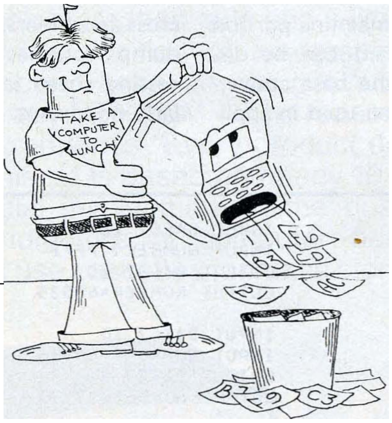
Cal R. Rasmussen
1758th St.
Idaho Falls ID 83401

Hexadecimal Memory Dump



One good article generates another. This is a nice companion to one of last year's.

Kudos to Mark Borgerson. His 6800 assembly-language program for fast loading machine-language programs as it appeared in the February 1977 issue of *Kilobaud* ("Cut 6800 Programming Time with this Extraordinary Program," p. 104) works extremely well. I reassembled that program to relocate it to a convenient location in my memory.

After implementing the loader program, it appeared that a hexadecimal dump program would be a very useful companion program. The MIKBUG P command can be used for that purpose; however, it has two disadvantages. First, it is necessary to use the M command to enter the starting and ending addresses at A002-A004. The second, and more serious, disadvantage is that the output is formatted for the MIKBUG tape Runch and is very difficult to read since there is no spacing between bytes.

Here is a program that overcomes those disadvantages. The display format is not new, by any means, but the only program I have seen for this format is written to run on an Altair 680b by Mits. Since Mits does not use MIKBUG, the program will not run on my SWTP 6800 system. My program will run

Program listing.

```
ØØØØ1                     NAM     HEXDUMP
ØØØØ2               **
ØØØØ3               * HEXADECIMAL MEMORY DUMP PROGRAM
ØØØØ4               **
ØØØØ5               * LOAD VIA MIKBUG "L" COMMAND
ØØØØ6               * USE MIKBUG "G" COMMAND TO START
ØØØØ7               **
ØØØØ8               * ENTER ADDRESS OF FIRST BYTE TO DUMP
ØØØØ9               **
ØØØ1Ø               * ENTER ADDRESS OF LAST BYTE TO DUMP
ØØØ11               **
ØØØ12               * PUSH "RESET" ON COMPUTER TO ABORT
ØØØ13               * CONTROl. RETURNS TO MIKBUG
ØØØ14               **
ØØØ15                     OPT     NOG
ØØØ16                     OPT     S
ØØØ17                     OPT     0

ØØØ19               * MIKBUG ROUTINES USED
ØØØ2Ø         EIDI  OUTEEE EQU    SEIDL
ØØØ21         EØBF  OUT 2H EQU    $EØBF
ØØØ22         EØ47  BADDR  EQU    $EØ47
ØØØ23         EØCC  OUTS   EQU    $EØCC
ØØØ24         EØC8  OUT4HS EQU    $EØC8
lilØØ25       EØE3  MONIT  EQU    $EØE3

ØØØ27               * START PROGRAM
ØØØ28 3F2Ø                ORG     $3F2Ø
ØØØ29 3F2Ø ØØØ2     TEMP   RMB    2          TEMP STORAGE FOR X REG.
ØØØ3Ø 3F22 ØØØ2     l.STBYT RMB   2          ADDRESS OF LAST BYTE TO DUMP
ØØØ31 3F24 ØØØ1     COUNT  RMB    I          COl.UMN COUNTER
```

on any 6800 system using MIKBUG, or one of the newer replacements for MIKBUG.

I have used the same basic dump technique as in the Altair program, but with input/output routines modified for MIKBUG. I have added some prompt messages at the beginning and have used Mr. Borgerson's technique of relocating the stack pointer to restart the program by simply typing G on the terminal.

To use the program after loading, set the program counter at A048-A049 to 3F25 (or the appropriate starting address if you have relocated the program) and use the MIKBUG G command to start execution. The program title will be printed, followed by a prompt message, FIRST BYTE TO PRINT. The address of the first byte to dump is entered and the computer responds with LAST BYTE TO PRINT. The address of the last byte is entered and the dump begins. The display format consists of 16 bytes per line with the address of the first byte being printed at the left (see Fig. 1).

There is no limit to the amount of memory that can be dumped at one time; any number of bytes from one to 65K can be dumped. (Hope you have a lot of paper for the larger numbers!) A word of caution: The address of the first byte to be dumped must be less than that of the last. If this is not the case, *all* memory locations *except* the region between the two addresses will be dumped! If both addresses are the same, only one byte will be displayed.

The dump shown in Fig. 1 is a dump of the dump program itself. This should prove to be a valuable debugging program; especially if your program has "gone to that never-never land known only to CPUs and covered its tracks in the process," to quote another *Kilobaud* author. ■

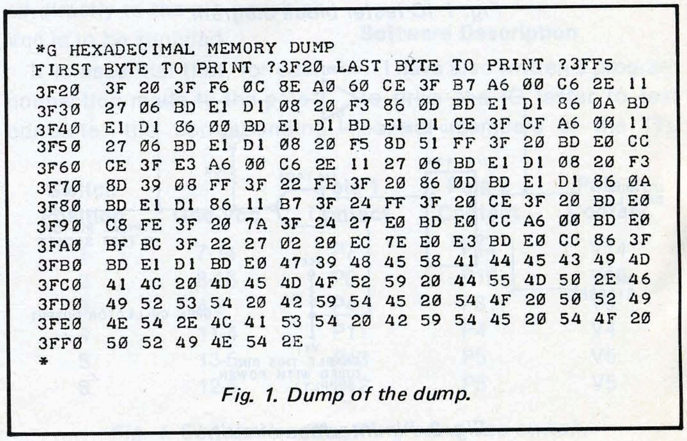
```
ØØØ32 3F25 8E AØ6Ø GO      LDS     #$AØ6Ø   RELOCATE STACK POINTER
ØØØ33 3F28 CE 3FB7         LDX     #TITLE   POINT TO 'TITLE' MESS.
ØØØ34 3F2B A6 ØØ    AA     LDA A   ØØ,X     GET CHARACTER TO PRINT
ØØØ35 3F2D C6 2E           LDA B   #'.      PUT ASCII PERIOD IN B
ØØØ36 3F2F 11              CBA              IS CHAR. IN A-REG A PERIOD?
ØØØ37 3F3Ø 27 Ø6           BEQ     ADRS1
ØØØ38 3F32 BD E1D1         JSR     OUTEEE   PRINT CHAR IN A REG
ØØØ39 3F35 Ø8              INX
ØØØ4Ø 3F36 2Ø F3           BRA     AA       LOOP FOR MORE
ØØØ41 3F38 86 ØD    ADRS1  LDA A   #$ØD     CARRIAGE RETURN
ØØØ42 3F3A BD E1D1         JSR     OUTEEE
ØØØ43 3F3D 86 ØA           LDA A   #$ØA     LINE FEED
ØØØ44 3F3F BD E1D1         JSR     OUTEEE
ØØØ45 3F42 86 ØØ           LDA A   #$ØØ     ASCII NULL
ØØØ46 3F44 BD E1D1         JSR     OUTEEE
ØØØ47 3F47 BD E1D1         JSR     OUTEEE
ØØØ48 3F4A CE 3FCF         LDX     #FIRST   POINT TO 'FIRST' MESS.
ØØØ49 3F4D A6 ØØ    BB     LDA A   ØØ,X     GET CHAR TO PRINT
ØØØ5Ø 3F4F 11              CBA              IS CHAR IN A-REG A PERIOD?
ØØØ51 3F5Ø 27 Ø6           BEQ     GET
ØØØ52 3F52 BD E1D1         JSR     OUTEEE   PRINT CHAR IN A REG
ØØØ53 3F55 Ø8              INX
ØØØ54 3F56 2Ø F5           BRA     BB       LOOP FOR MORE
ØØØ55 3F58 8D 51    GET    BSR     GETADR   GET FIRST ADDRESS
ØØØ56 3F5A FF 3F2Ø         STX     TEMP     STORE IT
ØØØ57 3F5D BD EØCC         JSR     OUTS
ØØØ58 3F6Ø CE 3FE3         LDX     #LAST    POINT TO 'LAST' MESS.
ØØØ59 3F63 A6 ØØ    CC     LDA A   ØØ,X     GET CHARACTER TO PRINT
ØØØ6Ø 3F65 C6 2E           LDA B   #'.      ASCII PERIOD IN B-REG
ØØØ61 3F67 11              CBA              IS CHAR IN A-REG A PERIOD?
ØØØ62 3F68 27 Ø6           BEQ     ADRS2
ØØØ63 3F6A BD E1D1         JSR     OUTEEE   PRINT CHAR IN A REG
ØØØ64 3F6D Ø8              INX
ØØØ65 3F6E 2Ø F3           BRA     CC       LOOP FOR MORE
ØØØ66 3F7Ø 8D 39    ADRS2  BSR     GETADR   GET LAST ADR
ØØØ67 3F72 Ø8              INX              ADJUST IT
ØØØ68 3F73 FF 3F22         STX     LSTBYT   STORE IT
ØØØ69 3F76 FE 3F2Ø         LDX     TEMP     POINT TO FIRST BYTE
ØØØ7Ø 3F79 86 ØD    CRLF   LDA A   #$ØD     SEND CR,LF
ØØØ71 3F7B BD E1D1         JSR     OUTEEE
ØØØ72 3F7E 86 ØA           LDA A   #$ØA
ØØØ73 3F8Ø BD E1D1         JSR     OUTEEE
ØØØ74 3F83 86 11           LDA A   #17
ØØØ75 3F85 B7 3F24         STA A   COUNT    INIT COUNTER
ØØØ76 3F88 FF 3F2Ø         STX     TEMP     STORE X REG
ØØØ77 3F8B CE 3F2Ø         LDX     #TEMP
ØØØ78 3F8E BD EØC8         JSR     OUT4HS   PRINT ADDRESS
ØØØ79 3F91 FE 3F2Ø         LDX     TEMP     RESTORE XREG
ØØØ8Ø 3F94 7A 3F24 NXTBYT  DEC     COUNT
ØØØ81 3F97 27 EØ           BEQ     CRLF
ØØØ82 3F99 BD EØCC         JSR     OUTS     SEND A SPACE
ØØØ83 3F9C A6 ØØ           LDA A   X        BYTE TO A
ØØØ84 3F9E BD EØBF         JSR     OUT2H    PRINT IT, & INCREMENT X-REG
ØØØ85 3FA1 BC 3F22         CPX     LSTBYT   ARE WE DONE?
ØØØ86 3FA4 27 Ø2           BEQ     JMONIT   YES, RETURN TO MIKBUG
ØØØ87 3FA6 2Ø EC           BRA     NXTBYT
ØØØ88 3FA8 7E EØE3 JMONIT  JMP     MONIT
ØØØ89 3FAB BD EØCC GETADR  JSR     OUTS     SEND SPACE
ØØØ9Ø 3FAE 86 3F           LDA A   #'?      SEND QUESTION MARK
ØØØ91 3FBØ BD E1D1         JSR     OUTEEE
ØØØ92 3FB3 BD EØ47         JSR     BADDR    GET ADDRESS
ØØØ93 3FB6 39              RTS              RETURN

ØØØ95 3FB7 48       TITLE  FCC     /HEXADECIMAL MEMORY DUMP./
ØØØ96 3FCF 46       FIRST  FCC     /FIRST BYTE TO PRINT./
ØØØ97 3FE3 4C       LAST   FCC     /LAST BYTE TO PRINT./

ØØØ99 AØ48                 ORG     $AØ48
ØØ1ØØ AØ48 3F2Ø            FDB     $3F2Ø    STARTING ADDRESS IN PROG CTR
ØØ1Ø1                      END
OUTEEE  E1D1
OUT2H   EØBF
BADDR   EØ47

OUTS    EØCC
OUT4HS  EØC8
MONIT   EØE3
TEMP    3F2Ø
LSTBYT  3F22
COUNT   3F24
GO      3F25
AA      3F2B
ADRS1   3F38
BB      3F4D
GET     3F58
CC      3F63
ADRS2   3F7Ø
CRLF    3F79
NXTBYT  3F94
JMONIT  3FA8
GETADR  3FAB
TITLE   3FB7
FIRST   3FCF
LAST    3FE3

TOTAL ERRORS ØØØØØ
```



```
*G HEXADECIMAL MEMORY DUMP
FIRST BYTE TO PRINT ?3F2Ø LAST BYTE TO PRINT ?3FF5
3F2Ø   3F 2Ø 3F F6 ØC 8E AØ 6Ø CE 3F B7 A6 ØØ C6 2E 11
3F3Ø   27 Ø6 BD E1 D1 Ø8 2Ø F3 86 ØD BD E1 D1 86 ØA BD
3F4Ø   E1 D1 86 ØØ BD E1 D1 BD E1 D1 CE 3F CF A6 ØØ 11
3F5Ø   27 Ø6 BD E1 D1 Ø8 2Ø F5 8D 51 FF 3F 2Ø BD EØ CC
3F6Ø   CE 3F E3 A6 ØØ C6 2E 11 27 Ø6 BD E1 D1 Ø8 2Ø F3
3F7Ø   8D 39 Ø8 FF 3F 22 FE 3F 2Ø 86 ØD BD E1 D1 86 ØA
3F8Ø   BD E1 D1 86 11 B7 3F 24 FF 3F 2Ø CE 3F 2Ø BD EØ
3F9Ø   C8 FE 3F 2Ø 7A 3F 24 27 EØ BD EØ CC A6 ØØ BD EØ
3FAØ   BF BC 3F 22 27 Ø2 2Ø EC 7E EØ E3 BD EØ CC 86 3F
3FBØ   BD E1 D1 BD EØ 47 39 48 45 58 41 44 45 43 49 4D
3FCØ   41 4C 2Ø 4D 45 4D 4F 52 59 2Ø 44 55 4D 5Ø 2E 46
3FDØ   49 52 53 54 2Ø 42 59 54 45 2Ø 54 4F 2Ø 5Ø 52 49
3FEØ   4E 54 2E 4C 41 53 54 2Ø 42 59 54 45 2Ø 54 4F 2Ø
3FFØ   5Ø 52 49 4E 54 2E
*
```

Fig. 1. Dump of the dump.