Testrom v2 for MPU boards level 3, 4, 6, 7 and corresponding driver boards

Author: André Boot

Author André Boot

Testrom v2.0 procedures

1. Introduction

In september 1998 I created a testrom for W cpu’s which gave it a “Bally” alike test capabilities. Now its 2008 and I wrote v2.0 because I had some trouble pinpointing the defective devices. (And thought it would be fun).

This is document is a brief summary of the W MPU testrom v2 for MPU boards 3,4,6,7 and corresponding driver boards. These Boards were used in pinball's Hot Tip (1977) until Laser Cue (1984). In order to perform the test this board should have a special boot ROM.

This document is structured as follows: The next chapter explains how to install the boot test ROM and the special key. Chapter 3 explains the MPU test with the Test flow chart with the repair on component level.

Although the content of the boot ROM has been thoroughly checked, the author holds no responsibility if any accident or damages which may occur to MPU boards, pinball, and any other living or non living subjects.

2. Installation van de Boot test ROM

The following steps should be taken for installation of the test ROM and the test key.

1. Switch of power

2. Remover connectors: 1J3, 1J4, 1J5, 1J6, 1J7, (for level 7 boards also 1J8).

3. Remove the flipper ROM 2 from the MPU board (location ROM 2, IC 17 see figure 1 for the different boards) (best to remove all roms). Place the boot ROM into the IC socket.

4. Only for level 7 boards: Make the following jumper's W23 (typical setting W22), W12 (typical setting W11), W 9 (typical setting W10).

IMPORTANT

Do not touch the SW2 (data switch /input enable switch/master command enter), as long testrom is present).Only use the diagnostic switch SW1

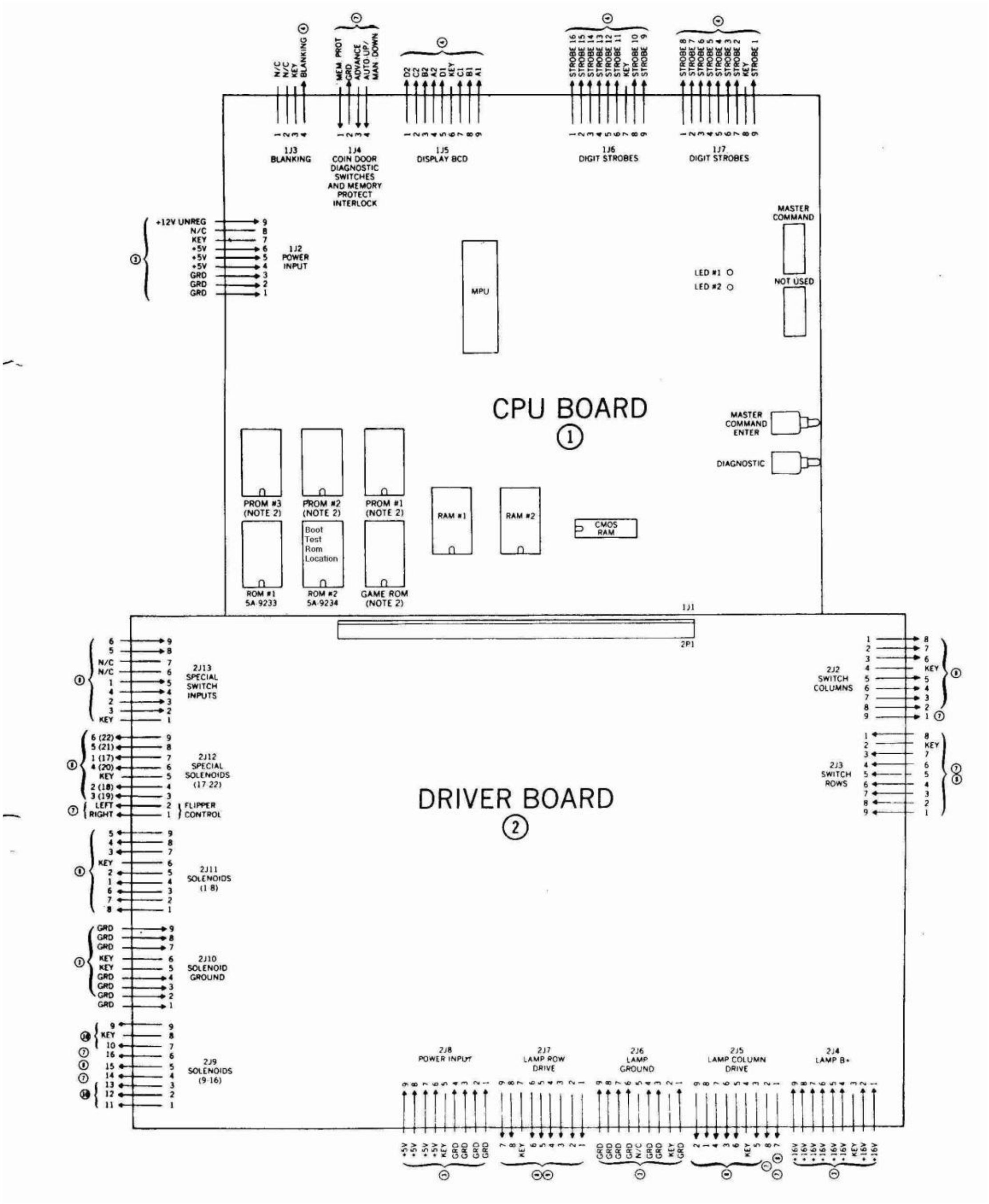


Figure A: identification of the different MPU boards and the location of the boot ROM.

3. Installation van de Boot test ROM

The following steps should be taken for installation of the test ROM and the test key.

3.1 Power up test

After inserting the boot ROM the test program is started by powering up.

During power up it will test hardware without making use of the memory (assembly level programmers no subroutines and no stack for pushing things!)

For system 7 boards it will use the display.

The software will test al the chips for system 7 configuration. For system 7 it will display the progress on the seven segment display. "One(1) through nine(9). Normal display means ok. Brief flickering/flashing digit means defective corresponding chip.

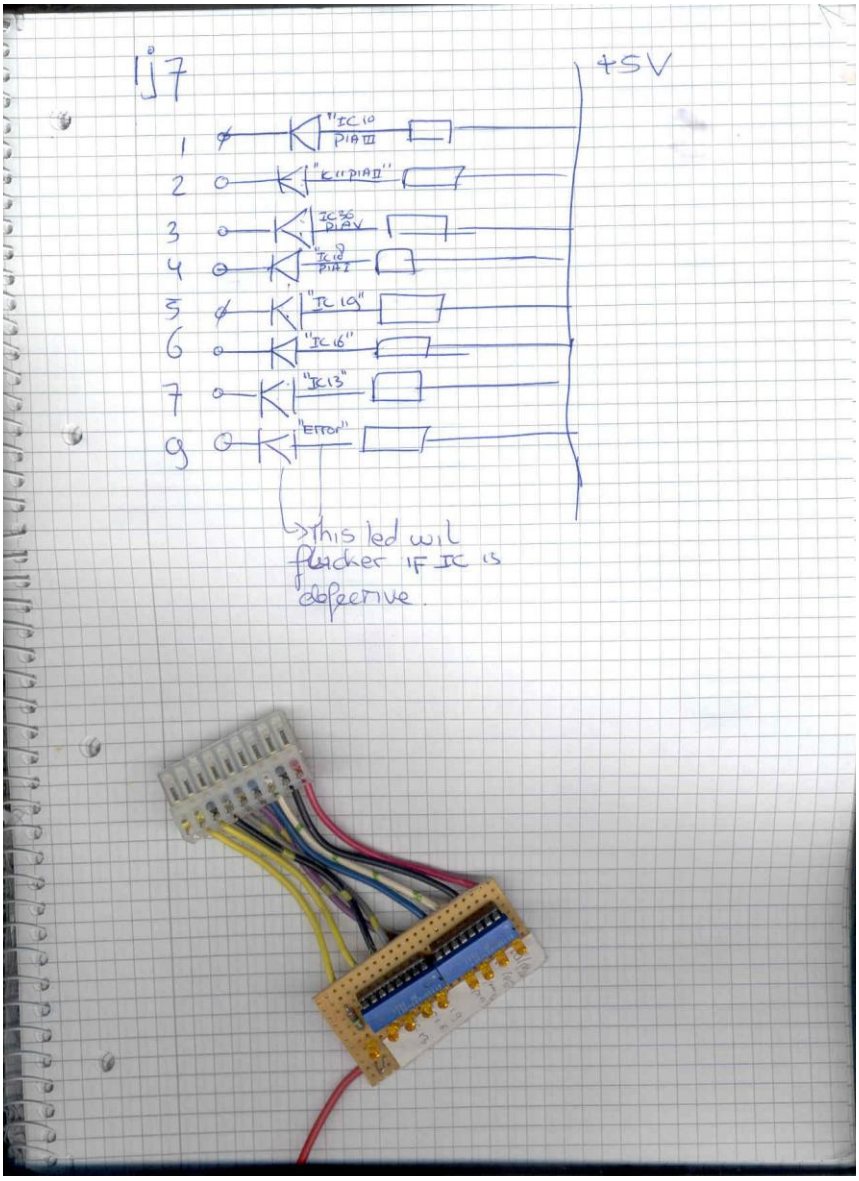
In addition 1J7 will output the display value also (see next section)

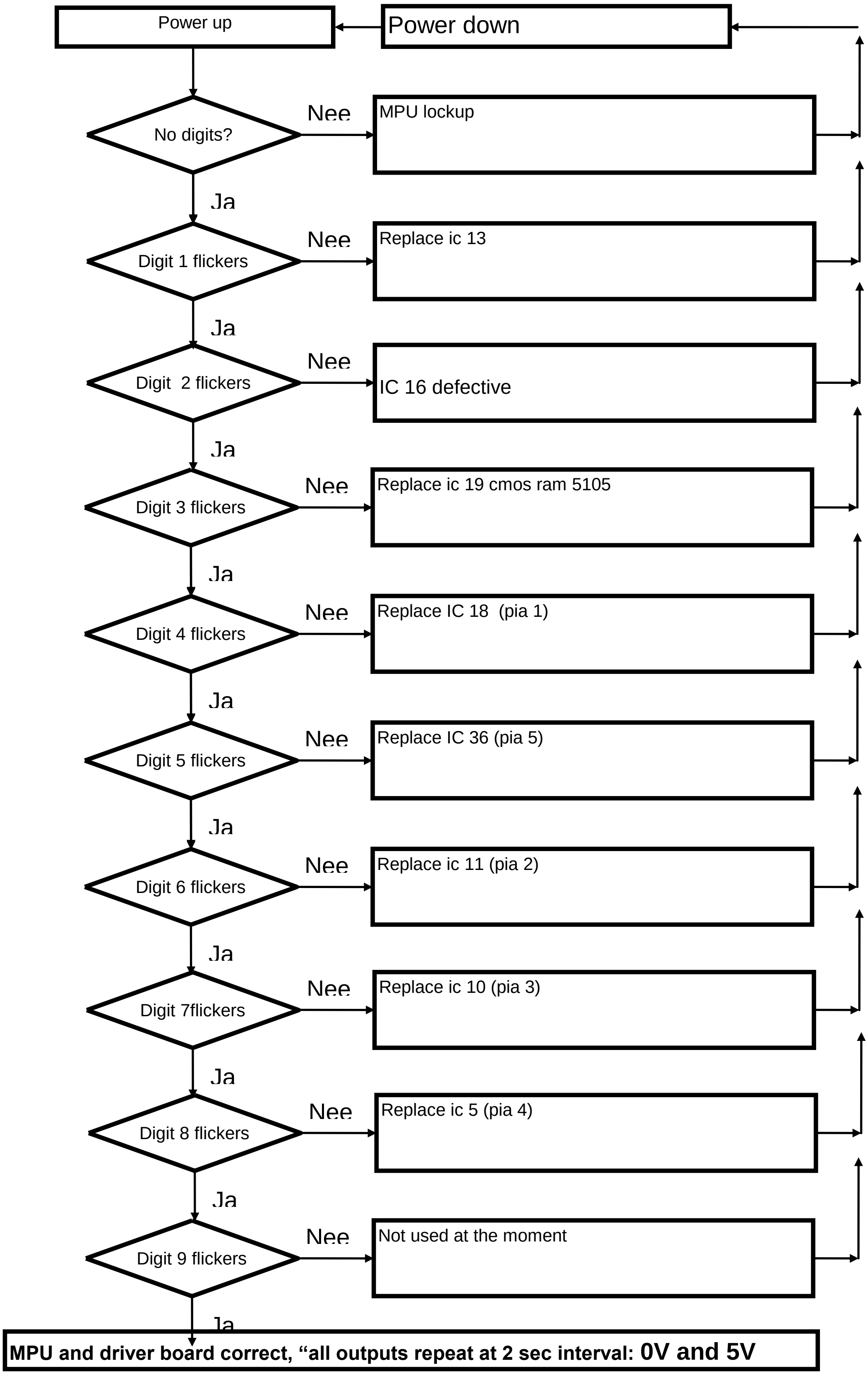
Furthermore the powerup test tests memory for syst 7 configuration. So it wil display the defective memory chip correctly

After the test it will do a "high/low" test on all pia outputs

3.2 Additional diagnostic leds.

Resistors are all 1 Kohm. Use small leds else they will not lite clearly.





Diagnostic key Test

3.3 Diagnosic switch test

Also pressing the diagnostic key will initiate the “old” test program. It allows more extensive testing of the pia’s and memory. But it will not stop flashing after it encounters a bad component (except ram failure can give strange effects)). It will flicker the led display. As said the diagnostic test uses extensively the stack so the memory chips should be ok for this test.

Furthermore the diagnostic test, tests memory for syst 3 through 6 (6810). So it will display the defective memory chip more correctly!

After the test it will do a “knight rider” loop on all pia outputs (except the one which should be input)

Attaching a switch to 2J3 (pull one input to ground) will display the 2J3 bits which are grounded to be visible on the 2J2 output. If nothing is connected to 2J3 “a “knight rider” loop should be present.

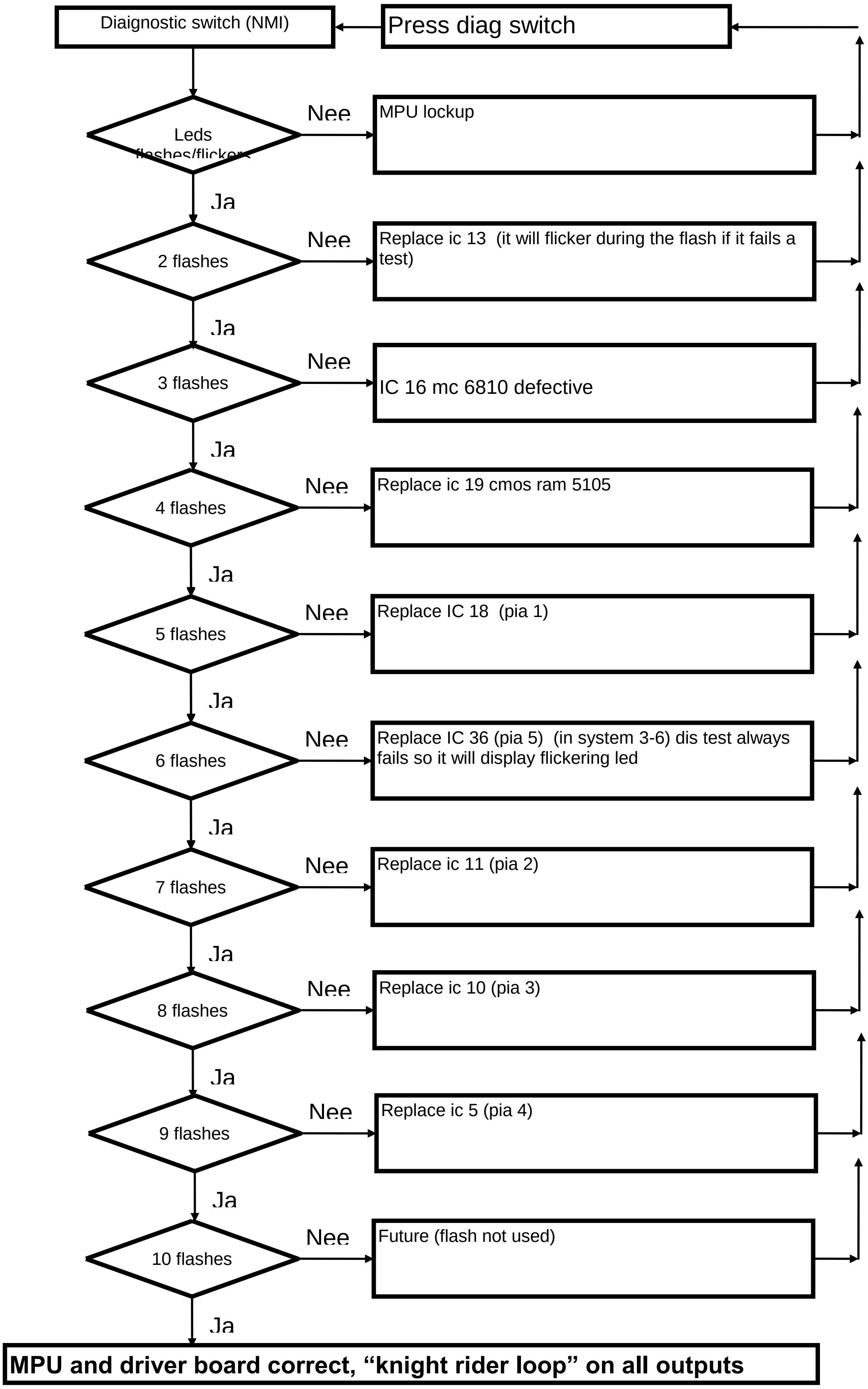


Figure B: Test flow chart diagnostic switch test

4. Testpoint voltages

Testpunt 1 Unregulated +12 Vdc

Testpunt 2 NMI:
5V (3.6 typical) switch not putshed
0 V VDC diagnostic switch pushed

Testpunt 3 Memory Protect:
5V VDC coindoor open
0 V VDC coindoor closed

Testpunt 4 Blanking:
Pulsing 5V boot test rom present

Testpunt 5 IRQ:
4.8 Vdc
Testpunt 6 Bus Φ2:
2.4 Vdc
Tespunt 7 Cmos Ram B+
> 4.3 at power on
> 3.9 at power off
Testpunt 8 Reset:
0 Vdc (t < 1s after power on)
5 Vdc (t > 1s after power on)
Testpunt 9: 5 Vdc.
Testpunt 10 GND
0 Vdc.

5. The source code

Use AS02 compiler .

```
;
;       The W  test rom
;       Created by Andre Boot
;          (c) Copyright 2009
;       Version 2.0 (januari 2009)
;
;       CPU:    Motorola 6800/6802/6808
;
;



ZERO    equ     $F0
ONE     equ     $E1
TWO     equ     $D2
THREE           equ     $C3
FOUR    equ     $B4
FIVE    equ     $A5
SIX     equ     $96
SEVEN           equ     $87
EIGHT   equ     $78
NINE    equ     $69

BOTH    equ     $32
BOTH_LEDS       equ     $32
UPPER_LED       equ     $22
LOWER_LED       equ     $11
NO_LED          equ     $00
NO_LEDS         equ     $00
NONE    equ     $00

waitloop  macro         value
        db      $CE
        dw      value
        db      $09
        db      $26
        db      $FD
           endm

;waitloop macro is:
;               ldx     #value
```

```
;WAIT:
;         dex
;         bne      WAIT


display   macro               getal
          db        $86
          db        getal
          db        $b7
          db        $28
          db        $00
          endm

;display macro is:
;         ldaa     #getal
;         staa     $2800


show      macro     number,delay_val
          display   number
          waitloop            delay_val
          endm

flicker   macro     f,s,w,n
          db        $c6
          db        n
          display   f
          waitloop            w
          display   s
          waitloop            w
          db        $5A
          db        $26
          db        $E7
          endm

;flicker macro contains
;ldab #n
;loopje
;display            f
;waitloop           w
;display            s
;waitloop           w
;decb
;bne  loopje

no_flicker          macro     f1,w1
          display   f1
          waitloop  w1
          endm

          org      $7800
```

```
nmi_entry:
int_entry:
swi_entry:
;restore pia since nmi can be in middle of pia check
;NMI does a system 6 memory check;
;also leds are flashed in stead of bcd display
;but pia port a (display strobe outputs) still same info.

        ldaa    #$38                ;First Set CA2 as output and set CA2 level high,
Disable IRQ by CA1/CB1, select DDRA and DDRB
        staa    $2801               ; write to control reg.
;       lda     #$3F            ;system 6 req. but if you do not touch upper switch its ok.
        lda     #$FF
        staa    $2800                ;Make PA4, PA5 as outputs to drive LEDS; PA0-
PA3 also output. PA6,7 are OUTPUT!!!!!
        ldaa    #$3C                ;Select Data register A
        staa    $2801               ;


        show ZERO,$FFFF
        show NONE,$FFFF




;============Led Blinked 1x


        lds     #$007F       ;initialize stackpointer



check_ram_ic13

nmos_6810_procedure_1:
        ldx     #$0000
        clra
L19B4_1:
        staa    $00,x
        cmpa    $00,x
L19B8_1:
        bne     ic_13defect
        inca
        bne     L19B4_1
        inx
        cpx     #$0080
        bne     L19B4_1

        ldaa            #(ONE | $30)
        staa    $2800
        jsr     flash_led_routine_2
        bra     ic13_good
```

```
ic_13defect
          ldaa    #(ONE | $30)
          staa    $2800
          jsr     flicker_subroutine
ic13_good



;==========led blinked 2x


check_ram_ic16
nmos_6810_procedure_2:
          ldx    #$0080
          clra
L19B4_2:
          staa   $00,x
          cmpa    $00,x
L19B8_2:
          bne    ic_16defect
          inca
          bne    L19B4_2
          inx
          cpx    #$0100
          bne    L19B4_2


          ldaa              #(TWO  | $30)
          staa    $2800
          jsr     flash_led_routine_2
          bra     ic16_good
ic_16defect
          ldaa    #(TWO  | $30)
          staa    $2800
          jsr     flicker_subroutine
ic16_good




;==========led blinked 3x

cmos_5101_procedure:
          ldx    #$0100
L19C8:
          ldab   $00,x
          pshb
          clra
L19CC:
          staa   $00,x
          tab
          eorb   $00,x
```

```
        andb   #$F
L19D3:
        bne    ic_19defect
        adda   #$1
        bne    L19CC
        pulb
        stab   $00,x
        inx
        cpx    #$0200
        bne    L19C8

        ldaa            #(THREE  | $30)
        staa    $2800
        jsr     flash_led_routine_2
        bra     ic19_good
ic_19defect
        ldaa    #(THREE  | $30)
        staa    $2800
        jsr     flicker_subroutine
ic19_good



;==========led blinked 4x

pia_1_procedure:
        ldaa    #(FOUR | $30)  ; four OR $30
        staa    $2800
        ldx      #$55F0              ;and enter wait loop;
        jsr     some_delay


        ldx    #$2803
        jsr    pia_check

        ldaa    #(FOUR & %11001111)  ;
        staa    $2800
        ldx      #$55F0              ;and enter wait loop;
        jsr     some_delay

        ldaa    #$3C; initialize pia 1 again
        staa    $2801



pia_5_procedure:
        ldaa    #(FIVE | $30)  ; four OR $30
        staa    $2800
        ldx      #$55F0              ;and enter wait loop;
        jsr     some_delay

        ldx    #$2103
        jsr     pia_check

        ldaa     #(FIVE & %11001111)  ;
        staa     $2800
        ldx       #$55F0                  ;and enter wait loop;
        jsr       some_delay



pia_2_procedure:
        ldaa     #(SIX| $30)  ; four OR $30
        staa     $2800
        ldx       #$55F0                  ;and enter wait loop;
        jsr       some_delay


        ldx     #$3003
        jsr     pia_check

        ldaa     #(SIX & %11001111)  ;
        staa     $2800

        ldx       #$55F0                  ;and enter wait loop;
        jsr       some_delay




pia_3_procedure:
        ldaa     #(SEVEN| $30)  ; four OR $30
        staa     $2800
        ldx       #$55F0                  ;and enter wait loop;
        jsr       some_delay

        ldx     #$2403
        jsr     pia_check

        ldaa     #(SEVEN & %11001111)  ;
        staa     $2800
        ldx       #$55F0                  ;and enter wait loop;
        jsr       some_delay

pia_4_procedure:
        ldaa     #(EIGHT| $30)  ; four OR $30
        staa     $2800
        ldx       #$55F0                  ;and enter wait loop;
        jsr       some_delay

        ldx     #$2203
        jsr     pia_check
        ldaa    #(EIGHT & %11001111)  ;
        staa    $2800
        ldx      #$55F0               ;and enter wait loop;
        jsr      some_delay




;display_interupt_procedure:
;       ldaa    $2800
;L1A42:
;       ldaa    $2801
;       bpl     L1A42
;       jsr     flash_led_routine

        ldaa  #$0
display74154
        staa $2800
        ldx     #$55F0               ;and enter wait loop
        jsr     some_delay
        inca
        cmpa #$10
        bne  display74154


;pia I $2800
        ldx     #$2800
        ldaa    #$38                        ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa    $03,x           ;control register B
        staa    $01,x           ;control register A
        ldab    #$FF            ;
        stab    $00,x           ;DDRA = all output
        stab    $02,x           ;DDRB = all output

        ldaa    #$3C                        ;Select OUTPUT register
        staa    $03,x           ;control register B
        staa    $01,x           ;control register B

;pia II (port A is input!!) $3003

        ldx     #$3000
        ldaa    #$38                        ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa    $03,x           ;control register B
        staa    $01,x           ;control register A
        ldab    #$00            ;
        stab    $00,x           ;DDRA = all input
        ldab    #$FF
        stab    $02,x           ;DDRB = all output
        ldaa    #$3C                    ;Select OUTPUT register
        staa    $03,x           ;control register B
        staa    $01,x           ;control register B

;pia III $2400
        ldx     #$2400
        ldaa    #$38                    ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa    $03,x           ;control register B
        staa    $01,x           ;control register A
        ldab    #$FF            ;
        stab    $00,x           ;DDRA = all output
        stab    $02,x           ;DDRB = all output

        ldaa    #$3C                    ;Select OUTPUT register
        staa    $03,x           ;control register B
        staa    $01,x           ;control register B

;pia IV $2400
        ldx     #$2200
        ldaa    #$38                    ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa    $03,x           ;control register B
        staa    $01,x           ;control register A
        ldab    #$FF            ;
        stab    $00,x           ;DDRA = all output
        stab    $02,x           ;DDRB = all output

        ldaa    #$3C                    ;Select OUTPUT register
        staa    $03,x           ;control register B
        staa    $01,x           ;control register B

;pia V $2100
        ldx     #$2100
        ldaa    #$38                    ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa    $03,x           ;control register B
        staa    $01,x           ;control register A
        ldab    #$FF            ;
        stab    $00,x           ;DDRA = all output
        stab    $02,x           ;DDRB = all output

        ldaa    #$3C                    ;Select OUTPUT register
        staa    $03,x           ;control register B
        staa    $01,x           ;control register B
        ldaa    #$01
        staa    $0000

end_of_all:


        ldaa  $0000
        staa  $2800
        staa  $2802
        staa  $2100
        staa  $2102

        ldab  $3000
        beq   no_key
        stab  $3002
        bra   skip_piaII

no_key

        staa  $3002     ;pia II only port B
skip_piaII
        staa  $2400
        staa  $2402
        staa  $2200
        staa  $2202

        ldaa   #$3C                    ;Select OUTPUT register CA2 and CB2 high
        staa  $2801
        staa  $2803
        staa  $2101
        staa  $2103
        staa  $3001     ;pia II only port B
        staa  $3003
        staa  $2401
        staa  $2403
        staa  $2201
        staa  $2203



        flicker  NONE,FOUR,$100,128  ; pulsing for Blanking to high state



        ldaa   #$34                    ;Select OUTPUT register CA2 and CB2 low
        staa  $2801
        staa  $2803
        staa  $2101
        staa  $2103
        staa  $3001
        staa  $3003
        staa  $2401
        staa  $2403
        staa  $2201
        staa  $2203

        waitloop $8000
        rol $0000
        bcc skip
        rol $0000
skip
        jmp end_of_all

some_delay:
        dex
        bne     some_delay
        rts

reset:
        sei
        lds     #$007F        ;initialize stackpointer


;only system 7 boards or dont touch input enable switch at level 3 thru 6 boards!!!!
;No subroutines are used since they use stack and that can be fautly (due to
malfunction ram IC's)


        ldaa    #$38                ;First Set CA2 as output and set CA2 level high,
Disable IRQ by CA1/CB1, select DDRA and DDRB
        staa    $2801               ; write to control reg.
;       lda     #$3F
        lda     #$FF
        staa    $2800               ;Make PA4, PA5 as outputs to drive LEDS; PA0-
PA3 also output. PA6,7 are OUTPUT!!!!!
        ldaa    #$3C                ;Select Data register A
        staa    $2801               ;


        show ZERO,$FFFF
        show NONE,$FFFF


;============this was the first flash

nmi_check_ram_ic13
        display ONE

;nmos_6810_procedure_1:
;check ic 13
        ldx     #$0000
        clra
L1a
```

```
        staa    $00,x
        tab
        eorb   $00,x
        lsrb
        lsrb
        lsrb
        lsrb




L2a
        bne    nmi_ic_13_defect
        inca
        bne     L1a
        inx
        cpx     #$0100
        bne     L1a
        jmp     nmi_ic_13_correct

nmi_ic_13_defect
        flicker  NONE,ONE,$1000,8
        jmp         end_nmi_check_ic_13

nmi_ic_13_correct

        no_flicker ONE,$8000

end_nmi_check_ic_13

        display TWO


;this whas the second flash
;check ic 16

        ldx     #$0000
        clra
L11a
        staa    $00,x
        tab
        eorb   $00,x
        aslb
        aslb
        aslb
        aslb

L22a
        bne    nmi_ic_16_defect
```

```
          inca
          bne     L11a
          inx
          cpx     #$0100
          bne     L11a
          jmp     nmi_ic_16_correct

nmi_ic_16_defect
          flicker  NONE,TWO,$1000,8
          jmp          end_nmi_check_ic_16

nmi_ic_16_correct

          no_flicker        TWO,$8000

end_nmi_check_ic_16



;this was the third flash

          display THREE


;this whas the second flash
;ic_19_cmos_5101_procedure:
;
;check ic 19

          ldx     #$0100
          clra
Lic19_11a
          staa    $00,x
          tab
          eorb   $00,x
          aslb
          aslb
          aslb
          aslb

Lic16_22a
          bne    nmi_ic_19_defect
          inca
          bne     Lic19_11a
          inx
          cpx     #$0200
          bne     Lic19_11a
          jmp     nmi_ic_19_correct

nmi_ic_19_defect
          flicker  NONE,THREE,$1000,8
          jmp          end_nmi_check_ic_19
```

```
nmi_ic_19_correct

        no_flicker      THREE,$8000

end_nmi_check_ic_19



;old ic_19_cmos_5101_procedure:
;
;       ldx     #$0100
;LC8:
;       ldab    $00,x
;       pshb
;       clra
;LCC:
;       staa    $00,x
;       tab
;       eorb    $00,x
;       andb    #$F
;LD3:
;       bne     ic_19_defect
;       adda    #$1
;       bne     LCC
;       pulb
;       stab    $00,x
;       inx
;       cpx     #$0200
;       bne     LC8
;       jmp     ic_19_correct
;
;ic_19_defect
;       flicker NONE,THREE,$1000,8
;       jmp         end_check_ic_19
;
;ic_19_correct
;
;       no_flicker      THREE,$8000
;
;end_check_ic_19

;this was the fourth flash



        display FOUR


;check pia I, ic 18  only ddrb is checked!
        ldx     #$2803
        ldaa   #$38                ;Select Data direction register B, and DDRB, CA2.
CB2 high
        staa   $00,x
        dex
        dex
        staa   $00,x
        inx
        inx
        dex                  ; ddra is now selected


        clra               ; make reg a zero
LA31:
        staa   $00,x        ;store a into ddrb
        ldab   #$03         ;load b with 3, and enter delay loop
LA35:
        decb
        bne    LA35
        cmpa    $00,x        ;read ddrb and compare to a
LA3A:
        bne    ic_18_defect         ;if not equal stop program
        inca               ;if correct increase do test until whole ddra is checked
        bne    LA31
        jmp    ic_18_correct
ic_18_defect
        ldaa   #$3C                ;Select OUTPUT register A
        staa   $2801               ;
        flicker  NONE,FOUR,$1000,16
        jmp        end_check_ic_18

ic_18_correct
        ldaa   #$3C                ;Select OUTPUT register A
        staa   $2801               ;

        no_flicker        FOUR,$FFFF

end_check_ic_18

        display FIVE

;check pia V, ic 36  only ddrb is checked!
        ldx    #$2103

        ldaa   #$38                ;Select Data direction register B, and DDRB, CA2.
CB2 high
        staa   $00,x
        dex
        dex
        staa   $00,x
        inx
        inx
        dex                  ; ddra is now selected
        clra                ; make reg a zero
LB31:
        staa   $00,x        ;store a into ddrb
        ldab   #$03         ;load b with 3, and enter delay loop
LB35:
        decb
        bne    LB35
        cmpa   $00,x        ;read ddrb and compare to a
LB3A:
        bne   ic_36_defect        ;if not equal stop program
        inca                ;if correct increase do test until whole ddra is checked
        bne    LB31
        jmp    ic_36_correct
ic_36_defect
        ldaa   #$3C              ;Select OUTPUT register A
        staa   $2801             ;
        flicker  NONE,FIVE,$1000,16
        jmp       end_check_ic_36

ic_36_correct
        ldaa   #$3C              ;Select OUTPUT register A
        staa   $2801             ;

        no_flicker      FIVE,$FFFF

end_check_ic_36

        display SIX


;check pia II, ic 11 but only ddrb is checked!
        ldx    #$3003

        ldaa   #$38              ;Select Data direction register B, and DDRB, CA2.
CB2 high
        staa   $00,x
        dex
        dex
        staa   $00,x
        inx
        inx
        dex                  ; ddra is now selected


        clra                ; make reg a zero
LC31:
        staa   $00,x        ;store a into ddrb
        ldab   #$03         ;load b with 3, and enter delay loop
LC35:
        decb
```

```
          bne     LC35
          cmpa    $00,x          ;read ddrb and compare to a
LC3A:
          bne    ic_11_defect          ;if not equal stop program
          inca                   ;if correct increase do test until whole ddra is checked
          bne     LC31
          jmp     ic_11_correct
ic_11_defect
              ;
          flicker  NONE,SIX,$1000,16
          jmp         end_check_ic_11

ic_11_correct


          no_flicker         SIX,$FFFF

end_check_ic_11


          display SEVEN

;check pia III, ic 10 but only ddrb is checked!
          ldx     #$2403

          ldaa    #$38                  ;Select Data direction register B, and DDRB, CA2.
CB2 high
          staa    $00,x
          dex
          dex
          staa    $00,x
          inx
          inx
          dex                     ; ddra is now selected


          clra                   ; make reg a zero
Lic10C31:
          staa    $00,x          ;store a into ddrb
          ldab    #$03           ;load b with 3, and enter delay loop
Lic10C35:
          decb
          bne     Lic10C35
          cmpa    $00,x          ;read ddrb and compare to a
Lic10C3A:
          bne    ic_10_defect          ;if not equal stop program
          inca                   ;if correct increase do test until whole ddra is checked
          bne     Lic10C31
          jmp     ic_10_correct
ic_10_defect
              ;
          flicker  NONE,SEVEN,$1000,16
```

```
        jmp       end_check_ic_10

ic_10_correct


        no_flicker      SEVEN,$FFFF

end_check_ic_10

        display EIGHT

;check pia IV, ic 5 but only ddrb is checked!
        ldx    #$2203

        ldaa   #$38               ;Select Data direction register B, and DDRB, CA2.
CB2 high
        staa   $00,x
        dex
        dex
        staa   $00,x
        inx
        inx
        dex                  ; ddra is now selected


        clra                ; make reg a zero
Lic5C31:
        staa   $00,x         ;store a into ddrb
        ldab   #$03          ;load b with 3, and enter delay loop
Lic5C35:
        decb
        bne    Lic5C35
        cmpa   $00,x         ;read ddrb and compare to a
Lic5C3A:
        bne   ic_5_defect         ;if not equal stop program
        inca                ;if correct increase do test until whole ddra is checked
        bne    Lic5C31
        jmp    ic_5_correct
ic_5_defect
           ;
        flicker  NONE,EIGHT,$1000,16
        jmp       end_check_ic_5

ic_5_correct


        no_flicker      EIGHT,$FFFF

end_check_ic_5
```

```
;initialize pia's (after reset they are inputs)


;pia I $2800
        ldx    #$2800
        ldaa   #$38                  ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa   $03,x         ;control register B
        staa   $01,x         ;control register A
        ldab   #$FF          ;
        stab   $00,x         ;DDRA = all output
        stab   $02,x         ;DDRB = all output

        ldaa   #$3C                  ;Select OUTPUT register
        staa   $03,x         ;control register B
        staa   $01,x         ;control register B

;pia II (port A is input!!) $3003

        ldx    #$3000
        ldaa   #$38                  ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa   $03,x         ;control register B
        staa   $01,x         ;control register A
        ldab   #$00          ;
        stab   $00,x         ;DDRA = all input
        ldab   #$FF
        stab   $02,x         ;DDRB = all output

        ldaa   #$3C                  ;Select OUTPUT register
        staa   $03,x         ;control register B
        staa   $01,x         ;control register B

;pia III $2400
        ldx    #$2400
        ldaa   #$38                  ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa   $03,x         ;control register B
        staa   $01,x         ;control register A
        ldab   #$FF          ;
        stab   $00,x         ;DDRA = all output
        stab   $02,x         ;DDRB = all output

        ldaa   #$3C                  ;Select OUTPUT register
        staa   $03,x         ;control register B
        staa   $01,x         ;control register B

;pia IV $2400
        ldx    #$2200
        ldaa   #$38                 ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa   $03,x         ;control register B
        staa   $01,x         ;control register A
        ldab   #$FF          ;
        stab   $00,x         ;DDRA = all output
        stab   $02,x         ;DDRB = all output

        ldaa   #$3C                 ;Select OUTPUT register
        staa   $03,x         ;control register B
        staa   $01,x         ;control register B

;pia V $2100
        ldx   #$2100
        ldaa   #$38                 ;Select Data direction register B, and DDRB,
CA2. CB2 high

        staa   $03,x         ;control register B
        staa   $01,x         ;control register A
        ldab   #$FF          ;
        stab   $00,x         ;DDRA = all output
        stab   $02,x         ;DDRB = all output

        ldaa   #$3C                 ;Select OUTPUT register
        staa   $03,x         ;control register B
        staa   $01,x         ;control register B


start_flip_outputs

        ldaa  #$ff
        staa  $2800
        staa  $2802
        staa  $2100
        staa  $2102
        staa  $3002    ;pia II only port B
        staa  $2400
        staa  $2402
        staa  $2200
        staa  $2202

        ldaa   #$3C                 ;Select OUTPUT register CA2 and CB2 high
        staa  $2801
        staa  $2803
        staa  $2101
        staa  $2103
        staa  $3001    ;pia II only port B
        staa  $3003
        staa  $2401
        staa  $2403
        staa  $2201
        staa  $2203



        flicker   NONE,FOUR,$100,255  ; pulsing for Blanking to high state
        flicker   NONE,FOUR,$100,255  ; pulsing for Blanking to high state
        flicker   NONE,FOUR,$100,255  ; pulsing for Blanking to high state


        lda  #$00
        sta  $2800
        sta  $2802
        sta  $2100
        sta  $2102
        sta  $3002       ;pia II only port B
        sta  $2400
        sta  $2402
        sta  $2200
        sta  $2202


        ldaa   #$34                         ;Select OUTPUT register CA2 and CB2 low
        staa  $2801
        staa  $2803
        staa  $2101
        staa  $2103
        staa  $3001
        staa  $3003
        staa  $2401
        staa  $2403
        staa  $2201
        staa  $2203

        waitloop $FFFF
        waitloop $FFFF
        waitloop $FFFF

        jmp      start_flip_outputs  ; blanking to zero







        jmp      nmi_entry
```

```
flash_led_routine_2:


          ldaa     $2800
          oraa    #$30
          staa    $2800                  ;
          ldx      #$FFFF                 ;and enter waitloop

L1:
          dex
          bne      L1
          ldaa     $2800
          anda    #%11001111
            staa             $2800
          ldx       #$FFFF                 ;and enter wait loop

L2:
          dex
          bne     L2




          rts                           ;exit routine
;

;flash_led_routine_both:

;         ldaa    #$3C                  ;Select Data register A
;         staa    $2801                  ;And make CA2 high
;         ldaa    #$00                  ;Turn both leds off
;         staa    $2800                  ;
;         ldx     #$FFFF                 ;and enter waitloop

;L1d:
;         dex
;         bne     L1d



;         ldaa    #$30                   ;Turn both leds on
;         staa    $2800
;         ldx     #$45F0                 ;and enter wait loop

;L2d:
```

```
;       dex
;       bne     L2d

;       ldaa    #$38            ;select
;       staa    $2801           ;DDRA


;       rts                     ;exit routine


;flash_led_routine_upper:


;       ldaa    #$3C            ;Select Data register A
;       staa    $2801           ;And make CA2 high
;       ldaa    #$00            ;Turn both leds off
;       staa    $2800           ;
;       ldx     #$FFFF          ;and enter waitloop

;L1e:
;       dex
;       bne     L1e



;       ldaa    #$10            ;Turn upper led on
;       staa    $2800
;       ldx     #$45F0          ;and enter wait loop

;L2e:
;       dex
;       bne     L2e

;       ldaa    #$38            ;select
;       staa    $2801           ;DDRA


;       rts                     ;exit routine


;flash_led_routine_lower:


;       ldaa    #$3C            ;Select Data register A
;       staa    $2801           ;And make CA2 high
;       ldaa    #$00            ;Turn both leds off
;       staa    $2800           ;
;       ldx     #$FFFF          ;and enter waitloop

;L1f:
;       dex
;       bne     L1f
;         ldaa    #$20                  ;Turn lower led on
;         staa    $2800
;         ldx    #$45F0                 ;and enter wait loop

;L2f:
;         dex
;         bne     L2f

;         ldaa    #$38                  ;select
;         staa    $2801                 ;DDRA


;         rts                       ;exit routine


;flash_led_routine_counter:



;         ldaa    #$30                  ;Turn both leds off
;         staa    $2800                 ;
;         ldx    #$FFFF                 ;and enter waitloop

;L1g:
;         dex
;         bne     L1g
;         ldaa    #$0F

;L3g

;         inca                      ;Turn lower led on
;         staa    $2800
;         ldx    #$45F0                 ;and enter wait loop

;L2g:
;         dex
;         bne     L2g

;         jmp     L3g
;         ldaa    #$38                  ;select
;         staa    $2801                 ;DDRA


;         rts                       ;exit routine
```

```
flicker_subroutine

          ldab    #$10
loopje
          staa    $2800
          waitloop  $1000
          clr     $2800
          waitloop  $1000
          decb
          bne     loopje
          staa    $2800
          rts




;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;original subroutines




flash_led_routine:

          ldaa   #$38              ;First Set CA2 as output
          staa   $2801              ;and set CA2 level high
          lda    #$30
          staa   $2800              ;Make PA4, PA5 as outputs to drive LEDS
          ldaa   #$3C              ;Select Data register A
          staa   $2801              ;And make CA2 high


          ldaa   #$30              ;Turn both leds on
          staa   $2800              ;
          ldx    #$7530             ;and enter waitloop

L19FC:
          dex
          bne    L19FC
turn_led_off:


          ldaa   #$0              ;Turn both leds off
          staa   $2800
          ldx    #$55F0             ;and enter wait loop
```

```
L1A06:
          dex
          bne     L1A06

          ldaa    #$38                 ;select
          staa    $2801                ;DDRA


          rts                          ;exit routine
;

;
pia_check:
L1A0B:
          bsr     L1A0D
L1A0D:
          ldaa    #$31
          bsr     L1A26          ;CRB check routine
          ldaa    #$39
          bsr     L1A26          ;CRB check routine
          dex
          bsr     L1A30          ;DDRb check routine
          inx
          ldaa    #$35
          bsr     L1A26          ;CRB
          ldaa    #$3D
          bsr     L1A26
          dex
;       bsr     L1A30
          dex
          rts
;
L1A26:
          staa    $00,x          ;check CRB
          ldab    $00,x          ;
          andb    #$3F
          cba
L1A2D:
          bne    pia_not_good        ;
          rts                  ;end check CRB


pia_not_good
          ldaa     $2800
          ora      #$30
          jsr      flicker_subroutine
          rts
```

```
;
L1A30:
        clra                ; make reg a zero
L1A31:
        staa   $00,x        ;store a into ddrb
        ldab   #$03         ;load b with 3, and enter delay loop
L1A35:
        decb
        bne    L1A35
        cmpa    $00,x       ;read ddrb and compare to a
L1A3A:
        bne    pia_not_good      ;if not equal flicker error
        inca                ;if correct increase do test until whole ddra is checked
        bne    L1A31
        rts
```

```
        org    $7FF8        ;define all the vecors
irq_vector:
        dw     int_entry
swi_vector:
        dw     swi_entry
nmi_vector:
        dw     nmi_entry
res_vector:
        dw     reset
```