NBSIR 77-1206 (R)

Example of A Numeric and Alphanumeric Technique for Conversion from A Small-Scale Computer to A Large-Scale Computer

Yui-May Chang
Daniel E. Rorrer

Center for Building Technology
Institute for Applied Technology
National Bureau of Standards
Washington, D.C. 20234

February 1977

Prepared for

Division of Energy, Building Technology and Standards
Office of Policy Development and Research
Department of Housing and Urban Development
Washington, D. C. 20410

NBSIR 77-1206

EXAMPLE OF A NUMERIC AND ALPHANUMERIC TECHNIQUE FOR CONVERSION FROM A SMALL-SCALE COMPUTER TO A LARGE-SCALE COMPUTER

Yui-May Chang
Daniel E. Rorrer

Center for Building Technology
Institute for Applied Technology
National Bureau of Standards
Washington, D. C. 20234

February 1977

Prepared for
Division of Energy, Building Technology and Standards
Office of Policy Development and Research
Department of Housing and urban Development
Washington, D. C. 20410

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, ***Secretary***

Dr. Betsy Ancker-Johnson, ***Assistant Secretary for Science and Technology***

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, ***Acting Director***

TABLE OF CONTENTS

EXAMPLE OF A NUMERIC AND ALPHANUMERIC TECHNIQUE FOR CONVERSION FROM A SMALL-SCALE COMPUTER TO A LARGE-SCALE COMPUTER

Yui-May Chang

and

Daniel E. Rorrer

This report describes the characteristic differences in word formats of two different computers, and the software interface technique for conversion from one to the other. Magnetic tapes produced from a small-scale compter were used as inputs to a large-scale computer. One interface program was developed for single-precision floating point numbers. Another interface program was modified from an existing program for alphanumerics. The program for reading real numbers is used as a subroutine in the main program for calculations. The program for reading an alphanumeric coded tape is used by itself to write an ASCII coded file. By using these programs, the UNIVAC 1108 system is able to read and accept data from the Raytheon 704 minicomputer.

Key words: Alphanumeric; computer system; conversion; interface; magnetic tape; program; single-precision floating-point numbers; word format.

1. INTRODUCTION

1.1 Purpose

Most small-scale computers are used for special experimental measurement in scientific and business applications or for data reduction. If the computers have larger memories, they can also be capable of handling most major computations. However, when minicomputers are dedicated as data acquisition systems to collect data, they are oftentimes unavailable for further analysis, processing and display of outputs. Since the purpose of informed data collection is to provide analysis within a reasonably short time, additional supplemental capability for supplying output is required. Also, the preparation of summary output in a format which is suitable for reporting may present a problem when using minicomputers if required peripherals are unavailable. These requirements can be fulfilled by utilizing a large-scale computer system for processing the data. Large computer systems are more efficient because of their high speed and the availability of a large number of system programs for summary calculations and peripheral control.

This report provides a description of programs which were used in the reading of non-standard format magnetic tapes. A technique is also described for converting the data from magnetic tapes written by computers with a 16-bit word format into a format that is acceptable by computers having 36-bit words.

This technical note is based on the development of two interface programs (in Fortran V) to be used specifically between a Raytheon 704 minicomputer (704 Computer) and a UNIVAC 1108 computer system (1108 system) located at the National Bureau of Standards. One program was used for preliminary data analysis while the other was used for documentation of system programs. These programs were prepared for the MIUS/(Modular Integrated and Utility Systems) Total Energy Project sponsored by the Department of Housing and Urban Development.

1.2 Background

The Total Energy Project was initiated to study the potential energy saving by providing electricity, heating and air conditioning from one central building to all other buildings within the complex. The site was chosen in Jersey City, New Jersey and it contains a central equipment building (CEB), four apartment buildings (485 apartments), an office building, an elementary school and a community swimming pool. The CEB and the individual buildings are instrumented to provide measured data from various systems--such as electrical, heating and cooling--for system performance analysis and evaluation. The data acquisition system (DAS), located inside the CEB consists 169 channels of CEB measurements and 112 channels of measurements from other buildings. These measurements are all related to physical parameters, such as flow, temperature, pressure, power, voltage, frequency, etc. The total energy system is being operated continuously and the data are being collected by the DAS scanning every five minutes. All signals coming from the transducers as analog voltages are digitalized by the analog to digital converter and written on the nine-track magnetic tape (raw tape) in EBCDIC*. The 704 computer is employed to obtain required information from the raw tape.

At the early stage of data collection, the 704 computer was committed to trouble shooting instrumentation of individual mechanical systems of different buildings. Simultaneously, monthly summary calculations, printouts and graphic plottings were also required in order to evaluate the total energy system performance. Therefore, after the raw tape was edited, the 704 computer was used to create another tape (either seven-track or nine-track) in binary code for utilizing the 1108 system.

One interface program was developed to provide a capability of printing or plotting real number data (in engineering units). The other (alphanumeric) was used for the dual purpose of document modification of software system programs of the 704 computer and in total energy system performance reporting.

1.3 Approach and Scope

The bit configurations of the 704 computer and the 1108 system are different from each other. There were no available interface programs to provide the necessary processing capability. This report presents a detailed description to meet the requirements for the 1108 system to accept information written on magnetic tapes by the 704 computer.

The program for numerical data is used as a subroutine in the single-precision floating-point conversion. A main program is required to call this subroutine and to perform subsequent processing for analysis, plotting or printing. The program for alphanumeric conversion will write an ASCII (American Standard Code for Information Interchange) file in the 1108 system. This file can be printed out by control cards or other programs.

---

*Extended Binary Coded Decimal Interchange Code. Coding alphanumeric data where 8 bits represent each character.

Magnetic tapes written by the 704 computer were analyzed for format on a bit-by-bit basis. After the tapes are read into the 1108 system, the bits are manipulated in order that the output can be directly accepted by the 1108 system format. For other projects, this work would be available as an illustrative working example of the approach necessary to transfer data from one system to another. It is also anticipated that similar techniques could be used for other specific computer interfacing.

2. SELECTION OF DATA FORMAT AND MAGNETIC TAPE DRIVES

Real numbers and alphanumeric characters produced by a 704 computer have various format representations when written on magnetic tapes. The single-precision floating-point output format for real numbers was chosen since it was sufficient to serve the purpose for future analysis, and since it was presently being written on a seven-track magnetic tape. Other numerical data formats such as integers, double-precision, and mid-precision* were not required, and interface programs for their conversions were not written. Additionally, numerical conversion routines for nine-track magnetic tape input were not considered; however, a preliminary conversion routine has been written. Alphanumeric characters were written in ASCII format on a nine-track magnetic tape by a 704 computer. The nine-track tape was chosen for alphanumerics because of the existence of a one-to-one correspondence in the bit structure of an ASCII character and a nine-track magnetic tape frame. If a seven-track tape had been chosen for alphanumerics, an additional conversion from octal fieldata to ASCII format would have to be made.

3. DESCRIPTION OF WORD FORMAT CHARACTERISTIC

3.1 Real Number

The 704 computer word is 16 bits in length and it takes two data words to form a single-precision floating-point number [1].** As shown in Figure 1, the first eight bits of word 1 contain the least significant part of the mantissa and the last eight bits contain the

---

* In the 704 computer, a mid-precision floating point number is formed by three data words and a range of $\pm 10^{38}$ with significant of 9 digits.

** See Reference at end of text.

the exponent biased by hexadecimal 80.* In word 2, the first bit is the sign bit (0 = positive, 1 = negative), followed by the most significant part of the mantissa. Negative quantities are represented as the two's complement of the positive number.

	Bits 0–7	Bits 8–15
WORD 1	LEAST SIGNIFICANT PART OF MANTISSA	EXPONENT BIASED BY HEXADECIMAL 80

	Bit 0	Bits 1–15
WORD 2	SIGN	MOST SIGNIFICANT PART OF MANTISSA

FIGURE 1. Single-Precision-Floating Point Data Word Format of the Raytheon 704 Minicomputer

---

* Hexadecimal 80 = $80_{16} = 128_{10} = 10000000_2$

A single precision floating-point real number Y ($Y \neq 0$) is represented as

$$Y = 2^{(x - 128)} M$$

where X is the exponent in decimal and M is a normalized fraction (or mantissa) within the following range,

$$.5 \leq |M| < 1.$$

A single-precision floating-point data word of the 1108 computer is held in a 36-bit one's complement format [2]*. The first bit is the sign bit, the next eight bits contain the biased exponent (by hexadecimal 80) and the rest are the mantissa, as shown in Figure 2.

0	1 — 8	9 — 35
SIGN	EXPONENT BIASED BY HEXADECIMAL 80	MANTISSA

FIGURE 2. Single-Precision Floating-Point Data Word Format of UNIVAC 1108 Computer System.

3.2 ASCII Character

An ASCII character is output by the 704 computer as a byte of eight bits in hexadecimal** with the first bit on the left always on [1]* . Each 16-bit 704 computer word will therefore contain two characters.

The 1108 system uses a 7-bit code for ASCII characters with convention in octal**[3].* The eighth bit (first bit on left) is reserved for additional future use and is at present always off. Furthermore, each 1108 computer ASCII character is stored in a 9-bit quarter word within the 36-bit binary word. The ninth bit is used as a stop control bit for peripheral output operations and is forced to a binary 0 on input operations.

Table 1 illustrates the differences in ASCII format bit structure of selected characters.

---

* See References at end of text.

** See Appendix A

SAMPLE ALPHANUMERIC CHARACTERS	RAYTHEON 704		UNIVAC 1108	
	ASCII CODE (HEXADECIMAL)	BITS	ASCII CODE (OCTAL)	BITS
A	C 1	11000001	101	01000001
⋮	⋮	⋮	⋮	⋮
M	C D	11001101	115	01001101
⋮	⋮	⋮	⋮	⋮
1	B 1	10110001	61	00110001
⋮	⋮	⋮	⋮	⋮
9	B 9	10111001	71	00111001

TABLE 1. Differences of selected ASCII characters of ASCII code convention and bit structures as used by the Raytheon 704 and UNIVAC 1108 computers.

4. SOFTWARE INTERFACE

4.1 Real Number

4.1.1 Seven Track Magnetic Tape

The output of the binary code written on a seven track magnetic tape (six tracks for data and one track for parity bits) by the 704 computer is shown in Figure 3 as follows:

1) The first two frames (12 bits) of the record are all 1's (12 bits).

2) Each four frames after the first two, when considered as 24 bits, contain a memory word of 16 bits followed by eight unused bits which are set to 0.

When the 1108 system reads the seven-track magnetic tape, it will read six frames at a time and pack the bits as input words of 36 bits in length (see Figure 4a). These input words do not have any meanings in the 1108 system since they are not in its word format. In order to read and convert real numbers generated by the 704 computer, a subroutine RD704T* has been developed for the 1108 system. This Fortran V routine reads the bit-by-bit configuration of the input tape, decodes it, and manipulates it to the word format of the 1108 Fortran V system (see Figure 4b).

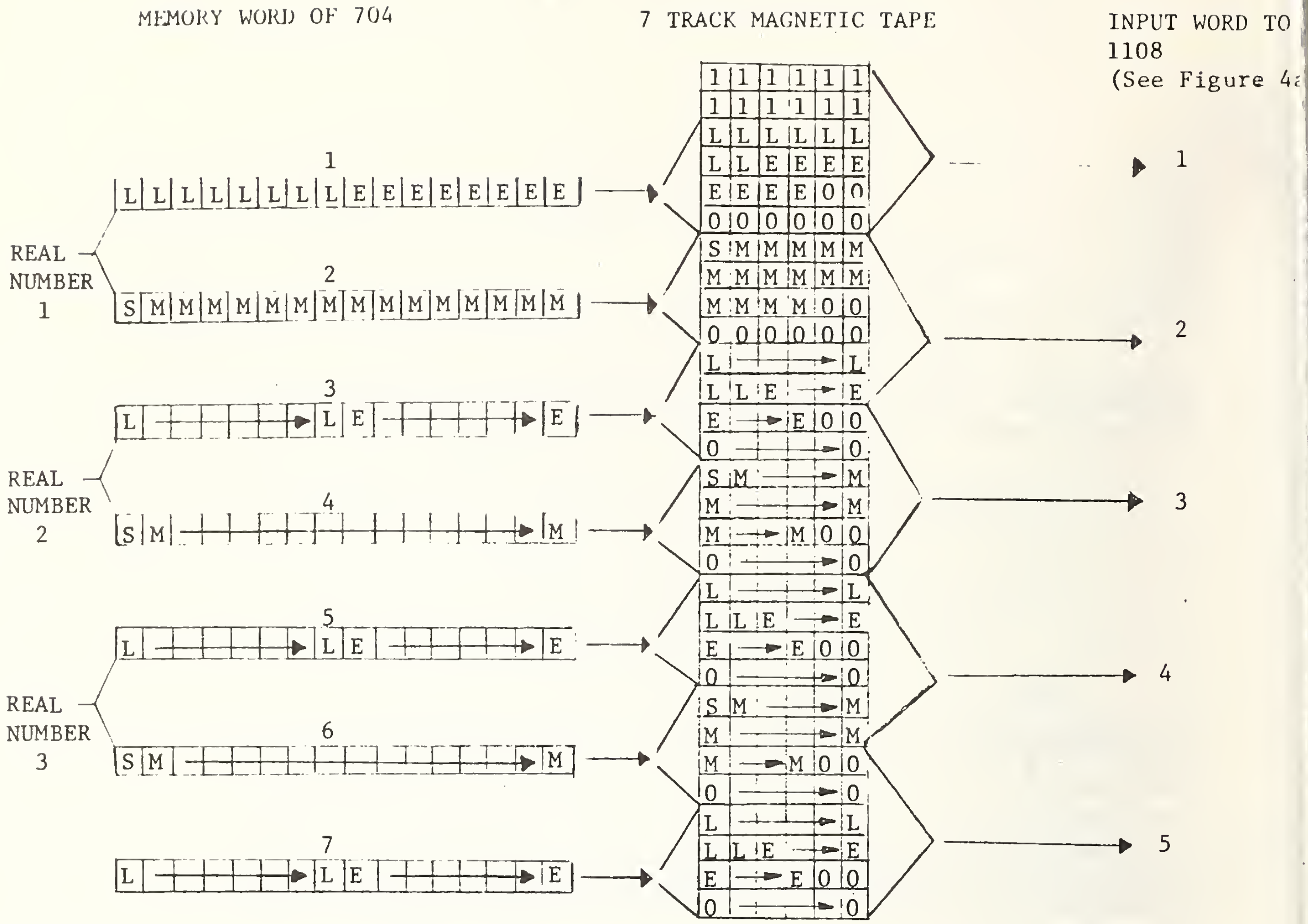
RD704T will read one record consisting of many frames from the tape whenever it is called. Subroutine NTRAN** and function FLD*** (field) are used by RD704T to read the tape and manipulate the bits respectively. As shown in Figure 4a, the first group of five input words (word 1 through word 5) to the 1108 system contains the first three complete real numbers. By using the FLD function, RD704T

---

* See Appendix B

** TEXTPROCESS * LIB. NTRAN is a Fortran subroutine to read a magnetic tape and pack all the bits into each 36-bit memory word of the 1108 system.

*** FLD is used to access the bit string of a computer word (36 bits). The form FLD (i,k,e) means to access k bits of bit string e starting from the i^{th} bit of e. The bits are counted from left to right with left most bit is bit 0.



L = bits making up the least significant part of the mantissa

E = bits making up the exponent

S = sign bit

M - bits making up the most significant part of the mantissa

Figure 3 Bit representation of real numbers as written on a seven-track tape by the 704 computer, and packing of the bits into the 36-bit 1108 computer word.

INPUT WORD TO 1108

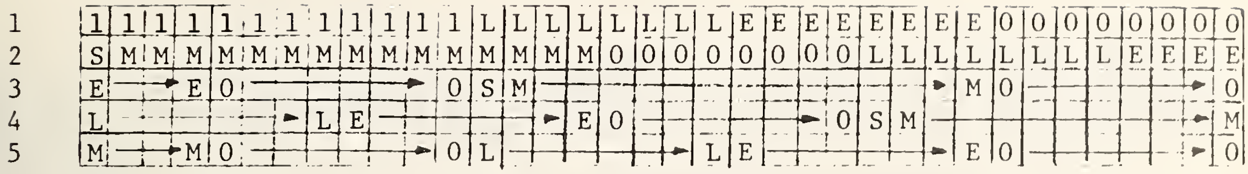


Figure 4a Bit representations after packing the seven-track tape into the 1108 system.

OUTPUT WORD FROM 1108

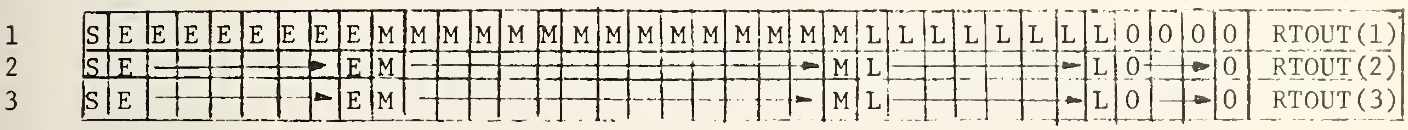


Figure 4b Bit representation of the original real numbers after decoding and manipulation.

arranges these real numbers into three output words (word 1 through word 3), shown in Figure 4b. As an example, the first output word (ROUT(1)) will have the arrangements as follows:

1) First sign bit (S) at bit 0.

2) First group of eight (8) exponent bits (E) at bit 1 through bit 8.

3) First group of fifteen (15) most significant part of mantissa bits (M) at bit 9 through bit 23.

4) First group of eight (8) least significant part of mantissa bits (L) at bit 24 through bit 31.

5) Leave bit 32 through bit 35 empty (or as 0).

The second and third output words are arranged in the same manner as the first one by using second group and third group of respective bits instead. The fifth input word has the same format as the first one and it also contains a part of the fourth real number. Therefore, the second group of five input words is from word 5 through word 9. After the manipulation, three output words (word 4 through word 6) will represent the fourth, fifth and sixth real numbers of the tape.

RD704T essentially takes five input words (with overlapping), and arranges them into three output data words, as shown in Figure 4b. Negative quantities are additionally changed from two's complement to one's complement format. The output of the conversion is placed in the array RTOUT (N).

4.1.2 Nine-Track Magnetic Tape

Each frame of the nine-track magnetic tape (eight tracks for data and one track for parity bits) produced by the 704 computer consisted of 8 bits (one byte) from the 16 bits of one word of memory. The 1108 system packs each word into 36 bits, which is the same in seven-track tapes. A preliminary program has been written to perform these nine-track input conversions; however, further debugging and test runs will be necessary before using.

4.2 ASCII Character

The bit structure of ASCII code produced by the 704 computer is only different from that used in the 1108 system when the left-most bit is considered (see Table 1). Each frame of the nine-track magnetic tape contains one ASCII character (eight bits), as shown in Figure 5. The program TEXT* in Fortran V is a modified program based on an existing program (ASCFLD)** for the 1108 system to read the tape produced by the 704 computer and to write an internal ASCII (1108) file. TEXT will read one record of the tape produced by the 704 computer, change the left-most bit of each frame (byte) and store it in a quarter-word ASCII file of the 1108 system (see Figure 6). The reading of the tape will continue until an end-of-file mark on tape is detected.

There are numerous ways for the 1108 system to output the ASCII file (for example, EDITOR (ED) or PRINT (PRT) could be used to print the entire file). In the case of report preparation or system program documentation, it is necessary to write a short program to control the printing format (with options to change number of lines per page, start a new page, etc.). If only upper-case characters are considered, an ordinary READ statement of FORTRAN will convert the ASCII code to Fieldata by the 1108 system internally for line printer outputs. However, if both upper- and lower-case characters are required, the subroutine AREAD*** could be used. It is therefore left to the user's imagination as to how the converted data could best be used.

ACKNOWLEDGEMENTS

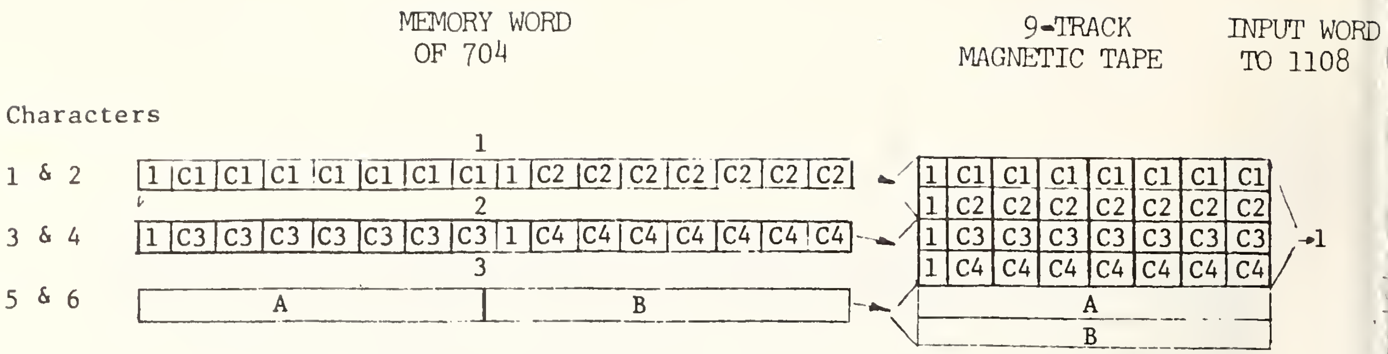
The authors would like to express grateful appreciation to Darcy P. Barnett who helped to identify and modify existing library programs of the 1108 system in order for them to be used. Special thanks to Larry Galowin and Paul Kopetka for their many helpful suggestions for clarity in the presentation of this technique.

---

* See Appendix C

** TEXTPROCESS*LIB. ASCFLD is a program developed by the Computer Service Division of NBS to read a nine-track magnetic tape written in ASCII or EBCDIC and to write a seven-or nine-track tape in Fieldata.

*** IN TEXTPROCESS*LIB.



C_N = the bits making up the ASCII character number N

Figure 5 BIT representation of ASCII characters written on nine-track tape by the 704 computer, and how they are transferred to the 1108 system word.

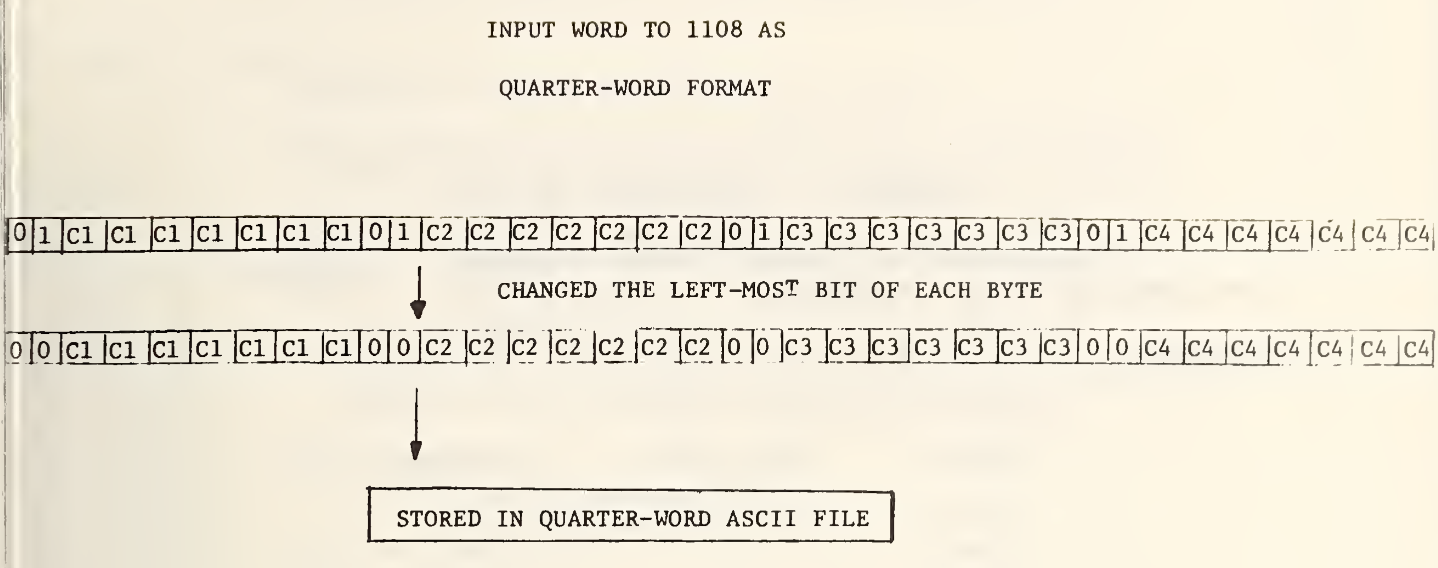


Figure 6 Bit representation of 4 ASCII characters in one memory word of the 1108 system.

APPENDIX A - HEXADECIMAL AND OCTAL

Hexadecimal is a number system of base 16. The symbols are 0 through 9 and A through F.

Examples:

Binary	Hexadecimal	Decimal
0000	0	0
1001	9	9
1010	A	10
1111	F	15
10000	10	16

Octal is a number system of base 8. The symbols are 0 through 7.

Examples:

Binary	Octal	Decimal
000	0	0
010	2	2
100	4	4
111	7	7
1000	10	8

APPENDIX B - SUBROUTINE FOR REAL NUMBER INTERFACING

```
EP*RDTAPE(1).RD704T(0)
C       SUBROUTINE RD704T (RTOUT,ISIZ,L,IMODE)
C
C       THIS SUBROUTINE IS USED TO READ 7-TRACK TAPES FROM
C       RAYTHEON 704 MINICOMPUTER WITH SPECIAL BINARY CODES.
C
C       RTOUT     OUTPUT OF ONE RECORD FROM TAPE
C       ISIZ      NUMBER OF WORDS (<=350) IN ONE RECORD
C       IMODE     THE FILE NUMBER (LUN) OF 1108 TO BE ASSIGNED,
C                 NORMALLY 8. .
C       L  WILL SET TO 1 IF END-OF-FILE IS READ--WRITE 'END-OF-FILE'
C          WILL SET TO 0 IF NORMAL READ OPERATION IS COMPLETED
C          WILL SET TO -1 IF DEVICE ERROR IS DETECTED
C
C       REQUIRES  @MAP
C                   IN TEP*RDTAPE.RD704T
C                   LIB TEXTPROCESS*LIB.
C       WITH MAIN PROGRAM
C
C
      SUBROUTINE RD704T(RTOUT,ISIZ,L,IMODE)
        INTEGER TIN(1200), TOUT(350), E(5),
     1  AML(5), SN(5), SN2(5), EML(5), FC(5)
       REAL RTOUT(ISIZ), RL(350)
        EQUIVALENCE  (TOUT(1),RL(1))
  200 INWD=ISIZ*4/3+2
      I1=0
      CALL NTRAN(8,28,INWD,TIN,L,INC,M,20,NW,25,9)
      IF (L.LT.0)  GOTO 750
      N=1
      DO 600 J=1,L,4
      FLD(1,8,E(1))=FLD(20,8,TIN(J))
      FLD(9,15,AML(1))=FLD(1,15,TIN(J+1))
      FLD(24,8,AML(1))=FLD(12,8,TIN(J))
      FLD(0,1,SN(1))=1
      FLD(1,4,E(2))=FLD(32,4,TIN(J+1))
      FLD(5,4,E(2))=FLD(0,4,TIN(J+2))
      FLD(9,15,AML(2))=FLD(13,15,TIN(J+2))
      FLD(24,8,AML(2))=FLD(24,8,TIN(J+1))
      SN(2)=2**23
      FLD(1,8,E(3))=FLD(8,8,TIN(J+3))
      FLD(9,11,AML(3))=FLD(25,11,TIN(J+3))
      FLD(20,4,AML(3))=FLD(0,4,TIN(J+4))
      FLD(24,8,AML(3))=FLD(0,8,TIN(J+3))
      SN(3)=2**11
      DO 530 K=1,3
       SN2(K)=AND(SN(K),TIN(K+J))
      IF (SN2(K)-SN(K)) 500,400,500
  400 AML(K)=AML(K)-1
      FLD(1,8,EC(K))=FLD(1,8,COMPL(E(K)))
      EML(K)=OR(EC(K),AML(K))
      TOUT(K+J-N)=OR(SN(1),EML(K))
      GOTO 530
  500 TOUT(K+J-N)=OR(E(K),AML(K))
  530  CONTINUE
  550 N=N+1
  600 CONTINUE
      DO 700 M=1,ISIZ
```

```
  700 RTOUT(M)=RL(M)
      IF(L) 704,704,702
  702 L=0
      GOTO 900
  704 L=1
      GOTO 900
  750 IF (L+3) 860,860,800
  800 WRITE(6,850)
  850 FORMAT(' END OF FILE')
      GOTO 704
  860 L=-1
  900 RETURN
      END
END PRT

@FIN
```

```
EP*RDTAPE(1).TEXT(0)
              IMPLICIT INTEGER(A-Z)
        C
        C READ A NINE TRACK TAPE WRITTEN IN ASCII (HEXADECIMAL)
        C AND WRITE A FILE (OR TAPE) IN ASCII (OCTAL)
        C
        C REQUIRES @MAP
        C              LIB TEXTPROCESS*LIB.
        C
        C USES ASCII/GPSDC TO FIELDATA TRANSLATION
        C       GPSDC(COL/ROW)                    FIELDATA
        C
        C                      ⌑   12/15    TO       ⌑  OCTAL 76
        C                      Δ   12/4     TO       Δ  OCTAL 04
        C                      RS  1/14     TO          OCTAL 77
        C
        C
        C
        C CAVEATS
        C
        C STOPS ON TAPE MARK ON INPUT TAPE, WRITES TWO SUCCESSIVE TAPE MARKS ON
        C OUTPUT, BACKSPACES OUTPUT TO BETWEEN TAPE MARKS.
        C
        C BLOCKS LESS THAN 1 CHARACTERS LONG ON INPUT ARE SKIPPED, STANDARD
        C 'NOISE' CRITERION
        C
        C STOPS ON END OF OUTPUT TAPE LEAVING INPUT TAPE EXTENDED
        C
        C
        C INPUT UNIT = 7
        C
        C OUTPUT UNIT = 8
        C
        C SET BUFFER SIZES
        C
              PARAMETER MAXCHR = 4500 @ MAX INPUT BLOCK IN CHARACTERS
        C
              PARAMETER IDIM=2*((MAXCHR-1)/9)+2
             PARAMETER OTDIM=MAXCHR/6+1
             PARAMETER WKDIM=MAXCHR/4+1
              DIMENSION INPUT(IDIM),OUTPUT(OTDIM),WRK(WKDIM)
                 DATA ABLANK/0040040040040/
        C
              NMAX=MAXCHR
        C
        C SET INPUT 7, OUTPUT 8
        C
              LI=7
              LO=8
        C
        C CHECK DEVICE TYPE , 9 TRACK UNIT 8C9 EQUIVALENT
        C
              CALL NTRAN(LI,21,ID)
              IF(ID.EQ.5 .OR. ID.EQ.6 .OR. ID.EQ.11
             . .OR. ID.EQ.12 .OR. ID.EQ.13)GO TO 302
              WRITE(6,300)
        300   FORMAT(1X,'FORCED HALT, INPUT DEVICE WRONG')
              RETURN 0
```

```
 301    FORMAT(1X,I10,' BLOCKS WRITTEN ONTO TAPE')
 302    CALL NTRAN(LI,24) @ ACCEPT FRAME COUNT ERRORS
        CALL NTRAN(LI,25,9) @ ATTEMPT 9 REREADS FOR PARITY ERRORS
        CALL NTRAN(LI,27,1) @ SET NOISE TO ONE CHARACTER
        NBLK=0
 400    CONTINUE
        CALL NTRAN(LI,28,IDIM,INPUT,L,MF,MP,20,NW)
        IF(L+1)404,402,408
 402    CALL STRACE @ STANDARD WALKBACK
 C
 C -1 NTRAN STATUS--SHOULD NOT HAPPEN WITH OPERATION 20 TO WAIT AND
 C UNSTACK.  CALL STRACE GIVES TRACE BACK, THEN
 C
        WRITE(6,301)NBLK
        STOP
 C
 C NEGATIVE STATUS

 C
 404    IF(L.EQ.-2)GO TO 800 @ TEST END OF FILE
        IF(L.EQ.-4)GO TO 900 @ DEVICE NOT OPERATIONAL
 C
 C DATA CHECK
 C
        NBLK=NBLK+1
        WRITE(6,406)NBLK
 406    FORMAT(1X,'DATA CHECK ON INPUT BLOCK',I6,', DATA ACCEPTED')
        GO TO 410
 C
 408    NBLK=NBLK+1
 C
 410    CONTINUE
 C
 C COMPUTE NUMBER OF BYTES IN INPUT
 C
        NB=9*(IABS(NW-1)/2)+MF
        IF(MF.EQ.0)NB=NB+9
 C
 C TEST IF NOISE BLOCK
 C
        IF(NB.GE.1)GO TO 414
 C
        WRITE(6,412)NBLK,NB
 412    FORMAT(1X,'BLOCK',I6,' SKIPPED,',I6,' NOISE CHARACTERS')
        GO TO 400
 C
 414    CONTINUE
 C
 C CHECK FOR POSSIBLE INPUT BLOCK TOO LONG
 C
        IF(NB.LT.MAXCHR)GO TO 420
        NB=MAXCHR
 C
        NIW=NIW+1
        WRITE(6,416)NBLK,NMAX
 416    FORMAT(1X,'WARNING*** BLOCK',I6,' MAY BE LONGER THAN',I6,' BYTES'
      .)
        IF(NIW.LT.20) GO TO 420
```

```
            WRITE(6,418)
418          FORMAT(1X,'*** LAST OF TOO LONG WARNINGS, ONLY 20 GIVEN')
             GO TO 800
420          CONTINUE
C
C COMPUTE LAST WORD TO BE USED IN OUTPUT AND SPACE FILL
C
                NOW=NB/4 @ OUTPUT WILL BE QUARTER WORD ASCII
                IF(MOD(NB,4).NE.0)NOW=NOW+1
                WRK(NOW)=ABLANK
                OUTPUT(NOW)=ABLANK
C
             CALL QWUP(INPUT,1,NB,WRK,1)
C
C TURN OFF FIRST BIT, KEEP AS QUARTER WORD ASCII
C
            CALL BITOFF(WRK,NOW,OUTPUT)
C
C WRITE OUTPUT
C
                CALL APRNTA(LO,1,NOW,OUTPUT) @ ALTERNATE ASCII PRINT FILE
            GO TO 400
C END OF FILE ON INPUT.
C ENDFILE OUTPUT TAPE AND MOVE BACK ONE
800          CONTINUE
                CALL ERTRAN(6,'@BRKPT 8 . ') @ CLOSE ALTERNATE ASCII FILE
             WRITE(6,301)NBLK
             STOP
C DEVICE ERROR
900          WRITE(6,901)L
901          FORMAT(' DEVICE ERROR ON INPUT TAPE'I3)
             GO TO 800
            SUBROUTINE BITOFF(IN,NOW,OUT)
            DIMENSION IN(NOW),OUT(NOW)
            DATA SEVENB/0177177177177/
            DO 1000 NN=1,NOW
1000        OUT(NN)=AND(SEVENB,IN(NN))
            RETURN
C
             END
END PRT

@BRKPT PRINT$
```

REFERENCES

[1] Raytheon 704 Computer User's Manual

[2] NBS Computer User's Guide

[3] UNIVAC 1100 Series EXEC 8 Hardware/Software Summary

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 77-1206 (R)	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE Example of a Numeric and Alphanumeric Technique for Conversion from a Small-Scale Computer to a Large-Scale Computer	5. Publication Date 6. Performing Organization Code
7. AUTHOR(S) Yui-May Chang and Daniel E. Rorrer	8. Performing Organ. Report No.
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234	10. Project/Task/Work Unit No. 4626382 11. Contract/Grant No.
12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)* Division of Energy, Building Technology and Standards Office of Policy Development and Research Department of Housing and Urban Development Washington, D.C. 20410	13. Type of Report & Period Covered NBSIR 3/75-5/75 & 2/76-3/76 14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

This report describes the characteristic differences in word formats of two different computers, and the software interface technique for conversion from one to the other. Magnetic tapes produced from a small-scale computer were used as inputs to a large-scale computer. One interface program was developed for Single-precision floating point numbers. Another interface program was modified from an existing program for alphanumerics. The program for reading real numbers is used as a subroutine in the main program for calculations. The program for reading an alphanumeric coded tape is used by itself to write an ASCII coded file. By using these programs, large-scale computers are able to read and accept data from small-scale computers.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Alphanumeric, computer system; conversion; interface; magnetic tape; program, single-precision floating-point numbers; word format.

18. AVAILABILITY	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
] Unlimited [X] For Official Distribution. Do Not Release to NTIS	UNCLASSIFIED	24
[ ] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151	20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price $3.50

USCOMM-DC 29042-P74