ITHACA InterSystems IA-3080 Real Time Clock

Neil B. Breeden II - nbreeden2@comcast.net
John King

Nov 2009 – Manual Version 1.2

Contents

About this manual:

This manual was created by reverse engineering the IA-3080. John King supplied me two unassembled IA-3080 boards along with one that was already assembled. With my trusty ohm meter and the TTL data book in hand I went over these boards to capture schematics and from those to interpret how the board's designers envisioned it working.

This manual is provided as is; it is NOT guaranteed to be correct. All risk in the use of the manual is yours. I have made my best effort to validate the contents including assembling an IA-3080 using the steps in this manual.

In today's world it is not common to find unassembled S-100 kits, as I encounter these kits I do my best to buy them. Often I will buy two kits if I can; one to build, one to keep unassembled. If you know of sources for bare S-100 boards or kits let me know. –Neil   nbreeden2@comcast.net

Conventions used in this manual:

/<name> - indicates the signal is active low

Hexadecimal numbers are represented as 00_{16} or 20_{16} or 40_{16}

Introduction:

The IA-3080 incorporates a MM58167N Clock chip into a battery backed up S-100 based Real Time Clock.

The clock circuit is crystal controlled and can be trimmed (C12 trimmer capacitor) meaning it should be able to be trimmed to within a few minutes a year or better.

The clock chip is backed up using a battery. The time will be maintained with the system power off for as long as the battery remains charged.

I have been running my assembled board for over a month with the system both running and turned off. Once the clock was trimmed it has kept excellent time.

The board can produce an interrupt to the CPU; the interrupt channel is switchable to any of the eight S-100 Vector Interrupts or can be disabled. The board can provide different kinds of interrupts including a 10hz interrupt, a 1hz interrupt etc.

The board uses an on-board 7805 regulator to supply power.

The board provided an optional power fail detection circuit or can use the S-100 /PWRFAIL signal.

The circuit presents itself to the host computer as 23 consecutive I/O ports. All I/O to and from the clock chip is through these ports. The board actually occupies 32 consecutive ports due to how it decodes the port addresses. Its base port will always be on a 32 port boundary, IE: port 00_{16} or 20_{16} or 40_{16} etc.

The board also has the ability to work as termination for the S-100 bus however this part of the board is not covered in detail in this manual.

Hardware identification:

Item	Device	Basic RTC	Power Fail Detect	2Mhz Clock	Extended Address Mode	Notes
R1	22K Ohm ¼ Watt		X			See Misc.
R2	1.1K Ohm ¼ Watt		X			See Misc.
R3	2.2K Ohm ¼ Watt		X			See Misc.
Q1	7805 1 amp Regulator	X				
Q2	2N3904		X			See Misc. – Can sub 2N2222
Q3	2N3904		X			See Misc. – Can sub 2N2222
D1	1N4148	X				
D2	1N4148	X				
D3	1N5234 6.2V Zener		X			See Misc.
UR1	4.7K Ohm Bussed 9-Pin	X				Can sub with 4.7-10.0K
UR2	1.1K Ohm Bussed 9-Oin				X	
Y1	32.768Khz Crystal	X				
Y2	<Unknown>			X		See Misc.
L1	<Unknown>			X		See Misc.
B1	3.0V Backup Battery	X				
S1	8-Pos DIP Switch	X				
S2	8-Pos DIP Switch	X				
S3	8-Pos DIP Switch				X	See Misc.
J1	3-Pos header and jumper	X				
J2	2-Pos header and jumper			X		See Misc.
J3	3-Pos header and jumper	X				
HS1	Heat sink for Q1	X				Heat sink is not really required.
SC1	Screw to attach Q1/HS1	X				Secure the regulator to the -
NT1	Nut to attach Q1/HS1	X				board with the screw and nut.

Hardware identification continued:

Item	Device	Basic RTC	Power Fail Detect	2Mhz Clock	Extended Address Mode	Notes
C1	.1uF Disc (104)	X				
C2	.1uF Disc (104)	X				
C3	.1uF Disc (104)	X				
C4	.1uF Disc (104)	X				
C5	.1uF Disc (104)	X?			X	
C6	.1uF Disc (104)	X				(Not needed for the BASIC RTC)
C7	.1uF Disc (104)	X				
C8	.1uF Disc (104)	X				
C9	.1uF Disc (104)	X				
C10	.1uF Disc (104)	X				
C11	20pF Disc	X				Can substitute with 10-20pF Disc
C12	6pF - 36pF trimmer	X				Can substitute with 10-20pF Disc
C13	.1uF Disc (104)	X?		X		(Not needed for the BASIC RTC)
C14	33uF Tantalum 20V	X				Can substitute 33-100uF Tantalum
C15	.1uF Disc (104)	X				
C16	.1uF Disc (104)	X				
U1	74LS74	X				
U2	7407	X				
U3	74L04	X				
U4	74LS244	X				
U5	74LS85	X				
U6	AM25LS251PC				X	See Misc.
U7	74LS244	X				
U8	74LS244	X				
U9	MM58167N	X				Can also use MM58167AN
U10	74S32	X				
U11	<Unknown>			X		See Misc.

How my build diverged from this documentation:

I used a CR2032 lithium battery and holder. I enlarged the lead hole for B1 in the ground plain area to accommodate the holder's negative lead. I drilled a new hole in the blank portion of the board above B1's upper two holes for the holders positive lead then used a jumper wire to connect the positive lead to B1's positive connection pad. The positive hole I drilled nearly hit a large trace on the back of the board; I should have drilled new holes about .125 inches lower on the board.

I had a trimmer cap (C12) supplied by John King that I managed to break. Its leads were larger than the holes in the PCB. In trying to force it in I managed to break off a lead. I ended up picking up a physically larger 6-36pF trimmer and drilling 3 holes in blank areas of the board to mount it. I was able to bend one lead over onto the trace/pad on the board; a jumper wire was used to connect the other pin to ground.

With a 22pF disc at C11 I was unable to trim the correct frequency, the clock always ran slow. I replaced C11 with a 10pF disc and can now trim from too slow to too fast. I was unable to find a 20pF in my inventory so I don't know if lowering to 20pF would have been enough.

I installed test points for Vss and Vcc to allow easier connection of my logic probe and scope to the board; this is a common practice of mine.

Initially I was unable to get the board to accept port writes. In the end this was due to U10. I originally used a 74LS32, after replacing it with a 74S32 the board worked fine. I have not determined if my original 74LS32 was defective or if the 'S' part is actually required?

Assembling the Basic Real Time Clock:

It is highly recommended that you use sockets for all ICs, this will make troubleshooting and repair of the board much easier.

Battery:

The battery is used to maintain the time in the clock chip with the computer power off. I have not been able to find a source for the original battery the board was designed for. The PCB can be retrofitted for a CR series battery holder and battery (the CR2032 would be a good choice). An alternative could be two AAA or AA batteries in a battery holder using extension leads to reach from the battery holder to the PCB.

It is highly suggest that the battery retrofit be completed before any other assembly work occurs.

Sockets:

1. Install 14-pin IC socket(s) observing polarity, the notch (pin 1) is towards the top of the board
    a. [ ] U1
    b. [ ] U2
    c. [ ] U3
    d. [ ] U10
2. Install 16-pin IC socket(s) observing polarity, the notch (pin 1) is towards the top of the board
    a. [ ] U5
3. Install 20 pin socket(s) observing polarity, the notch (pin 1) is towards the top of the board
    a. [ ] U4
    b. [ ] U7
    c. [ ] U8
4. Install 24-pin socket(s) observing polarity, the notch (pin 1) is towards the top of the board
    a. [ ] U9

Note: U6 and U11 will not need a socket for the Basic RTC.

DIP Switches:

5. Install 8 position DIP switch(es)
    a. [ ] S1 - only needed if you plan to use interrupts
    b. [ ] S2

Capacitors:

6. Install 0.1uF disc (typically marked 104) cap(s)
   a. [ ] C1
   b. [ ] C2
   c. [ ] C3
   d. [ ] C4
   e. [ ] C5
   f. [ ] C6 – Not strictly needed if U6 is not installed
   g. [ ] C7
   h. [ ] C8
   i. [ ] C9
   j. [ ] C10
   k. [ ] C13 – Not strictly needed as U11 is not used
   l. [ ] C15
   m. [ ] C16
7. Install 20pF disc cap(s)
   a. [ ] C11
8. Install 6-36pF trimmer cap(s)
   a. [ ] C12
9. Install 33uF 20V Tantalum cap(s)
   a. [ ] C14 – This is a polarized part, observe the '+' symbol on the cap – The '+' lead is installed away from the edge of the board; to the right with the top of the board up.

Resistors:

10. Install 4.7K Bussed 9-pin resistor
    a. [ ] UR1 – This is a polarized part, observe PIN 1 on the part; PIN 1 is towards the top of the board.

Voltage Regulator:

11. Install 7805 regulator (LM340T-5)
    a. [ ] Q1 – This is a polarized part – It should be installed with a heat sink
       i. See the photo in this manual for installation details

Headers/Jumpers:

You can install PIN headers / jumpers to make reconfiguration easier or you can simply use a short section of wire as the jumper.

12. Install 3-Pin Headers/Jumpers OR wire jumper
    a. [ ] J3 – Jumper B-C for default (see board layout diagram)
    b. [ ] J1 – Jumper A-B for default (see board layout diagram)

Diodes:

13. Install diode(s)
    a. [ ] D1 – This is a polarized part, observe the strip on the diode
    b. [ ] D2 – This is a polarized part, observe the strip on the diode

Crystal:

14. Install 32.768Khz crystal(s)
    a. [ ] Y1

Initial checkout:

Do not install the ICs at this time:

1. Inspect the board closely under a bright light. You are looking for solder bridges between adjacent solder joints. You are looking for missed or cold solder joints. Cold solder joints are typically gray and dull instead of silver and bright; they can also appear as a ball of solder.
2. Test the 5VDC regulator.
   a. Install the board in the computer, preferably on an extender card.
   b. Carefully power up the computer, it should act as if the card is not installed. If the computer acts oddly power it down and look at the troubleshooting tips.
   c. Using a quality multi-meter check for +5VDC (4.8VDC to 5.2VDC is an acceptable range). Make sure the negative lead of the meter is making good contact to ground; this could be the heat sink on the voltage regulator or the GND pin on the IC sockets. Measure the following locations for +5VDC; you should get the same reading for each test:
      i. [ ] U1 pin 14 – Ground on pin 7
      ii. [ ] U2 pin 14 – Ground on pin 7
      iii. [ ] U3 pin 14 – Ground on pin 7
      iv. [ ] U4 pin 20 – Ground on pin 10
      v. [ ] U5 pin 16 – Ground on pin 8
      vi. [ ] U7 pin 20 – Ground on pin 10
      vii. [ ] U8 pin 20 – Ground on pin 10
      viii. [ ] U9 pin 24 – Ground on pin 12 – This may measure a bit low, this is OK
      ix. [ ] U10 pin 14 – Ground on pin 7
   d. Do NOT proceed until the +5VDC test passes.

Install the ICs:

3. With the computer powered down remove the RTC board.
4. Install the following ICs observing polarity, pin 1 is towards the top of the board:
   a. [ ] U1 pin 14 – 74LS74
   b. [ ] U2 pin 14 – 7407
   c. [ ] U3 pin 14 – 74L04
   d. [ ] U4 pin 20 – 74LS244
   e. [ ] U5 pin 16 – 74LS85
   f. [ ] U7 pin 20 – 74LS244
   g. [ ] U8 pin 20 – 74LS244
   h. [ ] U9 pin 24 – MM58167N
   i. [ ] U10 pin 14 – 74S32

5. Using a bright light inspect the chips in the sockets. Make sure PIN 1 is towards the top of the board and that there are no pins bent under a chip or outside the socket.
6. Configure the board:
    a. The board must be properly configured to work and to allow your computer to operate. The 32 PORT block the board is configured to occupy must NOT overlap the port setting on any other board in your system.

SW2-8 (A7)	SW2-7 (A6)	SW2-5 (A5)	Ports (HEX)	Ports (Decimal)
CLOSED	CLOSED	CLOSED	00_{16}-$1F_{16}$	0-31
CLOSED	CLOSED	OPEN	20_{16}-$3F_{16}$	32-63
CLOSED	OPEN	CLOSED	40_{16}-$5F_{16}$	64-95
CLOSED	OPEN	OPEN	60_{16}-$7F_{16}$	96-127
OPEN	CLOSED	CLOSED	80_{16}-$9F_{16}$	128-159
OPEN	CLOSED	OPEN	$A0_{16}$-BF_{16}	160-191
OPEN	OPEN	CLOSED	$C0_{16}$-DF_{16}	192-223
OPEN	OPEN	OPEN	$E0_{16}$-FF_{16}	224-255

    b. SW2-1 thru SW2-4 OPEN (these are unused).
    c. SW2-5 enables the extended address mode (when closed) – for the basic RTC leave it open.
    d. For the initial checkout it is advised to leave SW1-1 thru SW1-8 open. This will disable any interrupts generated by the RTC from making to the CPU.
    e. J1 – Jumper A-B (with the board top UP the LEFT two pins are connected)
    f. J2 – Not required – leave open
    g. J3 – Jumper B-C (with the board top UP the RIGHT two pins are connected)
7. Install the board in the computer and power it up. If you notice odd behavior power it down and consult the troubleshooting section.
    a. Check the +5VDC power bus again to make sure none of the chips is causing the regulator to shut down.
        i. If you check U9-24 it may appear lower by as much as 0.8V as compared to the other +5VDC test points. This is due to voltage drop across D2 which is part of the battery backup circuit and is expected.
    b. If you are not seeing +5VDC consult the troubleshooting section.

Troubleshooting:

Normal system operation is blocked:

There are many things this could be, many of which have already been mentioned in this manual.

- Remove all ICs, test the power supply, check for solder bridges

System hangs with all ICs removed:
- Check for solder bridges

- Potential conflict with your version of the S-100 bus

System hangs with ICs installed but Ok with ICs removed:
- Port overlap – the block of 32 I/O ports the board is using conflict with another board, try using different settings on S2-6,7 and 8 to select different blocks.

- Bad IC(s) – 74LS244s highly suspect

- Remove one 74LS244 at a time, if the removal of one allows the system to run this is a big clue, use the schematics to trace back to the chips that control the specific 74LS244

- Interrupts are enabled when the OS is not setup to use them – make sure all positions of S1 are 'OPEN' – use an ohm meter top confirm this.

More general notes, the board is not acting right but overall system runs:

While running 'READRTC.BAS' use a logic probe to check pins 1 and 2 on the clock chip.

If you are unable to read the time AND you don't see activity on pin 1 (/CS) and pin 2 (/RD) when reading it:

- Pin 1 (/CS) issue - Software is using different ports then the board is configured for, bad IC in the control logic for /CS

- Pin 2 (/RD) issue – J1 not installed, U3 defective.

Modify 'SETRTC.BAS'adding line "`191 GOTO 190`"

With it running and looping on line 190 (setting the seconds) use a logic probe to check pins 1 and 3 on the clock chip.

If you are unable to write the time AND you don't see activity on pin 1 (/CS) and pin 3 (/WR) when setting it:

- Pin 1 (/CS) issue - Software is using different ports then the board is configured for, bad IC in the control logic for /CS

- Pin 3 (/WR) issue – U10, U1 or U3 defective.

Example MBASIC programs:

SETRTC.BAS

```
5   REM
10 REM Get the time from the user and set the RTC
15 REM
20 REM Base port address is 192 - edit line 30 if different
25 REM
30 BASE=192
40 REM
50 REM Prompt the user, get the time and set it
60 REM
70 PRINT "Enter the time HHMMSS ";
80 INPUT T$
90 IF LEN(T$) <> 6 THEN GOTO 70
100 OUT BASE+18,255:  REM Reset clock
110 UH=(ASC(MID$(T$,1,1))-48)*16
120 LH=(ASC(MID$(T$,2,1))-48)
130 OUT BASE+4,UH+LH:    REM Set Hours
140 UM=(ASC(MID$(T$,3,1))-48)*16
150 LM=(ASC(MID$(T$,4,1))-48)
160 OUT BASE+3,UM+LM:   REM Set Minutes
170 US=(ASC(MID$(T$,5,1))-48)*16
180 LS=(ASC(MID$(T$,6,1))-48)
190 OUT BASE+2,US+LS:    REM Set Seconds
200 REM
210 REM Now loop and read the time displaying it
220 REM
230 SEC=INP(BASE+2): REM Read Seconds
240 MIN=INP(BASE+3): REM Read Minutes
250 HOUR=INP(BASE+4):REM Read Hours
260 T$="": REM Build up t$ with the current time
270 T$=T$+CHR$((HOUR/16)+48)
280 T$=T$+CHR$((HOUR AND 15)+48)
290 T$=T$+":"
300 T$=T$+CHR$((MIN/16)+48)
310 T$=T$+CHR$((MIN AND 15)+48)
320 T$=T$+":"
330 T$=T$+CHR$((SEC/16)+48)
340 T$=T$+CHR$((SEC AND 15)+48)
350 PRINT T$;CHR$(13);
360 GOTO 230
```

READRTC.BAS

```
10 REM Example program to read the time and display it
20 BASE=192: REM Base port address of 192 - edit for other configurations
30 SEC=INP(BASE+2): REM Read Seconds
40 MIN=INP(BASE+3): REM Read Minutes
50 HOUR=INP(BASE+4):REM Read Hours
60 T$="": REM Build up T$ with the current time
65 REM CONVERT BCD ENCODED TIME TO ASCII CHARACTERS
70 T$=T$+CHR$((HOUR/16)+48)
80 T$=T$+CHR$((HOUR AND 15)+48)
90 T$=T$+":"
100 T$=T$+CHR$((MIN/16)+48)
110 T$=T$+CHR$((MIN AND 15)+48)
120 T$=T$+":"
130 T$=T$+CHR$((SEC/16)+48)
140 T$=T$+CHR$((SEC AND 15)+48)
150 PRINT T$;CHR$(13);
160 GOTO 30
```

Example Assembly Language Programs:

Intersystems RTC routines:

```
; Intersystems RTC routines
; Set clock to 0
; Set clock to time
; BCD/Binary conversion
; Send time to display on VDM

;convert 3 BCD digits to 3 binary digits
;the address of the BCD digits arrives in HL
; and the address for the binary digits arrives in DE

tobin3:
      mov  b,m            ;get 1st BCD digit
      call bcdbin ;convert to binary
      stax d             ;store converted
      inx  d             ;set up next digit
      inx  h
      mov  b,m            ;get next BCD digit
      call bcdbin ;convert to binary
      stax d             ;store converted
      inx  d
      inx  h
      mov  b,m            ;get next BCD digit
      call bcdbin ;convert to binary
      stax d             ;store converted
      ret
bcdbin:
      mov  a,b           ;BCD arrives in b
      ani  0fh           ;strip upper digit
      mov  c,a           ;save lower digit in c
      mov  a,b           ;get BCD
      ani  0f0h          ;shift upper digit
      rrc                ;X4
      rrc
      rrc
      rrc
      inr a
      mov b,a
      mov a,c
tmore:
      dcr b
      rz
      adi 10
      jmp tmore
sett0:                  ;Set clock to 0\
      mvi  a,0
      out  rtchr
      out  rtcmin
      out  rtcsec
      out  rtchun
      mvi  a,01h
```

```
        sta  statflag     ;signal time has been set
        ret
setrtc:
;       di                ;disable interrupts if needed
        lxi  h,bntcadr    ;get address
        mov  a,m
        out  rtcmin ;set minutes
        inx  h
        mov  a,m
        out  rtcsec ;set seconds
        inx  h
        mov  a,m
        rrc               ;shift tenths to high nibble
        rrc               ;rtc has tenths in high nibble
        rrc
        rrc
        ani 0f0h          ;zero out low nibble
        out rtchun  ;set hundreths
;       ei
        ret

bxbcd3:                   ;binary to BCD converter
        mov  b,m
        call bxbcd
        stax d
        inx  d
        inx  h
        mov  b,m
        call bxbcd
        stax d
        inx  h
        inx  d
        mov  b,m
        call bxbcd
        stax d
        ret
bxbcd:            ;binary arrives in b, BCD leaves in A
        inr b
        mvi c,0
        mvi a,0
xbcd1:
        dcr b
        jz  xbcdone
        inr a
        cpi 0ah
        jnz xbcd1
        mov a,c
        adi 10h
        mov c,a
        mvi a,0
        jmp xbcd1
xbcdone:
        add c
        ret
;******************** END OF SYNC SECTION********************
```

```
;time display on vdm screen


        DSEG
SHOWRETURN: DS 1
chklflg:    ds 1
        CSEG

timeloop::
        call dotime
showtime::
        in   rtcsec
        mov  b,a
        mvi  h,2                ;row (change to desired row)

        mvi  l,55               ;column (change to desired column
        shld conadr##
        call showcon##
        in   rtcmin
        mov  b,a
        mvi  h,2                ;row (change to desired row)
        mvi  l,52               ;column (change to desired column)
        shld conadr##
        call showcon##
        ret
showhun::
        in   rtchun
        mov  b,a
        mvi  h,2                ;row
        mvi  l,58               ;column
        shld conadr##
        jmp showcon##
showtnth:
        ret
during::
        call showtnth
        ret

        end
```

Sound Farm Memory Mapped screen and CP/M console driver:

```
;Sound Farm Memory Mapped screen and CP/M console driver
;1/13/90  4.1 version (cleans things up)
;initscrn writes blanks (20H) to the memory mapped video screen
;scrnadr  is the character out screen address
;get2     calls the console for two hex digits; its binary
;         equivilent leaves in A
;show2    converts a binary number in B to 2 ascii hex digits
;         and displays them on the vdm at the scrnadr
;show4    converts two binary numbers in BC to 4 ascii hex
;         digits and displays them on the vdm at the scrnadr
;showbin  converts a binary number to 8 ascii bits and displays
;         them at (hl). Saves hl at scrnadr
;const        returns with FF in A if character is available
;conin    returns with char in A
;conot    sends char in A to console
;pcmsg    prints the to the consolestring starting at the next
;         address after the call and ending with a $
;pvmsg    prints the string as above but to the MM video
;         starting at the bottom line after clearing the line
;clrmsg   clears the bottom line of the MM video
;vconot   places the character in A at the scrnadr
      title screen
      MACLIB Z80.LIB
      MACLIB V50.LIB
diag  equ 0
screen  equ 0f800h
scrend  equ 0fch
cmdline equ screen+03c0h
cstat   equ  01h
cdata   equ  00h
ckbrdy  equ  01h

      DSEG
chrtwo:: ds 2
scrnadr::ds 2
conadr::ds 2
vdmadr::ds 2

      CSEG
INITSCRN::
      LXI H,screen
      MVI  B,020H ;clear screen
      MVI  A,0FCH
CLR1: MOV  M,B
      INX  H
      CMP  H
      JNZ  CLR1
      lxi h,screen
      shld  scrnadr
      if    diag
      lxi  h,0f000h     ;clear vio screen
```

```
            lxi   b,0f7f0h
            lxi   d,0f001h
            mvi a,020h
            mov m,a
            ldir
            endif
            ret
const::
            in   01h
            ani 01
            cpi 0
            rz
            mvi a,0ffh
            ret
conin::
conin1:      in 01h
            ani 01
            cpi 0
            jz   conin1
            in    0
            ret
conot::
            ret
            push psw
co1:  in    01h
            ani   04h
            jz    co1
            pop   psw
            out   00h
            ret
vconot::             ;print message starting at scrnadr
            lhld scrnadr
            mov m,a
            inx h
            shld scrnadr
            ret
pcmsg::                   ;print message using conadr for
            call vdm2  ;   screen address
pcmsg1:      ldax d
            cpi   '$'
            rz
            lhld vdmadr
            mov  m,a
            inx  h
            shld vdmadr
            inx  d
            jmp pcmsg1
pvmsg::                    ;print message at bottom of video screen
            push d
            lxi h,cmdline
            lxi d,cmdline+1
            lxi b,32
            mvi a,020h
            mov m,a
            ldir
```

```
              lxi  h,cmdline
              pop d
pvmsg3:
              ldax d
              cpi  '$'
              rz
              mov  m,a
              inx h
              inx d
              jmp pvmsg3
clrmsg::          ;clear bottom line of video screen
              lxi h,cmdline
              lxi d,cmdline+1
              lxi b,03fh
              mvi a,020h
              mov m,a
              ldir
              ret
vdm2:             ;change row and column (conadr) to vdm
              push h         ;address (vdmadr)
              push b
              push d         ;save 2 ascii chrs from bxash
              lxi  h,screen    ;load start of vdm screen
              lxi  b,0040h     ;load one line
              lded conadr ;get row and column
              inr d
vdm21:     dcr d
              jz vdm22
              dad b       ;add one line
              jmp vdm21
vdm22:  mov a,e
              add l
              mov l,a
              shld vdmadr
              shld scrnadr
              pop d
              pop b
              pop h
              ret
vdm23:
              mov m,d
              inx h
              mov m,e
              ret
SHOW4::
              CALL BXASH
              lhld scrnadr
              mov  m,d
              inx  h
              mov  m,e
              inx  h
              MOV  B,C
              CALL BXASH
              mov  m,d
              inx  h
```

```
        mov  m,e
        inx  h
        shld scrnadr
        RET
SHOW2::
        CALL BXASH
        lhld scrnadr
        mov  m,d
        inx  h
        mov  m,e
        inx  h
        shld scrnadr
        ret

;CONVERT 8 BIT BINARY TO ASCII HEXADECIMAL FOR VIDEO DISPLAY
BXASH::    MVI  A,0F0H ;MASK
        ANA  B              ;GET FIRST CHARACTER
        RRC
        RRC
        RRC
        RRC
        ADI  030H
        CPI  03AH   ;SEE IF ITS OVER ASCII 9
        JM   J1              ;CHAR IS IN A
        ADI  7H
J1:     MOV  D,A    ;D = ASCII OF HIGHER 8 BITS
        MVI  A,0FH
        ANA  B
        ADI  030H
        CPI  03AH   ;SEE IF ITS OVER ASCII 9
        JM   J2             ;CHAR IS IN A
        ADI  7H
J2:     MOV  E,A    ;E = ASCII OF LOWER 4 BITS
        RET

GET2::     CALL CONIN  ;get 2 ascii hex chars from console,convert
        STA  CHRTWO ;and convert them to binary
        CALL CONIN
        STA  CHRTWO+1
ASXHEX:            ;two ascii chars are in
        LDA  CHRTWO ;chrtwo ,hex equivilent
        SBI  030H
        CPI  010H
        jm   ash1
        SBI  07H
ash1: SLAR A
        SLAR A
        SLAR A
        SLAR A
        MOV  C,A
        LDA  CHRTWO+1
        SBI  030H
        CPI  010H
        jm   ash2
        SBI  07H
```

```
ash2: ANI 00FH
      ORA C
      RET
showbin::
;     shld scrnadr
      push b
      push d
      mvi  d,030h ;ascii zero
      mvi  e,031h ;ascii one
      mvi  c,9    ;shift counter
      mov a,b            ;get byte to convert
bin1: dcr c       ;reduce counter
      jz  bindone ;done after 8 loops
      ral         ;shift next bit to carry
      jc itsone
      mov m,d
      inx h
      jmp bin1
itsone: mov m,e
      inx h
      jmp bin1
bindone:shld scrnadr
      pop d
      pop b
      ret

;************* RS232 TERMINAL CONSOLE DRIVER *****************
showcon::
;     push h
;     push d
;     push b
      call bxash  ;B has char to convert
      call vdm2
      lhld vdmadr
      inx  h
      mov  a,m
;     cmp  e
;     jz   sc4
      mov  m,e
sc4:  dcx  h
      mov  a,m
      cmp  d
      jz   sc2
      mov  m,d
      call sc1
sc2:
      lded conadr
      inx  d
      inx  d
      sded conadr
;     pop b
;     pop d
;     pop h
      ret
sc1:
```

```
        push d
        mvi  b,setfor
        call terdvr
        lded conadr
        mvi  b,curadr
        call terdvr
        pop  d
        mov  a,d
        call conot
        mov  a,e
        call conot
        mvi  b,setbak
        call terdvr
        ret

terdvr::
        mvi a,leadin
        call conot
        mov  a,b
        cpi  curadr
        jz   teradr
        call conot
        ret
teradr::                    ;on entry, d contains row, e contains column
        call conot
        mov  a,d    ;get row
        adi  020h
        call conot
        mov  a,e    ;get column
        adi  020h
        call conot
        ret
terclr::
        mvi b,clscr
        call terdvr
        ret
tereol::
        mvi b,cleol
        call terdvr
        ret
tertab::
        mvi b,tab
        call terdvr
        ret

        END
```

Misc:

Extended Address Mode:

This looks to be unique to IA boards. Basically the board is designed to be able to decode ports using a 16-bit port address, a normal 8080 or Z-80 based system provides 8-bit port addressing. The extended mode can be disabled by leaving SW2-5 open.

The original 8080 design did something interesting in that for port I/O the port address on A0-A7 was repeated on A8-A15. Many later Z-80 based designs could 'emulate' this mode as some older S-100 cards actually looked for the port address on A8-A15.

My guess (and this is totally a guess) is that an IA CPU card could set the bits to be presented on A8-A15, possibly by writing to a single port defined on A0-A7. Suppose that A0-A7 being 0's (00_{16}) is defined to ALWAYS be the port block setting, there could then be 256 port blocks. Within each port block then there would be 255 available ports ($01_{16}..FF_{16}$).

This would increase the total ports available to the system to 65280. We lose 256 ports as any address with all zeros for A0-A7 is used to define the block to talk too.

 Based on my limited knowledge of S-100 systems it appears that this 'extended address mode' has no practical use on standard S-100 systems.

Power Fail Detect:

This looks to be a simply circuit to detect that the power is failing, this in turn allows the /POWERDOWN pin on the clock chip to go low, putting the clock chip into 'battery mode'. As the +5VDC rail shuts down the diodes D1 and D2 will allow the battery to power the clock chip. As the S-100 bus defines a /PWRDN signal this circuit appears to not be needed for the typical S-100 system.

2MHz clock:

This is TOTAL guess work and has not been tried. U11, Y2 and L1 look like they could be setup to create a 2MHz clock signal for the S-100 bus using an i8224. Some Z-80 based CPU cards running a 4MHz (or faster) clock this S-100 signal at the CPU clock speed. Other boards that rely on this clock (such as UART and Floppy Controller) will not work if this signal is not at 2MHz.

U11 and Y2 could be setup to deliver a standard 2MHz clock with a bit of work on the board.

Note: I am NOT 100% convinced U11 uses an i8224, this is also a guess. Although the board is close to being correct for an 8224 there are issues: Phase1 (Pin 11) is connected to VCC. VDD (Pin 10) is floating. For an 18MHz crystal there should be a small cap (10pf) in series with the crystal. The coil (L1) would be driven by the RESET output which makes no sense.

Termination:

The silk screen for the termination on the board looks very similar to the silk screen on the IA-1718-A motherboard suggesting it can be stuffed with the same value components as are found on a terminated IA-1718-A motherboard.

Note that if you elect to use this termination a heat sink should be used on Q1 as it will be supplying VCC to the termination array.

I have not validated this works, use at your own risk.
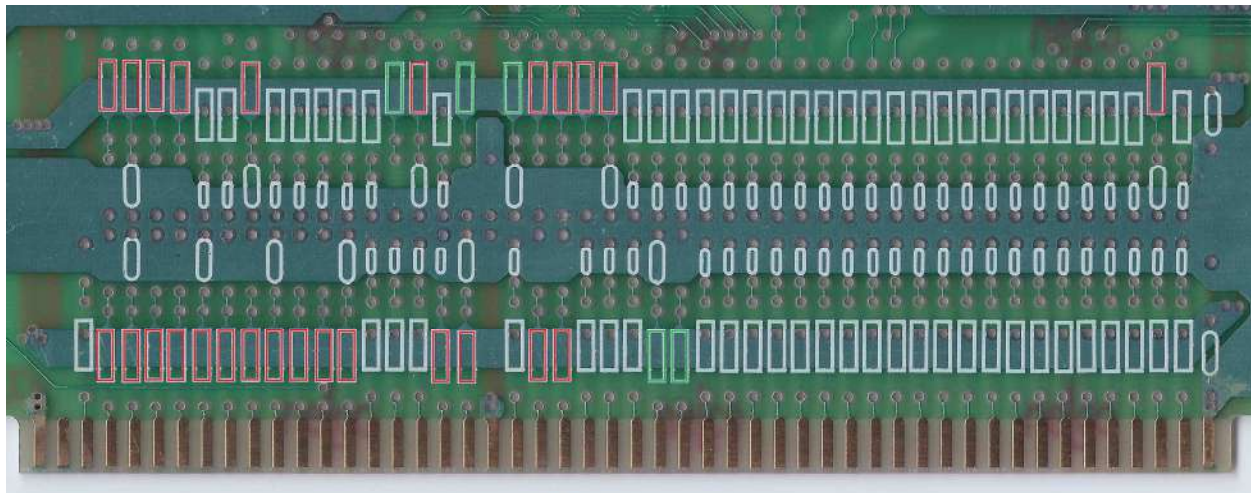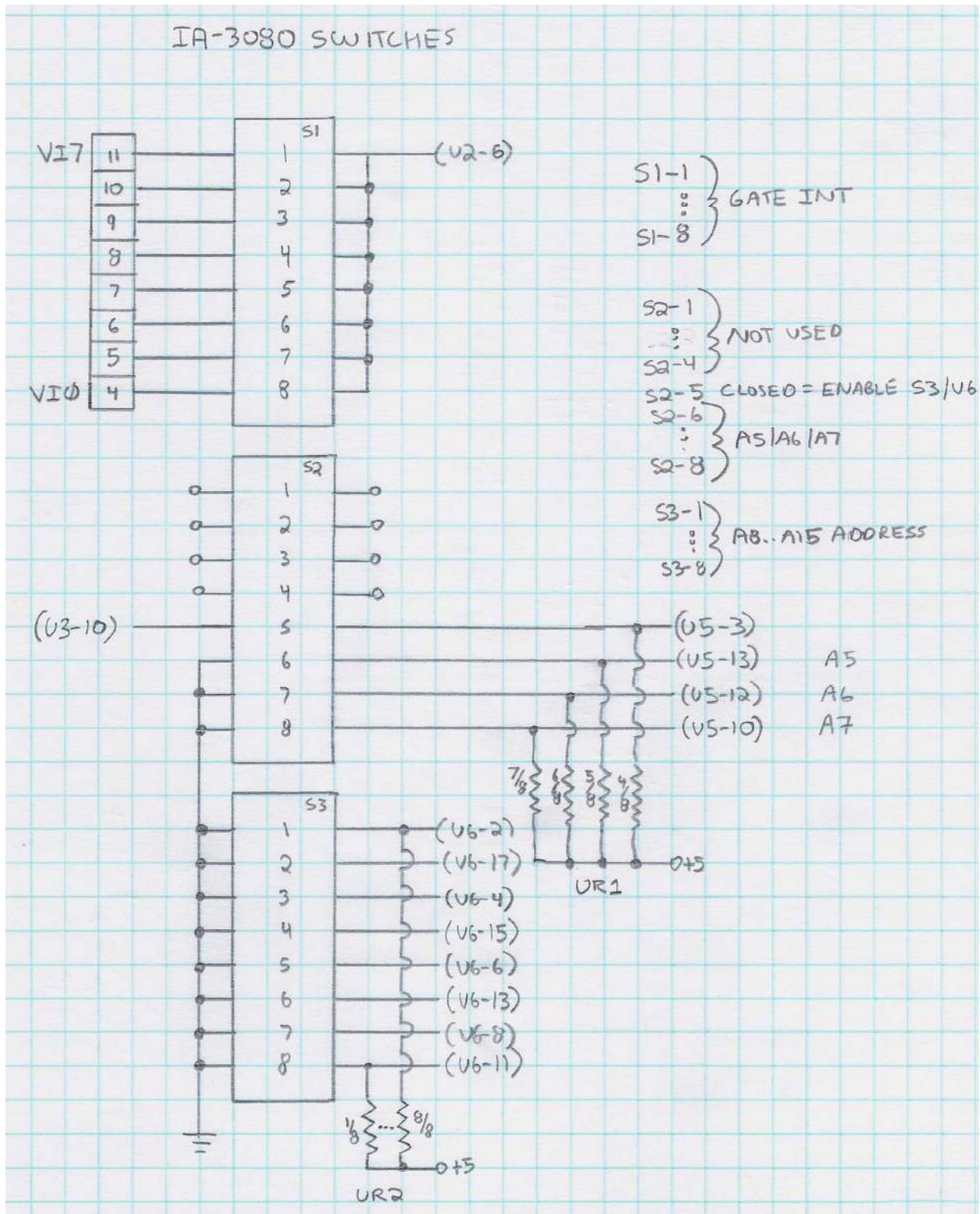

180 Ohm

470 Ohm

220 Ohm

0.01uF Disc (103)

56pF Disc Cap

Schematics:

Interrupt Circuit:
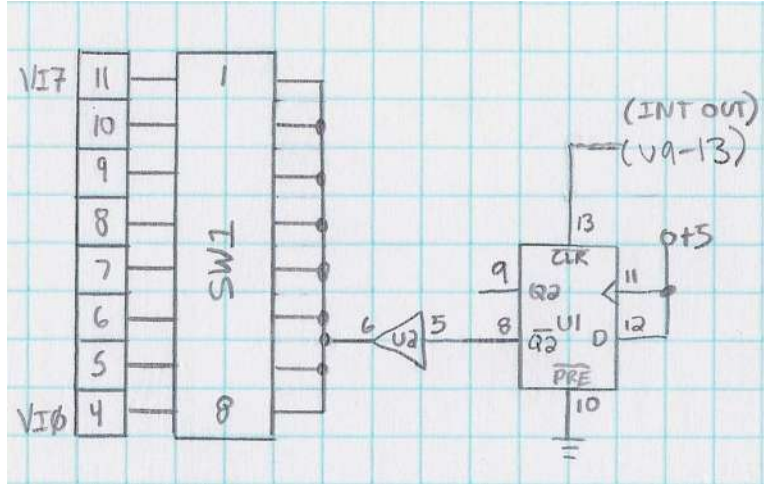
Power Supply:
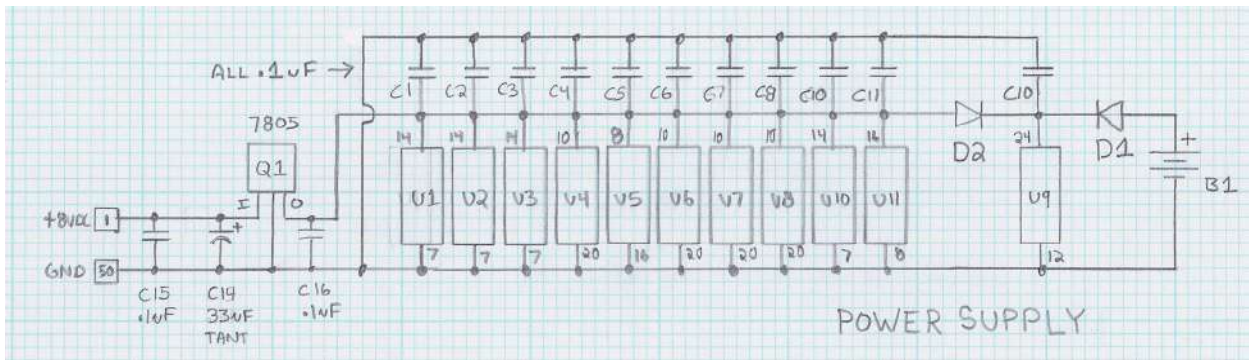
DIP Switches:

Clock, Power Switching, Power Fail, Address buffering:
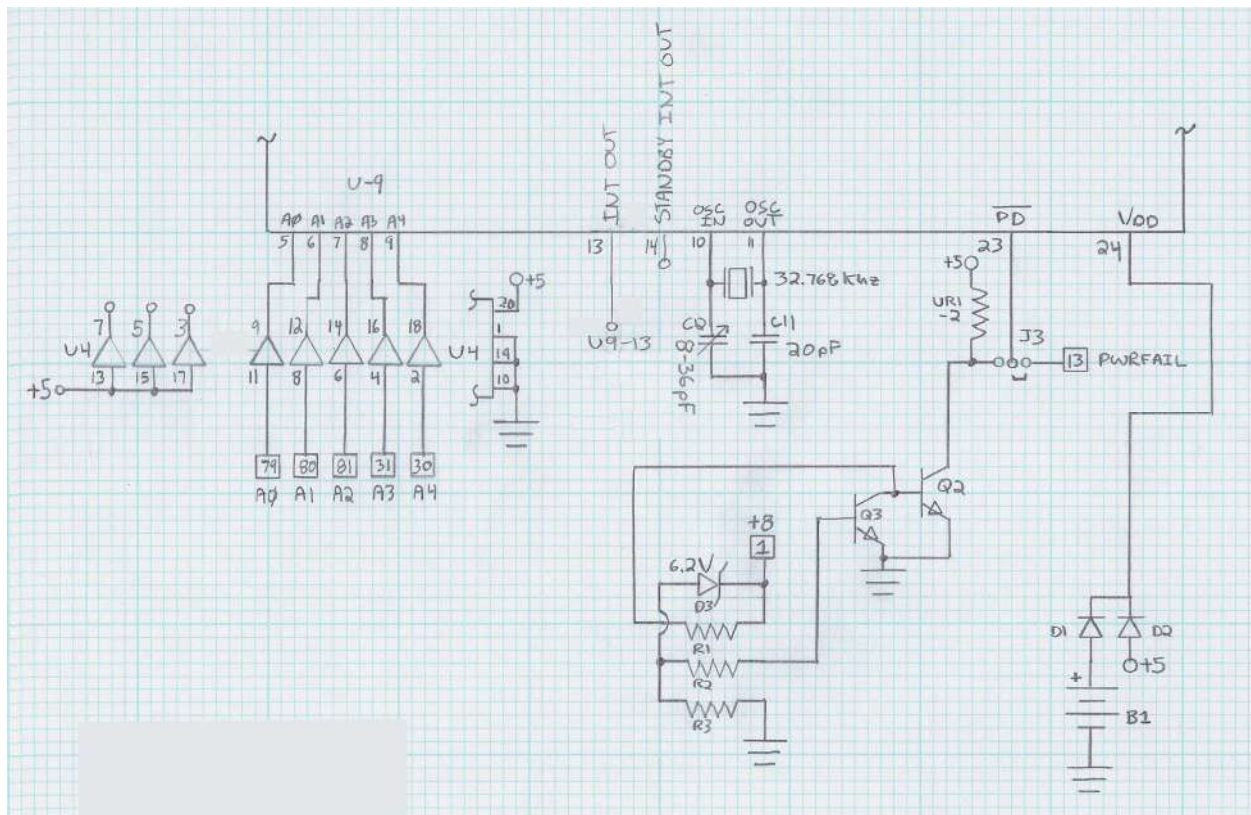
Data Buffering:
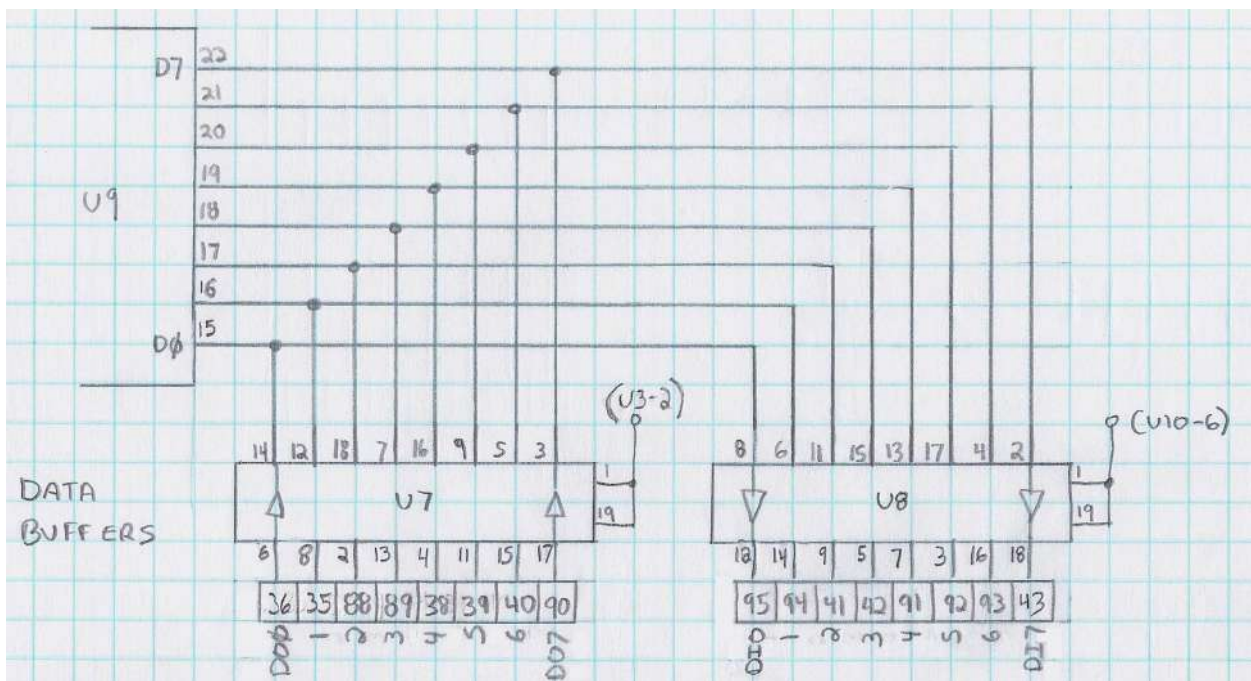
Port Decoder, Extended Address Decoder:
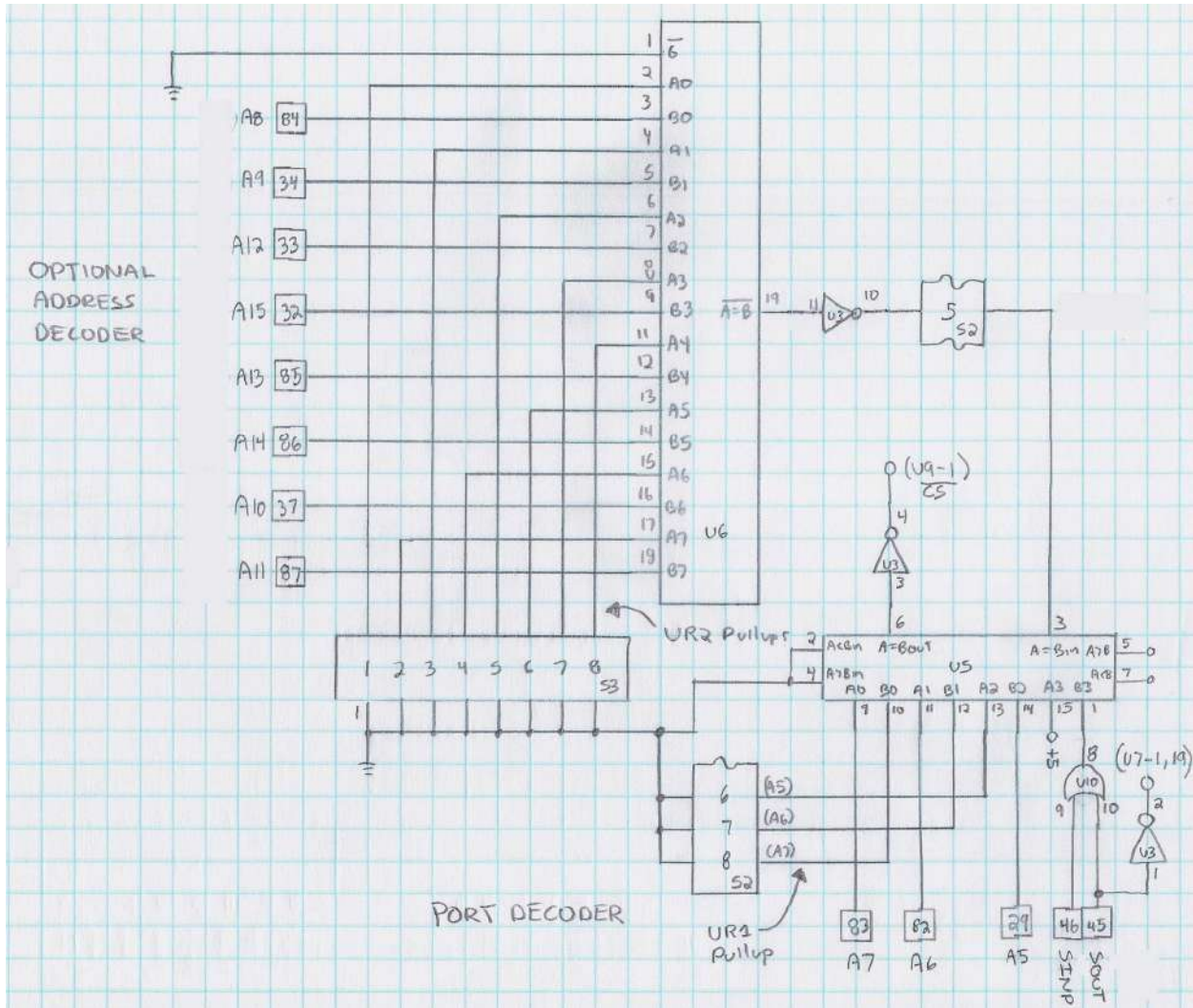
U11 – Unknown IC:
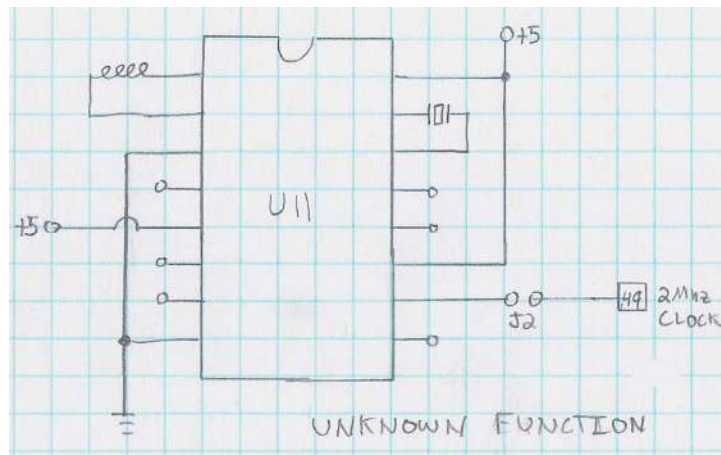
Control Logic:

CONTROL LOGIC
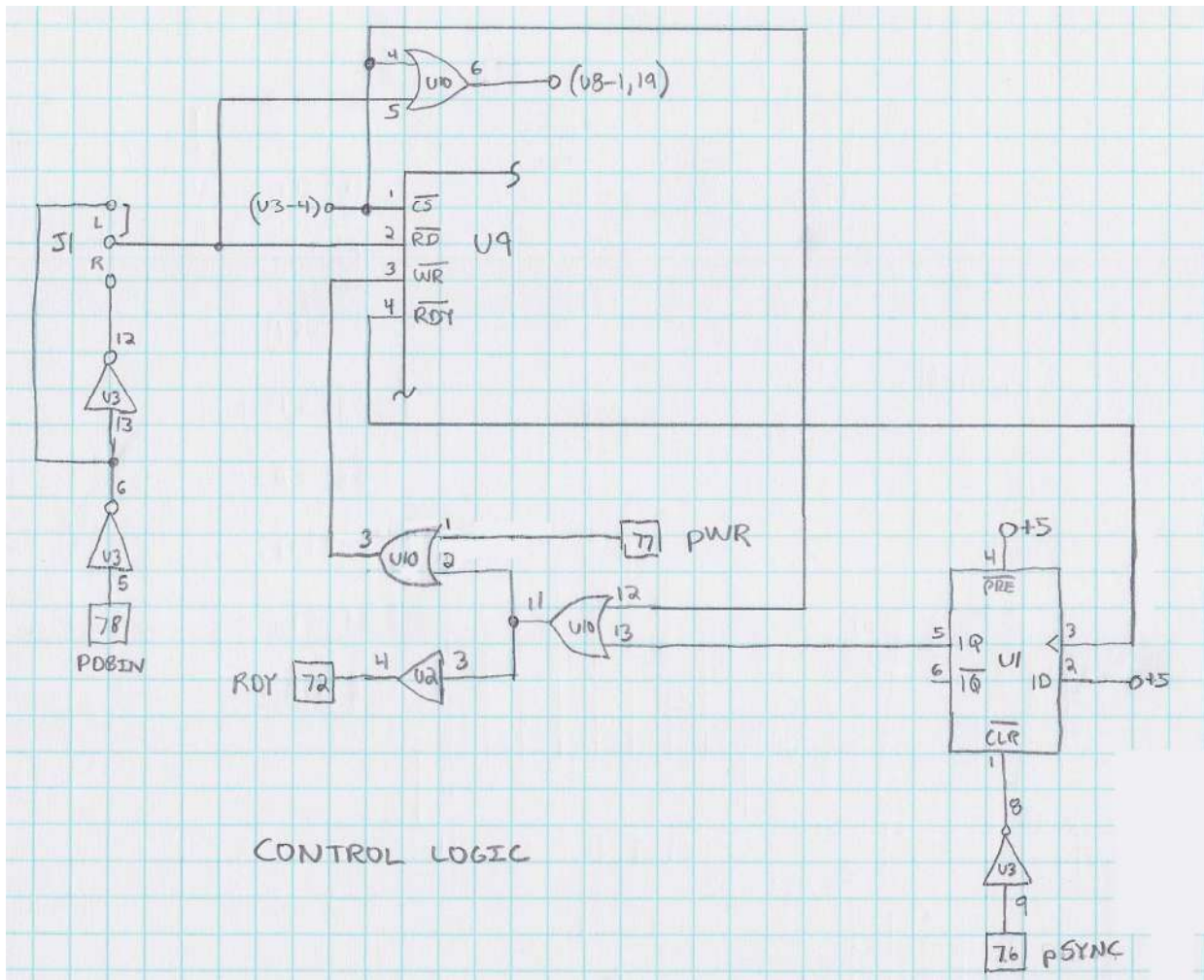
Basic Board Layout:
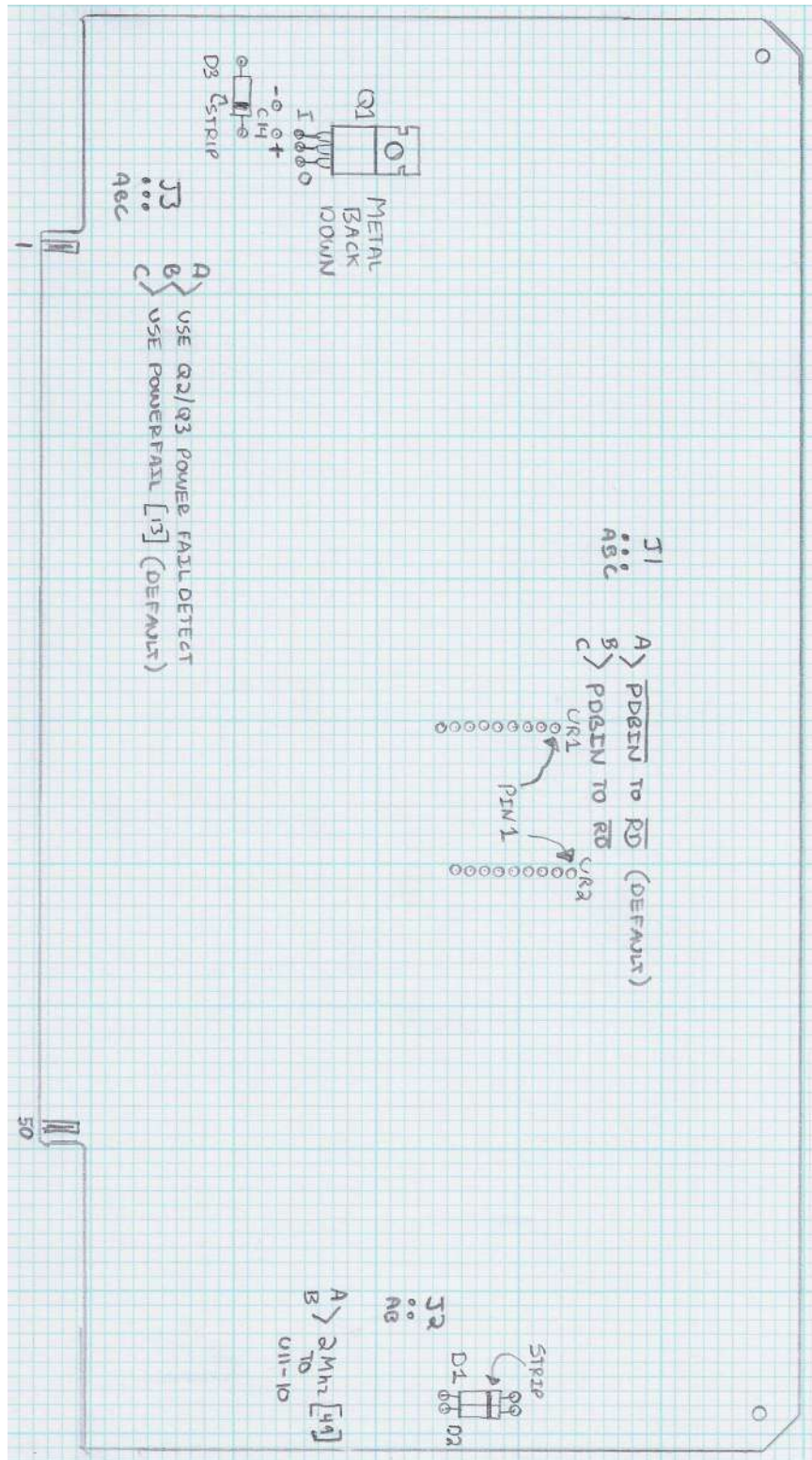
Board layout Front:
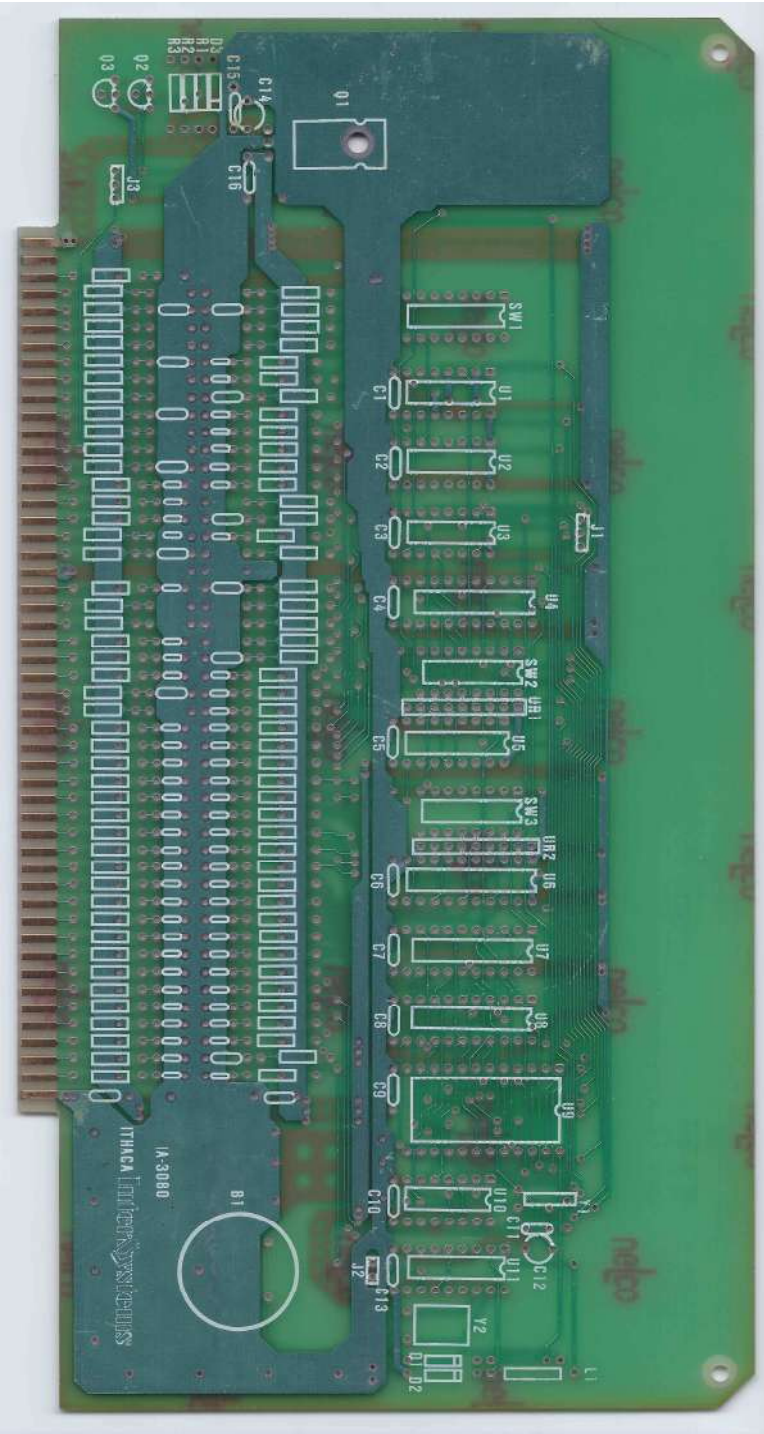
Original scan by Neil Breeden

Board Layout Rear:
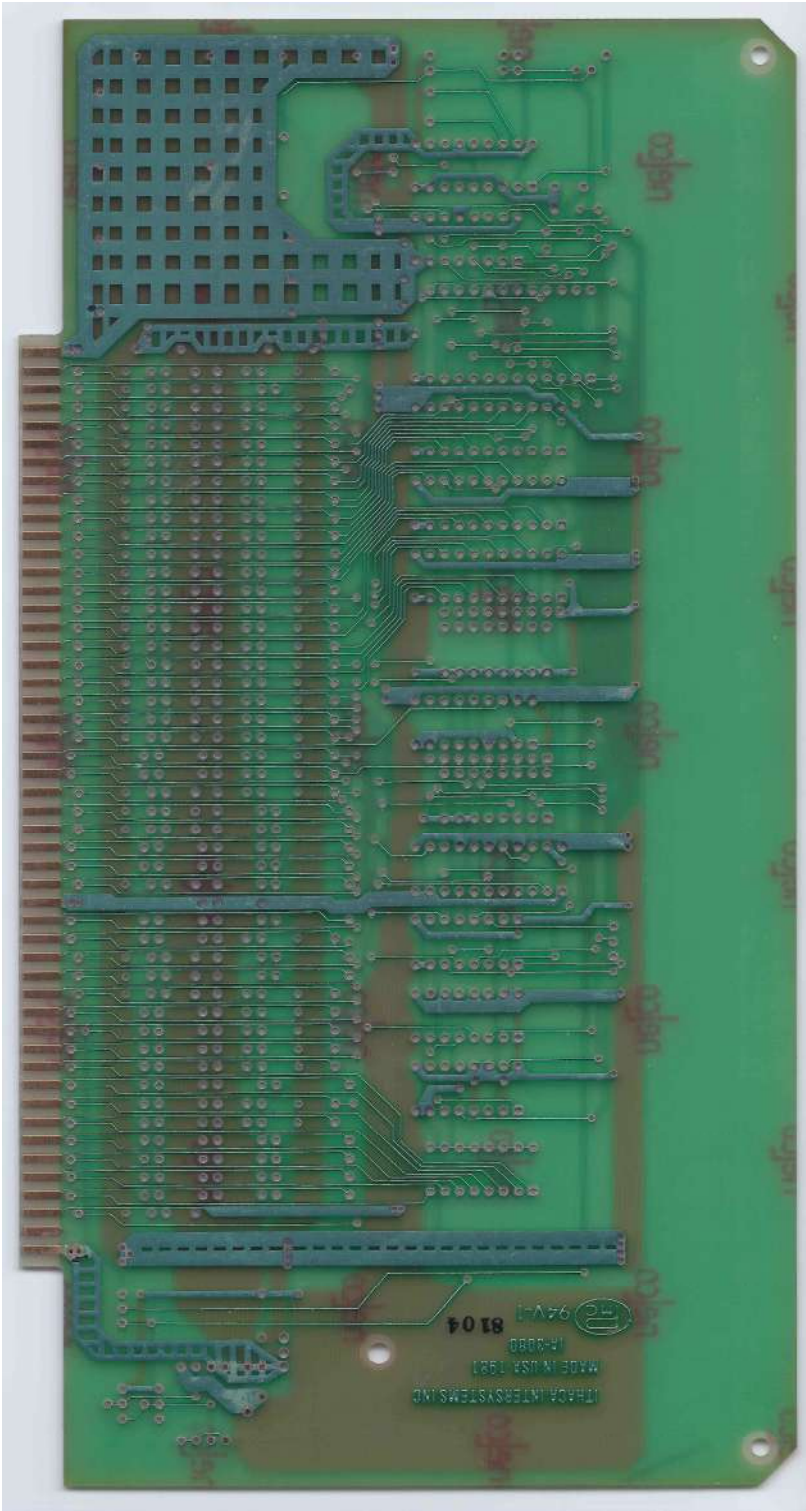
Original scan by Neil Breeden

Example Populated Boards:
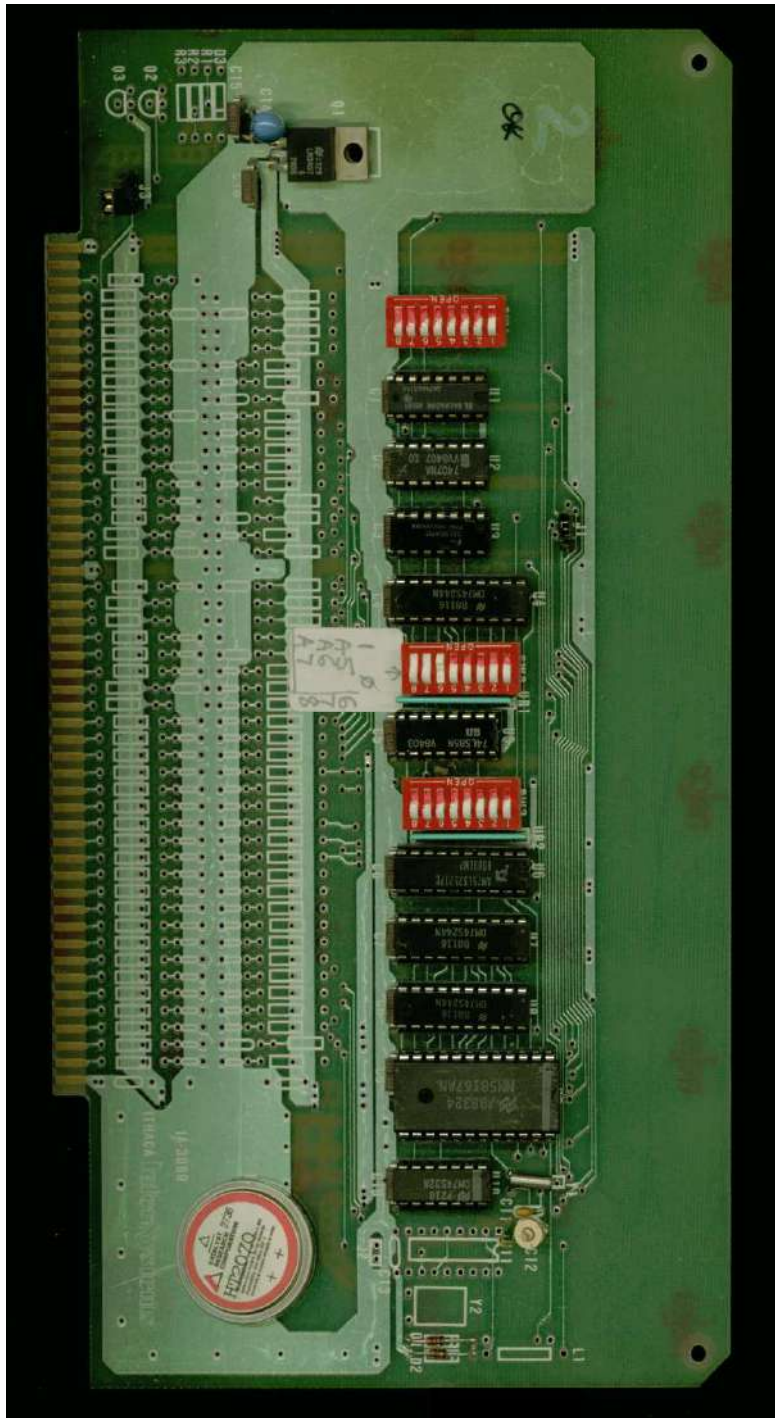
Photo courtesy of Dan's Collection web site:
http://libsrv1.lib.iastate.edu/CPMINFO/boards.html
This board was originally assembled by John King and sold on EBay.
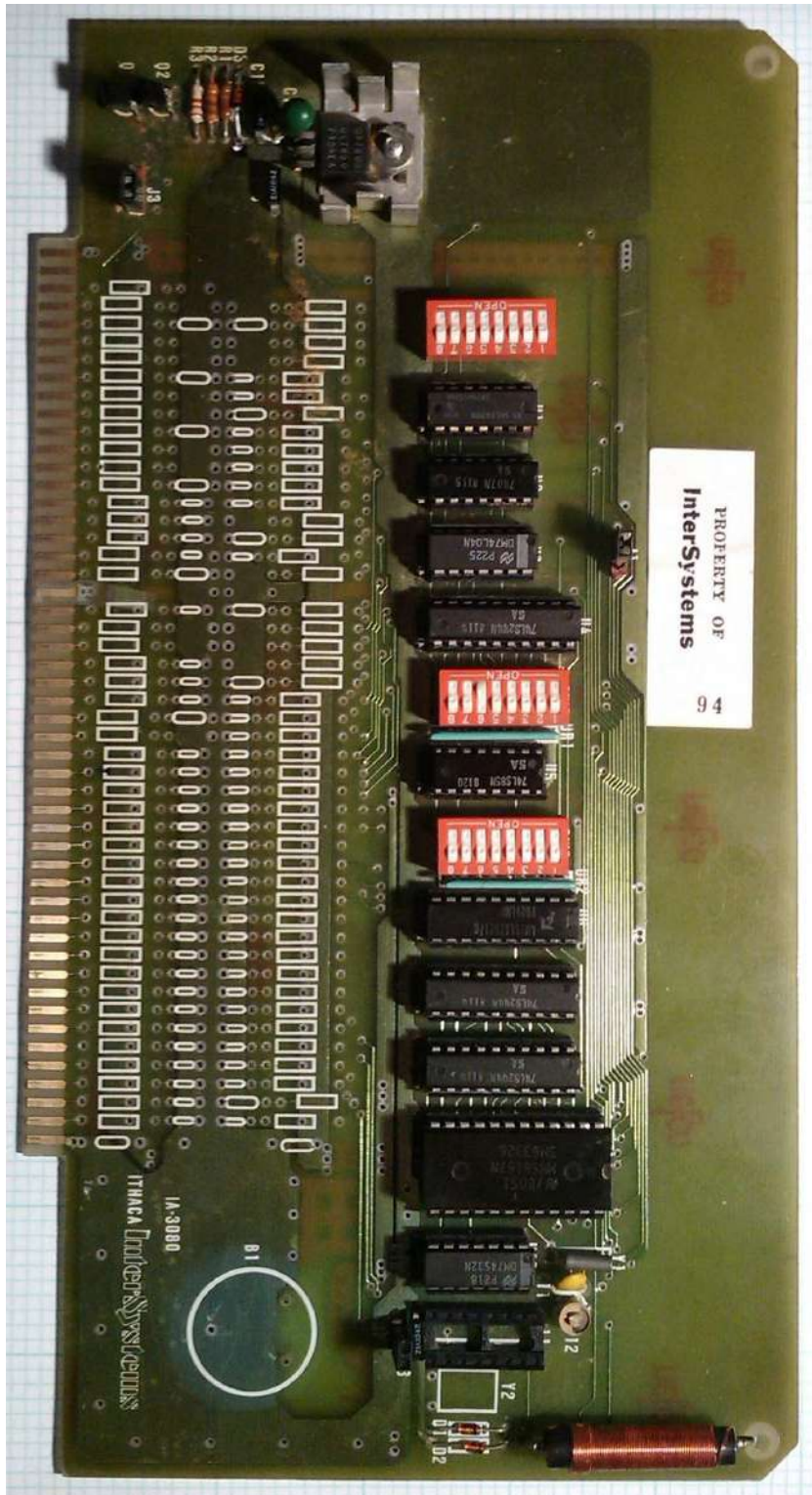
The assembled board John King loaned me.
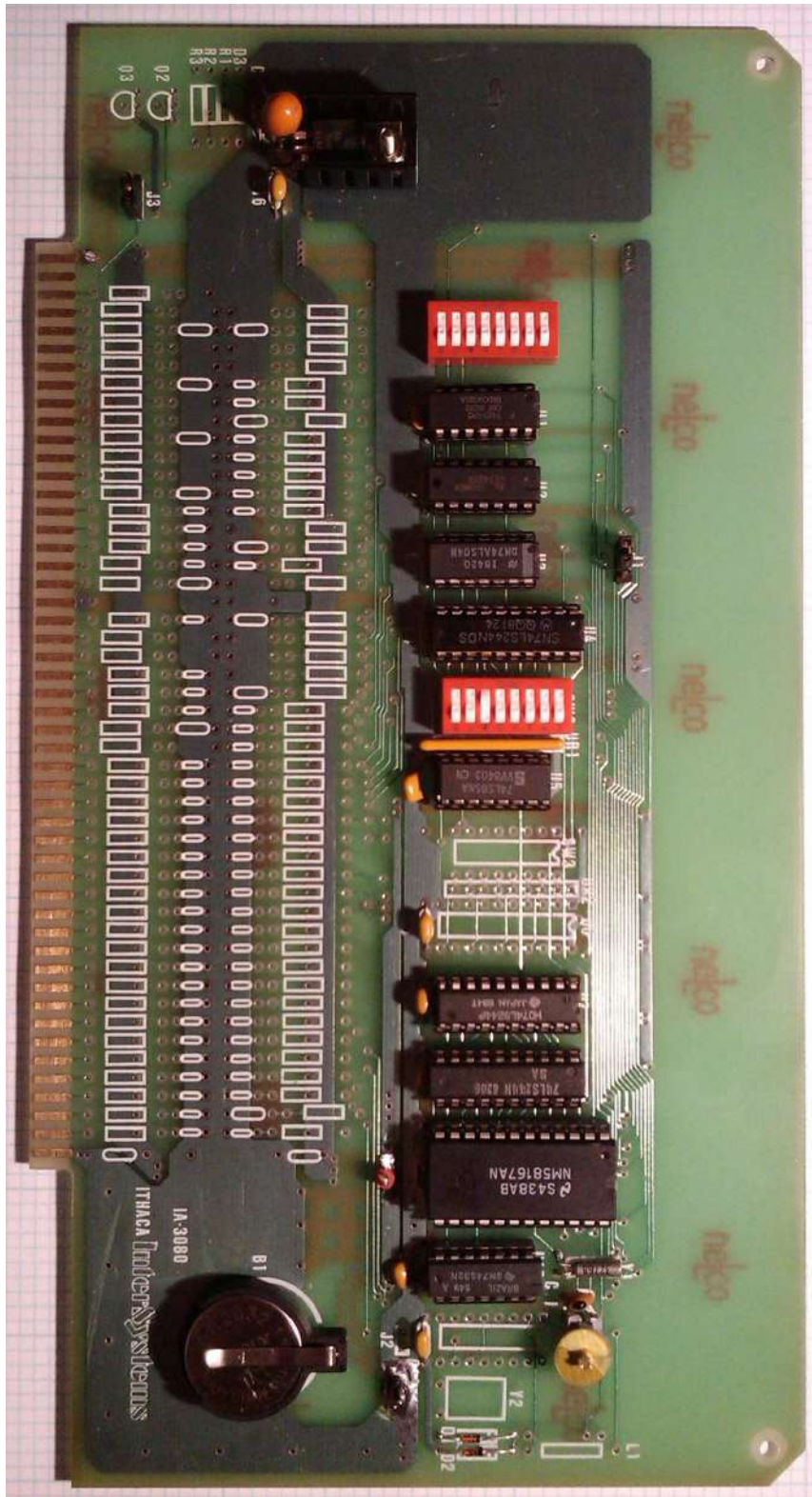
The basic RTC I assembled as I created this manual.

2736 Battery:

MM58167 Datasheet:

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Voltage at All Pins	$V_{SS} - 0.3V$ to $V_{DD} + 0.3V$
Operating Temperature	0°C to 70°C
Storage Temperature	−65°C to +150°C
$V_{DD} - V_{SS}$	6.0V
Lead Temperature (Soldering, 10 sec.)	300°C

Electrical Characteristics $V_{SS} = 0V$, $0°C \leq T_A \leq 70°C$

Parameter	Conditions	Min	Max	Units
Supply Voltage				
V_{DD}	Outputs Enabled	4.5	5.5	V
V_{DD}	POWER DOWN Mode	2.2	5.5	V
Supply Current				
I_{DD}, Dynamic	Outputs TRI-STATE® $f_{IN} = 32.768$ kHz, $V_{DD} = 5.5V$ $V_{IH} \geq V_{DD} - 0.3V$ $V_{IL} \leq V_{SS} + 0.3V$		20	μA
I_{DD}, Dynamic	Outputs TRI-STATE $f_{IN} = 32.768$ kHz, $V_{DD} = 5.5V$ $V_{IH} = 2.0V$, $V_{IL} = 0.8V$		5	mA
Input Voltage				
Logical Low		0.0	0.8	V
Logical high		2.0	V_{DD}	V
Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{DD}$	−1	1	μA
Output Impedance	I/O and INTERRUPT OUT			
Logical Low	$V_{DD} = 4.5V$, $I_{OL} = 1.6$ mA		0.4	V
Logical High	$V_{DD} = 4.5V$, $I_{OH} = -400 \mu A$	2.4		V
	$I_{OH} = -10 \mu A$	$0.8 V_{DD}$		V
TRI-STATE	$V_{SS} \leq V_{OUT} \leq V_{DD}$	−1	1	μA
Output Impedance	RDY and STANDBY INTERRUPT (Open Drain Devices)			
Logical Low, Sink	$V_{DD} = 4.5V$, $I_{OL} = 1.6$ mA		0.4	V
Logical High, Leakage	$V_{OUT} \leq V_{DD}$		10	μA

2

Functional Description

Real Time Counter

The real time counter is divided into 4-bit digits with 2 digits being accessed during any read or write cycle. Each digit represents a BCD number and is defined in Table I. Any unused bits are held at a logical zero during a read and ignored during a write. An unused bit is any bit not necessary to provide a full BCD number. For example tens of hours cannot legally exceed the number 2, thus only 2 bits are necessary to define the tens of hours. The other 2 bits in the tens of hours digit are unused. The unused bits are designated in Table I as dashes.

The addressable portion of the counter is from milliseconds to months. The counter itself is a ripple counter. The ripple delay is less than 60 μs above 4.5V and 300 μs at 2.2V.

RAM

56 bits of RAM are contained on-chip. These can be used for any necessary power down storage or as an alarm latch for comparison to the real time counter. The data in the RAM can be compared to the real time counter on a digit basis. The only digits that are not compared are the unit ten thousandths of seconds and tens of days of the week (these are unused in the real time counter). If the two most significant bits of any RAM digit are ones, then this RAM location will always compare. The rule of thumb for an "alarm" interrupt is: All nibbles of higher order than specified are set to C hex (always compare). All nibbles lower than specified are set to "zero". As an example, if an alarm is to occur everyday at 10:15 a.m., configure the bits in RAM as shown in Table II.

The RAM is formatted the same as the real time counter, 4 bits per digit, 14 digits, however there are no unused bits.

The unused bits in the real time counter will compare only to zeros in the RAM.

An address map is shown in Table III.

Interrupts and Comparator

There are two interrupt outputs. The first is the INTERRUPT OUTPUT (a true high signal). This output can be programmed to provide 8 different output signals. They are: 10 Hz, once per second, once per minute, once per hour, once a day, once a week, once a month, and when a RAM/real time counter comparison occurs. To enable the output a one is written into the interrupt control register at the bit location corresponding to the desired output frequency (*Figure 1*). Once one or more bits have been set in the interrupt control register, the corresponding counter's rollover to its reset state will clock the interrupt status register and cause the interrupt output to go high. To reset the interrupt and to identify which frequency caused the interrupt, the interrupt status register is read. Reading this register places the contents of the status register on the data bus. The interrupting frequency will be identified by a one in the respective bit position. Removing the read will reset the interrupt.

The second interrupt is the STANDBY INTERRUPT (open drain output, active low). This interrupt occurs when enabled and when a RAM/real time counter comparison occurs. The STANDBY INTERRUPT is enabled by writing a one on the D0 line at address 16$_H$ or disabled by writing a zero on the D0 line. This interrupt is not triggered by the edge of the compare signal, but rather by the level. Thus if the compare is enabled when the STANDBY INTERRUPT is enabled, the interrupt will turn on immediately.

TABLE I. Real Time Counter Format

Counter Addressed		Units				Max BCD Code	Tens				Max BCD Code
		D0	D1	D2	D3		D4	D5	D6	D7	
Milliseconds	(00$_H$)	—	—	—	—	0	D4	D5	D6	D7	9
Hundredths and Tenths Sec	(01$_H$)	D0	D1	D2	D3	9	D4	D5	D6	D7	9
Seconds	(02$_H$)	D0	D1	D2	D3	9	D4	D5	D6	—	5
Minutes	(03$_H$)	D0	D1	D2	D3	9	D4	D5	D6	—	5
Hours	(04$_H$)	D0	D1	D2	D3	9	D4	D5	—	—	2
Day of the Week	(05$_H$)	D0	D1	D2	—	7	—	—	—	—	0
Day of the Month	(06$_H$)	D0	D1	D2	D3	9	D4	D5	—	—	3
Month	(07$_H$)	D0	D1	D2	D3	9	D4	—	—	—	1

(—) indicates unused bits

3

Functional Description (Continued)

TABLE II. Clock RAM Bit Map for Alarm Interrupt Everyday at 10:15 a.m.

Function	Address					Data							
						Hi Nibble				Lo Nibble			
	4	3	2	1	0	7	6	5	4	3	2	1	0
Milliseconds	0	1	0	0	0	0	0	0	0	No RAM Exists			
Hundredths and Tenths of Seconds	0	1	0	0	1	0	0	0	0	0	0	0	0
Seconds	0	1	0	1	0	0	0	0	0	0	0	0	0
Minutes	0	1	0	1	1	0	0	0	1	0	1	0	1
Hours	0	1	1	0	0	0	0	0	1	0	0	0	0
Day of Week	0	1	1	0	1	No RAM Exists				1	1	X	X
Day of Month	0	1	1	1	0	1	1	X	X	1	1	X	X
Months	0	1	1	1	1	1	1	X	X	1	1	X	X

TABLE III. Address Codes and Function

A4	A3	A2	A1	A0	Function
0	0	0	0	0	Counter—Milliseconds
0	0	0	0	1	Counter—Hundredths and Tenths of Seconds
0	0	0	1	0	Counter—Seconds
0	0	0	1	1	Counter—Minutes
0	0	1	0	0	Counter—Hours
0	0	1	0	1	Counter—Day of Week
0	0	1	1	0	Counter—Day of Month
0	0	1	1	1	Counter—Month
0	1	0	0	0	RAM—Milliseconds
0	1	0	0	1	RAM—Hundredths and Tenths of Seconds
0	1	0	1	0	RAM—Seconds
0	1	0	1	1	RAM—Minutes
0	1	1	0	0	RAM—Hours
0	1	1	0	1	RAM—Day of Week
0	1	1	1	0	RAM—Day of Month
0	1	1	1	1	RAM—Months
1	0	0	0	0	Interrupt Status Register
1	0	0	0	1	Interrupt Control Register
1	0	0	1	0	Counters Reset
1	0	0	1	1	RAM Reset
1	0	1	0	0	Status Bit
1	0	1	0	1	GO Comand
1	0	1	1	0	STANDBY INTERRUPT
1	1	1	1	1	Test Mode

All others unused

The comparator is a cascaded exclusive NOR. Its output is latched 61 μs after the rising edge of the 1 kHz clock signal (input to the milliseconds counter). This allows the counter to ripple through before looking at the comparator. For operation at less than 4.5V, the thousandths of seconds counter should not be included in a compare because of the possibility of having a ripple delay greater than 61 μs. (For output timing see Interrupt Timing.)

Power Down Mode

The POWER DOWN input is essentially a second chip select. It disables all inputs and outputs except for the STANDBY INTERRUPT. When this input is at a logical zero, the device will not respond to any external signals. It will, however, maintain timekeeping and turn on the STANDBY INTERRUPT if programmed to do so. (The programming must be done before the POWER DOWN input goes to a

4

Functional Description (Continued)

logical zero.) When switching V_{DD} to the standby or power down mode, the POWER DOWN input should go to a logical zero at least 1 µs before V_{DD} is switched. When switching V_{DD} all other inputs must remain between V_{SS} − 0.3V and V_{DD} + 0.3V. When restoring V_{DD} to the normal operating mode, it is necessary to insure that all other inputs are at valid levels before switching the POWER DOWN input back to a logical one. These precautions are necessary to insure that no data is lost or altered when changing to or from the power down mode.

Counter and RAM Resets; GO Command

The counters and RAM can be reset by writing all 1's (FF) at address 12_H or 13_H respectively.

A write pulse at address 15_H will reset the thousandths, hundredths, tenths, units, and tens of seconds counters. This GO command is used for precise starting of the clock. The data on the data bus is ignored during the write. If the seconds counter is at a value greater than 39 when the GO is issued, the minute counter will increment; otherwise the minute counter is unaffected. This command is not necessary to start the clock, but merely a convenient way to start precisely at a given minute.

Status Bit

The status bit is provided to inform the user that the clock is in the process of rolling over when a counter is read. The status bit is set if this 1 kHz clock occurs during or after any counter read. This tells the user that the clock is rippling through the real time counter. Because the clock is rippling, invalid data may be read from the counter. If the status bit is set following a counter read, the counter should be reread.

The status bit appears on D0 when address 14_H is read. All the other data lines will zero. The bit is set when a logical one appears. This bit should be read every time a counter read or after a series of counter reads are done. The trailing edge of the read at address 14_H will reset the status bit.

Using the Rollover Status Bit

If a single read of any clock counter is made, it should be followed by reading the rollover status bit.

Example: Read months, then read rollover status.

If a sequential read of the clock counters is made, then the rollover status bit should be read after the last counter is read.

Example: Read hours, minutes, seconds, then read the rollover status.

Oscillator

The oscillator used is the standard Pierce parallel resonant oscillator. Externally, 2 capacitors, a 20 MΩ resistor and the crystal are required. The 20 MΩ resistor is connected between OSC IN and OSC OUT to bias the internal inverter in the linear region. For micropower crystals a resistor in series with the oscillator output may be necessary to insure the crystal is not overdriven. This resistor should be approximately 200 kΩ. The capacitor values should be typically 20 pF−25 pF. The crystal frequency is 32,768 Hz.

The oscillator input can be externally driven, if desired. In this case the oscillator output should be left floating and the oscillator input levels should be within 0.3V of the supplies.

A ground line or ground plane between pins 9 and 10 may be necessary to reduce interference of the oscillator by the A4 address.

Control Lines

The READ, WRITE, AND CHIP SELECT signals are active low inputs. The READY signal is an open drain output. At the start of each read or write cycle the READY line (open drain) will pull low and will remain low until valid data from a chip read appears on the bus or data on the bus is latched in during a write. READ and WRITE must be accompanied by a CHIP SELECT (see *Figures 3* and *4* for read and write cycle timing).

During a read or write, address bits must not change while chip select and control strobes are low.

Test Mode

The test mode is for production testing. It allows the counters to count at a higher than normal rate. In this mode the 32.768 kHz oscillator input is connected directly to the ten thousandths of seconds counter. The chip select and write lines must be low and the address must be held at $1F_H$.

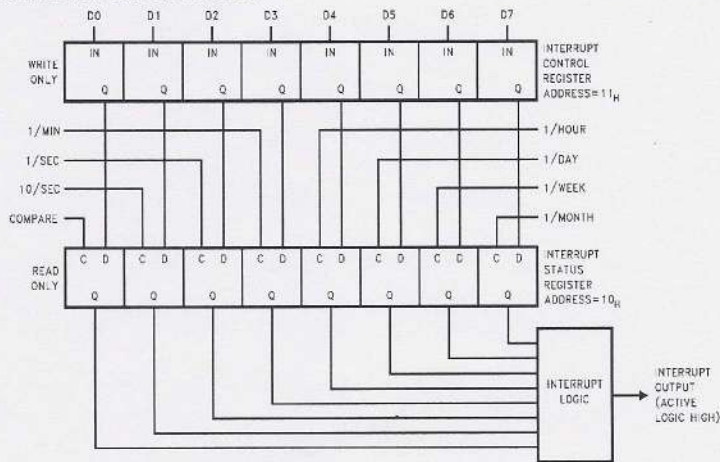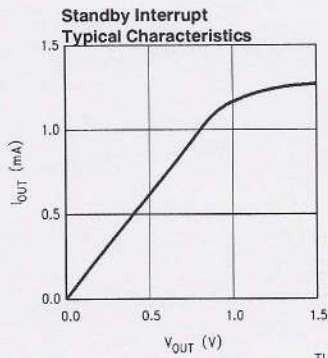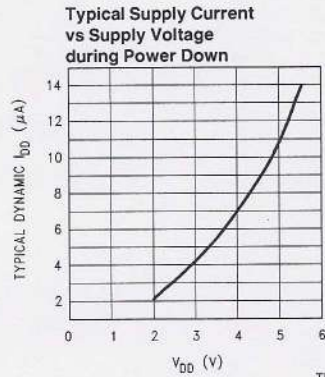FIGURE 1. Interrupt Register Format

TL/F/11070−3

5

Functional Description (Continued)

Standby Interrupt Typical Characteristics

Typical Supply Current vs Supply Voltage during Power Down

TL/F/11070–4   TL/F/11070–5

FIGURE 2

Interrupt Timing $0°C \leq T_A \leq 70°C, 4.5V \leq V_{DD} \leq 5.5V, V_{SS} = 0V$

Symbol	Parameter	Min	Max	Units
t_{INTON}	Status Register Clock to INTERRUPT OUTPUT (Pin 13) High (Note 1)		5	μs
t_{SBYON}	Compare Valid to STANDBY INTERRUPT (Pin 14) Low (Note 1)		5	μs
t_{INTOFF}	Trailing Edge of Status Register Read to INTERRUPT OUTPUT Low		5	μs
t_{SBYOFF}	Trailing Edge of Write Cycle (D0 = 0; Address = 16$_H$) to STANDBY INTERRUPT Off (High Impedance State)		5	μs

Note 1: The status register clocks are: the corresponding counter's rollover to its reset state or the compare becoming valid. The compare becomes valid 61 μs after the 1/10,000 of a second counter is clocked, if the real time counter data matches the RAM data.

Read Cycle Timing $0°C \leq T_A \leq 70°C, 4.5V \leq V_{DD} \leq 5.5V, V_{SS} = 0V$

Symbol	Parameter	Min	Max	Units
t_{AR}	Address Bus Valid to Read Strobe (Note 3)	100		ns
t_{CSR}	Chip Select to Read Strobe (Note 2)	0		ns
t_{RRY}	Read Strobe to Ready Strobe		150	ns
t_{RYD}	Ready Strobe to Data Valid		800	ns
t_{AD}	Address Bus Valid to Data Valid		1050	ns
t_{RH}	Data Hold Time from Trailing Edge of Read Strobe	0		ns
t_{HZ}	Trailing Edge of Read Strobe to TRI-STATE Mode		250	ns
t_{RYH}	Read Hold Time after Ready Strobe	0		ns
t_{RA}	Address Bus Hold Time from Trailing Edge of Read Strobe	50		ns
t_{RYDV}	Rising Edge of Ready to Data Valid		100	ns

Note 2: When reading, a deselect time of 500 ns minimum must occur between counter reads. Deselect is: $\overline{CS} = 1$ or $(\overline{WR}) \cdot (\overline{RD}) = 1$.

Note 3: If $t_{AR} = 0$ and Chip Select, Address Valid or Read are coincident then they must exist for 1050 ns.

6

Write Cycle Timing $0°C \leq T_A \leq 70°C$, $4.5V \leq V_{DD} \leq 5.5V$, $V_{SS} = 0V$

Symbol	Parameter	Min	Max	Units
t_{AW}	Address Valid to Write Strobe	100		ns
t_{CSW}	Chip Select to Write Strobe	0		ns
t_{DW}	Data Valid before Write Strobe	100		ns
t_{WRY}	Write Strobe to Ready Strobe		150	ns
t_{RY}	Ready Strobe Width		800	ns
t_{RYH}	Write Hold Time after Ready Strobe	0		ns
t_{WD}	Data Hold Time after Write Strobe	110		ns
t_{WA}	Address Hold Time after Write Strobe	50		ns

Note 4: If data changes while $\overline{CS}$ and $\overline{WR}$ are low, then they must remain coincident for 1050 ns after the data change to ensure a valid write. Data bus loading is 100 pF. Ready output loading is 50 pF and 3 kΩ pull-up.

Input and output AC timing levels:
Logical one = 2.0V
Logical zero = 0.8V

Read and Write Cycle Timing Diagrams
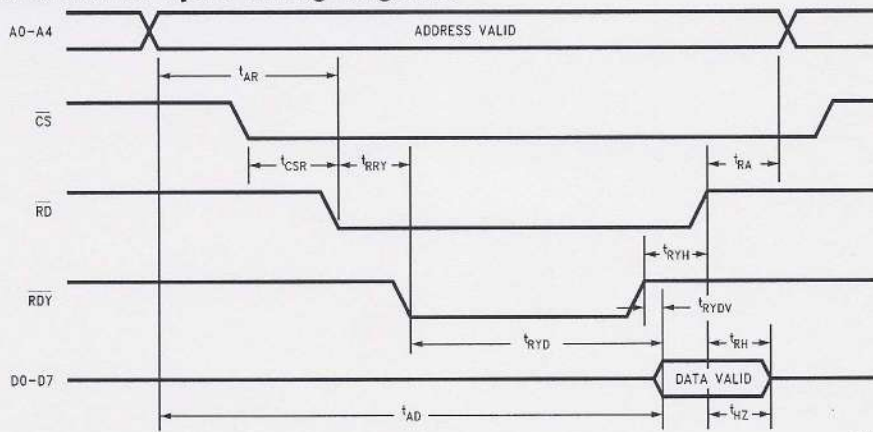
FIGURE 3. Read Cycle Timing

TL/F/11070–6
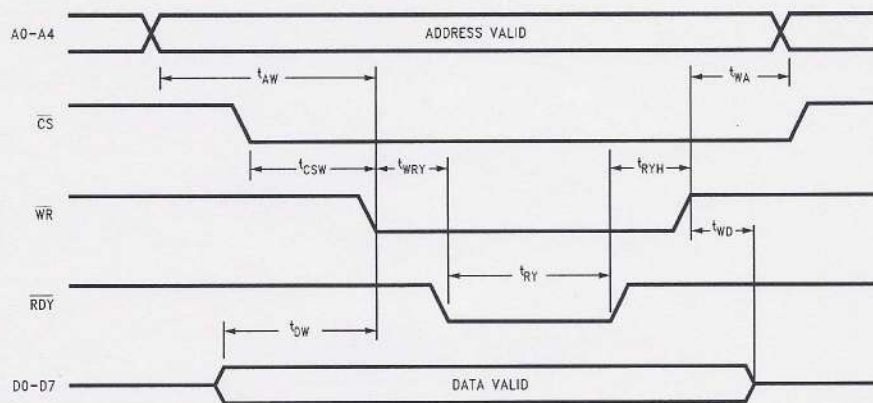
FIGURE 4. Write Cycle Timing

TL/F/11070–7

7

Block Diagram
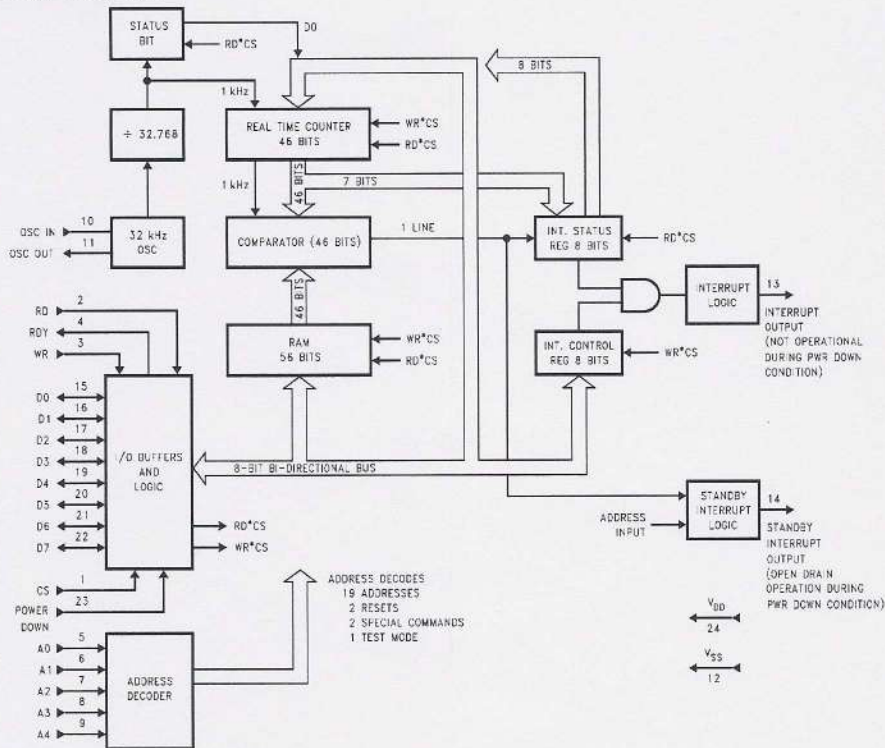
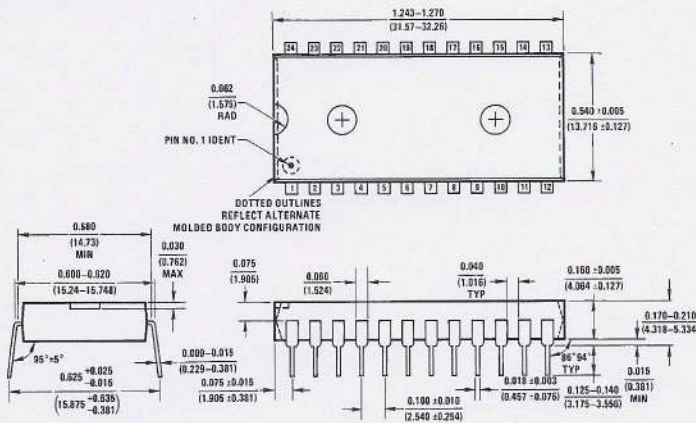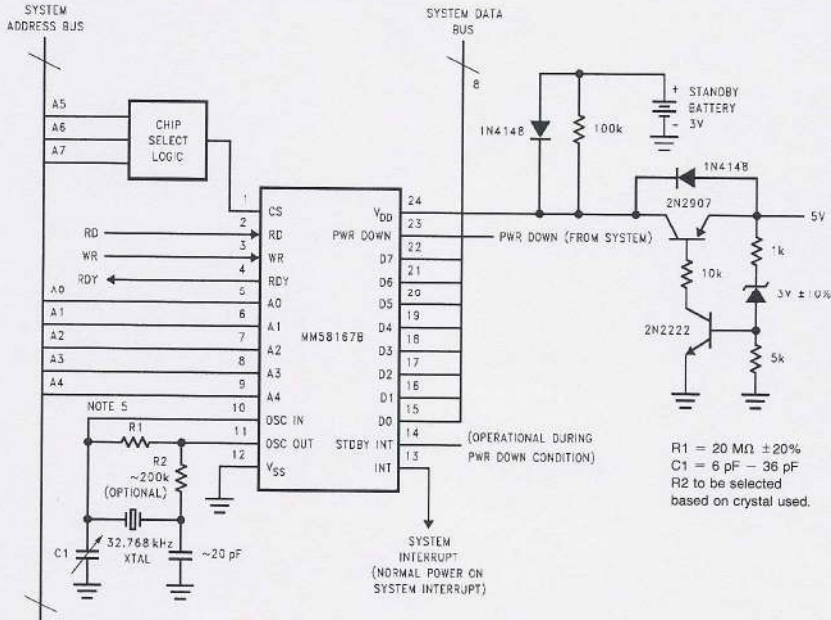
FIGURE 7

TL/F/11070–10

Physical Dimensions inches (millimeters)

Molded Dual-In-Line Package (N)
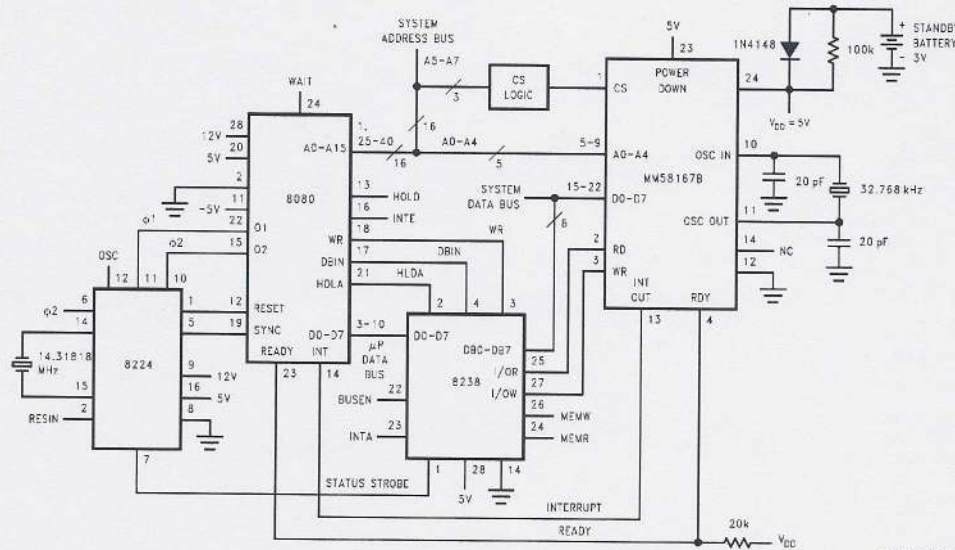Order Number MM58167BN
NS Package Number N24A

9

Typical Applications

Note 5: A ground line or ground plane guard trace should be included between pins 9 and 10 to insure the oscillator is not disturbed by the address line.

FIGURE 5. Typical Connection Diagram

Note 6: Must use 8238 or equivalent logic to insure advanced I/OW pulse; so that the ready output of the MM58167B is valid by the end of φ2 during the T2 microcycle.

Note 7: $t_{\phi2} \geq t_{RS8080} + t_{DL8238} + t_{WRY58167B}$.

FIGURE 6. 8080 System Interface with Battery Backup

8

MM58167B Microprocessor Real Time Clock

Physical Dimensions inches (millimeters) (Continued)
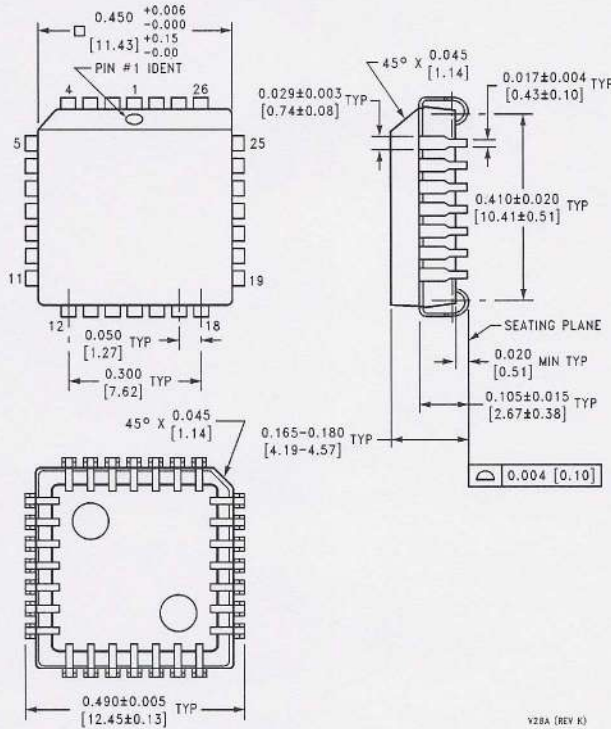
PCC Package (V)
Order Number MM58167BV
NS Package Number V28A

V28A (REV K)

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.