Extension of Automatic Flow Charting Capabilities

R. J. Margolin and W. O. Paine
Quality Assurance DSN and Mechanical Hardware Section

A new macro generation facility within the AUTOFLOW II flow charting system was used to process assembly language programs for the MODCOMP II computer. This article describes the nature of this new facility and how it was used, as well as describing other capabilities for automatic flow charting.

In its search for tools to aid in auditing DSN software, the Software Quality Assurance Group has made use of graphic macro facilities within AUTOFLOW II to enable automatic flow charting of assembly language source programs for the MODCOMP II computer used in the Deep Space Network.

This facility in AUTOFLOW II is known as macro definition. It permits the representation of the individual instructions of a target computer (in this case, the MODCOMP II) in terms of flow chart symbols with accompanying text information. A group of macro definition statements is used to express an instruction and each time that instruction occurs:

1. The type and number of flow chart symbols for that instruction are generated.
2. The text associated with each symbol is generated, including literal expressions and indicated parts of the source instructions.
3. In the case of decision, branch, and subroutine symbols, the location of the field which specifies the destination data is given.
4. For decision symbols, the in-line and out-of-line path labels are supplied.

Over 500 lines of macro definition statements were used to form 236 macro definitions of assembler directives and instructions for the MODCOMP assembler with some of the macro assembler features added. Twenty-three of the remaining 213 macro definitions were for

---

1 AUTOFLOW II is a program product of Applied Data Research, Inc., Princeton, New Jersey. There are many modules in AUTOFLOW II. One module, the chart/assembly module, accepts programs written in IBM 360/370 assembly language and automatically provides flow charts and analytical listings which make the structure of the program clearer. This system has been installed on the IBM 360/75 since early 1973.

2 The MODCOMP II computer is a product of Modular Computer Systems, Fort Lauderdale, Florida.
separate definitions of those instructions (indicated in MODCOMP code by an asterisk as the last character of the operation) where the memory reference was indirect. In addition, a group of specific MODCOMP executive service macros was defined.

A simple and conventional approach to graphic macro definition would have been to select the appropriate boxes and merely repeat the source coding along with the comment. It was our desire to provide a type of documentation suitable even for one not reasonably familiar with MODCOMP coding. In an effort to do this, we have expressed the instructions in a combination of English, logical, and program language-related symbolic notation. This enables a new reader to understand the operations more easily without knowing the many details of addressing modes. A certain lack of flexibility in the macro definition feature may cause pairs of parentheses to appear with null contents. This serves to indicate non-use of indexing where potentially available.

Where an assembler itself offers macro capabilities, it is possible to provide parallel AUTOFLOW macro definitions, depending on the complexity of the assembler macros used. This was done by Quality Assurance in the case of the Communication Buffer and Quad Standard Interface Adaptor (SIA) Test Program, part of which is shown as a sample. Part of the macro definitions, source code, and chart output are illustrated in Figs. 1, 2, and 3. Normally, the parallel AUTOFLOW macro definitions would be prepared to the general standard used for the original assembler macros.

In addition to handling 360/370 assembly language programs, AUTOFLOW II also has additional capabilities. AUTOFLOW has a built-in facility known as chart code. It is unrelated to any programming language, but is a vehicle for indicating the structure of the program at a design level. AUTOFLOW can also handle assembly language programs for a number of other computers and a variety of languages, such as PL/1, COBOL, JCL. Some of these are options which may be delivered with the basic AUTOFLOW II, while others are preprocessors which dovetail with AUTOFLOW or produce output that may be input to AUTOFLOW.

The status of other AUTOFLOW activities is as follows:

1. The current options in extensive use are FORTRAN and XDS Sigma Assembly Language.
2. The installed preprocessors are SDS 920, UNIVAC 1108, and CDC 3100.
3. Other preprocessors (which use the macro definition facility) are for the INTERDATA 4 and INTERDATA 70.
4. Other uses have been examined and it is possible to automatically chart assembly code for the PDP-8, NOVA, MAC-16, and PDP-11.

Bibliography


Fig. 1. Part of macro definition statements for MODCOMP instructions
1397  LDI  R2   #4020  PICK UP OX INSTRUCTION  HND04110
1398  ORS  R2,0  ADD GROUP AND UNIT NUMBERS  HND04120
1399  STS  R2   M3COND  STORE OX INSTR IN CONDITIONING  HND04130
1400  RST  R2   M3WRIT  AND WRITE MODE INSTRUCTIONS  HND04140
1401  LDS  R2,5  #0100  PICK UP I0 AND FUNCTION CODE  HND04150
1402  MLR  R2, R2  ZERO I/O STATE  HND04160
1403  ORI  R2   #0100  SET 1/0 STATE TO 1 WRITING MODE  HND04170
1404  STS  R2,5  SET TIMER  HND04190
1405  BLK  R14  #0000  PICK UP CONDITIONING COMMAND  HND04200
1406  LDS  R2   OUTPUT CONDITIONING COMMAND  HND04210
1407  M3COND  OCB  R2,0  PICK UP TI COMMAND  HND04220
1408  LDS  R2,0  M3WRIT  OUTPUT TI COMMAND  HND04230
1409  BRU   DOINT  DEQUEUE NEXT INTERRUPT  HND04240
1410  BLK  R14  EUT  HND04250
1411  M3TERM  BLK  R14  BLK  R14  CHECK STATUS  HND04260
1412  LDI  R2   #0100  PICK UP I/O STATE  HND04270
1413  LDS  R2,5  LTS  R2, R2  HND04280
1414  LTS  R2,2  M3HLL  LTS  R2,2  HND04290
1415  CRN  R2  CINE  R14, R2  M3HLL  HND04300
1416  CRN  R2  CRN  R2  M3HLL  HND04310
1417  CRN  R2  CRN  R2  M3HLL  HND04320
1418  M3HLL  CRN  R2  M3HLL  HND04330
1419  M3CKSM  TCR  R4, R2  M3ST3  HND04340
1420  BRU   SIMUL  GO PROCESS SIMULTANEOUS I/O  HND04350
1421  M3ST3  LDI  R2  #FFFB  PICK UP LENGTH OF 6 WORDS  HND04360
1422  STS  R2,1  16  PICK UP LENGTH OF 16 BYTES  HND04370
1423  LDI  R2  M3BUF  STORE IN TABLE  HND04380
1424  STS  R2,3  2  STORE IN TIB LOCATION  HND04390
1425  LDI  R2  M3BUF  STORE IN TABLE  HND04400
1426  STS  R2,4  2  STORE IN TIB LOCATION  HND04410
1427  LDI  R2  #4020  PICK UP OX INSTRUCTION  HND04420
1428  ORS  R2,0  M3READ  STORE OX IN READ INSTRUCTION  HND04430
1429  ORR  R2  #0010  PICK UP I0 AND FUNCTION CODE  HND04440
1430  M3READ  ORR  R2,0  ZERO I/O STATE  HND04450
1431  ORT  R2  #0010  SET I/O STATE TO 1 READING MEM  HND04460
1432  LTS  R2, 5  BLK  R14  SET TIMER  HND04470
1433  BLK  R14  #0000  PICK UP TI COMMAND  HND04480
1434  M3READ  BLK  R14  OUTPUT TI COMMAND  HND04490
1435  BRU   DOINT  DEQUEUE NEXT INTERRUPT  HND04500
1436  M3STRD  BLK  R14  SET INDICATOR FOR MEMORY PRINT  HND04510
1437  STS  R2,15  M3ST3  STORE BUFFER ADDR IN TABLE  HND04520
1438  LDI  R2,3  M3READ  STORE OX IN READ INSTRUCTION  HND04530
1439  M3STRD  LDI  R2,3  PICK UP I0 AND FUNCTION CODE  HND04540
1440  LDI  R2,3  M3BUF  ZERO I/O STATE  HND04550
1441  M3ST3  ORT  R2  #0200  SET IGS TO 2 WAITING  HND04560
1442  STS  R2,5  BLK  R14  DELAY  HND04570
1443  LTS  R2,5  BRU   DOINT  HND04580
1444  EUT  BRU   HND04600
1445  M3TIME  EUT  PICK UP I0 AND FUNCTION CODE  HND04610
1446  BRU   M3TIME  HND04620
1447  LDS  R2, 5  HND44630
1448  M3RIE  R2,5  CRN  R2  FOUR, M3RIE  HND04640
1449  M3RIE  R2,5  CRN  R2  M3RIE  HND04650
1450  M3RIE  R2,5  CRN  R2  M3RIE  HND04660
1451  CRN  R2  CRN  R2  HND04670
1452  CRN  R2  CRN  R2  HND04680
1453  CRN  R2  CRN  R2  HND04690

Fig. 2. Part of sample source program
Fig. 3. Part of generated flow chart corresponding to sample in Fig. 2