Clock Generator Synchronized from GPS

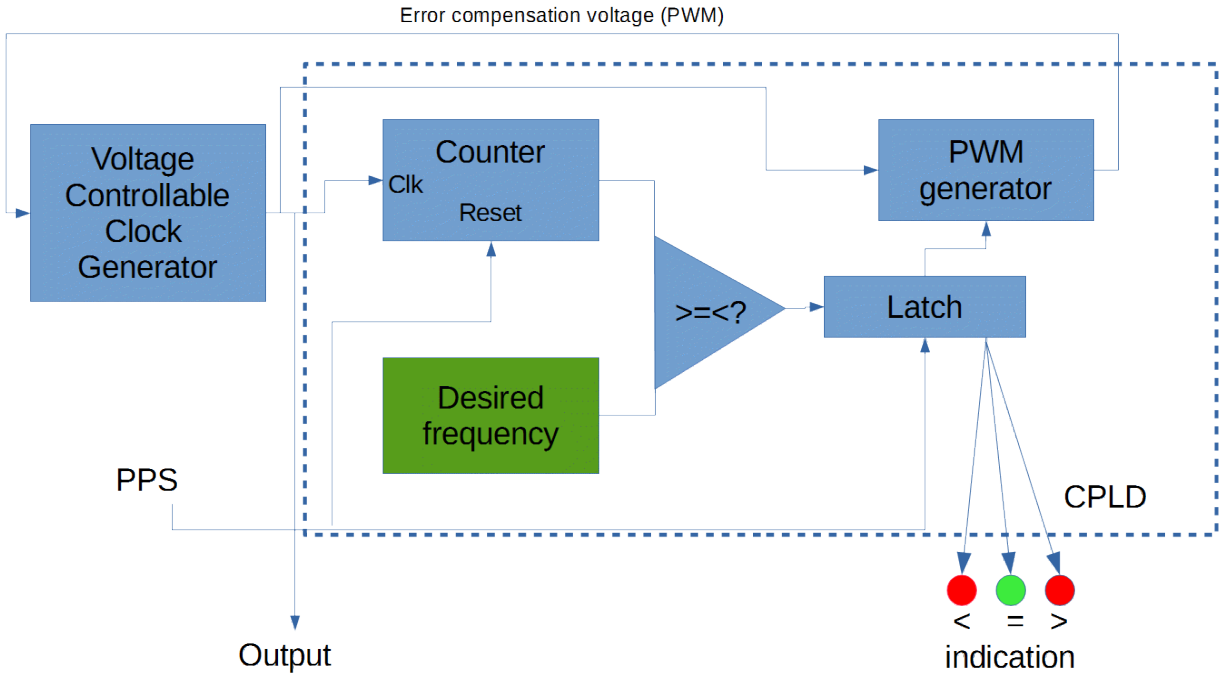
By Dmitry Korobkov <dkorobkov at mail dot ru>

Introduction

The goal was to build clock generator that provides better frequency match than a cheap crystal oscillator. Off-the-shelf TCXO (temperature compensated crystal oscillator) for desired frequency as well as VCXO (voltage controlled crystal oscillator) was not available at the moment. This is also an excuse for the components used in the project.

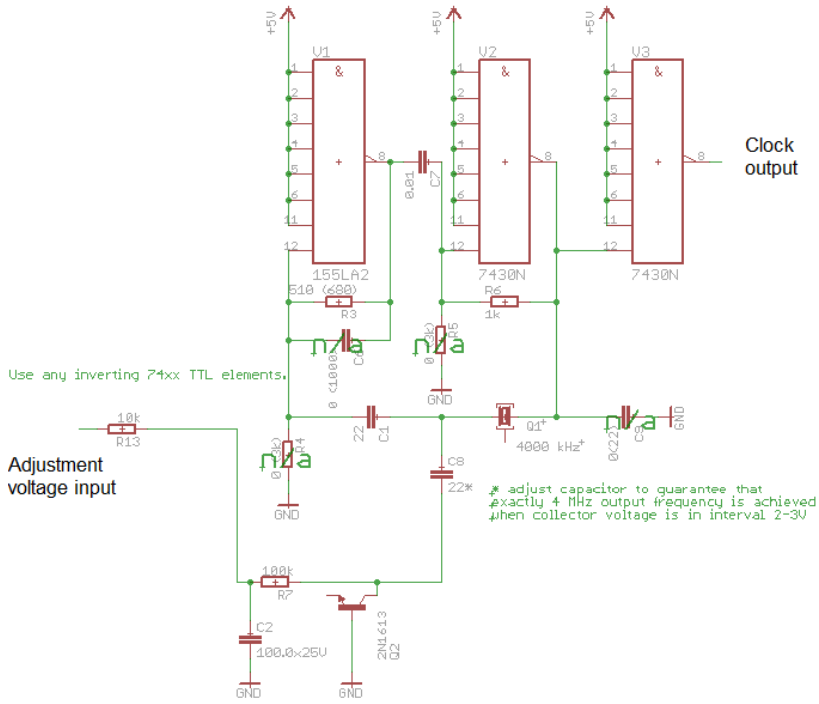
Design and Operation

The idea is to use PPS (pulse per second) signal from GPS, to count pulses from crystal oscillator and slightly adjust crystal frequency to match the desired frequency more or less exactly. Using this approach one can achieve 2-5 ppm (parts per million) stability. Further improvement can be achieved by building a true PLL loop that will match a front edge of PPS with a front edge of every 4,000,000th clock pulse (for 4 MHz frequency) but the simplistic VCXO circuit does not allow this level of precision and the design won't fit the chosen CPLD. This will work fine (and much better) on FPGAs (or at least XC2C128) but I hadn't one readily available, while CPLD was collecting dust in my nearby locker.



Functional schematic of clock generator

Voltage controlled clock generator is built on 74 series inverters. Just one of many possible circuits is used, it is possible to replace it with any type. Frequency is adjusted by changing the voltage on collector-base junction of a high frequency bipolar transistor, which will change capacity of the junction. This additional capacity, being applied to crystal, slightly changes its oscillating frequency. The capacitor connected in series with the crystal lowers its frequency below 4 MHz so full swing of PWM voltage will adjust it around 3.99995-4.00005 MHz. The values can be approximate since everything is compensated by feedback. When adjusting your frequency consider possible temperature drift to provide generator normal operation in temperatures other than normal.



Schematic of voltage controlled clock generator

Counter counts clock pulses until PPS front edge comes. At this moment we compare the actually counted number of pulses to desired one, adjust error compensation voltage accordingly and latch the value. Voltage is generated with 10-bit PWM and filtered by RC filter.

Since XC2C64a is too small it was not possible to fit a 24-bit counter that will be capable of counting all 4000000 pulses, so a 10 bit counter with overflow is used. After 4000000 clocks it will output the value of 256, so its output is compared with 256, incrementing or decrementing PWM value every second (PPS pulse). This approach requires that initial oscillator frequency deviation under any conditions is less than 500 Hz otherwise feedback loop will catch the wrong target frequency.

Obviously, if no PPS is coming, generator is not adjusted. To see synchronization state there are LEDs that roughly show counter value (“less than 256-minus-something”, “256 plus-minus something” and “more than 256-plus-something”). With 10 bit PWM precision, initial locking of frequency can take up to 1023 seconds since first PPS (one bit per second).

The generator can be adjust to any frequency less than 100 MHz (limited by CPLD performance).

Schematic

Schematic can be found at https://github.com/dkorobkov/sync_from_gps .

Verilog Code and CPLD Programming

Source code written in Verilog is in main.v file available at https://github.com/dkorobkov/sync_from_gps . It is rather simple and well commented.

To adjust your generator for another frequency you will need to calculate counter value (256 in code above) and replace it with your value. Example of calculation: 4000000 pulses will overflow 10-bit counter (4000000/1024)=3906.25 times so counter will show 256. Depending on quality and sensitivity of your generator you may want to imcrease precision, then adjust the lines

```
difference = (counter-256)>>2 ;
```

and

```
difference = (256-counter)>>2;
```

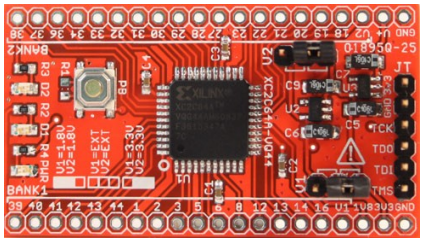
by changing 2 to 1 or 0. This will minimize tolerance but probably slow down frequency lock or make it impossible at all, oscillating up and down around target frequency.

Verilog file was compiled and CPLD was programmed from Xilinx ISE Webpack 14.7. Bitfile is loaded into CPLD via JTAG connector on the breakout board. A simple LPT cable did exist in the times of LPT enabled PCs, more convenient option is JTAG download cable available on Aliexpress.

Components

The following off-the-shelf components were used:

1) CPLD breakout board



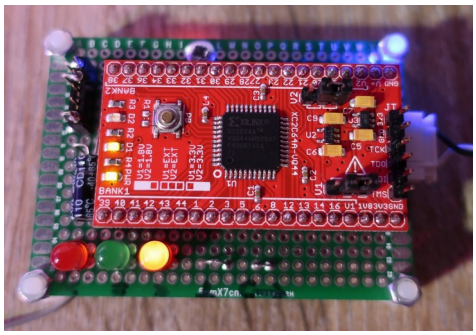
As of Dec 2019, available for purchase for $15 at http://dangerousprototypes.com/docs/CoolRunner-II_CPLD_breakout_board

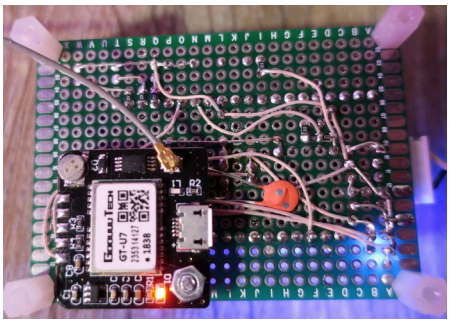
2) uBlox-7 based GPS

Available from Aliexpress for $6-7. Any GNSS receiver with dedicated 3.3V PPS output can be used.

Results

Schematic assembled on a breadboard looks as below:





74Xx logic is under CPLD board.