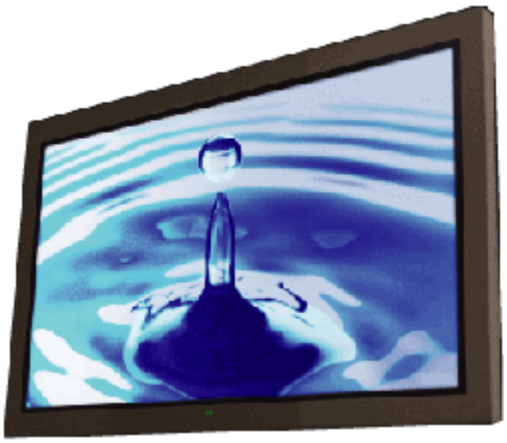
Flash Innovation 2003 Design Contest

CIRCUIT CELLAR®



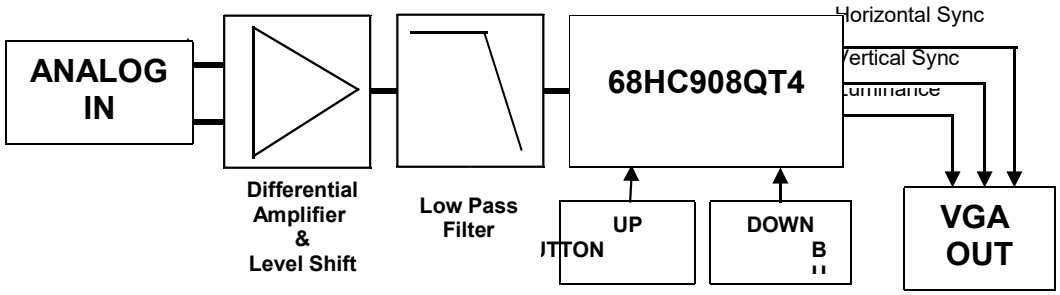
VGA SIGNAL PROBE

BY LINDSAY MEEK

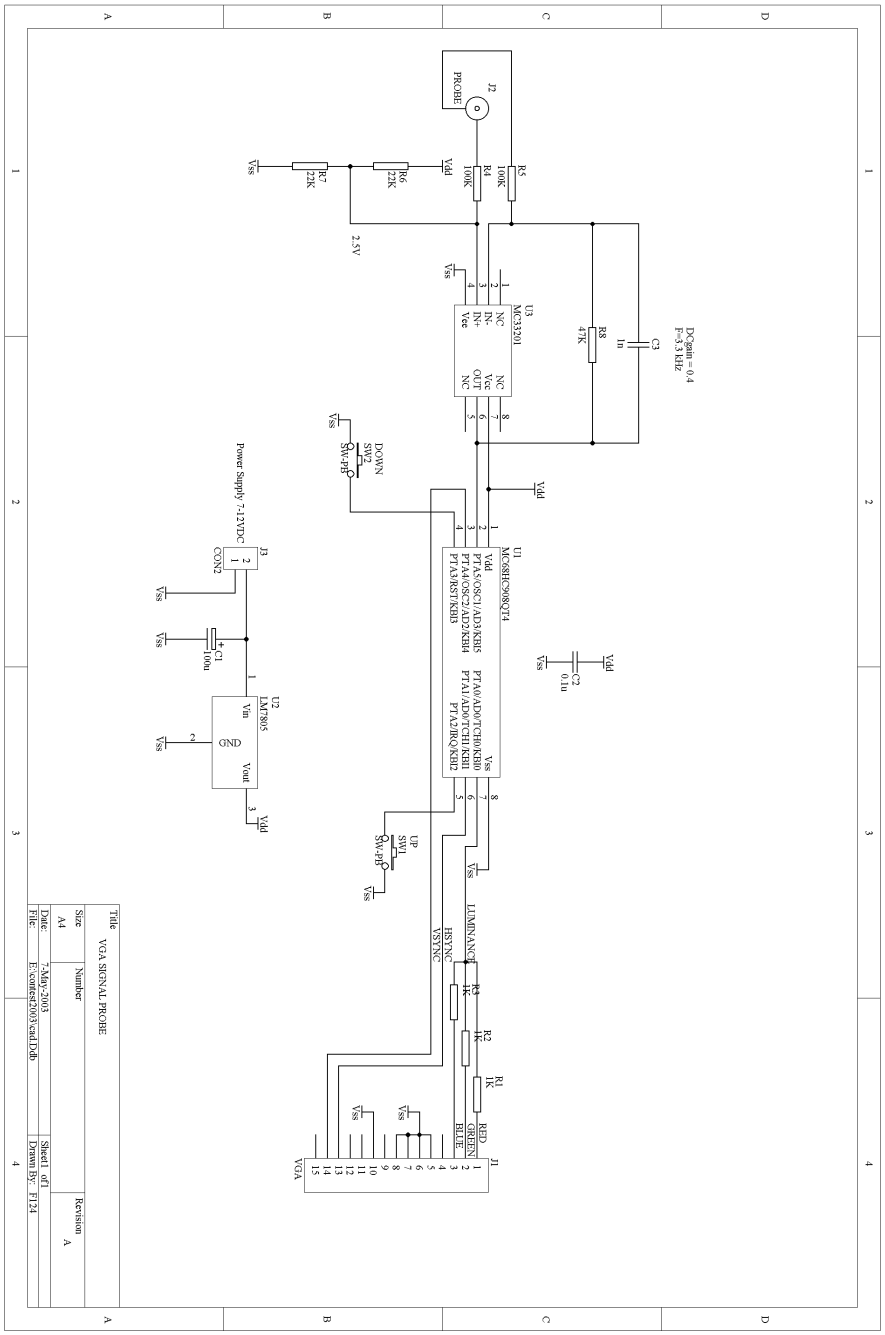
1. ABSTRACT

This project utilises a 68HC908QT4 to digitise an analog signal, and generate a corresponding VGA video signal representing the time sequence of the samples. This allows a simple audio-bandwidth CRO to be implemented using two 8-pin ICs. The device includes two buttons for toggling the acquisition mode between free running, and digital storage. In free running mode, the acquisition is synchronised to the horizontal display frequency. In digital storage mode, the buttons also act to adjust the time base, triggering voltage level and edge polarity.

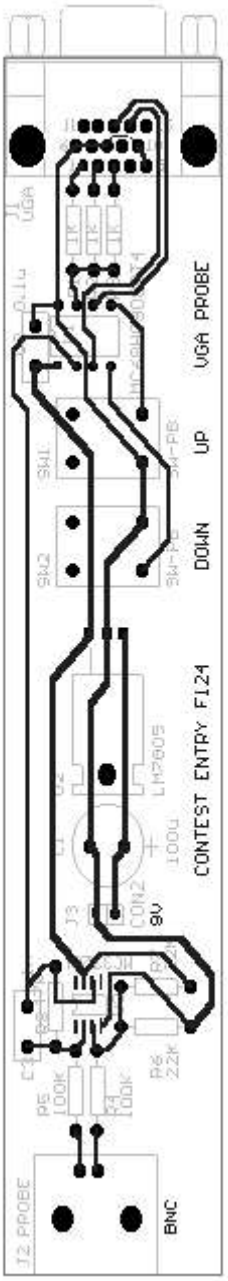
2. BLOCK DIAGRAM



3. SCHEMATIC


J2
PROBE
R4
100K
R5
100K
R7
22K
R6
22K
Vdd
Vss
2.5V
U3
MC33201
R8
47K
C3
1n
DC@min = 0.4
F=3.3 kHz
C2
0.1u
DOWN
SW2
SW-PB
UP
SW1
SW-PB
U1
MC9SHC908QT4
PTA5/OSC1/AD3/KBI5
PTA4/OSC2/AD2/KBI4
PTA3/RST/KBI3
PTA0/ADO/TCH0/KBI0
PTA1/ADO/TCH1/KBI1
PTA2/IRQ/KBI2
Power Supply 7-12VDC
J3
CON2
C1
100u
U2
LM7805
Vin
GND
Vout
LUMINANCE
HSYNC
VSYNC
R3
1K
R2
1K
R1
1K
RED
GREEN
BLUE
J1
VGA
Title
VGA SIGNAL PROBE
Size
A4
Number
Revision
A
Date:
7-May-2003
Sheet of
1
File:
E:\cortex\2003\vga\.Ddb
Drawn By:
FT24

4. PCB LAYOUT



5. VGA SIGNAL GENERATION

Theory Of Operation

The basic VGA signal consists of a set of horizontal scan lines, each 31.77 μs long, which commence with a 3.77 μs TTL pulse known as the horizontal sync.

The horizontal scan line timing is generated by the QT4's timer module, with the timer modulo value set to reset the timer at the end of each scan line, which works out to be exactly 101 timer counts with a 3.2 MHz internal clock.

The horizontal sync pulse is generated using timer channel 1 in unbuffered output compare mode, with the channel controlling the horizontal sync pin directly. The compare value is set to the length of the horizontal sync, and deasserts sync by driving the pin high. The toggle-on-overflow feature is also used to assert sync when the timer resets, hence no CPU intervention is required to generate this signal. The output compare status bit is polled by the CPU, to detect the start of a new scan line, and synchronise the ADC sampling.

This sync pulse is followed by a small timing gap known as the 'front porch', after which the colour video image is sent in its red, green and blue components. These components are encoded as analog signals, each with a peak amplitude of 0.7V, driving into a 75 ohm load.

For simplicity, the circuit encodes the same luminance level on red, green and blue, generating a white pixel. The TTL output from the processor drives into a 1k resistor on each component, resulting in an effective analog voltage of 5V*75Ω/(1075Ω) = 0.35V, or 50% of the maximum amplitude.

The luminance signal generation is achieved using timer channel 0, also in unbuffered output compare mode, with the channel controlling the luminance pin directly. The compare value is set to the position of the horizontal pixel, and asserts the luminance signal by driving the pin high. The toggle-on-overflow feature is used to deassert the luminance signal at the end of the scan line. This effectively renders an image of horizontal bars, extending from a pre-programmed horizontal position extending to the right of the screen.

In free running sample mode, the luminance output compare register is reloaded at the start of each scan line with a sample from the ADC. The sample is scaled such that the 8-bit value is converted to a 6-bit value, and offset to active video region. This mode uses a 640 x 480 VGA image to maximise the number of visible lines.

In digital storage mode, the luminance output compare register is reloaded every third line, as there is a limited amount of sample storage RAM. The remaining two lines are used to operate the triggering and subsampling algorithms. The same 8-bit to 6-bit scaling algorithm is applied to the samples from the ADC. This mode uses a 640 x 400 VGA image, with 350 lines used to display storage memory.

At the end of the visible image, the QT4 stops luminance and horizontal sync generation, and waits for a blanking period called

the ‘front porch’. This is followed by a 62 µs TTL pulse called the vertical sync. The polarity of the vertical sync is reversed depending on whether the QT4 is generating a 640 x 480 or 640 x 400 image.

The vertical sync is followed by another blanking period called the ‘back porch’, after which a new image commences. The QT4 takes advantage of this blanking period to poll the up and down buttons and adjust the triggering level, timebase, or operation mode.

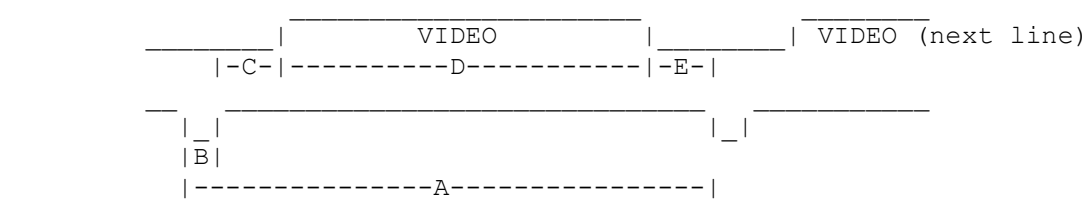
Pressing the up and down buttons simultaneously toggles between free running and digital storage mode. Whilst in digital storage mode, pressing both buttons will toggle between adjusting the time base and the trigger voltage.

When adjusting the trigger voltage, pressing up once will set the trigger polarity to rising edge triggered, and pressing down once will set the trigger polarity to falling edge triggered.

Detailed Horizonal Timing

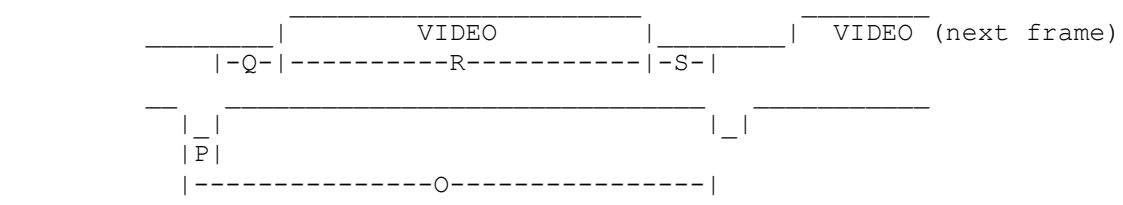
Source http://www.epanorama.net/documents/pc/vga_timing.html

Horizonal Dots	640	640	
Vertical Scan Lines	400	480	
A (µs)	31.77	31.77	Scanline time
B (µs)	3.77	3.77	Sync pulse length
C (µs)	1.89	1.89	Back porch
D (µs)	25.17	25.17	Active video time
E (µs)	0.94	0.94	Front porch

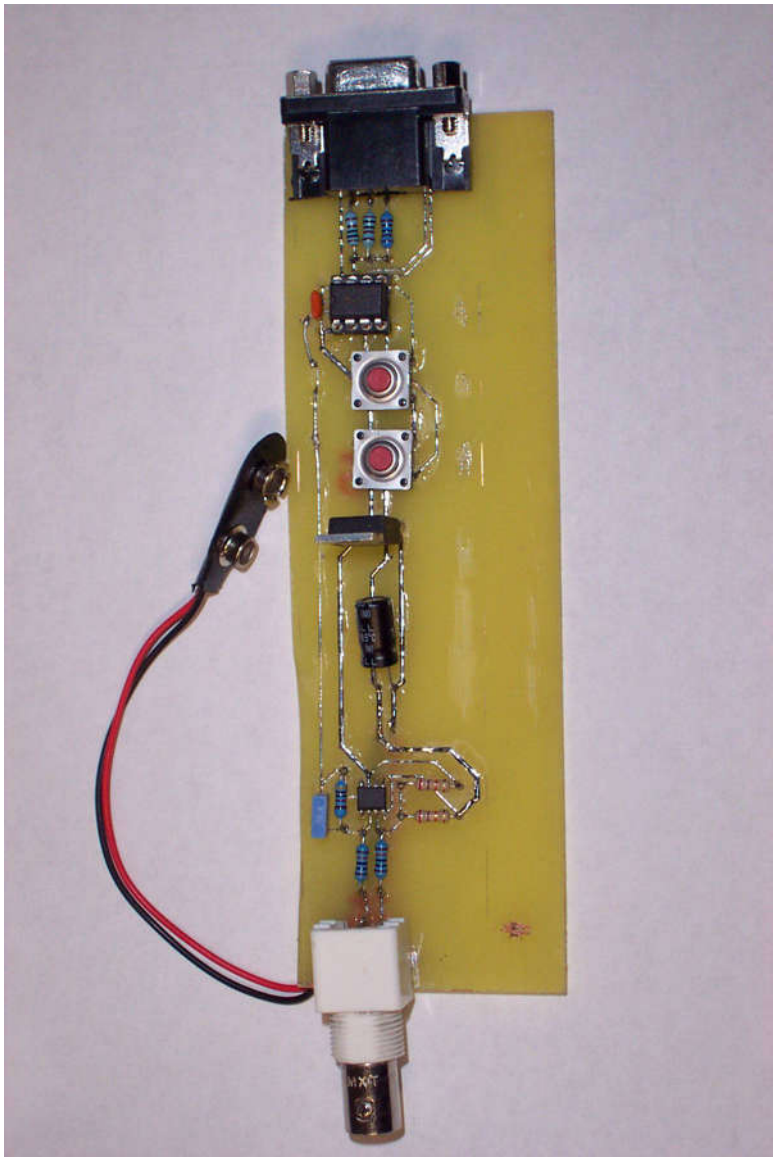


Detailed Vertical Timing

Horizonal Dots	640	640	
Vertical Scan Lines	400	480	
Vert. Sync Polarity	POS	NEG	
Vertical Frequency	70Hz	60Hz	
O (ms)	14.27	16.68	Total frame time
P (ms)	0.06	0.06	Sync length
Q (ms)	1.08	1.02	Back porch
R (ms)	12.72	15.25	Active video time
S (ms)	0.41	0.35	Front porch



6. PROJECT PICTURE



7. SUMMARY SPECIFICATIONS

INPUT SECTION	
SUPPLY VOLTAGE	7-12 VDC
SUPPLY CURRENT	15 mA
PROBE VOLTAGE	DIFFERENTIAL ± 5.0V
PROBE IMPEDANCE	100 KΩ
PROBE 3dB FREQUENCY	3.3 KHz
OUTPUT SECTION	
LUMINANCE	0.35 V
HORIZONTAL SYNC	TTL ACTIVE LOW
VERTICAL SYNC	TTL ACTIVE HIGH & LOW
VGA TIMING	640 x 480 (FREE RUN) 640 x 400 (STORAGE MODE)

8. SOURCE LISTING

```
;
;Motorola/Circuit Cellar
;
;Flash Innovation Contest 2003
;
;Contest Entry F124
;
;VGA Signal Probe
;

               org $FFC0

trim_val:  DC.B 105     ;FLASH trim default value

;
;RAM variable definitions
;

BUFSIZ:                equ            350/3   ;size of sample storage buffer
STACKSIZ:              equ            8       ;size of stack

MY_RAM                 SECTION SHORT

LUM_TIMER:             ds.b    1              ;luminance timer latch
FLAGS:                 ds.b    1              ;general flag bits
TMR_ACC:               ds.b    1              ;subsampling fractional accumulator
SAMPLE_IDX:            ds.b    1              ;sample buffer index
STORAGE:               ds.b    BUFSIZ-2       ;sample buffer .. 116 entries
TRIGGER:               ds.b    1              ;trigger sample level
TMR_STEP:              ds.b    1              ;subsampling fractional step (timebase)

MY_ROM                 SECTION


;
;QT4 relevant hardware definitions
;
_PTA:          equ            0
_DDRA:         equ            4
_PTAPUE:       equ            0bh
_KBSCR:        equ            1ah
_KBIER:        equ            1bh
_TSC:          equ            20h
_TCNTH:        equ            21h
_TCNTL:        equ            22h
_TMODH:        equ            23h
_TMODL:        equ            24h
_TSC0:         equ            25h
_TCH0H:        equ            26h
_TCH0L:        equ            27h
_TSC1:         equ            28h
_TCH1H:        equ            29h
_TCH1L:        equ            2ah
_OSCTRIM:      equ            38h
_ADSCR:        equ            3ch
_ADR:          equ            3eh
_ADICLK:       equ            3fh
_COPCTL:       equ            0ffffh

;
;Port A pin allocations
;
LUM_PIN:       equ            0              ;Luminance output (0=black 1=grey)
HSYNC_PIN:     equ            1              ;Horizontal sync output (0=active)
UP_PIN:        equ            2              ;Up pin (0=pressed)
DOWN_PIN:      equ            3              ;Down pin (0=pressed)
VSYNC_PIN:     equ            4              ;Vertical sync output (0=active)
```

```
;
;Flag bits
;
RISING:         equ             0 ;set when rising edge triggering
ADJTRIG:        equ             1 ;set when trigger is being adjusted, else timebase
FREERUN:        equ             2 ;set when free run active

;
;VGA Horizontal timing constants
;
HORIZ_WIDTH:            equ     101                             ;31.5625 us    @ 3.2 MHz
SYNC_WIDTH:             equ     12                              ;3.75 us       @ 3.2 MHz
BACKPORCH_WIDTH:        equ     6                               ;1.875 us      @ 3.2 MHz
FRONTPORCH_WIDTH:       equ     3                               ;0.9375 us     @ 3.2 MHz
ACTIVE_START:   equ     SYNC_WIDTH+BACKPORCH_WIDTH              ;18 image left margin
ACTIVE_END:     equ     HORIZ_WIDTH-FRONTPORCH_WIDTH            ;98 image right margin
ACTIVE_WIDTH:   equ     ACTIVE_END-ACTIVE_START                 ;80 width of image
RENDER_START:   equ     ACTIVE_START+(ACTIVE_WIDTH-64)/2        ;19 rendering start

                xdef            _Reset

_Reset:
                lda             trim_val        ;load the TRIM value
                sta             _OSCTRIM

                sei
                ldhx            #LUM_TIMER      ;initialise stack pointer to end of RAM
                txs
                jsr             InitIO          ;crank up the I/O ports
                jsr             InitADC         ;crank up the ADC
                jsr             InitTimers      ;crank up the VGA horizontal timers
                cli
                bra             DoFreerun       ;start free running sampler by default

;
;Initialise ADC subsystem
;
InitADC:
                lda             #64             ;ADC clock= bus clock/4=0.8 MHz @ 3.2 MHz
                sta             _ADICLK
                lda             #32+1+2         ;turn on ADC, continuous conversion on
                                                ;PTA5, no IRQ
                sta             _ADSCR
                rts

;
;Initialise I/O subsystem
;
InitIO:
                lda             #(1<<LUM_PIN)|(1<<HSYNC_PIN)|(1<<VSYNC_PIN)
                ;LUM, HSYNC, VSYNC = output, ADC, DOWN, UP = input
                sta             _DDRA
                lda             #2+16+4+8
                sta             _PTA            ;deassert HSYNC and VSYNC, LUM=0V
                lda             #(1<<UP_PIN)|(1<<DOWN_PIN)
                sta             _PTAPUE         ;enable pullups for input switches

                bset            1,_KBSCR        ;mask keyboard interrupts
                bset            UP_PIN,_KBIER
                bset            DOWN_PIN,_KBIER ;enable up and down keys
                bset            0,_KBSCR        ;keyboard is edge and level triggered
                bset            2,_KBSCR        ;ack any pending keyboard interrupts

                rts

;
;Initialise VGA timers and system mode
;
InitTimers:

                lda             #16+32
                sta             _TSC            ;stop timer & reset it

                clr             _TMODH
                lda             #HORIZ_WIDTH
                sta             _TMODL          ;timer overflows every horizontal line
```

```
                lda             #16+8           ;unbuffered operation, no IRQ
                                                ;output compare, LUM clear on match
                sta             _TSC0
                lda             #16+8+4+2       ;unbuffered operation, no IRQ
                                                ;output compare, HSYNC set on match,
                                                ;toggle on overflow
                sta             _TSC1

                clr             _TCH1H
                lda             #SYNC_WIDTH
                sta             _TCH1L          ;HSYNC width
                clr             _TCH0H
                lda             #ACTIVE_START+(ACTIVE_WIDTH)/2
                sta             _TCH0L          ;LUM trigger initially at mid point
                sta             LUM_TIMER

                bclr            7,_TSC0         ;clear LUM output compare flag
                bclr            7,_TSC1         ;clear HSYNC output compare flag

                clr             FLAGS

                bclr            RISING,FLAGS    ;falling edge by default
                bset            FREERUN,FLAGS   ;free running by default

                bclr            LUM_PIN,_PTA    ;LUM off (low)
                bset            HSYNC_PIN,_PTA  ;HSYNC inactive high
                bset            VSYNC_PIN,_PTA  ;VSYNC inactive high

                lda             #128+64
                sta             TRIGGER         ;set trigger level to 50% full scale

                lda             #BUFSIZ-3
                sta             SAMPLE_IDX      ;start sampling

                clr             TMR_ACC         ;reset sample timer accumulator
                lda             #255
                sta             TMR_STEP        ;maximum sample rate

                lda             #64
                sta             _TSC            ;timebase = bus clock, start,
                                                ;overflow IRQ

                rts

;
;Timer overflow IRQ. This occurs at the start of every horizontal line
;

                xdef _TIMER_IRQ

_TIMER_IRQ:
                                                ;
                                                ;9 latency
                                                ;
                lda             LUM_TIMER       ;3
                sta             _TCH0L          ;3 load luminance timer with new value
                bclr            7,_TSC          ;4 ack overflow IRQ
                rti                             ;7

;
;Process keypresses macro
;
;In trigger adjust mode:
;
;UP             increase trigger level, set trigger direction to rising level
;DOWN           decrease trigger level, set trigger direction to falling level
;
;In timebase adjust mode:
;
;UP             increase sampling frequency
;DOWN           decrease sampling frequency
;
;UP+DOWN        toggles between trigger adjust, timebase adjust and freerun
;
;Total execution time < 47 cycles
;
Keypress:       MACRO
```

```
                brclr           3,_KBSCR,\@KeypressDone       ;5 check for a keypress
flag
                bset            2,_KBSCR                      ;4 ack keyboard interrupt

                lda             _PTA                          ;3 fetch port state
                and             #(1<<UP_PIN)|(1<<DOWN_PIN)    ;2 mask pins
                beq             \@Toggle                      ;3 both held = toggle mode
                and             #(1<<DOWN_PIN)                ;2 down pin active?
                beq             \@GoUp                        ;3 no...must be up
                bra             \@GoDown                      ;3 yes..must be down

\@Toggle:
                brclr           FREERUN,FLAGS,\@NotFreerun    ;5
                bclr            FREERUN,FLAGS                 ;4
                bset            ADJTRIG,FLAGS                 ;4 free run -> adj trigger
                bra             \@KeypressDone                ;3

\@NotFreerun:
                brclr           ADJTRIG,FLAGS,\@GoFreerun     ;5
                bclr            ADJTRIG,FLAGS                 ;4 adj trig ->
                                                              ;  adj timebase
                bra             \@KeypressDone                ;3

\@GoFreerun:
                bset            FREERUN,FLAGS                 ;4 adj timebase ->
                                                              ;  free run
                bra             \@KeypressDone                ;3

\@GoDown:
                brset           FREERUN,FLAGS,\@KeypressDone  ;5 only adjust if DSO mode

                brset           ADJTRIG,FLAGS,\@DecTrig       ;5
                tst             TMR_STEP                      ;3
                beq             \@KeypressDone                ;3
                dec             TMR_STEP                      ;4
                bra             \@KeypressDone                ;3

\@DecTrig:
                bclr            RISING,FLAGS                  ;4 falling edge triggered
                tst             TRIGGER                       ;3
                beq             \@KeypressDone                ;3
                dec             TRIGGER                       ;4
                bra             \@KeypressDone                ;3

\@GoUp:
                brset           FREERUN,FLAGS,\@KeypressDone  ;5 only adjust if DSO mode

                brset           ADJTRIG,FLAGS,\@IncTrig       ;5
                inc             TMR_STEP                      ;4
                bne             \@KeypressDone                ;3
                dec             TMR_STEP                      ;4
                bra             \@KeypressDone                ;3

\@IncTrig:
                bset            RISING,FLAGS                  ;4 rising edge triggered
                inc             TRIGGER                       ;4
                bne             \@KeypressDone                ;3
                dec             TRIGGER                       ;4
                bra             \@KeypressDone                ;3

\@KeypressDone:
                ENDM

;
;Wait for start of horizontal line, and clear flag
;
HWAIT:  MACRO
\@WtHsync:
                brclr           7,_TSC1,\@WtHsync             ;5 wait for HSYNC compare
                bclr            7,_TSC1                       ;4 ack it
                ENDM


;
;Untriggered unstored 31kHz sample + display
;
```

```
;Uses 480 line VGA mode
;
DoFreerun:

                sta             _COPCTL                 ;watchdog reset
                bset            VSYNC_PIN,_PTA          ;deassert VSYNC

                lda             #16+8+4+2               ;unbuffered operation, no IRQ
                                                        ;output compare, LUM set on match
                                                        ;clear on overflow
                sta             _TSC0

                ;generate 480 VGA lines

                ldx             #480/3                  ;2

LineLoop1:
                lda             _ADR                    ;3 read most recent ADC result
                lsra                                    ;1
                lsra                                    ;1 convert to 6-bit
                add             #RENDER_START           ;2 offset
                sta             LUM_TIMER               ;3 load into LUM timer
                HWAIT
                lda             _ADR                    ;3 read most recent ADC result
                lsra                                    ;1
                lsra                                    ;1 convert to 6-bit
                add             #RENDER_START           ;2 offset
                sta             LUM_TIMER               ;3 load into LUM timer
                HWAIT
                lda             _ADR                    ;3 read most recent ADC result
                lsra                                    ;1
                lsra                                    ;1 convert to 6-bit
                add             #RENDER_START           ;2 offset
                sta             LUM_TIMER               ;3 load into LUM timer
                HWAIT
                dbnzx           LineLoop1               ;3 next line if applicable

                ;deactivate the luminance generator

                lda             #16+8                   ;unbuffered operation, no IRQ
                                                        ;output compare, LUM clear

                sta             _TSC0
                clr             _TSC                    ;stop overflow interrupt

                ldx             #11                     ;front porch = 0.35 ms = 11 lines
WtFrontPorcha:
                HWAIT
                dbnzx           WtFrontPorcha

                bclr            VSYNC_PIN,_PTA          ;assert VSYNC
                ldx             #2
WtVSYNCa:
                HWAIT
                dbnzx           WtVSYNCa                ;vsync = 0.06 ms = 2 lines

                bset            VSYNC_PIN,_PTA          ;deassert VSYNC

                Keypress                                ;scan keypad

                ldx             #32                     ;back porch = 1.02 ms = 32 lines

WtBackPorchA:
                HWAIT
                dbnzx           WtBackPorchA

                bclr            7,_TSC                  ;ack overflow IRQ

                lda             #64
                sta             _TSC                    ;enable overflow interrupt

                brclr           FREERUN,FLAGS,DoDSO     ;entering DSO mode?
                jmp             DoFreerun               ;no, next screen

;
;Sample buffer display. One sample per three horizontal lines
;
```

```
;Uses 400 line VGA mode
;
DoDSO:
                sta             _COPCTL                 ;clear watchdog
                bclr            VSYNC_PIN,_PTA          ;deassert VSYNC

                ldx             #50                     ;gap of 50 lines

LineLoop3a:
                HWAIT
                dbnzx           LineLoop3a              ;next line

                lda             #16+8+4+2               ;unbuffered operation, no IRQ
                                                        ;output compare, LUM set on match
                                                        ;clear on overflow
                sta             _TSC0

                ;generate 350 lines in digital storage mode

                ldx             #BUFSIZ-1

                ;First line of three

LineLoop2:
                lda             STORAGE,x               ;3 read sample from buffer
                lsra                                    ;1
                lsra                                    ;1 convert to 6-bit
                add             #RENDER_START           ;2 offset
                sta             LUM_TIMER               ;3 load into LUM timer
                HWAIT

                ;worst case execution below = 33 clocks

                lda             TMR_ACC                 ;3
                add             TMR_STEP                ;3
                sta             TMR_ACC                 ;3 time to take a sample?
                                                        ;(rate = TMR_STEP/256 *
                                                        ;scan rate/3)?
                bcc             DoneSampling                    ;3 no, skip

                lda             SAMPLE_IDX              ;3 fetch current sample index
                beq             DoneSampling            ;3 at start of buffer..
                                                        ;  then we are not sampling

                pshx                                    ;2 save current display index
                tax                                     ;1 index = sample pointer
                lda             _ADR                    ;3 fetch ADC result
                sta             STORAGE,x               ;3 store new sample in buffer
                dec             SAMPLE_IDX              ;3 advance sample pointer

                pulx                                    ;2 restore current display index

DoneSampling:

                ;Second line of three

                HWAIT

                ;worst case execution below = 35 iclocks

                lda             SAMPLE_IDX              ;3 at the start of the buffer?
                bne             DoneTrigger             ;3 no, not looking for trigger

                lda             _ADR                    ;3
                sub             TRIGGER                 ;3
                beq             DoneTrigger             ;3
                brset           RISING,FLAGS,ChkRising  ;5 rising or falling edge?
ChkFalling:
                bcs             DoneTrigger             ;3
                bra             GotTrigger              ;3
ChkRising:
                bcc             DoneTrigger             ;3

                ;fall thru

GotTrigger:
```

```
                lda             #BUFSIZ-3               ;2
                sta             SAMPLE_IDX              ;3 start sampling

DoneTrigger:

                ;Third line of three

                HWAIT
                dbnzx           LineLoop2               ;3 next lines set, if applicable

                ;deactivate the luminance generation

                lda             #16+8                   ;unbuffered operation, no IRQ
                                                        ;output compare, LUM clear
                sta             _TSC0
                clr             _TSC                    ;disable overflow interrupt

                ldx             #50                     ;remaining lines

LineLoop3:

                HWAIT
                dbnzx           LineLoop3               ;next lines

                ldx             #13                     ;front porch = 0.41ms = 13 lines
WtFrontPorch:
                HWAIT
                dbnzx           WtFrontPorch

                bset            VSYNC_PIN,_PTA          ;assert VSYNC
                ldx             #2                      ;vsync = 0.06ms = 2 lines
WtVSYNC:
                dbnzx           WtVSYNC
                bclr            VSYNC_PIN,_PTA          ;deassert VSYNC

                Keypress                                ;scan keypad

                ldx             #34                     ;back porch = 1.08ms = 34 lines
WtBackPorch:
                HWAIT
                dbnzx           WtBackPorch

                bclr            7,_TSC                  ;ack overflow IRQ

                lda             #64
                sta             _TSC                    ;enable overflow interrupt

                brset           FREERUN,FLAGS,GoFreerun       ;exiting to freerun mode?
                jmp             DoDSO                   ;loop to next DSO screen
GoFreerun:
                jmp             DoFreerun               ;no, flip to freerun

                END
```