VGA check PicoMiteVGA

Vertical timing

Sync is negative polarity

Frame part	Lines	Time [ms]
Visible area	480	15.253227408143
Front porch	10	0.31777557100298
Sync pulse	2	0.063555114200596
Back porch	33	1.0486593843098
Whole frame	525	16.683217477656

Frame	Time [ms] measured	Error
Visible area		
Front Porch		
Sync Pulse	0.06353968254	
Back Porch		
Whole Frame	16.68333333	

Horizontal timing

Sync is negative polarity.

VGA 1 pixel (25MHz) is: 40ns
Measurement resolution is (PIO @126MHz/2) : 16ns

Scanline part	Pixels	Time [µs]
Visible area	640	25.422045680238
Front porch	16	0.63555114200596
Sync pulse	96	3.8133068520357
Back porch	48	1.9066534260179
Whole line	800	31.777557100298

Scanline	Time [us] measured	Error (%)
Visible area	25.42857143 (+/- 0.016)	+0.02%
Front Porch	0.6507936508 (+/- 0.016)	+2.3% (15ns too long)
Sync Pulse	3.793650794 (+/- 0.016)	-0.51% (20ns too short)
Back Porch	1.904761905 (+/- 0.016)	-0.1%
Whole Line	31.77777778 (+/- 0.016)	0%

Assuming the VGA PIO in VGApicoMite is running 126MHz to achieve the 25MHz pixel clock, the vertical sync pulse looks 2 cycles to short, causing the backporch 2 cycles to long. This would however not cause the TheBackShed user “nbrok” complaining about blurring at right side of the VGA screen.

The above tolerances may cause “loosing” the leftmost pixel. But since the visible (horizontal) area is exact same size, there should be no shift in sampling during the visible area.

Above numbers are measured using a VGA picomite running V5.07.07 release. The HSYNC and VSYNC are generated by the picomite, and measured by the picomite. This will outrule crystal tolerances. So the numbers are clock cycle accurate.

Measuring HSYNC and VSYNC:
HSYNC and VSYNC are not connected to a monitor, but directly to GP0 with a flying wire, and PIO 1 is used to measure the timing.
Measuring visible area and backporch/frontporch
No monitor is connected, the VGA RED pin is connected to either GP0 or GP1 and generates a 3.3V logic signal when the screen RED. (there is no terminating 75 ohm resistor).
The HSYNC signal is connected to the other GPpin.

The program used for testing is:

You select a measurement mode by changing the label after RESTORE. This is very elementary, but does the job. The PWM is used to calibrate the measurement results.

```
'various measurements for digital signals using PIO

'generate test signal on GP0/1
  'SetPin gp0,pwm  'CH 0a
  'SetPin gp1,pwm  'CH 0b
  'PWM 0, 1000, 60, 20
'

'pio setup, assign GP0 as JMP pin
p=Pio(pinctrl 0,0,0,gp0,gp0,gp0,gp0)
e=Pio(execctrl gp0,0,31)
s=Pio(shiftctrl 0,0,0,0,0,0)
f=126e6
PIO init machine 1,0,f,p,e,s,0

'pio programs and start
Dim a%(7)
Restore fafa
Read inst,offs
For i=0 To 7: Read a%(i):Next
PIO program 1,a%()

PIO start 1,0

'mmbasic data handler from fifo's
'Check the the read data in MMBasic and print
Dim d%


Do
  k=0
  CLS RGB(red)
  For i=1 To 10
    PIO read 1,0,1,d%
    k=k+d%
  Next i
  d%=k/10
  Print "time =";1e6*(d%+offs)*inst/f;" us"
  Pause 1000
Loop
End

'--------------------------- PIO programs ----------------------------
'data format: inst,offs,program
' inst = number of instructions per loop
' offs = adjustment for instructions outide the loop

pul:
```

```
' measure pos pulse width on GP0
' line     code     mnemonics
' &h00     A02B    Mov X=-NULL                     'clear X
' &h01     2080    Wait until GP0=1                'wait for pulse rising edge to start counting
' &h02     0043    JMP (x<>0) x--, &h03            'always go to next instruction, decrement x
' &h03     00C2    JMP (GP0=1), &h02               'high count loop
' &h04     A0C9    MOV ISR = -X                    'X invert result -> ISR
' &h05     8000    Push noblock                    'ISR->FIFO
' &h06     0000    JMP 0
Data 2,1,&hc200432080a02b,&h8000a0c9,0,0,0,0,0,0

pau:
'measure neg pulse width on GP0
' line     code     mnemonics
' &h00     A02B    Mov X=-NULL                     'clear X
' &h01     2000    Wait until GP0=0                'wait for falling edge to start counting
' &h02     00C4    JMP (GP0=1), &h04               'high count loop, exit when GP0 high
' &h03     0042    JMP (x<>0) x--, &h02            'count loop, decrement x, goto 2
' &h04     A0C9    MOV ISR = -X                    'X invert result -> ISR
' &h05     8000    Push noblock                    'ISR->FIFO
' &h06     0000    JMP 0
Data 2,-1,&h4200c42000a02b,&h8000a0c9,0,0,0,0,0,0

per:
' measure period on gp0
' line     code     mnemonics
' &h00     A02B    Mov X=-NULL                     'clear X
' &h01     0042    JMP (x<>0) x--, 2               'always go to next instruction, decrement x
' &h02     00C1    JMP (GP0=1), 1                  'high count loop
' &h03     00C5    JMP (GP0=1) &h05                'jump out of the low count loop when GP0=1
' &h04     0043    JMP (X<>0) X--, &h03            'decrement X and always jump back to 3
' &h05     A0C9    MOV ISR = -X                    'X invert result -> ISR
' &h06     8000    Push noblock                    'ISR->FIFO
' &h07     0000    JMP 0
Data 2,3,&h00c500c10042a02b,&h8000a0c90043,0,0,0,0,0,0

fari:
' measure fall on gp1 to rise on gp0
' line     code     mnemonics
' &h00     A02B    Mov X=-NULL                     'clear X
' &h01     2081    Wait until GP1=1                'wait for rising edge GP1
' &h02     2001    Wait until GP1=0                'wait for falling edge GP1 to start counting
' &h03     00C5    JMP (GP0=1), &h05               'high count loop
' &h04     0043    JMP (x<>0) x--, &h03            'always go to next instruction, decrement x
' &h05     A0C9    MOV ISR = -X                    'X invert result -> ISR
' &h06     8000    Push noblock                    'ISR->FIFO
' &h07     0000    JMP 0
Data 2,-1,&hc520012081a02b,&h8000a0c90043,0,0,0,0,0,0

riri:
' measure rise on gp1 to rise on gp0
' line     code     mnemonics
' &h00     A02B    Mov X=-NULL                     'clear X
' &h01     2001    Wait until GP1=0                'wait for falling edge GP1
' &h02     2081    Wait until GP1=1                'wait for rising edge GP1 to start counting
' &h03     00C5    JMP (GP0=1), &h05               'high count loop
' &h04     0043    JMP (x<>0) x--, &h03            'always go to next instruction, decrement x
' &h05     A0C9    MOV ISR = -X                    'X invert result -> ISR
' &h06     8000    Push noblock                    'ISR->FIFO
' &h07     0000    JMP 0
Data 2,-1,&hc520812001a02b,&h8000a0c90043,0,0,0,0,0,0

fafa:
' measure rise on gp1 to rise on gp0
' line     code     mnemonics
' &h00     A02B    Mov X=-NULL                     'clear X
' &h01     2081    Wait until GP1=1                'wait for rising edge GP1
' &h02     2001    Wait until GP1=0                'wait for falling edge GP1 to start counting
' &h03     0044    JMP (x<>0) x--, &h04            'always go to next instruction, decrement x
' &h04     00C3    JMP (GP0=1), &h03               'high count loop
' &h05     A0C9    MOV ISR = -X                    'X invert result -> ISR
' &h06     8000    Push noblock                    'ISR->FIFO
' &h07     0000    JMP 0
Data 2,0,&h4420012081a02b,&h8000a0c900c3,0,0,0,0,0,0
```