PHILIPS

Electronic components and materials

Technical note 069

2650 microprocessor keyboard interfaces

In many computer systems, the vital link between computer and operator consists partly of a keyboard. The keyboard may be part of a larger unit, such as a teletype or VDU; in other cases, a simple keyboard is sufficient.

Standard keyboard units, containing full decoding and control circuitry, are readily available. However, these units are expensive.

A cheaper solution is to use the required number of keyboard switches and perform the tasks of detection, contact-bounce suppression and switch encoding by software. This also results in a keyboard which is custom built for the microcomputer system. This publication describes four such keyboards.

Of the four keyboard types described here, two require regular sensing of the inputs by the software, while the other two rely on hardware-generated interrupts and an interrupt program to scan the keyboard. A listing of the software required for each system is given in Appendix A.

System 1

System 1 is designed to detect and encode up to eight key-switches, using regular software scanning of the inputs. The hardware and connections to the 2650 microprocessor are shown in Fig. 1 and a flow chart of the software in Fig. 2.

A depressed key causes a '1' on the corresponding bit of the data bus. The program accepts the data from the data bus and generates the appropriate code for the key by a simple algorithm, after a short delay to eliminate the effects of contact-bounce.

The program now loops until it senses that no key is depressed and then waits a further 11 milliseconds to allow for contact-bounce before scanning the switch inputs again. This program thus holds the computer in a loop while a key remains depressed. If two keys are depressed at once, the key of lesser significance is detected.

The bounce-suppression delay of 11 milliseconds corresponds to a clock frequency of 1 MHz. The program assumes that there are no other I/O devices connected to the data bus: otherwise the keyboard buffers must be inhibited during I/O to other devices.

The keyboard is treated as a peripheral connected to the non-extended I/O port C of the 2650.

signetics

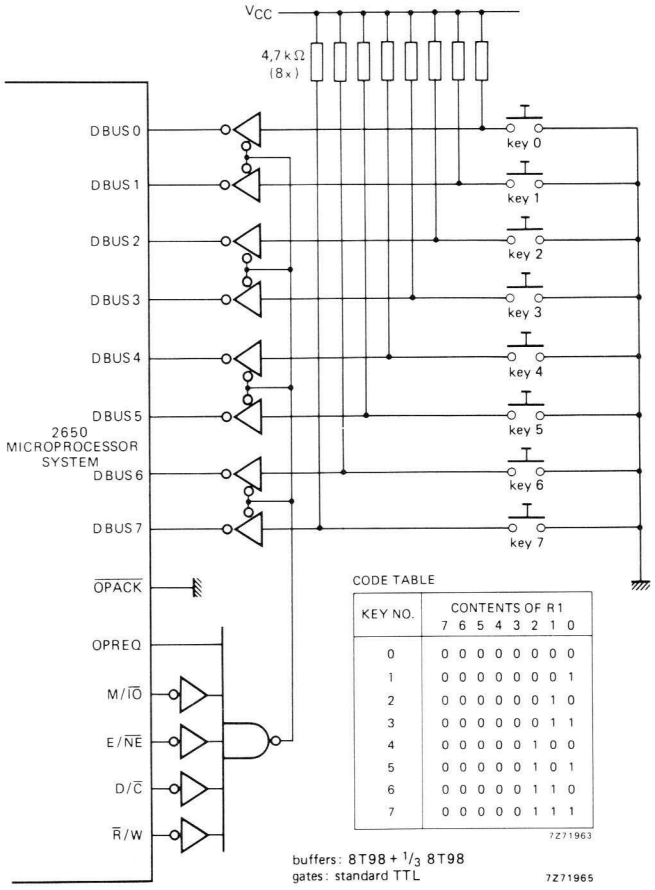

Fig. 1 Hardware for eight-key keyboard, system 1.

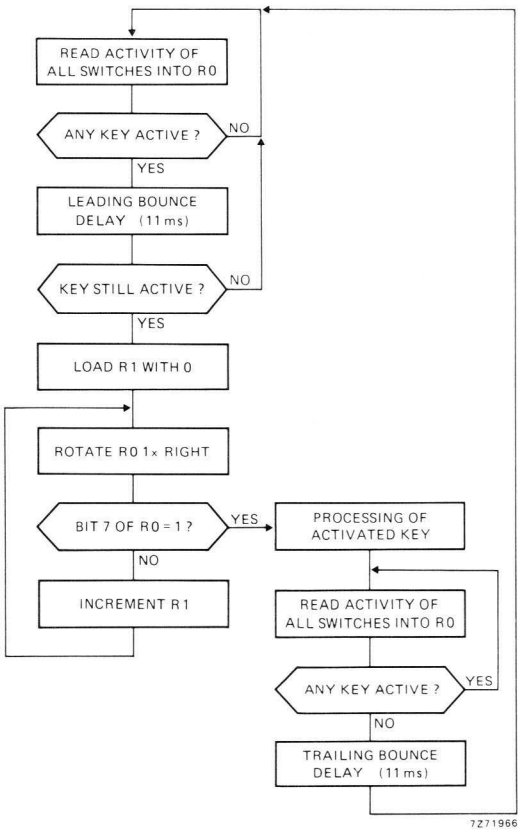

Fig. 2 Flow chart of the software for keyboard system 1.

System 2

System 2 is designed to detect and encode up to sixteen keys. It uses the same method as system 1, but employs the FLAG output to select alternately the two banks of eight switches. More sophisticated software allows the subsequent detection of a second key if two are depressed at the same time and the first is released after detection. If the FLAG output is already in use, other signals can be used in its place, e.g. D/$\overline{C}$, E/$\overline{NE}$, with an appropriate change in the program.

Figure 3 shows the code table for the circuit of Fig. 4 which shows the hardware and connections to the 2650, while the flow chart of the software is given in Fig. 5.

CODE TABLE

KEY NO.	CONTENTS OF R1 7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	1	0
7	0	0	0	0	0	1	1	1
8	0	0	0	0	1	0	0	0
9	0	0	0	0	1	0	0	1
10	0	0	0	0	1	0	1	0
11	0	0	0	0	1	0	1	1
12	0	0	0	0	1	1	0	0
13	0	0	0	0	1	1	0	1
14	0	0	0	0	1	1	1	0
15	0	0	0	0	1	1	1	1

7Z71964

Fig. 3 Code table for the circuit of Fig. 4.

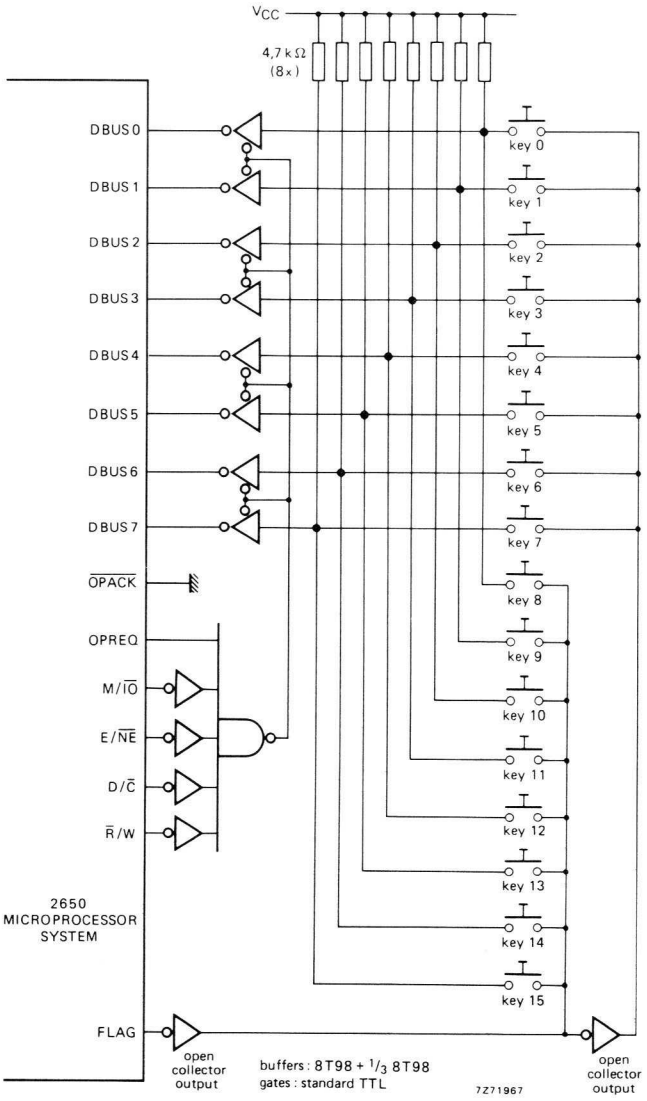

Fig. 4 Hardware for the 16-key keyboard, system 2.

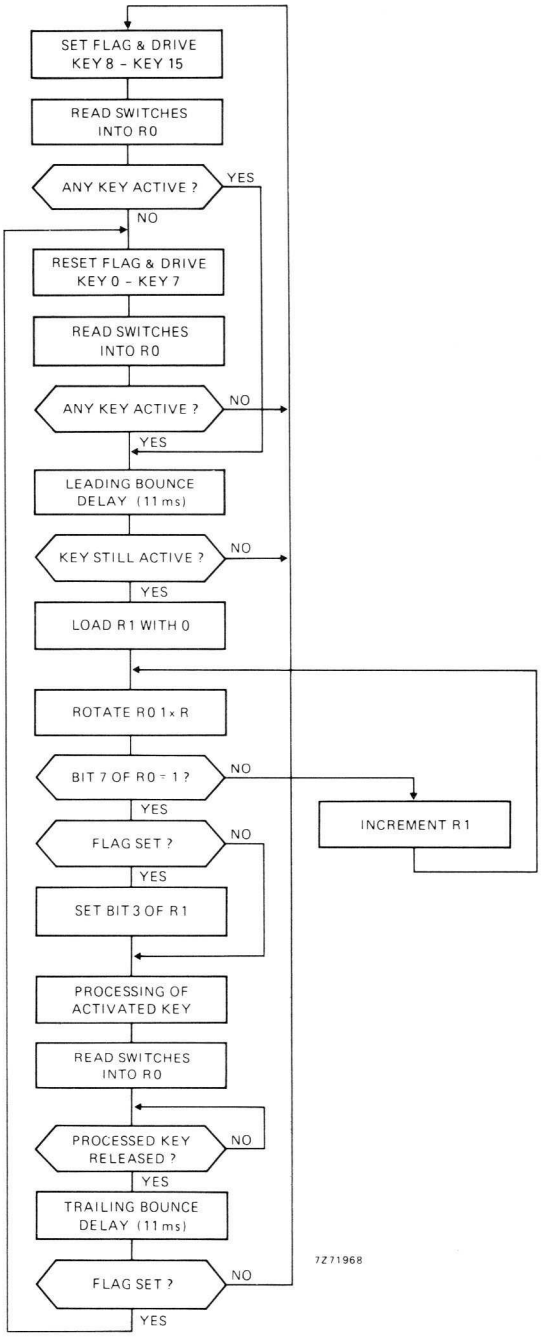

Fig. 5 Flow chart of the software for keyboard system 2.

System 3

System 3 detects and encodes 32 keys in a 4 x 8 matrix. Processing time is minimized by the use of an interrupt program to scan the keyboard input. The interrupt is generated by hardware when a key is depressed.

The hardware is shown in Fig. 6, with the appropriate timing waveforms in Fig. 7. A flow chart of the software is given in Fig. 8.

When a key is depressed, one of the eight inputs to the OR gate becomes low and causes the INTREQ flip-flop to be set. The four scan lines are driven low by the signal ADR13-E/$\overline{\text{NE}}$ (ADR13 is assumed to be unused). After receiving the interrupt request, the microcomputer responds with the interrupt acknowledge signal (INTACK). The INTACK signal selects the A inputs of the multiplexers, setting the interrupt vector address determined by the wire links onto the data bus. This address is then used to locate the keyboard interrupt routine.

The interrupt routine contains four REDE instructions which sequentially sense the four scan lines. After bounce suppression and processing, the INTREQ flip-flop is reset. When the key is released, contact bounce can cause setting of the INTREQ flip-flop and entry to the interrupt routine. The bounce-suppression delay will prevent a second detection of the key.

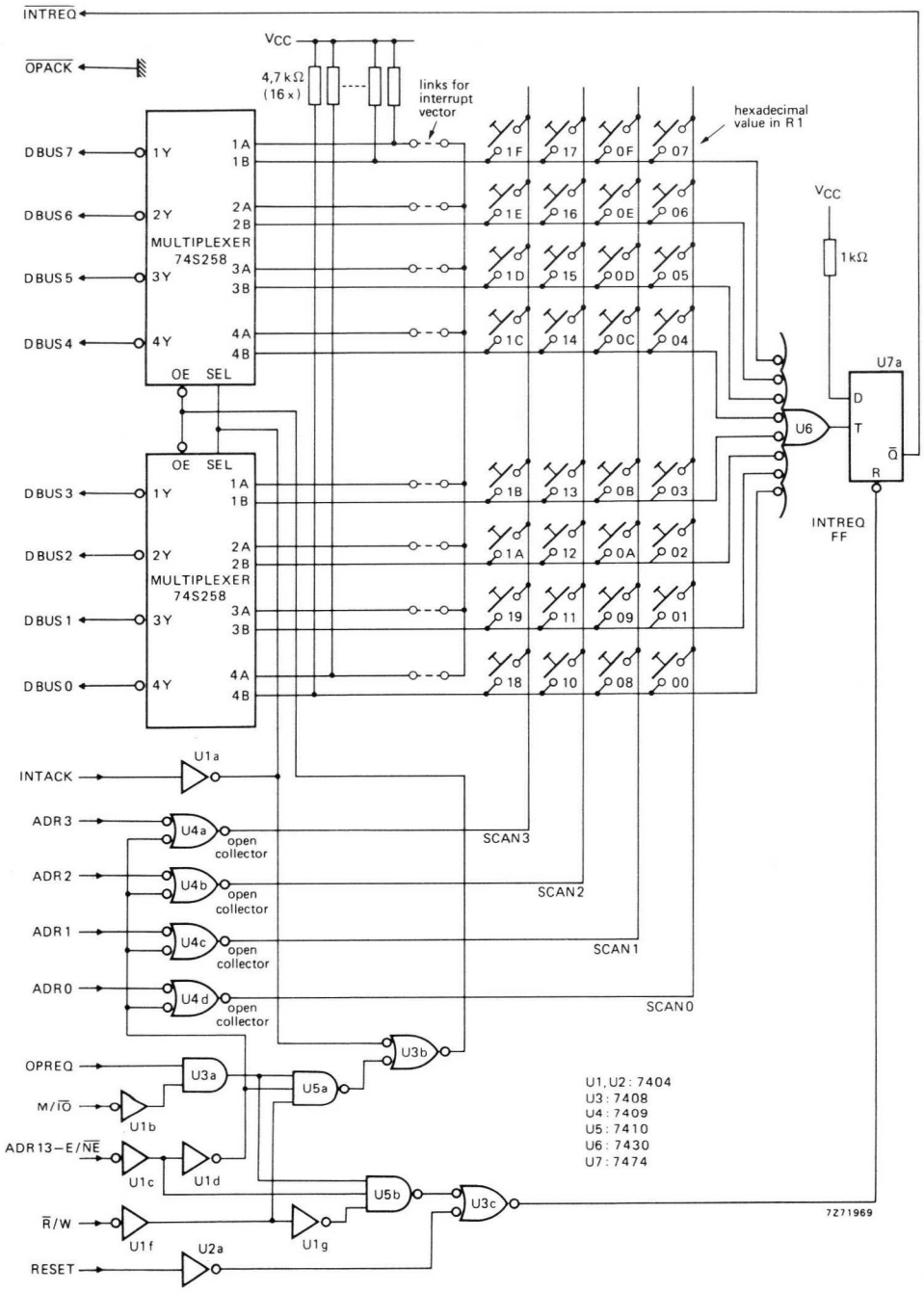

Fig. 6 Hardware for 32-key keyboard, system 3.

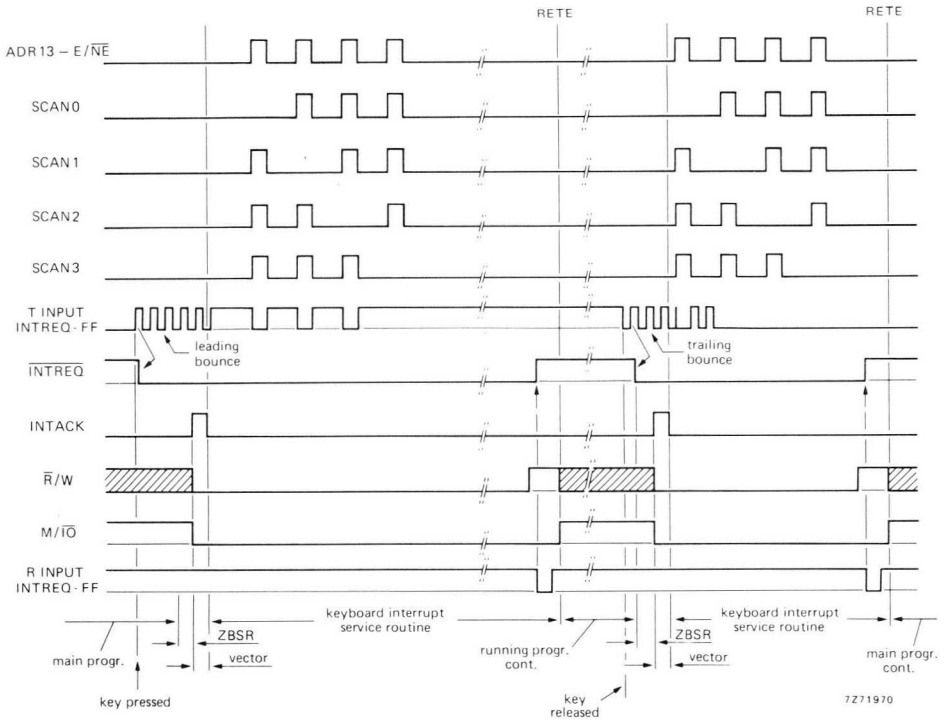

Fig. 7 Timing diagram for keyboard system 3.

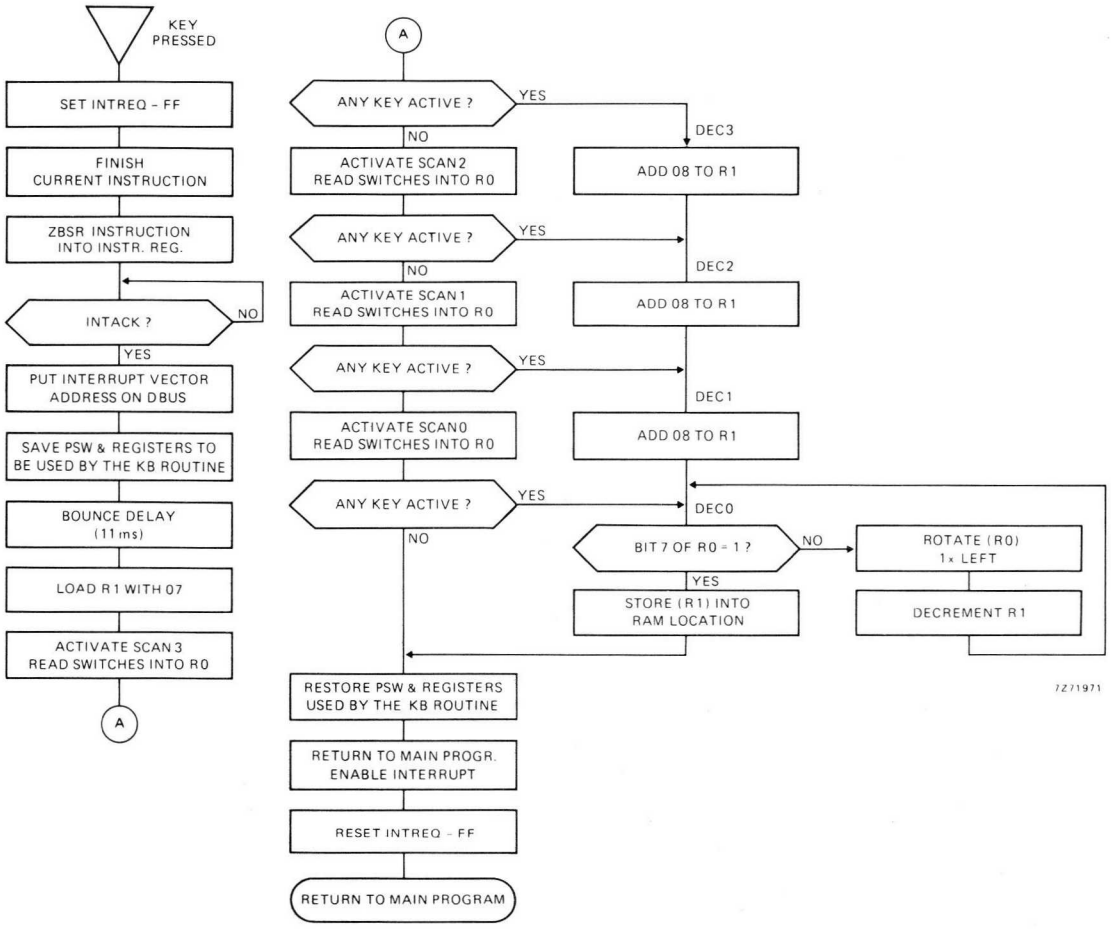

Fig. 8 Flow chart of the software for keyboard system 3.

System 4

System 4 uses an 8 x 8 matrix to detect and encode 63 keys. Figure 9 shows the hardware while Fig. 10 shows the flow chart of the software. The matrix is scanned every 100 milliseconds by an interrupt routine: a 10 Hz oscillator drives the $\overline{\text{INTREQ}}$ line.

When the INTREQ flip-flop is set, the microcomputer replies with the INTACK signal, which causes the multiplexers to set the interrupt vector address on the data bus. This gives the start address of the keyboard interrupt routine.

In the interrupt routine, each column is sequentially scanned by the instructions WRTC, R0 and REDC, R2, R0 contains a single '1' which is rotated one bit position after each scan. Register R2 is loaded with the information from the column scanned. The bounce-suppression delay prevents key-bounce from causing another key detection.

The system has three memory locations to store the depressed keys, so that up to three keys can be detected, awaiting processing. It is assumed that the main program fetches key-codes from the key buffers (LOC1, LOC2 and LOC3) and then performs the appropriate key action. When this action is complete, bit 7 of the key buffer is

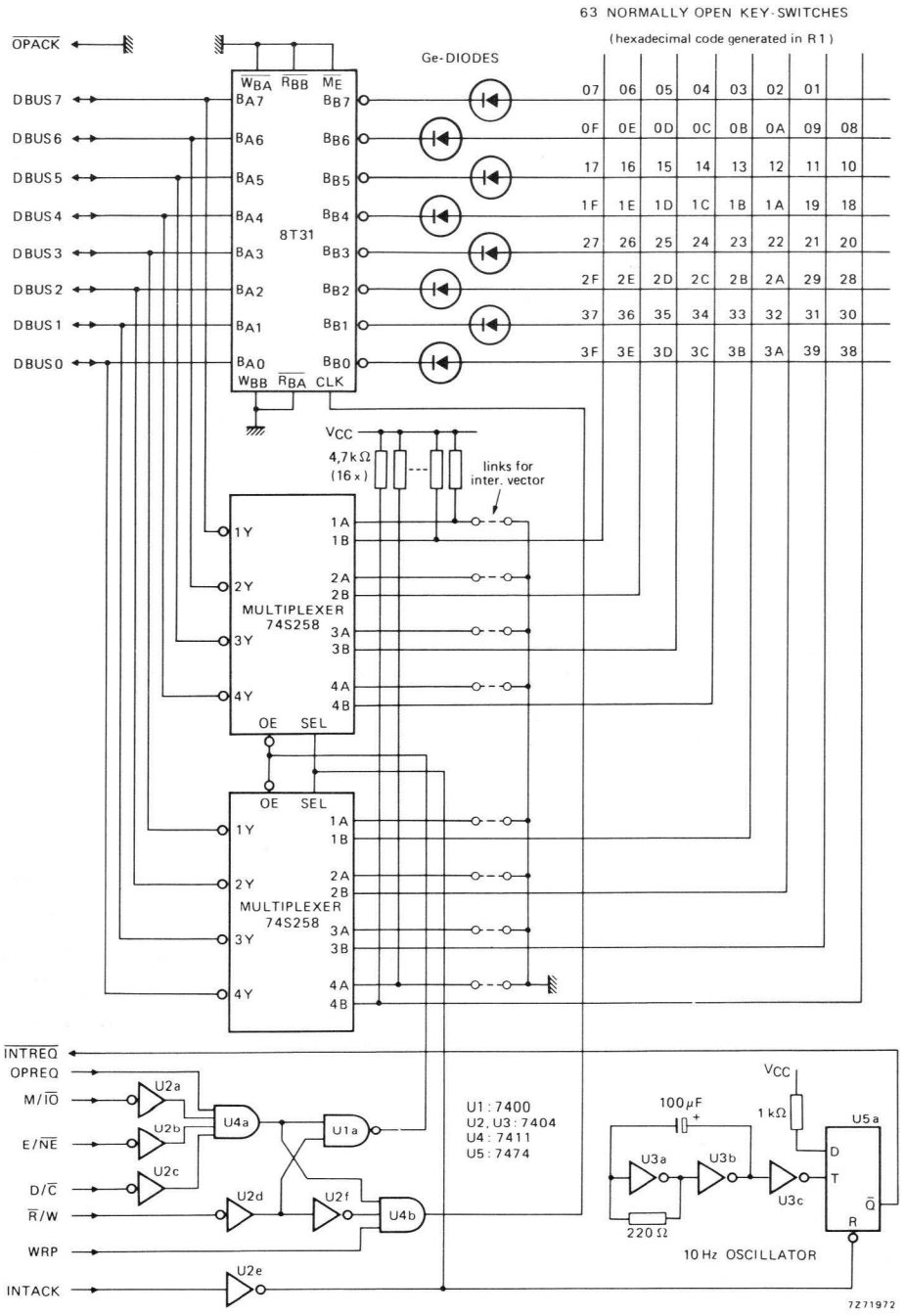

Fig. 9 Hardware for 63-key keyboard, system 4.

set, informing the interrupt program that the key-code is no longer required. After release of that key, the next execution of the interrupt program will result in the contents of that buffer being erased and the remaining key-codes in the other buffers being shifted up in the buffers. For example, if the key-code in LOC1 has been processed (LOC1 bit 7=1) the contents of LOC2 are moved to LOC1 and the contents of LOC3 are moved to LOC2. LOC3 then contains zeros. The next key to be entered will then be stored in LOC3.

Bit 6 of a key buffer is set whenever a key is released before its action has been executed, to provide discrimination between a key that has already been stored and is still pressed and a key that has been pressed for a second time, requiring its code to be stored as the next key-code.

The 10 Hz interrupt clock allows the user to enter characters at a rate of 10 per second; this rate can be increased by using a faster interrupt clock. To prevent incorrect key detection when three keys are depressed at the same time, a diode, not shown on Fig. 9, should be connected in series with each key-switch.

Position 00 of the key matrix is not used because this code is already used to indicate an empty buffer.

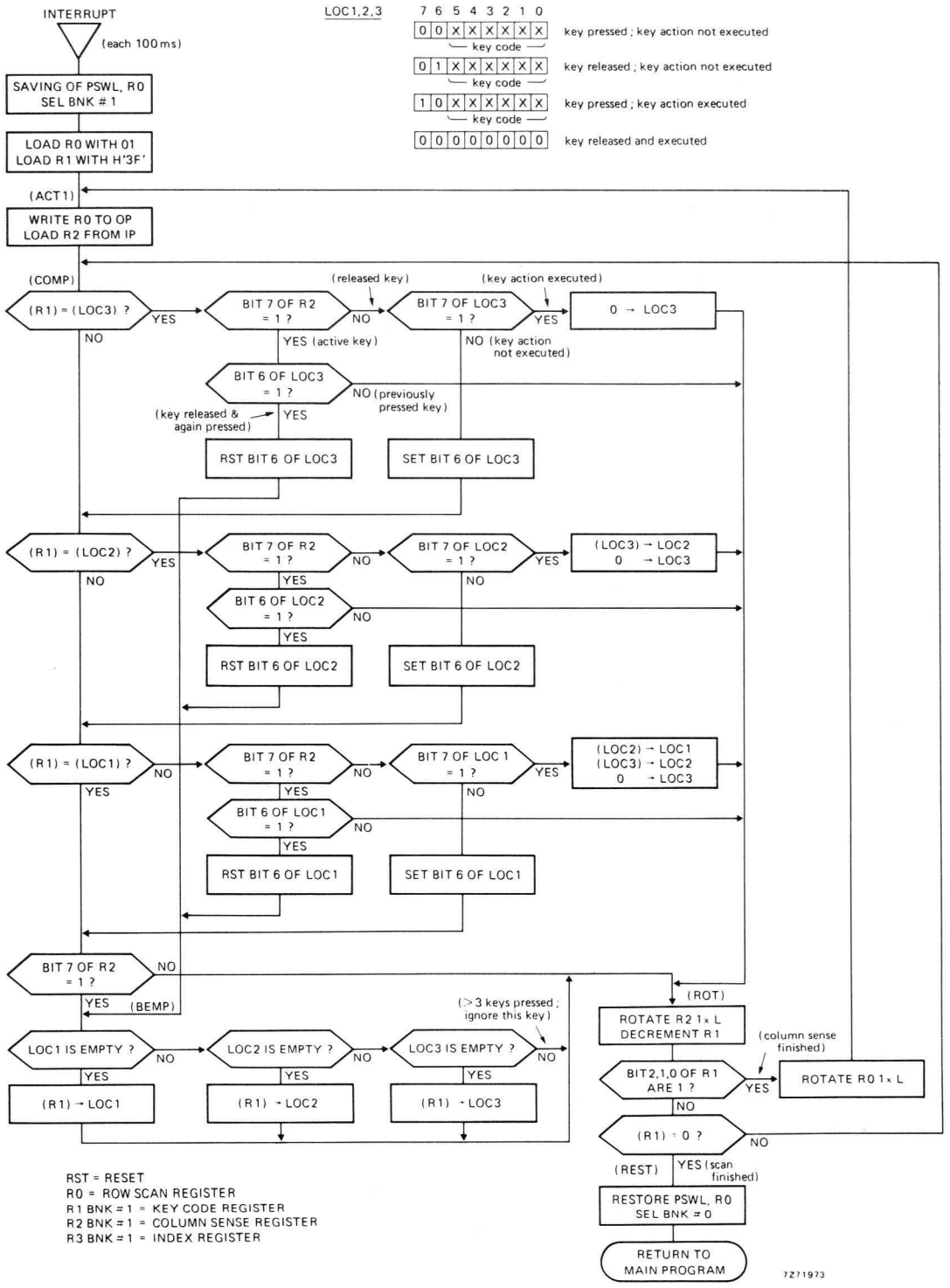

Fig. 10 Flow chart of the software for keyboard system 4.

APPENDIX A. Listings of the software for the keyboard systems.

```
LOC OBJECT   ADDR E STMT   SOURCE LINE

                     2   *
                     3   * DEFINITIONS OF SYMBOLS:
                     4   *
             0000    5   R0   EQU   0        PROCESSOR REGISTERS
             0001    6   R1   EQU   1
             0002    7   R2   EQU   2
             0003    8   R3   EQU   3
             0080    9   S    EQU   H'80'    PSU: SENSE
             0040   10   F    EQU   H'40'         FLAG
             0020   11   II   EQU   H'20'         INTERRUPT INHIBIT
             0007   12   SP   EQU   H'07'         STACKPOINTER
             00C0   13   CC   EQU   H'C0'    PSL: CONDITION CODE
             0020   14   IDC  EQU   H'20'         INTER DIGIT CARRY
             0010   15   RS   EQU   H'10'         REGISTER BANK SELECT
             0008   16   WC   EQU   H'08'         1=WITH, 0=NO CARRY
             0004   17   OVF  EQU   H'04'         OVERFLOW
             0002   18   COM  EQU   H'02'         1=LOG., 0=ARITH COMP
             0001   19   C    EQU   H'01'         CARRY/NO BORROW
             0000   20   Z    EQU   0        BRANCH COND: ZERO
             0001   21   P    EQU   1                     POSITIVE
             0002   22   N    EQU   2                     NEGATIVE
             0000   23   EQ   EQU   0                     EQUAL
             0001   24   GT   EQU   1                     GREATER THAN
             0002   25   LT   EQU   2                     LESS THAN
             0003   26   UN   EQU   3                     UNCONDITIONAL
             0000   27   A1   EQU   0                     ALL BITS ARE 1
             0002   28   N1   EQU   2                   NOT ALL BITS ARE 1
                    29   *
```

Fig. A1. Definitions common to all software.

```
TWIN ASSEMBLER VER 2.1

 LOC OBJECT   ADDR E STMT   SOURCE LINE

0450                 40        ORG    H'450'   START ADDRESS OF PROGRAM
0450 7508            41        CPSL   WC
0452 30              42   DET  REDC,R0          TEST FOR KEYACTIVITY
0453 187D    0452    43        BCTR,Z  DET      BRANCH IF NO ACTIVE KEY
0455 3F0479  0479    44        BSTA,UN DLY      BRANCH TO SUBR. DLY IF KEY ACT.
                     45   *
                     46   *
0458 30              47        REDC,R0          TEST IF KEY IS STILL PRESSED
0459 5802    045D    48        BRNR,R0 KEY       IF NOT RETURN TO KEYDETECTION
045B 1B75    0452    49        BCTR,UN DET
                     50   *
                     51   *
045D 0500            52   KEY  LODI,R1 00
045F 50              53   LOOP RRR,R0           INCREMENT R1 UNTILL A SINGLE
0460 1E0465  0465    54        BCTA,N EXEC       ONE IS FOUND IN BIT 7 OF R0
0463 D97A    045F    55        BIRR,R1 LOOP
                     56   *
                     57   *
0500                 58   EXEC ORG    H'500'
                     59   ****************************
                     60   *PROCESSING OF ACTIVATED KEY*
                     61   ****************************
0500 1F0470  0470    62        BCTA,UN KREL
                     63   *
                     64   *
0470                 65        ORG  H'470'
0470 30              66   KREL REDC,R0          TEST IF KEY IS RELEASED
0471 587D    0470    67        BRNR,R0 KREL     BRANCH IF KEY ACTIVE
0473 3F0479  0479    68        BSTA,UN DLY      TRAILING BOUNCE DELAY
0476 1F0452  0452    69        BCTA,UN DET      RETURN TO DET FOR NEW KEYACT.
                     70   *
                     71   *
                     72   *SUBROUTINE DELAY
0479 0605            73   DLY  LODI,R2 05
047B 20              74        EORZ   R0
047C F87E    047C    75        BDRR,R0 $
047E FA7C    047C    76        BDRR,R2 $-2
0480 17              77        RETC,UN
                     78   *
                     79   *
                     80        END

 TOTAL ASSEMBLY ERRORS   0
```

Fig. A2. Assembly listing of the software for keyboard system 1.

```
TWIN ASSEMBLER VER 2.1

 LOC OBJECT   ADDR E STMT   SOURCE LINE

                      31   *JMB 751223-12.00
                      32   *
                      33   ********************************************
                      34   ********************************************
                      35   *KEYENCODING
                      36   ********************************************
                      37   *THIS PROGRAM ENCODES 16 SINGLE NORMALLY OPEN KEYS
                      38   *TO A 4 BIT CODE IN THE 4 LOW ORDER BITS OF R1;
                      39   *MAX. KEYBOUNCEDELAY IS 11 MSEC;
                      40   *2-KEY ROLL-OVER.
                      41   ********************************************
                      42   *
0450                  43        ORG     H'450'   START ADDRESS OF PROGRAM
                      44   *
0450 7508             45        CPSL    WC
0452 12               46   DET  SPSU             TOGGLE FLAG IN ORDER TO DRIVE
0453 2440             47        EORI,R0 F          HI-ORDER/LO-ORDER KEYS
0455 92               48        LPSU
0456 30               49        REDC,R0
0457 1879     0452    50        BCTR,Z  DET      BRANCH IF NO KEY ACTIVE
0459 3F047C   047C    51        BSTA,UN DLY      BRANCH TO SUBR. DLY
                      52   *
045C 30               53        REDC,R0          TEST IF KEY IS STILL PRESSED
045D 5802     0461    54        BRNR,R0 KEY       IF NOT RETURN TO KEYDETECTION
045F 1B71     0452    55        BCTR,UN DET
                      56   *
0461 C2               57   KEY  STRZ    R2
0462 0500             58        LODI,R1 00
0464 50               59   LOOP RRR,R0           INCREMENT R1 UNTILL A SINGLE
0465 1A02     0469    60        BCTR,N   RDY       ONE IS FOUND IN BIT 7 OF R0
0467 D97B     0464    61        BIRR,R1  LOOP
                      62   *
0469 B440             63   RDY  TPSU    F        IF FLAG SET SET BIT 3
046B 9802     046F    64        BCFR,A1 EXEO      OF R1
046D 6508             65        IORI,R1 H'08'
046F 1F0700   0700    66   EXEO BCTA,UN EXEC
                      67   *
0472 30               68   KREL REDC,R0
0473 42               69        ANDZ    R2       BRANCH IF EXECUTED KEY RELEASED
0474 587C     0472    70        BRNR,R0 KREL        (NEW KEY MAY BE ENTERED)
0476 3F047C   047C    71        BSTA,UN DLY
0479 1F0452   0452    72        BCTA,UN DET
                      73   *
                      74   *SUBROUTINE DELAY
047C 0505             75   DLY  LODI,R1 05
047E 20               76        EORZ    R0
047F F87F     047F    77        BDRR,R0 $
0481 F97C     047F    78        BDRR,R1 $-2
0483 17               79        RETC,UN
                      80   *
                      81   *
0700                  82        ORG     H'700'
                      83   ****************************
                      84   *PROCESSING OF ACTIVATED KEY*
                      85   ****************************
0700 1F0472   0472    86   EXEC BCTA,UN KREL
                      87   *
                      88        END

 TOTAL ASSEMBLY ERRORS   0
```

Fig. A3. Assembly listing of the software for keyboard system 2.

```
                        31  *JMB 760123-14.00
                        32  *
                        33  ***********************************************
                        34  ***********************************************
                        35  *KEYENCODING
                        36  ***********************************************
                        37  *THIS ROUTINE ENCODES 32 NORMALLY-OPEN SWITCHES,
                        38  *POSITIONED IN A 4 X 8 MATRIX TO A 5 BIT CODE IN THE
                        39  *5 LOW-ORDERBITS OF R1;
                        40  *THE ROUTINE STARTS AFTER RECEIVING AN INTERRUPT
                        41  *REQUEST SIGNAL FROM THE KEYBOARD-STROBE;
                        42  *THE MATRIX IS SCANNED AND SENSED BY USING READ
                        43  *EXTENDED I/O INSTRUCTIONS;THUS EACH SCANLINE WITH
                        44  *SWITCHES REPRESENTS AN EXTERNAL I/O DEVICE;
                        45  *THE MAIN PROGRAM IS CONTINUED AFTER DETECTING,
                        46  *DECODING AND STORING OF THE PRESSED KEY;
                        47  *REGISTER R3 IN BOTH RANKS IS ASSUMED TO BE FREE;
                        48  *MAX. KEYBOUNCEDELAY IS 11 MSEC.
                        49  ***********************************************
                        50  *
0477                    51       ORG    H'477'    START OF KEYBOARD INTERRUPT
                        52  *                     SERVICE ROUTINE
                        53  ***********************************************
                        54  *SAVING OF USED REGISTERS
0477 CDA600   0600      55       STRA,R0 LOC1      SAVE R0 IN LOC1
047A CDA601   0601      56       STRA,R1 LOC2      SAVE R1 IN LOC2
047D 13                 57       SPSL              (PSL) TO R0
047E C3                 58       STRZ   R3         (R0) TO R3
                        59  ***********************************************
                        60  *START OF KEYSCAN
047F 7508               61       CPSL   WC         INITIALIZE
0481 0505               62       LODI,R1 05        PRESET FOR BOUNCEDELAY
0483 20                 63       EORZ   R0
0484 F87E     0484      64       BDRR,R0 $         BOUNCEDELAY (11MSEC)
0486 F97C     0484      65       BDRR,R1 $-2
                        66  *
0488 0507               67       LODI,R1 H'A7'     SET R1 TO 7
048A 54FE               68       REDE,R0 H'FE'     ACTIVATE/SENSE SCAN 0 LINE
048C 5814     04A2      69       BRNR,R0 DEC0      BRANCH IF ACTIVE KEY FOUND
048E 54FD               70       REDE,R0 H'FD'     ACTIVATE/SENSE SCAN 1 LINE
0490 580E     04A0      71       BRNR,R0 DEC1      BRANCH IF ACTIVE KEY FOUND
0492 54FB               72       REDE,R0 H'FB'     ACTIVATE/SENSE SCAN 2 LINE
0494 5808     049E      73       BRNR,R0 DEC2      BRANCH IF ACTIVE KEY FOUND
0496 54F7               74       REDE,R0 H'F7'     ACTIVATE/SENSE SCAN 3 LINE
0498 5802     049C      75       BRNR,R0 DEC3      BRANCH IF ACTIVE KEY FOUND
049A 1BAF     04AB      76       BCTR,UN REST      BRANCH TO RESTORE
                        77  *
049C 8508               78  DEC3 ADDI,R1 H'08'     MODIFY BITS 4 AND 3
049E 8508               79  DEC2 ADDI,R1 H'08'     MODIFY BITS 4 AND 3
04A0 8508               80  DEC1 ADDI,R1 H'08'     MODIFY BITS 4 AND 3
                        81  *
04A2 60                 82  DEC0 IORZ   R0         DECREMENT R1 UNTILL A SINGLE
04A3 1A03     04A8      83       BCTR,N  STOR        ONE IS FOUND IN BIT 7 OF R0
04A5 D0                 84       RRL,R0
04A6 F97A     04A2      85       BDRR,R1 DEC0
                        86  *
04A8 CDA602   0602      87  STOR STRA,R1 LOC3      STORE (R1) IN LOC 3
                        88  ***********************************************
                        89  *RESTORING OF USED REGISTERS
04AB 03                 90  REST LODZ   R3         (R3) TO R0
04AC 93                 91       LPSL              (R0) TO PSL
04AD ACA600   0600      92       LODA,R0 LOC1      RESTORE R0; CC IS AFFECTED
04B0 0DA601   0601      93       LODA,R1 LOC2      RESTORE R1; CC IS AFFECTED
04B3 47C0               94       ANDI,R3 H'C0'     RECONSTRUCT CC IN PSL
                        95  *
04B5 B0                 96       WRTC,R0           RESET INTREQ FF
04B6 37                 97       RETE,UN           RETURN TO MAIN PROGRAM; ENABLE
                        98  *                      INTERRUPT
0600                    99       ORG H'600'
0600                   100  LOC1 RES 1
0601                   101  LOC2 RES 1
0602                   102  LOC3 RES 1
                       103       END

 TOTAL ASSEMBLY ERRORS   0
```

Fig. A4. Assembly listing of the software for keyboard system 3.

```
TWIN ASSEMBLER VER 2.1

 LOC OBJECT   ADDR E STMT   SOURCE LINE

                        31   *JMB 760212-10 30
                        32   *
                        33   ************************************************
                        34   ************************************************
                        35   **KEYENCODING
                        36   ************************************************
                        37   *THIS ROUTINE ENCODES 63 KEY-SWITCHES POSITIONED IN
                        38   *A 8 X 8 MATRIX INTO A 6 BIT CODE;
                        39   *EACH 100 MSEC THE MAIN PROGRAM IS INTERRUPTED BY
                        40   *THIS KEYBOARD-SERVICE ROUTINE;
                        41   *UP TO 3 KEYCODES CAN BE STORED; SO A MAXIMUM OF 3
                        42   *KEY-SWITCHES CAN BE DOWN AT A TIME;
                        43   *SUCCESSIVELY ENTERED KEYS WILL BE STORED IN RAM
                        44   *LOCATIONS LOC1, LOC2 AND LOC3;
                        45   *LOC1 HAS FIRST PRIORITY TO BE SERVED BY THE MAIN
                        46   *PROGRAM FOLLOWED BY LOC2 AND LOC3;
                        47   *REGISTERBANK31 IS USED DURING INTERRUPT;
                        48   *REG R3 BNK30 IS SUPPOSED TO BE FREE FOR SAVING PSL
                        49   ************************************************
                        50   *
                        51   *POWER ON INITIALIZE OF MAINPROGRAM
0450                    52        ORG H'450'
0450 7620               53        PPSU    II          INHIBIT INTERRUPT
0452 7510               54        CPSL    RS          SELECT MAINPROGRAM REG BANK(0)
0454 20                 55        EORZ    R0          CLEAR R0
0455 0706               56        LODI,R3 06          SET INDEX TO 06
0457 CF4600   0600      57   INIT STRA,R0 LOC1,R3,-   CLEAR LOC1---LOC6
045A 5B7B     0457      58        BRNR,R3 INIT
045C 7420               59        CPSU    II          ENABLE INTERRUPT
                        60   ************************************************
0477                    61        ORG     H'477'      START OF KEYBOARD INTERRUPT
                        62   *                          SERVICE ROUTINE
                        63   ************************************************
                        64   *SAVING OF USED REGISTERS
0477 CC0605   0605      65        STRA,R0 LOC6        SAVE R0 IN LOC6
047A 13                 66        SPSL                (PSL) TO R0
047B C3                 67        STRZ    R3          (R0) TO R3
047C 7710               68        PPSL    RS          SELECT INTERRUPT REG BANK (31)
                        69   ************************************************
                        70   *START OF KEYSCAN
047E 7508               71        CPSL    WC          OPERATIONS WITHOUT CARRY
0480 7702               72        PPSL    COM         LOGICAL COMPARE
0482 053F               73        LODI,R1 H'3F'       SET R1 TO THE MAX  KEYCODE
0484 0401               74        LODI,R0 01          SET LSB OF R0
0486 B0                 75   ACT1 WRTC,R0             ACTIVATE ONE ROWLINE
0487 32                 76        REDC,R2             READ COLUMNINFO INTO R2
                        77   *
0488 0703               78   TES  LODI,R3 03          SET INDEX TO 3
048A CC0604   0604      79        STRA,R0 LOC5        SAVE SCANREG  IN LOC5
048D 0F4600   0600      80   COMP LODA,R0 LOC1,R3,-   FETCH KEYCODEMEM LOCATION
0490 443F               81        ANDI,R0 H'3F'       CLEAR UPPER 2 BITS
0492 E1                 82        COMZ    R1          COMPARE LOC AND KEYCODEREG
0493 1807     049C      83        BCTR,EQ EQL         BRANCH IF EQUAL
0495 5B7A     048D      84   COM1 BRNR,R3 COMP
                        85   *
```

```
TWIN ASSEMBLER VER 2.1

 LOC OBJECT   ADDR E STMT   SOURCE LINE

0497 02                 86        LODZ    R2          SET CC CODE
0498 9A3A     04D4      87        BCFR,N  ROT         BRANCH IF NO ACTIVE KEY FOUND
049A 1B27     04C3      88        BCTR,UN BEMP        BRANCH IF NEW KEY FOUND
                        89   *
049C 02                 90   EQL  LODZ    R2          SET CC CODE
049D 1A18     04B7      91        BCTR,N  FLAT        BRANCH IF ACTIVE KEY FOUND
049F 0F6600   0600      92        LODA,R0 LOC1,R3     FETCH KEYCODEMEM LOCATION
04A2 9A0C     04B0      93        BCFR,N  FLAS        BRANCH IF KEY NOT EXECUTED
                        94   *
04A4 0F2600   0600      95   UPDA LODA,R0 LOC1,R3,+
04A7 CF65FF   05FF      96        STRA,R0 LOC1-1,R3 UPDATE KEYCODEMEMORYLOCATIONS
04AA F703               97        TMI,R3  03
04AC 9876     04A4      98        BCFR,A1 UPDA        BRANCH IF INDEX NOT EQ  3
04AE 1B24     04D4      99        BCTR,UN ROT
                       100   *
04B0 6440              101   FLAS IORI,R0 H'40'        SET BIT 6 OF R0
04B2 CF6600   0600     102        STRA,R0 LOC1,R3
04B5 1B5F     0495     103        BCTR,UN COM1
                       104   *
04B7 0F6600   0600     105   FLAT LODA,R0 LOC1,R3      FETCH KEYCODEMEM LOCATION
04BA F440              106        TMI,R0  H'40'
04BC 9816     04D4     107        BCFR,A1 ROT         BRANCH IF BIT 6 IS 0
04BE 44BF              108        ANDI,R0 H'BF'       CLEAR BIT 6 OF R0
04C0 CF6600   0600     109        STRA,R0 LOC1,R3
                       110   *
04C3 07FF              111   BEMP LODI,R3 H'FF'
04C5 0F2600   0600     112   LOOP LODA,R0 LOC1,R3,+
04C8 1806     04D0     113        BCTR,Z  STOR        BRANCH IF KEYBUFFER EMPTY
04CA F702              114        TMI,R3  02
04CC 9877     04C5     115        BCFR,A1 LOOP        BRANCH IF INDEX NOT EQ  2
04CE 1B04     04D4     116        BCTR,UN ROT         A FOURTH KEY HAS BEEN ENTERED
                       117   *                        AND CAN BE IGNORED
04D0 01                118   STOR LODZ    R1
04D1 CF6600   0600     119        STRA,R0 LOC1,R3     STORE KEYCODE IN RAMLOCATION
04D4 0C0604   0604     120   ROT  LODA,R0 LOC5        RESTORE SCANREG
04D7 D2                121        RRL,R2              ROTATE COLUMNINFO 1 X LEFT
04D8 F902     04DC     122        BDRR,R1 NEC
04DA 1B09     04E5     123        BCTR,UN REST        BRANCH IF ALL KEYPOSITIONS
                       124   *                        ARE SCANNED
04DC F507              125   NEC  TMI,R1  H'07'
04DE 9C0488   0488     126        BCFA,A1 TES         BRANCH IF COLUMNSCAN NOT READY
04E1 D0                127        RRL,R0              ROTATE SCANREGISTER 1 X LEFT
04E2 1F0486   0486     128        BCTA,UN ACT1
                       129   *
                       130   ************************************************
                       131   *RESTORING OF USED REGISTERS
04E5 7510              132   REST CPSL    RS          SELECT MAINPROGRAM REG BANK(30)
04E7 03                133        LODZ    R3          (R3) TO R0
04E8 93                134        LPSL                (R0) TO PSL
04E9 0C0605   0605     135        LODA,R0 LOC6        RESTORE R0; CC IS AFFECTED
04EC 470C              136        ANDI,R3 H'0C'       RECONSTRUCT CC
04EE 37                137        RETE,UN             RETURN TO MAIN PROGRAM;
                       138   *                        ENABLE INTERRUPT
                       139   *
0600                   140        ORG H'600'          DEFINITION OF RAMLOCATIONS
0600                   141   LOC1 RES 1               BUFFER FOR FIRST ENTERED KEY
0601                   142   LOC2 RES 1               BUFFER FOR SECOND ENTERED KEY
0602                   143   LOC3 RES 1               BUFFER FOR THIRD ENTERED KEY
0603                   144   LOC4 RES 1               USED FOR UPDATE (=0)
0604                   145   LOC5 RES 1               TEMPORARY SAVING OF R0
0605                   146   LOC6 RES 1               SAVING OF R0 BEFORE INTERR
                       147        END

 TOTAL ASSEMBLY ERRORS   0
```

Fig. A5. Assembly listing of the software for keyboard system 4.

Related 2650 publications

no.	title	summary
AS50	Serial Input/Output	Using the Sense/Flag capability of the 2650 for serial I/O interfaces.
AS51	Bit & Byte Testing Procedures	Several methods of testing the contents of the internal registers in the 2650.
AS52	General Delay Routines	Several time delay routines for the 2650, including formulas for calculating the delay time.
AS53	Binary Arithmetic Routines	Examples for processing binary arithmetic addition, subtraction, multiplication, and division with the 2650.
AS54	Conversion Routines	• Eight-bit unsigned binary to BCD • Sixteen-bit signed binary to BCD • Signed BCD to binary • Signed BCD to ASCII • ASCII to BCD • Hexadecimal to ASCII • ASCII to Hexadecimal
AS55	Fixed Point Decimal Arithmetic Routines	Methods of performing addition, subtraction, multiplication and division of BCD numbers with the 2650.
SP50	2650 Evaluation Printed Circuit Board (PC1001)	Detailed description of the PC1001, an evaluation and design tool for the 2650.
SP51	2650 Demo System	Detailed description of the Demo System, a hardware base for use with the 2650 CPU prototyping board (PC1001 or PC1500).
SP52	Support Software for use with the NCSS Timesharing System	Step-by-step procedures for generating, editing, assembling, punching, and simulating Signetics 2650 programs using the NCSS timesharing service.
SP53	Simulator, Version 1.2	Features and characteristics of version 1.2 of the 2650 simulator.
SP54	Support Software for use with the General Electric Mark III Timesharing System	Step-by-step procedures for generating, editing, assembling, simulating, and punching Signetics 2650 programs using General Electric's Mark III timesharing system.
SP55	The ABC 1500 Adaptable Board Computer	Describes the components and applications of the ABC 1500 system development card.
SS50	PIPBUG	Detailed description of PIPBUG, a monitor program designed for use with the 2650.
SS51	Absolute Object Format	Describes the absolute object code format for the 2650.
MP51	Initialization	Procedures for initializing the 2650 microprocessor, memory, and I/O devices to their required initial states.
MP52	Low-Cost Clock Generator Circuits	Several clock generator circuits, based on 7400 series TTL, that may be used with the 2650. They include RC, LC and crystal oscillator types.
MP53	Address and Data Bus Interfacing Techniques	Examples of interfacing the 2650 address and data busses with ROMs and RAMs, such as the 2608, 2606 and 2602.
MP54	2650 Input/Output Structures and Interfaces	Examines the use of the 2650's versatile set of I/O instructions and the interface between the 2650 and I/O ports. A number of application examples for both serial and parallel I/O are given.
TN 064	Digital cassette interface for a 2650 microprocessor system	Interface hardware and software for the Philips DCR digital cassette drive.

Printed in The Netherlands Date of release: 2 January 1978 T.D. 9399 500 06901