records are more compact and do not need sorting. 3. Printing of an LREF is not dependent on the LIST or XREF options.

The XREF is processed before the RLD. XREF handling is completely revised since: 1. Definition records are taken from symbol table in core. 2. The reference records are more compact. 3. The reference records do not need sorting since F7 cannot iterate. 4. Two printed formats are available: squished and full. 5. Printing is not dependent on the LIST option.

The RLD is printed only if requested. The RLD is written on SYSUT2 if EXECUTE is specified. Utility I/O is done through BUFF.

FD - Some extra error switches in ASM are tested. Printing of the errors is not dependent on the LIST option. Utility I/O is done through BUFF. If BATCH and NOEXECUTE are specified, then a eight byte entry is constructed using BUFF for a summary to be printed at the end of the batch. This summary is printed if the end of file switch in ASM is on. FEX is branched to if EXECUTE specified. If BATCH is specified, FD determines if another assembly is to be done. A register is set telling RTA whether or not it should XCTL directly to F1 based on the occurrence of EOF on SYSIN. Otherwise, RTA will return to ASMGASM and terminate the step.

F. MEMORY MAP FOR 100K ENVIRONMENT

VALUES IN BRACKETS ARE TYPICAL

```
  0|___F1___|___F2___|___F3___|___F7___|___F8___|___FPP___|
   |                                                      |
   |              ASMGASM & BUFF                          |
   |                        8.0K                          |
   |______________________________________________________|
 10|        |        |        |____RTA_____2.0K___________|
   |   F1   |   F2   |   F3   |        |        |         |
   |   &    |   &    |  8.9K  |   F7   |   F8   |   FPP   |
   | COMMON | COMMON |________|   &    |        |   FD,   |
 20|        |        | Global |   FI   |        |   &     |
   |        |        |__Dict__|        |        |   FEX   |
   | 26.0K  | 26.0K  |UT3 buff|        |        |         |
   |        |        |_(2.2K)_| 22.7K  | 22.8K  |  21.9K  |
 30|        |        |  Not   |        |        |         |
   |________|________|__Used__|________|________|_________|
   |___ASMGISxx Module (only if BATCH)  (2K or 4K)________|  _
   | UT3 Buffer      |SYSUT1  |UT3 buff|_unused_|         |  |
   |____(2.2K)_______|buffers |_(2.2K)_|adjtb 1K|  XREF   |  |
 40| UT1 Buffer      |_(4.0K)_|UT2 buff|UT3 buff|  sort,  |  |
   |____(2.0K)_______|        |_(2.2K)_|_(2.2K)_|  RLD    |  |
   |   SYSLIB        |        |                 |  sort,  |  |
   |   Buffers       | Local  |                 | EXECUTE |  |
   |____(6.7K)_______| Dict   |                 |  load   |  |
 50|        |        | Area   |  Symbol Table   |  area   |  |
   |        |        |        |                 |         |  |
   |        |  Dict  |        |                 |         |  S
   |        |  Area  |        |                 |         |  P
   |        |        |        |                 |         |  A
 60|        | Min=6K |        |                 |         |  C
   |________|________|________|_________________|_________|  E
   |          (Floating Boundary)                         |  |
   |                                                      |  |
 70|                                                      |  |
   |        Dynamic Buffer area controlled by BUFF        |  |
   |                                                      |  |
   |          Min=1K + blksize of each utility in         |  |
   |              Read Mode + blksize of largest          |  |
 80|              Utility in Write Mode.                  |  |
   |______________________________(Min=8K)________________|  |
   |                 |                                    |  |
   |     Macro       |   Macro directory if BATCH         |  |
 90|   Directory     | dynamic buffer area if NOBATCH     |  |
   |_____(6K)________|____________________________________|  |
   |___OS free core (2K or 4K) for default SP=____________|
   |          Access Methods      (5K)                    |
   |          and QSAM buffers    (1K)                    |
100|______________________________________________________|
```

G. USER MODIFICATIONS

The following procedure should be followed in making changes to the assembler. Update the source member in ASMG.SOURCE. Assemble it into the corresponding member of ASMG.OBJMODS. Linkedit the necessary object decks into ASMG.LOADMODS using ASMG.LKEDIN as a guide to which object decks go into which LOADMODS.

Near the start of many decks appears the statement 'COPY ASMGSET'. This statement copies the definition of two global set symbols from ASMG.MACROLIB. These set symbols are &DEBUG which controls the outputting of some debugging information, and &STAT which controls the outputting of some statistics information. If either of these switches are turned on in ASMGSET and one or more of the decks that uses them is reassembled, then a //STATDBUG DD SYSOUT=A card is needed. Everything printed on STATDBUG is done by a OPEN, PUT, CLOSE, FREEPOOL sequence, which results in very slow execution.

The default EXEC parms are set in deck ASMGF1, CSECT COMMON at location 'EXTMWD' and following. The default DD names are set in deck ASM near location 'DEFDDNAM' and also in the DCB definitions in ASM. The default DCB BLKSIZE and BUFNO are specified in deck F1 near location 'DCBTAB'.

If 1000 macros in SYSLIB is too restrictive for an installation, i.e. if too many OS FINDS are being done to complete an assembly and thus reducing performance then GBLA variable &MAXMAC in ASMGSET may be changed from 1000 to a more suitable value. The option of attempting to use SYSGO if SYSLIN is missing is controlled by variable &SYSGO in ASMGSET in the assembly of ASMGF1.

The name of the batch-execute separator card '$JOB' is defined in deck ASM at location '$JOB'.

To add a new instruction set examine one of the present ASMGISXX source listings, particularly the comments at the end of the INSTSET macro. Copy one of the present ASMGISXX source decks into a new source member (using IEBGENER for example) and modify it to suit your requirements.

Explicit GETMAINS are done in deck F1 near locations 'BUFAGAIN', 'GETMAIN', and 'GETVAR'. Implicit GETMAINs are done by the OPEN in deck F1 near 'OPENMAC'. Explicit FREEMAINs are done in F1 near 'ERROR998' and in ASM near 'RETURN' and 'FREEPOOL'. The only explicit WTO in ASMG is in deck ASM near 'AB2'.

To generate support for the model 67 RPQ instructions AX, DX, LX, MX, SX, ADD, MDD, SDD, SLT, ADDR, MDDR, SDDR, and SWPR you should turn on the appropriate symbolic switch in ASMGSET, reassemble decks F7X, F8M and IS67 and relinkedit these decks. If this change is made, then ASMG will not flag R2 of AXR, MXR, or SXR for being 2 or 6.

H. PERFORMANCE

The following bench mark tests were made, running stand-alone under HASP/MVT, on a model J75 with three 2314 disk drives on separate channels and a 2301 drum on another. The system was 18.6 MVT with HASP 2.3. Two series of tests were done. The first was run after an IPL to include two megabytes of Ampex LCS with an effective cycle time of 1.8 times the cycle time of the megabyte of fast core. The LCS was used for the Link Pack area and part of HASP. The second series was run on the machine IPLed with only the megabyte of fast core.

The version of Assembler (F) was Release 20 (15OCT70), of Assembler (G) was V2L4 (19FEB71) and of Assembler (H) was V01. The following parameters were used for all assemblies, where applicable and not already the default.

```
PARM='LIST,XREF,NODECK,LOAD,ESD,RLD,IS=0'
```

Source was read from disk and output was spooled by HASP. All utilities and SYSLIN were pre-allocated disk data sets. An optimal allocation of data sets for each processor was attempted, including the following:

```
ASMF -     separate channels for all three utilities.
     -     UT1 separate channel from SYSIN and SYSGO.
     -     UT3 separate channel from SYSLIB.

ASMG -     separate channels for all three utilities.
     -     UT1 separate channel from SYSLIB.
     -     UT3 separate channel from SYSIN and SYSLIN.

ASMH -     UT1 separate channel from everything.
```

Two test decks were used. The ASMGF2 deck of Assembler (G) is about 4500 statements long and contains almost no macros. The LANDR deck of WATFIV is about 8000 statements long and is 'filthy with many, many' macros.

The Real time, CPU time, number of non-HASP EXCPs and Cost are simply the figures given by HASP for the job. The Cost in University of Waterloo dollars was based on the accounting formula at the time:

Cost (in $) = 700*CPU + .58*C*ERT + .0016*PRT + .0066*PCH + .0014*RD
where:

```
CPU = hours of actual central processor attention.
C   = .833*(R + .002*R*R); a weighted measure of core usage.
R   = kilobytes of core assigned for job step.
ERT = estimated 'real' time that the job step
      would require if there were no higher
      priority jobs stealing CPU attention.
PRT = number of lines printed.
PCH = number of cards punched.
```

RD = number of cards read.

In the final cost comparison ASMG is cheating slightly because it is printing its squished format XREF, thus saving on printer costs. This difference in printer cost amounts to about $1.30 for ASMGF2 (5191 lines instead of 6013) and $2.50 for LANDR (5292 lines instead of 6906). Assemblers (F) and (H) printed 6001 and 5997 lines for ASMGF2, respectively, and 6883 and 6895 lines for LANDR, respectively. Also, no attempt has been made to include the actual cost of any of the assemblers in the cost of running the sample programmes.

The minimum region for Assembler (G) assembling ASMGF2 was 88K and for LANDR was 122K. The minimum region for Assembler (H) assembling ASMGF2 was 176K and for LANDR was 212K. These minimums are not absolute, but depend on what routines are included in the Link Pack area.

Some of the conclusions that can be drawn from the tests are:

The most economical region for Assembler (F) is about 80K.

The most economical region for Assembler (G) is up to 150K.

The most economical region for Assembler (H) is up to 250K. Economical refers to University of Waterloo dollars in the above.

Assembler (F) performance is highly dependent on the number of macros in an assembly.

Macros do not affect Assembler (G) very much.

Assembler (H) loses much of its performance advantage when processing macros.

Assembler (G) suffers least when access methods are in LCS, and (F) suffers the most.

Assembler (H) is faster than Assembler (G) which is faster than Assembler (F).

Assembler (F) requires less core than Assembler (G) which requires less core than Assembler (H).

Little is to be gained running any of the Assemblers in a region far in excess of its minimum requirement.

Assembler performance with LCS

	Real Time			CPU Time			# non HASP			Cost		
ASMGF2	(min.)			(min.)			EXCPs			(U of W $)		
Region	F	G	H	F	G	H	F	G	H	F	G	H
64K	1.44	-	-	.58	-	-	2902	-	-	17.51	-	-
80K	1.21	-	-	.55	-	-	1781	-	-	17.08	-	-
88K	1.24	1.09	-	.56	.53	-	1801	2467	-	17.30	15.93	-
100K	1.21	1.08	-	.55	.53	-	1716	2366	-	17.31	16.03	-
150K	1.16	.83	-	.54	.49	-	1552	933	-	17.97	15.63	-
176K	1.16	.76	.57	.54	.49	.38	1552	776	398	18.37	15.65	15.16
200K	1.16	.74	.56	.54	.49	.38	1553	708	391	18.84	15.92	15.32
250K	1.16	.77	.56	.54	.49	.38	1552	701	389	19.75	16.54	15.77
300K	1.15	.77	.56	.54	.49	.38	1552	687	388	20.84	17.26	16.22
400K	1.15	.80	.56	.54	.50	.38	1552	666	389	23.21	18.86	17.33
750K*	1.16	.71	.55	.54	.48	.38	1552	412	326	34.74	24.09	22.00
LANDR												
64K	8.72	-	-	1.60	-	-	18800	-	-	35.55	-	-
80K	9.13	-	-	1.50	-	-	12825	-	-	33.95	-	-
100K	8.69	-	-	1.40	-	-	11508	-	-	33.75	-	-
122K	8.67	1.67	-	1.40	1.05	-	11395	2496	-	35.31	23.50	-
150K	8.62	1.46	-	1.38	1.03	-	11258	1589	-	37.28	23.27	-
206K	8.63	1.34	-	1.39	1.01	-	11306	969	-	41.83	23.63	-
212K	8.63	1.33	1.67	1.40	1.01	.89	11318	957	1152	43.15	23.87	24.90
250K	8.52	1.34	1.36	1.37	1.02	.88	10944	950	798	46.06	24.83	25.09
300K	8.52	1.36	1.33	1.36	1.03	.88	10944	938	770	51.29	26.16	26.13
400K	8.53	1.39	1.32	1.37	1.05	.88	10944	917	754	63.55	29.27	28.44
750K*	8.52	1.34	1.28	1.35	1.04	.87	10944	702	653	121.77	41.19	38.81

*NOTE: Assembler (F) would not use more than 548K in a 750K region.

Assembler Performance without LCS

	Real Time			CPU Time			# non HASP			Cost		
ASMGF2	(min.)			(min.)			EXCPs			(U of W $)		
Region	F	G	H	F	G	H	F	G	H	F	G	H
64K	1.44	-	-	.53	-	-	2900	-	-	16.91	-	-
80K	1.21	-	-	.50	-	-	1781	-	-	16.54	-	-
88K	1.23	1.13	-	.51	.49	-	1801	2515	-	16.75	15.40	-
100K	1.19	1.10	-	.50	.49	-	1716	2401	-	16.75	15.57	-
150K	1.14	.82	-	.50	.46	-	1552	996	-	17.39	15.13	-
176K	1.14	.75	.55	.50	.45	.33	1552	776	398	17.79	15.18	14.52
200K	1.15	.74	.55	.50	.46	.33	1552	708	391	18.21	15.43	14.69
250K	1.14	.76	.54	.49	.45	.33	1552	698	389	19.06	15.99	15.06
300K	1.13	.77	.54	.50	.46	.33	1552	688	388	20.07	16.71	15.46
400K	1.14	.80	.54	.50	.47	.33	1552	667	388	22.40	18.27	16.45
500K	1.14	.81	.54	.49	.47	.33	1552	628	360	25.09	20.04	17.50
LANDR												
64K	8.72	-	-	1.48	-	-	18801	-	-	34.17	-	-
80K	8.26	-	-	1.41	-	-	12825	-	-	32.87	-	-
100K	7.81	-	-	1.31	-	-	11508	-	-	32.62	-	-
122K	7.79	1.66	-	1.31	1.01	-	11393	2496	-	34.18	22.95	-
150K	7.75	1.46	-	1.30	.99	-	11258	1589	-	36.16	22.74	-
206K	7.77	1.32	-	1.30	.97	-	11306	969	-	40.64	23.09	-
212K	7.78	1.32	1.65	1.31	.97	.83	11318	957	1152	41.85	23.33	24.07
250K	7.65	1.34	1.32	1.27	.98	.82	10944	950	800	44.60	24.26	24.10
300K	7.66	1.35	1.29	1.27	.99	.81	10944	938	769	49.86	25.55	25.05
400K	7.64	1.38	1.28	1.28	1.01	.81	10944	817	754	61.91	28.60	27.26
500K	7.65	1.40	1.28	1.27	1.02	.81	10945	892	747	75.87	31.96	29.82

I. RELEASE PROSE

This section replaces the normal separate Prose document that accompanies each release. Its purpose is to summarize the changes for this release and to show the growth of Assembler (G) since its first release.

ASMG V1L0 Release=18OCT67

ASMG V1L1 Release=03NOV67

1. Assembler (F) updates from Release 11 to Release 12.
2. Fix LTORG problem.

ASMG V1L2 Release=04DEC67

1. Assembler (F) updates from Release 12 to Release 13.
2. True blocksize written on SYSUT1 by ASMGF2.

ASMGV1L3 Release=12JAN68

1. UTBUFF=3 for entire assembly.
2. No iterating allowed in assembly phases.
3. ASMGFI incorporated into ASMGF7.
4. New XREF handling.
5. Implementation of the EXECUTE option.
6. Macro directory built in core.

ASMG V1L4 Release=15JAN68

1. Fixes to bugs in ESD handling.
2. Fixes to bugs in XREF handling.

ASMG V1L5 Release=29FEB68

1. Double buffered reads of SYSLIB.
2. Subsetting of dictionaries in core.
3. Phase ASMGF3 doing less I/O.
4. INSTSET= feature implemented.
5. Fixed bugs in BUFF and F7.
6. BATCH option working.
7. Updates to Release 14 level.

ASMG V2L0 Release=11MAR68

1. Comments and deck names changed to ASMG.
2. BUFF tries to check spillouts to get more core.

ASMG V2L1 Release=12JUL68

1. Two bugs existed in the BUFF routine. These bugs were extremely partition-size dependent and thus appeared quite randomly. The symptoms were variable.
2. The macro directory reading routine in F1 did not work correctly if there were no unused directory blocks on each SYSLIB dataset. Symptoms were undefined opcode error messages for some macros in SYSLIB.
3. ASMG did not work under MVT due to three distinct core management problems and one return/base-register problem.
4. As part of the heading line ASMG now prints 'SYSTEM=XXX', where XXX is 'PCP', 'MFT' or 'MVT'. It gains this information from an undocumented byte in the CVT. Warning -- this byte may not exist in releases prior to 13.
5. The Implementation Guide specified that 2K of core was left free for OS. This was incorrect as 4K was actually left free. In V2L1 ASMG 2K is left free for PCP and MFT, and 4K left free for MVT, the decision based on item 4 above.
6. A program with no END card produced an 0C6 in F2, except that a null program produced an 0C1 in F3.
7. A routine in F2 was sometimes called without the proper base register setting, resulting in an 0C4 abend.
8. If the storage required for subscripted variable symbols was greater than 1K, the dictionary was not correctly subsetted. Possible symptoms include 0C5 or 0C6 in F2, or undefined macro of undefined sequence symbol error messages.
9. If copy code contained an END card, and macros were needed from SYSLIB, an 0C4 occurred in F2.
10. The restore job in ASMG.JCL did not block datasets correctly.
11. BUFF would sometimes use excessive disk space because it did not correctly rewind the utility file.
12. BUFF is altered slightly to reduce the chances of doing unnecessary spillouts.
13. The text output of F7 is single buffered instead of double buffered to save core. This will adversly affect performance if UTBUFF¬=3 is specified.
14. The symbol table grow increment is changed from a fixed 4K to equal .25*SYSUT2&3 blksize to reduce the probable amount of unused core.
15. BUFF's RCL is changed to grow in a fixed increment of 256 bytes rather than by the size of the I/O buffer that happened to be adjacent to it. BUFF's core management is altered so that it is no longer necessary to have an unused maximum length buffer reserved. As well as saving a buffer, the probable unused amount of the RCL is reduced.
16. The rewinding of UT1 and UT2 is moved from the end of F2 to the start of F3 to reduce the chances of BUFF not having enough core to do the rewind.
17. The ASMGISxx modules are made smaller to reduce the minimum dictionary requirement and also the load time. Also they are marked re-entrant in the hopes that MVT may keep them in core during BATCH assemblies.
18. Miscellaneous minor changes have been made to save core.

19. As a result of items 13 to 18, the minimum core check is changed from 98K to 90K (ASMG may actually become G-level someday).
20. Debugging code has been added to BUFF. This code is not generated in the distribution version.
21. ASMG will accept several parms which have not yet been (and maybe never will be) implemented. No mention was made of this fact in either the documentation or the source code.
22. Miscellaneous source cleanup.
23. Miscellaneous documentation cleanup.
24. Heading line changed to 'RELEASE=12JUL68'.

ASMG V2L2 Release=16JUL69'

1. The text is single buffered rather than double buffered for F3 output, F7 input and F8 input to reduce core requirements. This will reduce performance if UTBUFF¬=3.
2. An addressability problem in an F3 error message routine is fixed.
3. The F3 text I/O buffering is improved by allocating three rather than two input buffers (one dedicated to open code). This reduces the number of reads necessary when assembling a program containing many macros.
4. A SPACE option has been added: SPACE=MAX, SPACE=nnn or SPACE=MAX-nnn may be specified. This specifies how much core the assembler is to use. The default is SPACE=MAX-2K for PCP and MFT, SPACE=MAX-4K for MVT. Core management is cleaned up in general.
5. All of the assembler's datasets are opened in parallel.
6. A card in F1 which was dropped by IEBUPDTE in the V2L1 update is reinserted.
7. Load module ASMGASM is aliased ASMG for ease of reference.
8. Seconds are added to the TIME= field on the header page.
9. F2 calculated the number of blocks in the subset dictionary incorrectly if the dictionary size was a multiple of the utility blocksize.
10. The SYNADAF macro for an I/O error on SYSLIB specified ACSMETH=BASM instead of ACSMETH=BPAM.
11. Deck ASMGMAC is deleted. Those parts of MAC still used have been moved into deck ASM.
12. F3 now XCTLs directly to RTA instead of returning to MAC which LINKed to RTA.
13. Message ASMG998A is changed to ASMG998I: INSUFFICIENT MEMORY TO SATISFY MINIMUM SPACE REQUIREMENTS. It is printed instead of typed and is produced by F1 instead of ASM. It indicates that less than 12736+nnn bytes are available for SPACE=MAX-nnn, or that less than nnn bytes are available if SPACE=nnn is specified. The check for sufficient core in ASM is removed. Loading ASMG into a small partition will now result in an 804 or 80A abend.
14. A common print error and abend routine is added to ASM. All abnormal completions are now issued by branching to it.
15. Phases F1 through F3 now do I/O on utilities directly through BUFF rather than indirectly through MAC. Initialization records are no longer written on UT1 and UT3.

16. The default DDNAMES are no longer initialized by code.
17. The ASMGISxx module is kept resident for BATCH assemblies. For non-BATCH assemblies it is temporarily written on a utility and then read back in to simplify core management.
18. Miscellaneous minor coding changes have been made to save core.
19. If default SYSLIN BLKSIZE was used, RECFM=B was not set.
20. NOOVERLAY/OVERLAY unimplemented, option removed.
21. Opcodes LM, STM, BXH, BXLE and EX are added to Model 44 INSTSET.
22. 0C4 abend in F3 if recursive macros.
23. 0C5 abend in F3 if SYSLIB card missing.
24. A COPY statement in a system macro which was incorrect or for which the member did not exist in SYSLIB caused an infinite generation loop.
25. In the ASMG.JCL dataset there were several sample update cards with the ./ in columns 2 and three.
26. The region sizes have been corrected on several jobs in ASMG.JCL and the documentation printing jobs now use a forms parameter instead of allocating to a printer.
27. The SYSPRINT DCB no longer specifies RECFM=S.
28. The chaining field in the unsubsetted local dictionary has been changed from two bytes to three, permitting larger programs to be assembled. Error message 'ASMG991I LOCAL DICTIONARY FULL' has been deleted. This modification can cost a person who doesn't need it up to about 4K in core. The change is due to Christine Packard and George Sjoberg of the Pennsylvania State University Computation Center.
29. Concatenation of unlike datasets on unlike devices is now supported for SYSIN.
30. An extension is implemented permitting PRINT statements to appear in macros and COPY code unless INSTSET=0 is specified.
31. IBM fixes made to Assembler (F) in Release 15/16 which were applicable to ASMG are included. The applicable APAR numbers are 12765, 13222, 12455, 14306, 15405, 16338, 16679 and 17168.
32. Using SYSGO instead of SYSLIN is now a severity 4 error instead of severity 16.
33. Miscellaneous source cleanup.
34. Miscellaneous documentation cleanup.
35. Heading line changed to 'RELEASE=16JUL69'

ASMGV2L3 Release=18JAN70

1. If the unsubsetted local dictionary of a macro exceeded 1K this could cause some of the subsetted local dictionary blocks to be incorrectly initialized. This happened regularly to one of the Release 18 SYSGEN macros.
2. Statement numbers greater than 32767 were printed incorrectly on the listing.
3. When using IS=67, the instruction LRA was not assembled correctly.
4. When RLD was specified, the RLD listing lines included some null (HEX 00) characters. This is acceptable to the printers, but not by some terminal systems.
5. Using Q-cons to make a DSECT into an External Dummy Section resulted in the wrong length being passed to the linkage

editor.

6. The second and succeeding lines of a multi-line MNOTE were not printed.
7. When using the EXECUTE option, CSECTs were not placed on double word boundaries in the LOAD MAP. This could result in incomplete dumps if an error occurred.
8. When using the EXECUTE option, the contents of each TXT card was placed in core followed by a blank. This could have adverse effects if an ORG back was done to place one constant on top of another.
9. When the programmer had a MACRO which was correctly flagged as being previously defined OP-CODE, then later in the assembly some other MACRO might get a similar diagnostic incorrectly.
10. Duplicate sequence symbols were not detected occasionally.
11. An EXTEN option is added. Several extensions to the assembler language are placed under its control. This includes the ability to have PRINT statements in macros, which was previously under control of the INSTSET parameter.
12. Three new system variable symbols are added (if EXTEN is on). These are &SYSDATE (returns the date of the assembly), &SYSTIME (returns the time at which the assembly was started) and &SYSSTYP (returns the section type - CSECT, DSECT, or COM).
13. If EXTEN is on a nested macro can determine the attributes of labels defined in outer macros as well as in the outer program.
14. The COM statement can have a label (if EXTEN is on) allowing you to define and reference blocks of NAMED COMMON. Thanks to Martin Raim at the University of Michigan Computing Center.
15. The execution monitoring program, FEX, has been improved to support imprecise interrupts as generated by Models 91 and 195.
16. By reassembling three of the assembler modules, a user may generate support for the following Model 67 instructions: AX, DX, LX, MX, SX, ADD, MDD, SDD, SLT, ADDR, MDDR, SDDR and SWPR. Again thanks to Martin Raim.
17. If neither BATCH nor EXECUTE is specified, then both SYSIN and SYSLIN may share the same buffers. This can result in an 8K saving when both SYSIN and SYSLIN are blocked datasets.
18. A CLI, BNE, LA, B loop in Phase F2 is changed to a TRT Instruction.
19. A new option, FULLLIST is implemented. This causes each library macro from SYSLIB to be printed as it is edited (i.e. following the END card). It is especially useful when trying to find the location of an error in a library macro.
20. When a line in error occurs under PRINT OFF, NOGEN or NODATA, ASMG now attempts to print the line in error anyway.
21. The error message for ASMG085I is changed to 'RE-ENTRANCY VIOLATION.' and is produced for each line in which the error occurred (only if RENT is specified).
22. The following IBM improvements made to Assembler (F) in Release 18 are in ASMG V2L3.

  Model 85 Op-codes - these are added to IS=0 and IS=1. IS=85 is deleted.

  DOS Option - this is implemented as a combination of a DOS option which invalidates Q-cons and L-cons and suppresses the RLD sort, and a new instruction Set (IS=9) which is compatible with the DOS Assembler (F) Instruction Set. ASMG does not check for any conflicts in the parm field - i.e. DOS may be specified with any other option.

Date-stamping the END card in the object deck. ASMG uses a different format but supplies the same information.

Variable symbol subscripts up to 2500.

ALGN/NOALGN parameter.

OPSYN instruction.

APAR #25195 - limited support of &SYSLIST in mixed mode and keyword macros.

APAR #21602 - warning message before errors from library macro editing. In ASMG this is message ASMG078I and is not produced if FULLLIST is specified.

23. The following Assembler (F) release 17 fixed were applicable to ASMG and have been applied.

    17835, 18268, 18695, 18743, 19077, 19074.

    The following APAR fixes in Release 17 of Assembler (F) are not in ASMG V2L3.

    17717 - chaining to invokers save area.

    18848 - return code 20 instead of abend 20.

24. The following Assembler (F) Release 18 APAR fixes were applicable to ASMG and have been applied.

    21760, 22351, 22860, 23218, 23749, 24527, 25196.

25. Miscellaneous source cleanup.
26. Miscellaneous documentation cleanup.
27. Heading line changed to 'RELEASE=18JAN70'

ASMGV2L4 Release=19FEB71

1. When in BATCH, EXECUTE a GET with an invalid save area address will no longer flush to end of file and then go into the WAIT state.
2. The $JOB control card in BATCH, EXECUTE must now have a blank immediately after $JOB, thus allowing more $ labels. Any installation is encouraged to change $JOB to)JOB (for example) with SUPERZAP which will not restrict the use of a valid label, if this better suits their needs. See label $JOB in ASMGASM.
3. Heading line changed to 'RELEASE=19FEB71'.
4. Conflict between TEST and EXTEN options has been removed.
5. The assembler will now utilize up to eight megabytes (if you've got it). The former limit of one megabyte was felt to be an inconvenient restriction by some.
6. When concatenated SYSLIBs are present, the inclusion of COPY code can lose the location on disk where the assembler must resume processing. Now the assembler finds its own concatenation numbers for NOTE saving it for a later POINT.
7. An AIF statement from SYSLIB with an invalid label could abend the assembler if it is the first error encountered while editing library macros.
8. The presence of self-defining terms in macro prototypes has been removed. This eliminates the possibility of non-existant or erroneous variable types being returned by the T' operator.
9. The processing of a null macro no longer results in 0C4s. This fix includes the IBM APAR P26354 in Release 19.
10. Miscellaneous documentation corrections and changes.
11. Miscellaneous code changes to save a byte here and there.
12. To prevent an 0C6 abend after a given sequence of macro calls,

an extra STM has been added to Phase F3. IBM APAR P27008.

13. To prevent overlaying of the bottom most CML when the current dictionary will hold the next entry in 51 bytes but the next entry is the last, extra dictionary space must be checked for. Thanks to Darrell Wilcox of Boeing Computer Services.
14. In large assemblies the SPACE instruction may not generate blank lines. IBM APAR P27084 in Release 19.
15. A single DSECT with outstanding literals will not initiate a blank CSECT. IBM APAR P27018 in Release 19.
16. Phase F7N has a type code included for L Constants. IBM Release 19.
17. The new System /360 instruction HDV (Halt Device) has been included in all instruction sets except that for a Model 20.
18. A new instruction set has been included. It is INSTSET=70 and includes the thirteen new System /370 instructions plus those instructions from INSTSET=01.
    Note that source modules ASMGF7X and ASMGF8M must be reassembled with GBLB variable &SYM370 set to 1, and load module ASMGIS70 must be included in your JOBLIB.
19. Exceptions to 'NOALGN' parameter not flagged as an alignment error. In F8M an incorrect test is made to make B, BAL, BCT, EX, BXH, BXLE and LPSW exceptions to the 'NOALGN' parameter. IBM APAR P28239 in Release 19.
20. Incorrect output when opcode longer than six characters is generated by a variable symbol. IBM APAR P28241 in Release 19.
21. At the beginning of cross-reference processing in Phase FPP a program check may occur for very large programs. Register one is destroyed by the code which invokes the SHRINK routine in BUFF when existing free core will not be enough to hold the XREF list. Thanks to Bruce Davidson of IBM Canada.
22. A change in error message ASMG057I to read: 'SUBSCRIPT NOT WITHIN DIMENSIONS.'
23. In EXECUTE mode fixed-point and decimal overflow exceptions as well as significance and underflow exceptions in a user's program are no longer trapped by the SPIE in FEX, unless the user sets his own program mask.
24. In EXECUTE mode any previous SPIE will be reinstated by FEX after executing the user's program. This may be useful to programs that LINK to ASMG.
25. In EXECUTE mode if a user's program exceeds his EXTIME then a dump of registers and core will follow. If this dump is not desired then see comments near ISTHATIT in FEX to ZAP a branch instruction.
26. Again in EXECUTE mode a successful STIMER exit is no longer dependent on register 13 not being altered.
27. Lastly in EXECUTE mode if the STIMER exit was taken in the middle of executing a module in the link pack area (e.g. an access method) then the attempt to create a program interrupt and remove the STIMER IRB would fail. Now the user's program area is zeroed before returning which, unless the program is self-regenerating, will eventually terminate in an interrupt and return control to FEX via SPIE.
28. Files 3 and 4 of ASMG24 contain the User's Guide and Implementation Guide, respectively. These files can be used to generate a master copy for the manuals using a TN print chain and full page carriage tape by using IEBGENER programs provided

in the ASMG24.JCL file.

ASMGV2L5 Release=20SEP71

1. Division of the ACTR counter by 2 for each error during macro expansion.
2. Printing both the before and after location addresses for an ORG statement.
3. Cross-reference of literals under control of an LREF parameter. The default is NOLREF.
4. A front end UPDATE facility has been added with ./ DELETE and ./ ENDUP support.
5. Under EXTEN, SETC variables may be declared as having maximum lengths other than eight. The default length may be set via the new LSETC= parameter or explicitly within the program:
   GBLC &A*100,&B*1(256),&C declares &A to have a maximum length of 100 bytes, &B an array each with a max length of one and &C with the default length, normally eight bytes. The maximum length specified may range from 1 to 255 bytes. The second expression of a substring notation may also be as large as 255.
6. Removal of the TRBAL SVC requirement for utilities.
7. Addition of the OS release number to the header page.
8. Addition of routing codes to WTO operations for MCS.
9. The object output on SYSLIN or SYSGO is determined by examining the TIOT instead of trying SYSLIN and if that fails producing 3 error messages and trying SYSGO.
10. Instead of terminating with an ABEND 20 in recognized terminal situations, now a return with a return code of 20 is done to ASMG's invoker.
11. A problem with pre-allocated utilities has been fixed. The utility blocksize may be specified on the DD card but will not be taken from an existing dataset.
12. A bug where a DDNAME was missing in an I/O error message has been fixed.
13. All non-comment MNOTEs are now tagged with '***MNOTE***'.
14. MULT is a synonym for the BATCH option, and NOMULT for NOBATCH.
15. Object cards are packed more efficiently if DS 0H etc., is used.
16. A BLDL is now done on six modules at the start of an assembly. If the BLDL fails a System 806 abend will occur.
17. Most of the entries in the LIST1 common area in the ASMGASM root segment are now references with symbolic displacements.
18. Fix a bug whereby an unblocked SYSIN dataset could not be concatenated to a blocked one.
19. For multiple BATCH assemblies, a summary table of errors will terminate the SYSPRINT listing.
20. A Phase F3 performance bug has been corrected. In given circumstances with macros, too many reads were done on UT3. Also the priority scheme in BUFF has been altered.
21. Under EXTEN, the K' (count) attribute of any SETC variable in a macro may be taken.
22. Under EXTEN, SETC variables which contain B, C or X type self-defining terms may be used in a SETA expression.
23. Under EXTEN, a new system variable symbol &SYSPARM has been

implemented in macros. Its value is defined by the SYSPARM= parameter and its default value is a null string.

24. Under EXTEN, MACRO and MEND statements may be copied using COPY, into the programmers source stream. Such copied programmer macros may not contain COPY.
25. LINECNT=0 is now a valid parm. This turns off the page heading mechanism for all portions of the listing except for TITLE and EJECT statements in the source.
26. A null parm on the EXEC card will no longer be treated as an error. Now PARM='LOAD,,NODECK' will be valid.
27. A generation parameter has been added to ASMGF1. &MAXMAC is a GBLA variable which defines the maximum number of macros allowed on SYSLIB. The default value is 1000.
28. OPSYN placement has been restructed by setting a switch more often in F2. This gives the same results as an OPSYN preprocessor in F1. Now only ICTL or another OPSYN may precede an OPSYN. IBM APAR P31010.
29. MNOTE, TITLE and PUNCH operands with quotes are restricted. IBM APAR P32950.
30. 337 ABEND if nested program macros with undefined variable symbol in macro call. IBM APAR P28259.
31. Variable with dimension greater than 255 and SYSLIST is used. APAR P32940.
32. Sublist passed to inner Macro Instruction is treated as character string by Inner Macro Definition. APAR P32953.
33. Entry statement not properly flagged when symbol referenced is equated to a symbol declared as external. APAR P32954.
34. Parm=TEST does not generate any SYM entry for private code. APAR P31001.
35. Fix up some restructions on 370 instructions in F7X.
36. Plus sign inserted in statement following a macro when using PRINT NOGEN. APAR P28587.
37. Message ASMG066I is incorrect. P28594.
38. Too low a severity code for ASMG107I. APAR P32938.
39. A DC with previously defined label and Parm=TEST generates no SYM card output for this statement. APAR P31000.
40. ZAP =(LIT1),=(LIT2) causes two error messages but only the first is in error. APAR P28590.
41. Removal of the SYSPRINT DD card requirement if Parm='NOLIST,TERM' specified and SYSTERM opens successfully.
42. Implementation of WXTRN (weak external) in all Assembler (G) instruction sets.
43. Inclusion of the thirteen System /370 instructions in INSTSET=0. Removal of the former INSTSET=70. The System /370 instructions have also been included in INSTSET=1 but they may be removed by setting GBLB variable &STN370 to 0 in ASMGIS01 and reassembling.
44. Implementation of SYSTERM support for TSO. Under HASP, the SYSTERM listing will precede the SYSPRINT listing. Very minor changes were made in the SYSTERM format.
45. An EXECUTE bug in FEX is fixed. If the user's program got a program interrupt, the assembler might go into a dumping loop because the maskable interrupt bits in the PSW were not turned off.
46. If INSTSET=20 is specified then only one ESD item per ESD record is placed in the object program to be compatible with

the Model 20 Loader.

47. The Release Date on the header page has been changed to '20SEP71'.

48. Concatenated SYSIN on unlike devices may fail if QSAM I/O modules are not resident in LINKPACK. Thanks to Ken Kashmarek of the University of Iowa.

49. Address constants with bit lengths may produce wrong object code. APAR P31016.

50. Under EXTEN, assembler opcodes EJECT, PRINT, SPACE and TITLE are allowed between GBLx, LCLx and ACTR statements.

51. Implementation of a fourteenth 370 instruction Monitor Call (MC) in ASMGIS00 and ASMGIS01.

ASMGV2L6 Release=21SEP72

1. Routing codes changed on WTO. ASM.

2. Zero subscript in macro operand not flagged. F3. P39586.

3. &SYSLIST(0) allowed. F3. P39640.

4. All of the common LIST1 area in ASMGASM referenced by symbolic displacements.

5. Order of DCB addresses and DDNAMEs in ASMGASM is symbolic.

5a. Overriding DDNAMEs no longer need be double word aligned. F1.

6. PRINT, SPACE, EJECT and TITLE allowed before GBLx declarations in open code. F2. P32973.

7. If first record of system Macro contains garbage. F2A. P39576.

8. Processing of ASMG996I 'INSUFFICIENT MEMORY TO PROCESS SYMBOL TABLE' would BAL to an incorrect location. FI.

9. If source record is longer than 236 and the record gets segmented on the utility then counts not updated correctly. F3. P45807.

10. Testran SYM code for L-constant incorrect. F7N. P39577.

11. Change to MC (Macro Call) instruction to allow literals as the 1st operand. F7X, IS00 and IS70.

12. Change to SRP (Shift and Round Packed) instruction not to allow literals as the 2nd operand. F7X, IS00 and IS70.

13. Fix a bug that allowed SRP mask to be greater than 9. F7X.

14. IDR support on END card. If only one ID is present the second slot is used for date and time stamps with the count marked one.

15. First machine instruction after S-constant with address error flagged. F8D. P39647.

16. Under EXECUTE, a user's program of size greater than 6K that uses too much CPU time may cause a three instruction loop in the FREECORE routine of BUFF. FD. Thanks to Michael Stack of Northern Illinois University.

17. LREF entry past statement 32,767 would have a garbage statement number. FPP.

18. A change in default parameter from NOLREF to LREF. F1.

19. Overriding parameters from the EXEC card (if any) are printed on the header page. F1.

20. RLD listing would print an unnecessary heading if the number of entries was an exact multiple of LINECNT. FPP.

21. An invalid label on AGO produces incorrect edited text during editing in F2. At best this error would produce an erroneous

ASMG059 UNDEFINED SEQUENCE SYMBOL and at worst would abend attempting to POINT to an invalid disk address. F3.

22. Under BATCH, EXECUTE an initial $JOB card will be supplied by the assembler if omitted. The $JOB card retains its function as a separator between jobs. F1.

23. A user under EXECUTE is expected to do output on SYSPRINT in a PUT MOVE manner with ASA carriage control with LRECL 121 or 133. This is in spite of the fact that the dataset is really PUT LOCATE and machine RECFM. FEX.

24. Record count statistics follow the number of diagnostic messages at the end of the assembly. ASM, F2 and FD.

25. Removal of the SYSLIB dataset requirement. If no SYSLIB DD card is present and macro calls are present then INVALID OPCODE errors will result. F1.

26. Removal of the 1000 macro limit in the SYSLIB dataset concatenations. If more than the maximum number of macros are found then the incore macro table will not be complete and warning diagnostic ASMG208 will be produced. Furthermore ASMG209 will be produced telling the number of conventional OS FINDs that were necessary to complete the assembly (if any). The default incore limit specified by &MAXMAC has been decreased to 800.

27. The title on the header page is changed from 'O/S 360 ASSEMBLER' to 'OS/360 ASSEMBLER'. The title on the UPDATE listing is changed to 'ASSEMBLER (G) UPDATE LOG'.

28. Page numbering for one assembly is now consecutive. In BATCH mode each separate assembly resets the page count as does the BATCH SUMMARY listing.

29. Errors following an 'ASMG078I FOLLOWING ERRORS OCCURRED WHILE EDITING LIBRARY MACROS' no longer point to the END statement if FULLLIST is not specified. Instead the diagnostics point to a generated comment statement following the END card which gives the library macro name being edited when the error occurred. Only one generated comment is present for each macro with errors. F2.

30. The ./ DELETE function of the UPDATE feature has been modified. The keyword arguments SEQ1= and SEQ2= may occur in any order and their operands may be from one to eight alphanumeric characters in length. High order character zeros will be supplied if necessary to make the arguments eight bytes in length. UP.

31. To prevent overlaying the bottom most CML in the dynamic region managed by BUFF when the current dictionary will hold the next entry in 51 bytes but the next entry is the last. Instead 54 bytes plus the length of the optional SYSPARM must be checked. F3.

32. When &SYSNDX is 256 or more and a local macro dictionary is being built at or near the end of a BUFF core block, the next CML block pointers may be destroyed by incorrectly inserting the &SYSPARM parameter. This leads to 0C5 and 0C6 abends in Stage 1 Sysgens in small regions. F3.

33. If the TERM option is specified under TSO, SYSTERM buffers will not be primed to avoid unnecessary line feeds.

34. Nine new System 370 instructions added to IS=0 and IS=70. These are instructions for clock comparator and address translation. IS00, IS70, F7X and F8M.

35. Release SYNADF buffer and save areas to the system in the case of permanent I/O errors. ASM.
36. A new keyword parameter COL= for multiple column output for XREF and RLD. Thanks to Chet Smith at Penn State who sent code to implement this idea and to Andrew Koenig at Columbia University whose code was incorporated almost verbatim. FPP.
37. Generated BNPR and BNZR not recognized in final assembly. F7X.
38. Release date changed to 21SEP72.
39. Lastly this is to acknowledge two ASMG enhancements that were not included. The first was the ability to read squished Wylbur files on SYSIN, SYSUP and SYSLIB from Andrew Koenig at Columbia. The second a mod to cross reference symbols in MACROs listed on SYSPRINT, and an instruction opcode cross reference from Michael Ward and John Keith at Miami-Dade College. Thanks to both installations for passing on their changes.

ASMGV2L7 Release=21FEB74

1. Changes to SYSIN monitoring routines in ASMGASM to allow change of QSAM I/O routine location when crossing concatenations. ASM.
2. Removal of 'I' from the end of all error messages. ASM. FD.
3. CALIGN=n parameter to govern alignment of macro comments in generated statements. ASM. F1. F2. F7X. F8P.
4. UPCOND=n parameter to abort an assembly with UPDATE option if errors are detected. ASM. UP. F3. FD.
5. Release date changed to '21FEB74'. ASM.
6. IDR information changed to 'ASMG21FEB 0207'. ASM.
7. Blksizes of utilities specified on DD cards are saved in a DCB extension, not in the DCBBUFL field. ASM. F1.
8. Utility read mod to circumvent CMS bug passing back maximum record length of data, not actual length of data. BUFF.
9. Unimportant update log messages are indented five columns. UP.
10. Changes to SYSIN read routine in ASMGUP to allow change of QSAM I/O routine location when crossing concatenations. UP.
11. ASMGF1 made a separate module from ASMGF2. ASMGF1 reorganized and resequenced. F1.
12. System checking enhanced to recognize VS1, VS2, and CMS. F1.
13. I/O exits supported for all files except utilities and SYSLIB. ASM. F1.
14. Parm field scan routine redone to work on minumum unique length principle for recognition. Result -- some short forms for example, RLD and ESD, have changed. F1.
15. XREF=SHORT and XREF=FULL support. F1. FPP.
16. UMAP (Using Map) support. F1. F8A. FPP.
17. Scanning errors when getting SYSPARM= argument have been corrected. F1.
18. CMS option supported as default under CMS, optionally under OS to suppress incore macro directory and optimized NOTE/POINT Syslib routines. F1. F2.
19. Assignment of default Utility Blksizes altered to better utilize the device they are on. F1.
20. Mod to move just the initial global dictionary length and no more to solve VS Fetch Protect abend. F1.
21. Assembler (F) and (G) compatible DDname mapping routine for

overriding ddnames. F1.

22. ICTL and OPSYN scanning routines moved from ASMGF1 to ASMGF2. F1. F2.
23. MACRO and MEND are invalid opcodes when COPYing within a system macro. F2.
24. Nested COPY to five levels is supported. Double buffered reads all the way down. F2. F2A.
25. PUSH and POP assembler opcodes are supported to five levels with USING and PRINT operands. F2. F7C. F8A. F8I.
26. Null positional arguments supported in macro prototypes. F2.
27. Extended EQU support. F2. F2A. F3. F7C.
28. System 200 fixed when ASMG992 in ASMGF2 fails to check SYSLIB reads before terminating. F2A.
29. To ignore arguments of PTLB and IPK in the XREF. F7C.
30. Labelled CNOP supported under EXTEN. F7C. F7X.
31. Labelled ORG supported under EXTEN. F7C. F7X.
32. Extended DROP supported under EXTEN. F7C. F7X. F8A.
33. Unlabelled DSECTs supported under EXTEN. F7C. F7X.
34. To ignore second argument of END for XREF. F7C.
35. Support 25 terms, 10 levels of parentheses and unary + and - in the assembly phases, under EXTEN. F7V. F8V.
36. SYM entry for labelled COMMON under EXTEN. F7N.
37. Generated comments in macro operand field when delimited by a blank. F7X.
38. Reorganization and resequence of ASMGF7X. F7X.
39. Support for up to 8 character TITLE labels everywhere except object deck sequence numbers, under EXTEN. F7C.
40. MNOTE operands print starting in column one of Source Statement field. F8P.
41. First Location counter generated by a macro call under PRINT NOGEN will print in LOC field beside last statement of Macro Call. F8P.
42. Operand value of EQU, ORG and USING will print in ADDR2 field. F8P.
43. ASMGFPP reorganized and resequenced. FPP.
44. Fix to ASMGFEX under EXECUTE when assembler abends printing user's dump, giving a dump loop. FEX.
45. Nine new instructions added to IS=00 and IS=70. IS00. IS70.
46. PUSH and POP assembler opcodes added to all instruction sets except IS=00 (Assembler (F)) and IS=09 (DOS).
47. IS=71 created for CMS, based on IS=70 and including the HVC instruction (Hypervisor Call). IS71.
48. Change of default incore macro directory size from 800 to 1000 entries. F1.
49. Miscellaneous source cleanup.
50. Documentation revision and conversion to SCRIPT for formatting.

ASMGV2L7a Release=21MAR76

1. Change Release date from 21FEB74 to 21MAR76. ASMGASM -- V7A01.
2. To allow a label on other than the first TITLE statement of an assembly, under EXTEN. ASMGF7C -- V7A02.
3. Addressing exception fix under NOEXTEN with Scale, Type and Length attributes. ASMGF3 -- V7A03.
4. To correct printing of TITLE source statement when TITLE is

continued. ASMGF2, ASMGF2A, ASMGF8P -- V7A04. Thanks to Greg Lancaster.

5. To correct placing comment of generated statements in source column one of listing when CALIGN=1. ASMGF2 -- V7A05.
6. To correct equal sign implying a keyword within quoted positional operand. ASMGF2 -- V7A06.
7. Invalid test of switch for PRINT GEN. ASMGF8P -- V7A07.
8. Fix for TITLE/REPRO/PUNCH operand check for quoted string. ASMGF2, ASMGF2A -- V7A08.
9. Align comments of generated CSECT, DSECT and COM statements in the correct column. ASMGF8P -- V7A09.
10. Successive spaces greater than 255. IBM APAR 52832. ASMGF8P -- V7A10.
11. Erronious literal is flagged but code is generated. IBM APAR 45818 and 67494. ASMGF8M -- V7A11.
12. 0C4 when MACRO is only statement of library member. IBM APAR 61450. ASMGF2A -- V7A12.
13. Expressions allowed for literal duplications and length factors, under EXTEN. ASMGF7D, ASMGF8M -- V7A13.
14. Current Location Counter '*' allowed in Duplication and Length factor calculations in DCs, DSs and literals, under EXTEN. ASMGF7D -- V7A14.
15. Support of 4 byte self-defining terms in Assembly Phases, under EXTEN. ASMGF7V, ASMGF8V -- V7A15.
16. Increase Assembler size by 2K to solve VS2 core management problems (804, 80A) under BATCH and to allow for extra code in this update. ASMGF3, ASMG.MACROLIB(COMMON) -- V7A16.
17. Under EXTEN, allow SETx variable dimension to be up to 9999. ASMGF2 -- V7A17.
18. If more than 255 statements are flagged in an assembly then the "nnn WAS HIGHEST SEVERITY CODE" message will show an incorrect value for "nnn'. ASMGFD -- V7A18.
19. Uninitialized dummy ESD name at start of table could lead to ASMG019 diagnostic for EXTRNs and WXTRNs in extraordinary circumstances. ASMGF7I -- V7A19. Thanks to WATBOL.
20. ./ NUMBER support in UPDATE feature. ASMGUP -- V7A20. Thanks to David Singer.
21. ./ * support in UPDATE feature for comments. ASMGUP -- V7A22.
22. ./ DELET support in UPDATE feature as in UPDAT. ASMGUP -- V7A22. Thanks to Greg Lancaster.
23. Possible erronious column number when first error was in opcode field of first statement. ASMGF7I -- V7A23.
24. 0C5 abend in ASMGF2 when editing an empty macro. ASMGF2A -- V7A24.
25. 0C4 or 0C6 ABEND in ASMGF3 after a MACRO has terminated with ACTR Counter Exceeded. ASMGF3 -- V7A25.
26. 0C5 ABEND in BUFF because KEYLEN value non-zero in DSCB or JFCB before assembler invoked. ASMGF1 -- V7A26.
27. For greater compatability with parm scanning routine of previous ASMG levels, 'N' is now a valid prefix equivalent for 'NO'. Also LD (LOAD), EDict (ESD), RDict (RLD) and LCount= (LINECNT=) made alternate forms. ASMGF1 -- V7A27.
28. Under EXTEN, 'END' statement allowed in COPY code. ASMGF2 -- V7A28.
29. Label field support for USING MAP Option. ASMGF8A, ASMGFPP -- V7A29.

30. Looping SYSTERM output with macro call error continued over six cards. ASMGF8I -- V7A30.
31. TESTRAN SYM card support for simply relocatable EQUs, labelled LTORGs, CNOPs and ORGs. ASMGF7N -- V7A31.
32. Change maximum value of EXTIME= parameter from 999 to 9999 to agree with the documentation. ASMG.MACROLIB(COMMON) -- V7A32.
33. Clear first column of SYSTERM error number in case Statement Number exceeds 9999. ASMGFD -- V7A33.
34. Bad label with extra ampersands in a Macro Prototype causes 0C6 in ASMGF3. ASMGF2 -- V7A34.
35. BAL to incorrect address in ASMGASM in case of ASMG997I error. Fixed by changing ASMG997I termination to ASMG114 error on one statement. Thus the assembly can continue but will almost surely run out of core in the XREF processing. ASMGF8A, ASMGFD -- V7A35.
36. Incorrect spelling of PROTOTYPE in ASMG093 diagnostic. ASMGFD -- V7A36.
37. Incorrect spelling of THAN in ASMG335 diagnostic. ASMGUP -- V7A37.
38. Label containing an equal sign on an undefined operation code caused 0C6. ASMGF2 -- V7A38.
39. A generated statement that is too long (ASMG040) produces 0C1 in print phase. ASMGF3 -- V7A39.
40. Documentation error where EXTIme= was shown with an incorrect minimum length. ETime= was added for compatability with previous ASMG levels. ASMGF1 -- V7A40.
41. Under EXTEN, treat an MNOTE with only a quoted string as an operand (i.e. Severity code is not specified or implied) as a comment when printing. ASMGF7C -- V7A41.
42. Direct support for Wylbur format input files on SYSIN, SYSUP and SYSLIB. Thanks to Andrew Koenig of Columbia. ASMGWYL, ASMGASM, ASMGF1, ASMGF2A, ASMGUP -- V7A42.
43. SEQuence/NOSEQuence parm for Wylbur input files. COMMON, ASMGWYL, ASMGF1 -- V7A43.
44. YFlag/NOYFlag parm support to optionally suppress the 'ASMG046 AT LEAST ONE RELOCATABLE Y TYPE CONSTANT IN ASSEMBLY' diagnostic. ASMGF1, ASMGFD -- V7A44.
45. After a user program interrupt in EXECUTE mode, the PIE may be altered before being formatted for print. ASMGFEX -- V7A45.
46. Fix possible bad DDNAME in an ASMG200 disgnostic. ASMGF1 -- V7A46.
47. ./ CHNGE support in UPDATE feature as in UPDAT as an alias for the ./ CHANGE function. ASMGUP -- V7A47.
48. Restore second base for ASMGF1 open exit when HARDCOPY option used with UPDATE feature. ASMGUP -- V7A48.
49. Make system variable symbols, such as SYSPARM, available in open code. ASMGF2, ASMGF3, COMMON -- V7A49.
50. Check for change in output access method address for QSAM files. ASMGASM, ASMGWYL -- V7A50.
51. Length attribute (L') of Extended EQU symbol with length of zero produced Addressability errors. ASMGF7V, ASMGF8V -- V7A51.
52. Change to operation of SPACE=MAX- parameter to free storage at high end of core. ASMGF1 -- V7A52.
53. Intermixing of Positional and Keyword parameters in Macro Prototype and Macro Instruction statements, under EXTEN.

ASMGF2, ASMGF3 -- V7A53.

54. Addressability adjustments. ASMGF2, COMMON -- V7A54.

55. LCLx, GBLx and ACTR statements can appear anywhere before the variable's use in Open code or within a Macro definition, under EXTEN. ASMGF2, ASMGF3 -- V7A55.

56. Under EXTEN it is valid to use the K' (count) operator to get the length of a SETA or a SETB variable. ASMGF2A, ASMGF3 -- V7A56.

57. Fix for a possible bad register when an ASMG305 diagnostic occurs under EXECUTE. ASMGFEX -- V7A57.

J. REPORTING BUGS

Present plans indicate that there will be active support of Assembler (G) until at least January 1977. Assistance in overcoming bugs in ASMG is available from the University of Waterloo.

Bugs should be reported to:

Mrs. Sandi Ward
Computing Centre
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

When reporting bugs the following material should be included:

1. SYSUDUMP if applicable.
2. Source tape or listing to reproduce the problem.

```
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
****  ********  ***          ***  ************            **************
****  ********  **            **  ************           ***************
****  ********  **  ********  **  ************  ******  ****************
****  ********  ************  **  *******************  *****************
****  ********  ************  **  ******************  ******************
****  ********  ***********  ***  *****************  *******************
****  ********  *********  *****  *****************  *******************
*****  ******  ********  *******  *****************  ******AAA**********
******  ****  *******  *********  *****************  *****A***A*********
*******  **  ******  ***********  *****************  *****AAAAA*********
********    ******            **            *******  *****A***A*********
*********  *******            **            *******  *****A***A*********
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
*******     ********     *****  ***  *****  **********    **  **********
******       ******       ****  **    ***    ********      *  **********
*****  ****   ****  *****  ***  *  **  *  **  ******  ****    **********
****  ****  *  ***  **********    ****   ****  ****  ******   **********
**** ****  **  ***  **********   *****  *****  ***  ********  **********
********  ***  ****     ******  ******  *****  ***  ********  **********
*******  ****  ******     ****  ******  *****  ***  ********  **********
******  *****  **********  ***  ******  *****  ***  ********  **********
*****  ******  **********  ***  ******  *****  ***  *******   **********
****  *******  **********  ***  ******  *****  ***  ******    **********
****  *****    ***  *****  ***  ******  *****  ****  ****  *  **********
*****       *   ***       ****  ******  *****  *****      **  **********
******     ***  ****     *****  ******  *****  ******    ***  **********
************************************************************  **********
************************************************************  **********
************************************************************  **********
******************************************************  ****  **********
*******************************************************  ***  **********
********************************************************      **********
*********************************************************    ***********
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
```