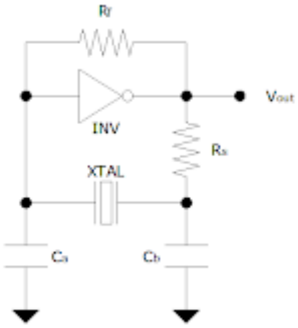
Checking out the Pierce oscillator

The Pierce oscillator is a very simple circuit and as with the ring oscillator also uses inverters (actually just one inverter will do). The principle of operation, however, is very different. While the ring oscillator exploits the incremental propagation delay through the series of inverters, the Pierce oscillator puts the inverter/gate into the linear region of operation and employs it as an analog amplifier.

To find out the details of the gate-based Pierce oscillator I recommend reading the references listed below. What follows is just a summary of the tips and formulas culled from the said references, pointers and tidbits which I found useful in getting a crystal-based oscillator up and running. I also did a number of breadboarded experiments and the results are shown below.

The following schematic shows the basic Pierce oscillator based on a CMOS inverter as its amplifier.



INV is of course the inverter (buffered or unbuffered). Oscillator signal is taken from the inverter output (V_{out}). To improve (decrease) its rise/fall time, another inverter or, better yet, a (fast!) Schmitt trigger can be used to spruce up the signal. The signal at the inverter input is a sinusoid and can also be used (some Microchip MCU datasheets show the clock signal being derived from the input of the inverter while other Microchip datasheets show it being taken from the inverter output).

R_f is a feedback resistor that puts the gate in linear (as opposed to digital) mode operation. The following table provides a rule of thumb value for R_f given crystal frequency:

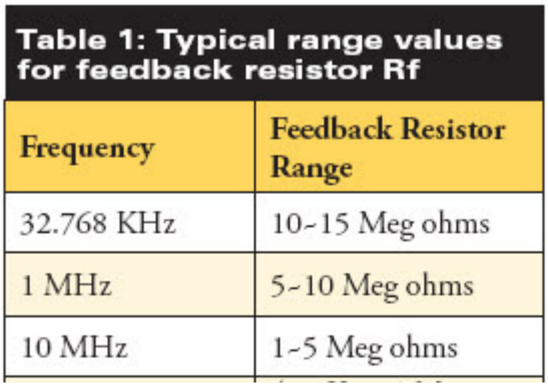
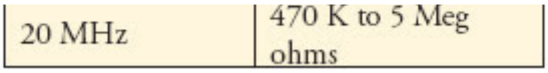


Table 1: Typical range values for feedback resistor Rf

Frequency	Feedback Resistor Range
32.768 KHz	10-15 Meg ohms
1 MHz	5-10 Meg ohms
10 MHz	1-5 Meg ohms



20 MHz	470 K to 5 Meg ohms

Among other things, R_s limits the amount of crystal drive--increasing R_s decreases drive. A ballpark figure or first cut value for R_s can be derived by computing for and equating R_s to the reactance of C_b:

> $X_{c_b} = 1/(2\pi f C_b)$, where f = crystal frequency

According to Microchip:

> R_s is typically 40 K ohms or less, but is almost never more than 100 K ohms. If the value for R_s is too high, then the high impedance input side of the amplifier may be more susceptible to noise, very much the same way a pull-up resistor on an input pin is normally kept below about 50 K ohms to prevent noise from having enough strength to override the input.

Freescale (Motorola) meanwhile claims that for low frequencies such as 32.768kHz watch crystals, R_s can go as high as 330Kohms. Because of a permissible maximum drive of just 1μW, a minimum R_s value for tuning fork crystals such as the 32.768kHz crystal is 10kohm.

C_a and C_b along with R_s and XTAL provide a 180 degree phase shift (INV provides the other 180 degrees for a total of 360). Moreover, the R_s and C_b network partly "acts as a low pass filter that discourages the crystal from running at a third or fifth harmonic, or other higher frequency" (Lancaster and Berlin). Increasing C_a and C_b decreases the gain. C_a and C_b are usually equal but C_b can be made larger than $C_{a.}$ The voltage at the input of INV is (partly) determined by C_b / C_a, so that increasing C_b relative to Ca increases the voltage at INV input. As a rule the values of C_a and C_b should satisfy the following condition:

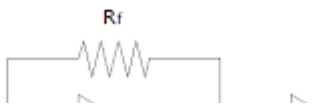
> $C_{load} = (C_a)(C_b)/(C_a + C_b)$ + stray capacitance

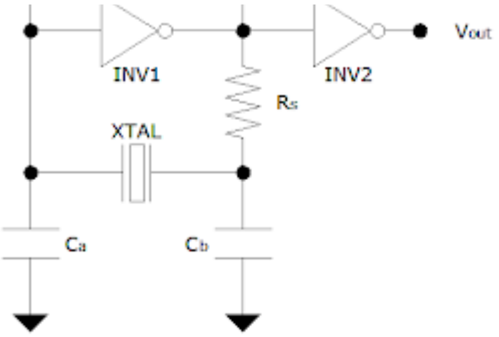
where C_{load} is the load capacitance of the crystal as per manufacturer's specifications. Typical load capacitances are 12 pF, 15 pF, 18 pF, 20 pF, 22 pF and 32 pF.

Finally, an advice worth keeping in mind: "Oscillator design is an imperfect art at best. Combinations of theoretical and experimental design techniques should be used." And so we move on to the experimental side of things.

I performed a few tests on a breadboard using 4.000 MHz and 32.768 kHz crystals. The values for the resistors and caps are the ones I arrived at after some trial and error. The values below seem to work better. Yes, that's a very subjective assessment and shall leave it at that.

I. Motorola MC14049UBCP hex inverting buffer





A. XTAL = 32.768 kHz

R_f = 10Mohm

Rs = 100kohm

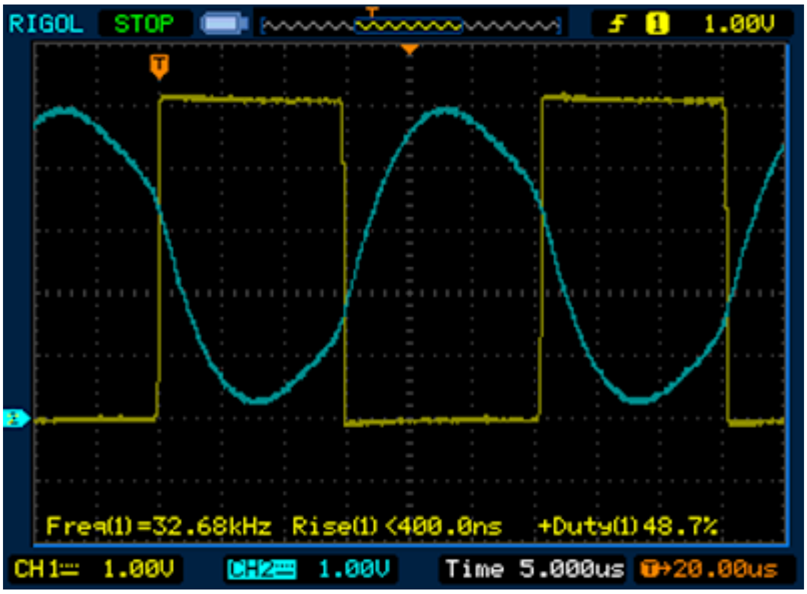
C_a = 33pF

C_b = 50pF

Oscilloscope setup

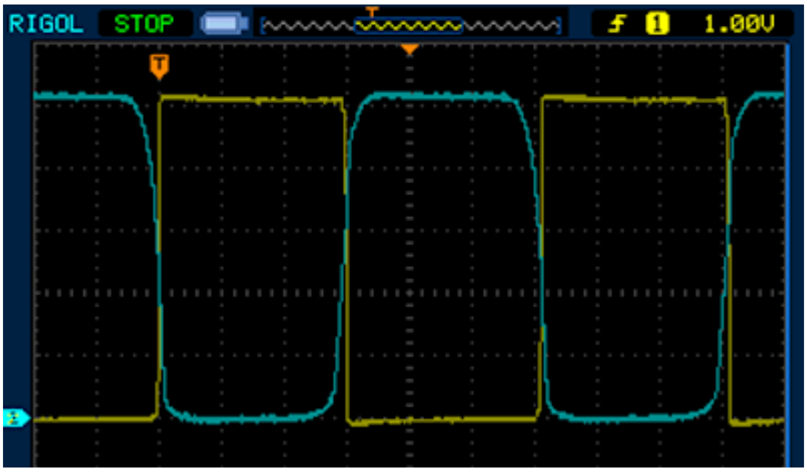
Channel 1 (yellow) hooked up to INV2 output

Channel 2 (cyan) probe hooked up to C_b



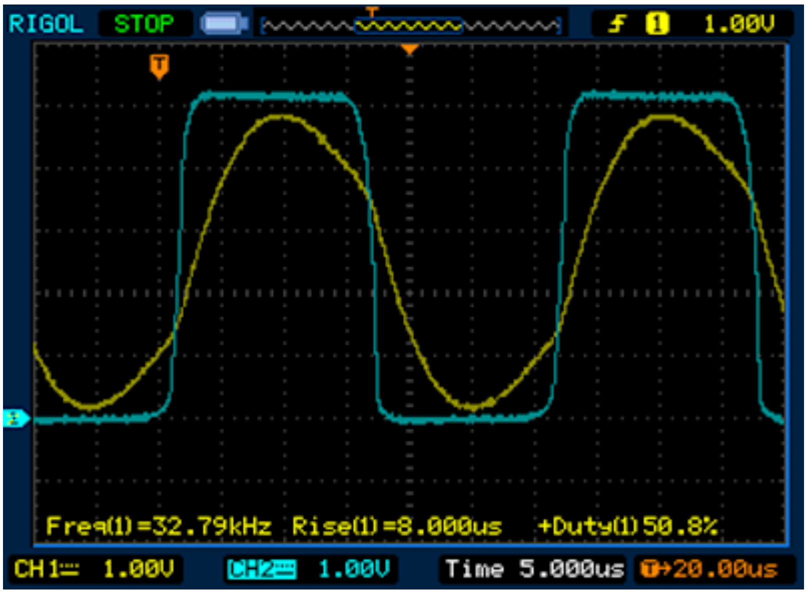
CH1: INV2 output

CH2: INV1 output



CH1: C_b

CH2: INV1 output

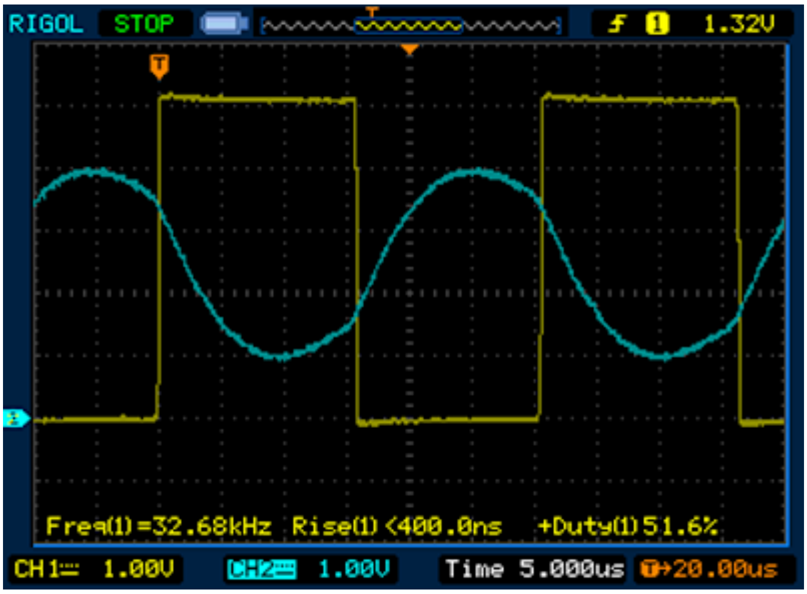


I increased the capacitances as follows
C_a = 18pF

C_b = 100pF

and I was pleasantly surprised to see that the waveforms became much more stable--it practically ceased "wiggling" (the frequency was stable but the duty cycle wasn't and so the falling edge would be shifting rapidly to and fro horizontally--I was triggering on the rising edge so that wasn't moving at all)

CH1: INV2 output
CH2: C_b



Notice how the increasing the C_b to C_a ratio has decreased the signal's amplitude at C_b.

B. XTAL = 4.000 MHz

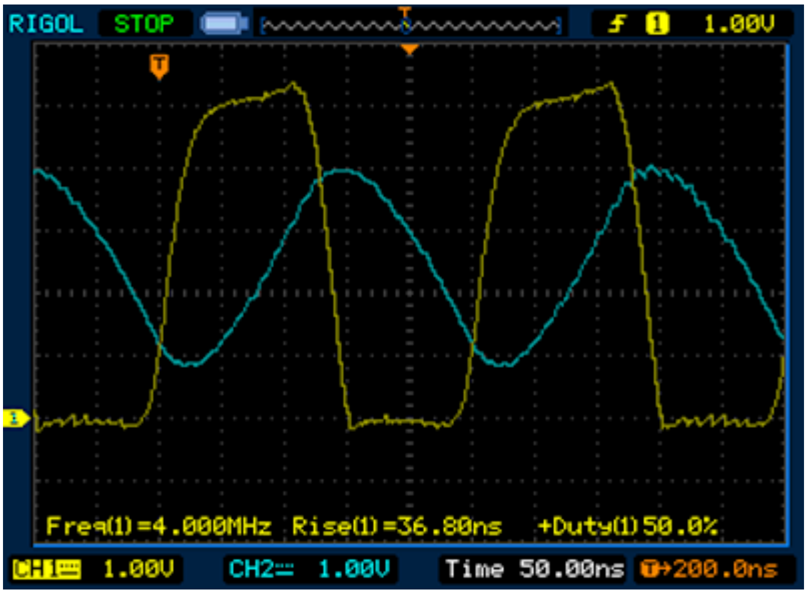
R_f = 5.1Mohm

Rs = 2kohm

$C_a = C_b$ = 18pF

CH1: INV2 output

CH2: C_b

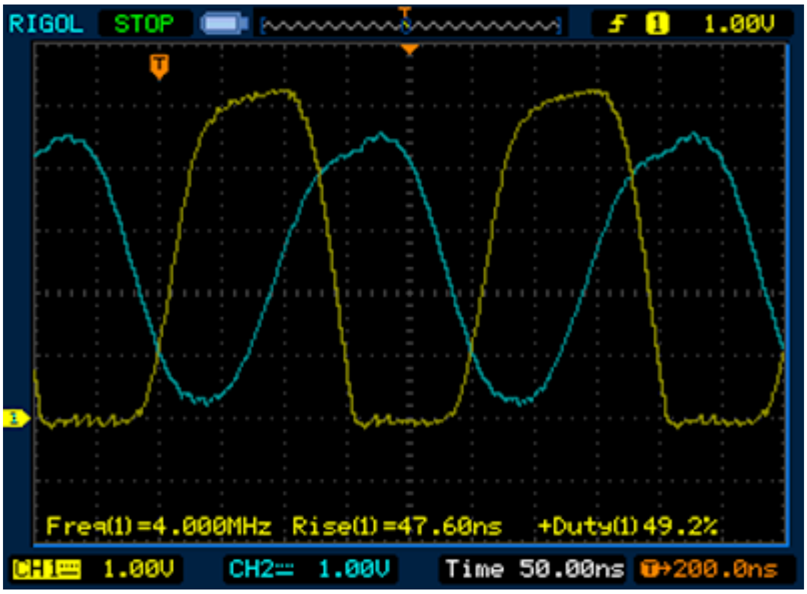


```
Freq(1)=4.000MHz  Rise(1)=36.80ns  +Duty(1)50.0%
CH1 1.00V   CH2 1.00V   Time 50.00ns  T→200.0ns
```

The CMOS 4000 series is relatively slow. In the reading above rise time is almost 37ns.

CH1: INV2 output

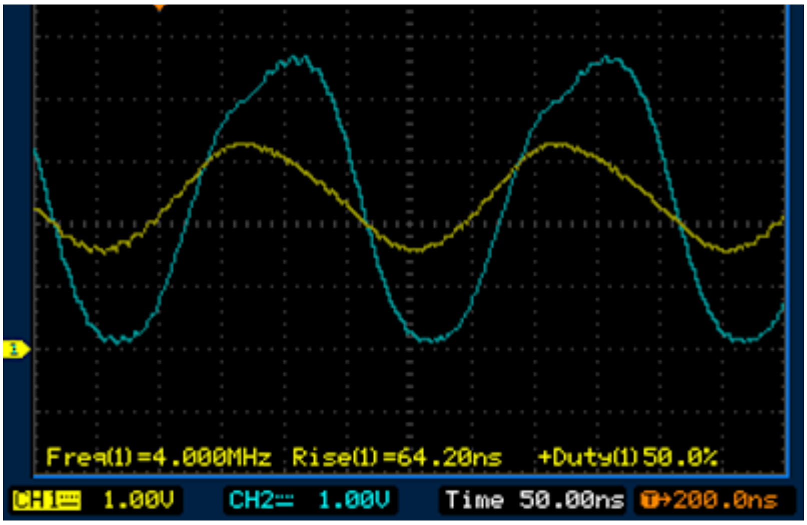
CH2: INV1 output



```
Freq(1)=4.000MHz  Rise(1)=47.60ns  +Duty(1)49.2%
CH1 1.00V   CH2 1.00V   Time 50.00ns  T→200.0ns
```

CH1: C_b

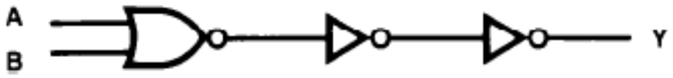
CH2: INV1 output



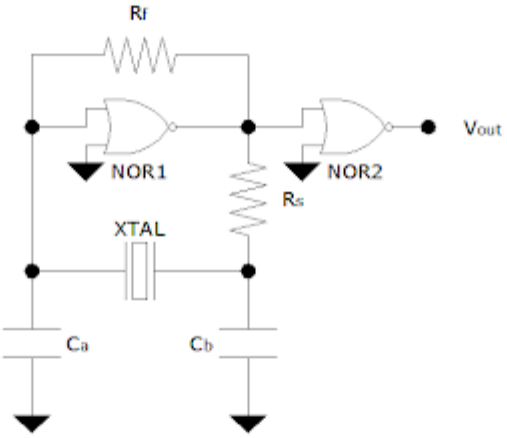
Freq(1)=4.000MHz Rise(1)=64.20ns +Duty(1) 50.0%

CH1 1.00V CH2 1.00V Time 50.00ns T→200.0ns

II. National Semiconductor MM74HC02N quad 2-input NOR gate. Unfortunately, I don't currently have any CMOS HC or AC series inverters. According to the datasheet each NOR gate is buffered as follows:



Using a buffered gate in a Pierce oscillator consumes less power and has a gain in the order of thousands compared to an unbuffered gate which has a gain of hundreds. The drawback of a buffered gate is that it is more sensitive to the values of the passive components and tends to be less stable.



A. XTAL = 32.768 kHz

R_f = 10Mohm

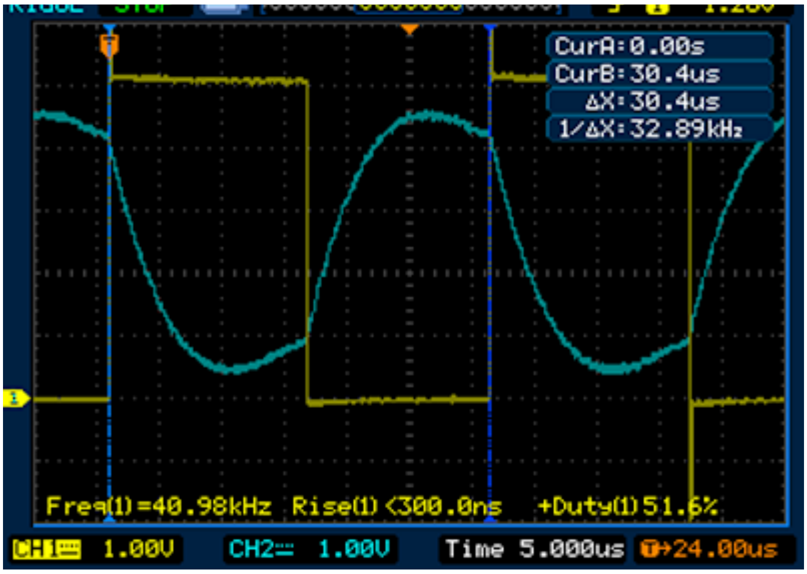
Rs = 51kohm

C_a = 33pF

C_b = 100pF

CH1: NOR2 output

CH2: C_b

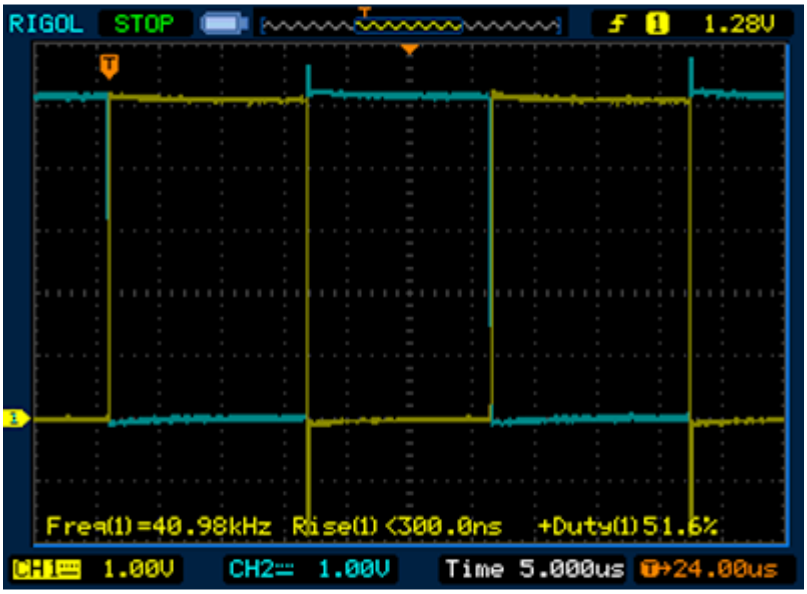


```
CurA:0.00s
CurB:30.4us
ΔX:30.4us
1/ΔX:32.89kHz
Freq(1)=40.98kHz  Rise(1)<300.0ns  +Duty(1)51.6%
CH1 1.00V  CH2 1.00V  Time 5.000us  T→24.00us
```

I believe the over- and undershoots (at the rising and falling edges) is confusing the scope. In the screenshot above it's 40.98kHz, but in real time it's all over the place, sometimes reaching as high as 80kHz. So I turned on the cursors. As you can see it says 32.89kHz. Resolution is such that moving the cursor just one pixel down results in a reading of 32.68kHz.

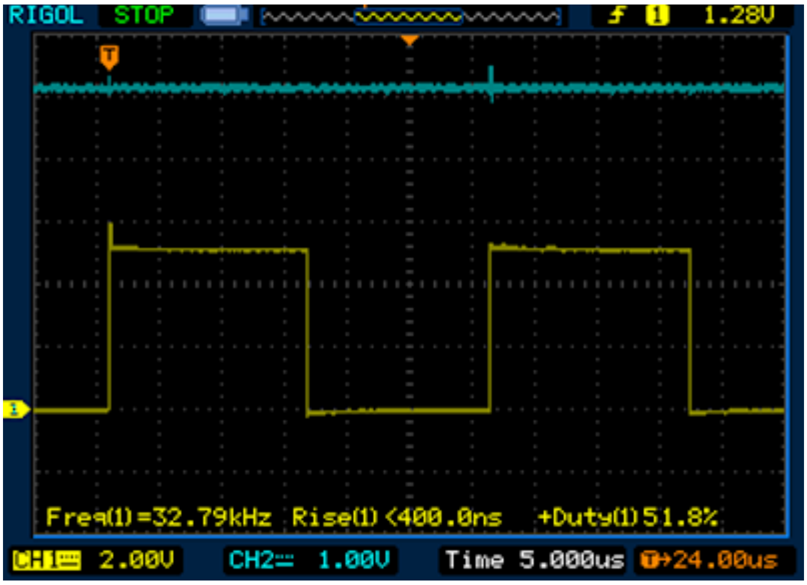
To obtain a slightly better reading I sent the INV2 output to a Goldstar FG-2002C function generator / frequency counter. The LED display has a resolution to two decimal places and said the signal was 32.77kHz. I guess the output is pretty close to the crystal's fundamental frequency. The frequency counter apparently loads the INV1 output because connecting it directly produces garbage readings (bounces around from 40 to 65kHz). A minimum of 15 ohms of resistance or a few picofarads (the smallest I have right now is 18pF) of capacitance in series is enough to satisfy INV1. The reading is the same as probing INV2 output.

CH1: NOR2 output
CH2: NOR1 output



```
RIGOL STOP  1.28V
Freq(1)=40.98kHz  Rise(1)<300.0ns  +Duty(1)51.6%
CH1 1.00V  CH2 1.00V  Time 5.000us  T→24.00us
```

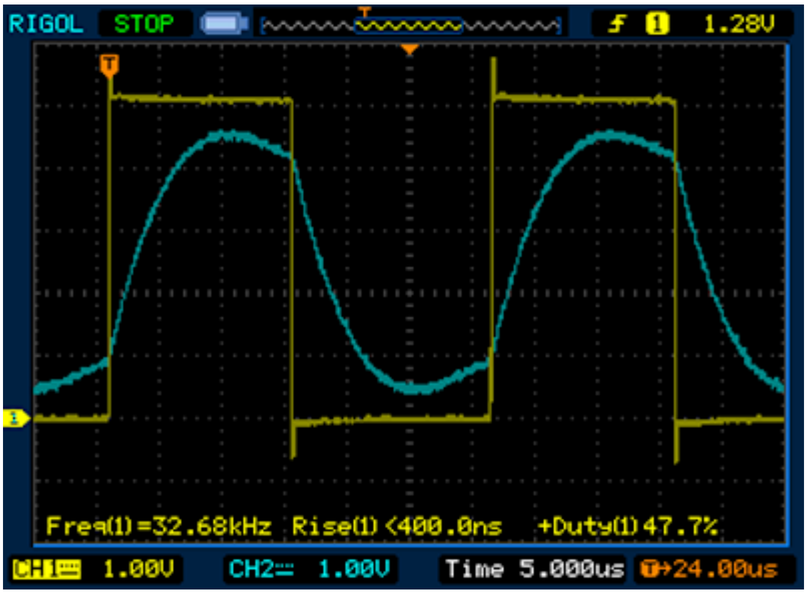
CH1: NOR2 output
CH2: V_{DD}


RIGOL STOP 1.28V

Freq(1)=32.79kHz Rise(1)<400.0ns +Duty(1)51.8%

CH1 2.00V CH2 1.00V Time 5.000us T→24.00us

CH1: NOR1 output
CH2: C_b


RIGOL STOP 1.28V

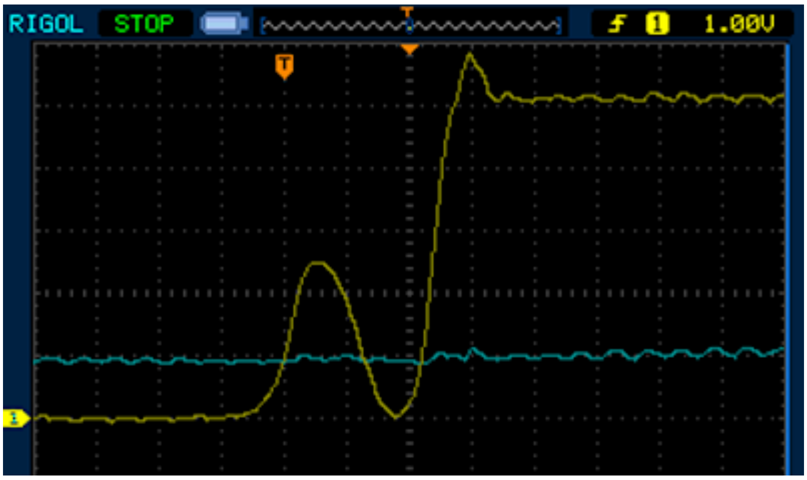
Freq(1)=32.68kHz Rise(1)<400.0ns +Duty(1)47.7%

CH1 1.00V CH2 1.00V Time 5.000us T→24.00us

Changing the time base and capturing those overshoots/undershoots:

CH1: NOR1 output
CH2: C_b

Trigger: rising edge, 1.0V

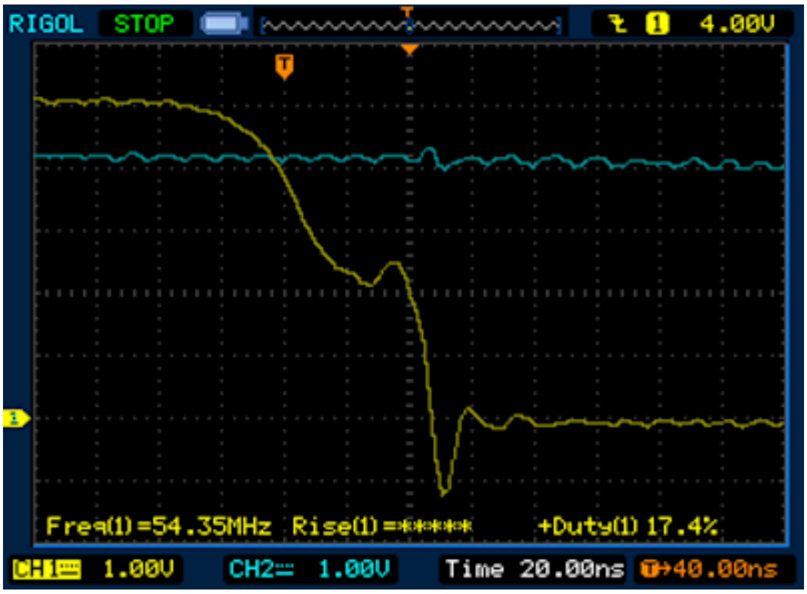

RIGOL STOP 1.00V

```
Freq(1)=*****        Rise(1)=10.56ns    +Duty(1)=*****
CH1  1.00V    CH2  1.00V    Time 20.00ns  T→40.00ns
```

CH1: NOR1 output

CH2: C_b

Trigger: falling edge, 4.0V



```
RIGOL  STOP                         1  4.00V
Freq(1)=54.35MHz  Rise(1)=*****    +Duty(1) 17.4%
CH1  1.00V    CH2  1.00V    Time 20.00ns  T→40.00ns
```

B. XTAL = 4.000 MHz

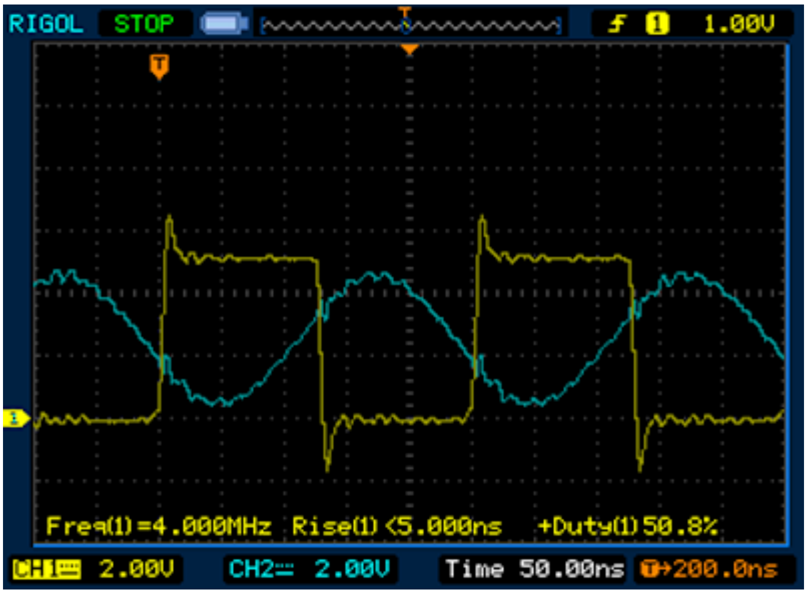
R_f = 5.1Mohm

Rs = 10kohm

C_a = C_b = 33pF

CH1: NOR2 output

CH2: C_b

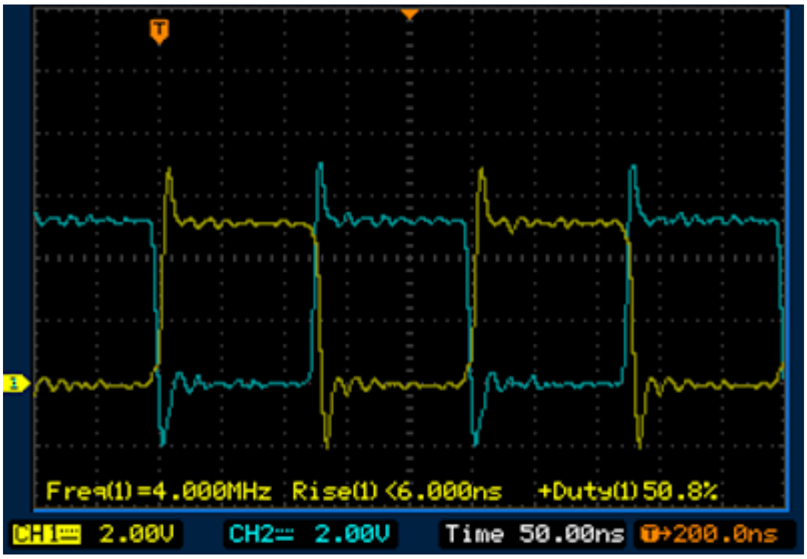


```
RIGOL  STOP                         1  1.00V
Freq(1)=4.000MHz  Rise(1)<5.000ns  +Duty(1)50.8%
CH1  2.00V    CH2  2.00V    Time 50.00ns  T→200.0ns
```

CH1: NOR2 output

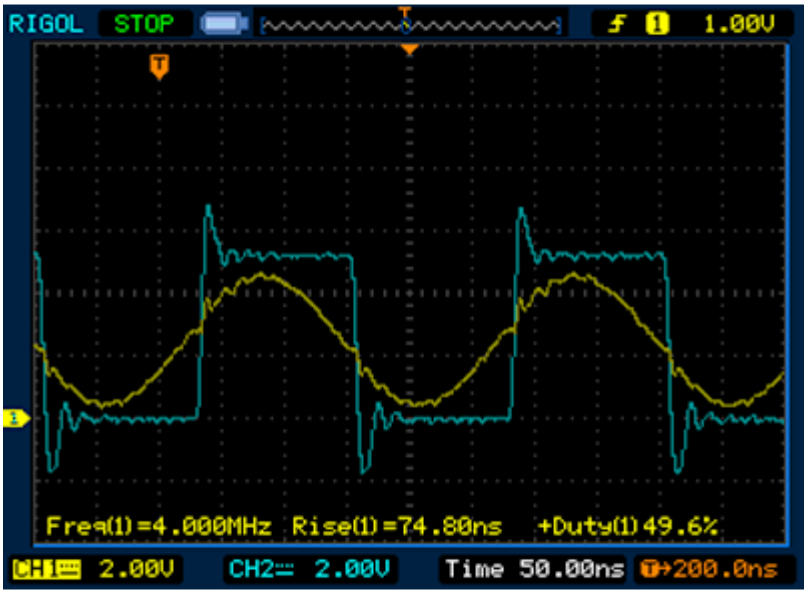
CH2: NOR1 output

```
RIGOL  STOP                         1  1.00V
```



CH1: C_b

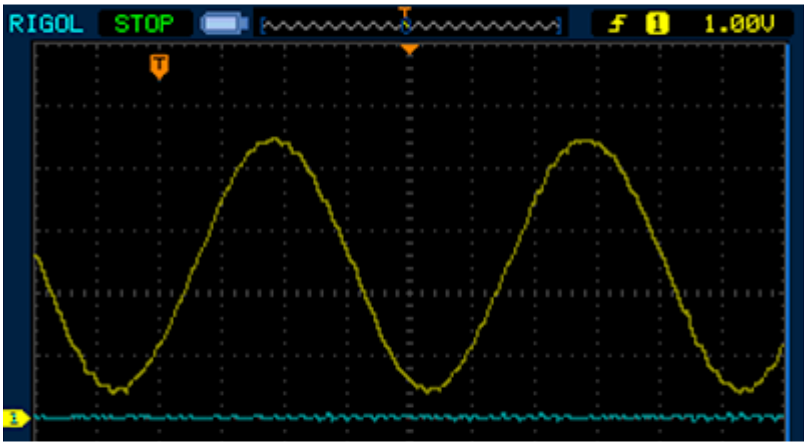
CH2: NOR1 output

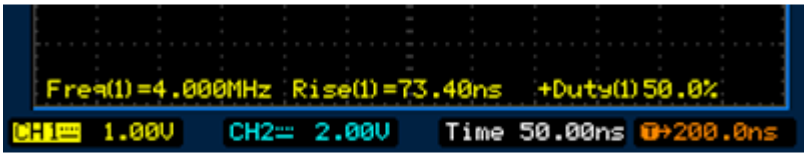


It turns out that the scope probe might be (largely) responsible for the ringing in the output. When not probing either INV1 or INV2, V_{DD} has much less ripple and the ripple in C_b signal practically disappears:

CH1: C_b

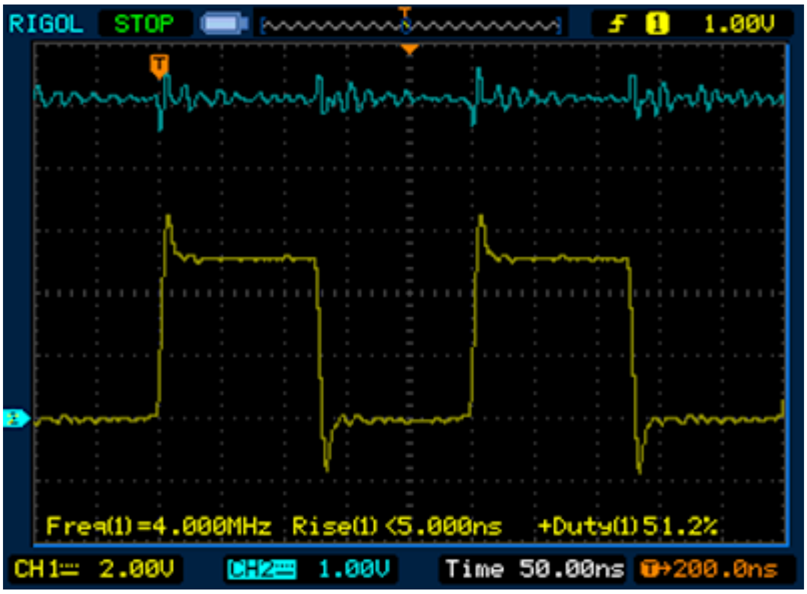
CH2: ground





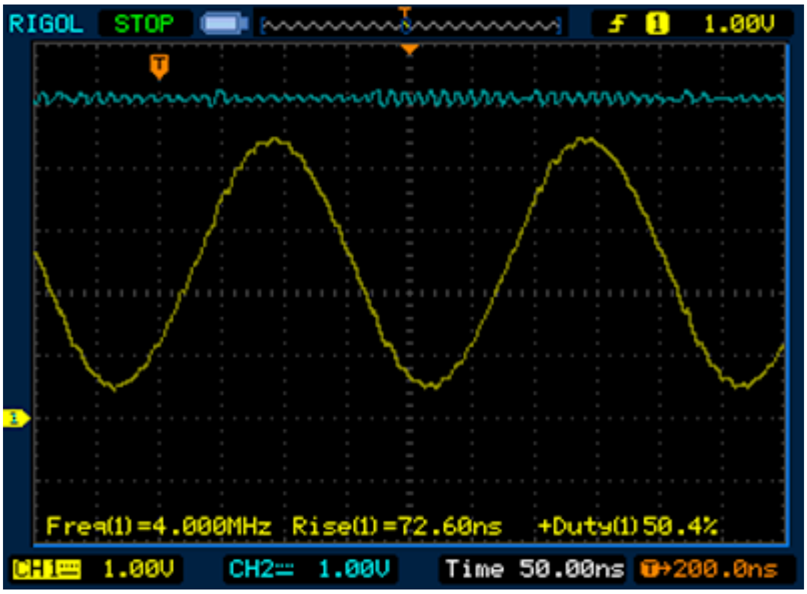
CH1: NOR2 output

CH2: V_{DD}



CH1: C_b

CH2: V_{DD}



References:

Don Lancaster and Howard M. Berlin, *CMOS Cookbook*, 2ed., Howard W. Sams, 1988, p.279-281

AN849: Basic PICmicro® Oscillator Design

Pierce-Gate Crystal Oscillator, an Introduction

AN1706/D Microcontroller Oscillator Circuit Design Considerations

SZZA043: Use of the CMOS Unbuffered Inverter in Oscillator Circuits

AN340: HCMOS Crystal Oscillators