Re: Accurate(ish) frequency measurement

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2007−03/msg02066.html

* *From*: "jure" <jure@xxxxxxxx>
* *Date*: 10 Mar 2007 09:52:08 −0800

On Mar 9, 9:36 am, "Michael Brown" <s...@xxxxxxxxxxxxxxx> wrote:

> I'm playing around with one of my projects at the moment, and what would be nice is to have built−in frequency calibration. The project essentially involves watching crystals to see how they age (and also is a fun experiment as to how stable of an environment − thermal and voltage − I can make). Which is probably only somewhat more interesting than watching paint dry to most people :) Currently I can do this, but only with an annoying amount of external equipment.
>
> Currently, the board under test has a 4 MHz crystal oscillator as found in the LT1016 datasheet. I would like to do fairly accurate frequency measurements of this crystal against a rubidium reference 1 PPS source I have access to (assumed to have at worst 5E−11 short term stability). I'd ideally like to make measurements of the frequency at the 1E−8 level or better. Since the board has a microcontroller on it, I would ideally like to simply plug in the 1 PPS source and read out a frequency through a serial cable.
>
> The simple method − counting cycles − would sort of work. If I count the output of the xtal oscillator for about 25 seconds I should get an error of 1 part in 10^8. But I'd like to make the measurement faster (for 'I doubt I can hold everything stable for that long' reasons and because faster = better) and ideally not simply limited in accuracy to how long I wait.
>
> My plan is to make a slight variant on a TAC. The number of full XTAL cycles in between the reference rising edges is obviously easy to measure. To measure the part cycles, I was going to use the discharge time of a capacitor. For the final part cycle: initially prepare the capacitor to ~1V, start charging it (+5V −> resistor −> capacitor −> GND) on the rising edge of the 1 PPS source, keep charging until the rising edge of the 4 MHz clock, then discharge it through a much bigger resistor and count how many 4 MHz cycles occur until it hits ~0.8V. There's a few complications to avoid dropping or collecting extra cycles, but that's the basic idea. Also, there's obviously non−linearity problems here but nothing a bit of microcontroller time can't fix. A similar method will be used for measuring the start partial cycle.
>
> Since this is living on a double−sided PCB, using a much higher frequency

> clock to measure things more accurately isn't too feasable. Another alternative would be to simply go out and buy a nice TDC chip from somewhere ... unfortunately obtaining one or two of these appears to be either impossible or extremely expensive − neither of which are helped by the fact that I'm in Australia.
>
> The questions (finally!) are
> 1) Is this the sensible way to do it?
> 2) Slew rate and switching delay for the capacitor charging (and to a letter degree, the discharging) circuit is obviously important. From a back of the envelope calculation even a BC109 seems to be able to do the job, but that seems too easy. A MOSFET + driver is mess that I'd rather avoid.
> 3) Any nasty hidden surprises in these types of circuits that I should keep an eye out for when designing it?
>
> −−
> Michael Brown
> Add michael@ to emboss.co.nz − My inbox is always open

Sounds interesting Michael ,

although you have to decide for yourself if it is worth the exercise.
You could improve better that 100x your single shot precision in 1 sec, using the TAC interpolators.
Certainly it may be quite a learning experience that could be applied to other fields....

So, the situation is:

a) there is an accurate 1 Hz gate signal.
b) there is an unknown clock of frequency close to 4 MHz.
c) an interpolator will find Delta_T1 = the time from the active edge of the gate to the next active edge of the unknown clock.
(Delta_T1 is in your case less than 25 ns)
d) a counter will count the integer number N of clock cycles between the first active clock edge after the first active edge of the gate,
and the first active edge of the clock after the last active edge of the gate.
e) a second interpolator will find Delta_T2 = the time from the 2nd active edge of the gate to the next active edge of the unknown clock.
(Delta_T2 is in your case less than 25 ns)

Note : here your case is a little easier than the more general solution:
you stop the second interpolator after the counter reaches N = 4 000 000

f) compute the time it took to run N = 4 000 000 complete cycles

T = 1 s + Delta_T2 − Delta_T1

then , the 1s averaged unknown frequency is: Fx = 4 000
000 / T

Extra Notes:

1) the interpolator is a gated constant current source charging a capacitor (=> linear ramp voltage on C), followed by a sample and hold, and an ADC.
2) there could be one single interpolator , given that there is 1s between its use for the gate start and gate stop (and you have 1 second to: integrate , hold, convert , read ADC , reset).
3) the system may be pipelined to get one frequency reading per second, but you will need two interpolators.
4) I would stay with one interpolator, so that you will have to calibrate only one of them , not two.
5) the interpolator(s) have to be calibrated so that the addition above is meaningful, (how many ADC counts per Fx period are there ?)
5) all the logic , the counter and the state machine controlling the system may be implemented in a fast CPLD.
6) the scheme presented above assumes that the unknown frequency is 4 MHz +/− 1Hz (ie +/− one count in 1 s),
if not you could change the state machine so that the interpolators span 2 or more clock cycles, then change everything else to match)

timing diagram (see with fixed point courier):

___________________... 1s ....
____
1s Gate signal | |
______| ....____|

____..
____..
Fx Clock |
|
..___| ..._|

−>| |<− DT1 −>| |<−
DT2

Clock Edge # 0 Clock
Edge # N−1

Thanks , Jure Z.

.