TACHOMETER – CYCLOMETER WITH PROGRAMMABLE LOGIC

Yavor Danchev Donkov, Todor Stoyanov Djamiykov, Atanas Georgiev Rusev

Department of electronics, TU-Sofia, Kl. Ohridsky blv. 8, 1756 - Sofia, Bulgaria,
(+3592)9652142, e-mail: tsd@tu-sofia.bg

This paper presents the technical decision of tachometer which is realized with program logic. A distinguish feature of the suggestion is the using functional transformation of the number next impulses in order to reach a result in rpm. This way there is no need oft mathematical operations multiplication and division. The device is programmed with language from upper level VHDL on Xilinx's crystals.

Keywords: Tachometer, programmable logic, CPLD, VHDL.

1. INTRODUCTION

The cyclometer is important part of every contemporary car. They use different principles of measurement. The information for the end-user is also displayed by many different ways. When we talk about the mass end-user manufactured cars the cyclometers are used mainly as additional source of information for the driver. If we take a look on the racing cars we will see that there this indication is really important, as it is used as main source of information for the decision when to switch to another gear. The main purpose here is to have the maximum acceleration of the car.

The input signal of the device is delivered from the ignition system of the engine by a coil. The frequency of the pulses of this signal is multiplied by the coefficient of four stroke gasoline engine with carburetor, which is 30. The received number is the actual number of rotations of the engine.

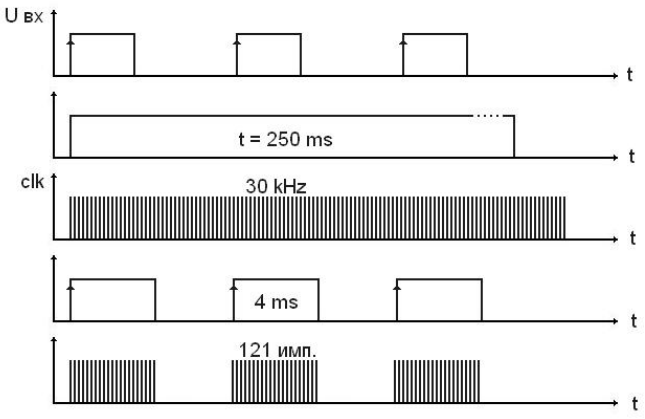

Fig. 1. Time diagram of input and clock signals

When in the input of the cyclometer comes a signal two time periods are started – one 4 ms long and one 250 ms long. At the same time a clock signal with frequency 30 kHz is generated. During the time of the long period (250 ms) every input pulse of the device start new 4 ms period. While this time window is opened the number of the 30 kHz pulses is counted. If at the end of the long time window (250 ms) a short period hasn't been finished the counting is stopped despite this fact. The total number of pulses is a number which is proportional to the number of revolutions of the engine per minute. E.g. if we have 100 revolutions of the engine per minute this number is 1033, if we have 4000

revolutions the number is 4033 etc. The little differences of the numbers do not count for the end result as additional processing is used before the information is displayed.

2. PROCESSING ALGORITHM

The controlling part of the device transforms the information from the coil (which is used as inductive sensor) in driving signals for the LED indication of the device. By the assistance of programmed counters, the main clock signal for the device is divided to an internal clock with frequency of 30 kHz. Two other counters create the 4 and 250 ms intervals. The first long time window is started with the first rising edge of the signal from the coil and it is opened for 250 ms exactly without considering the other changes of this signal. After this the window is closed and opened again with the next rising edge of the signal from the coil. The 4 ms windows are opened every time when new rising edge of the coil signal is detected. After the end of this period the window is closed and when new rising edge of the same signal comes new short window is opened. While both windows (4 and 250 ms) are opened a 14 bit counter is used to count the number of pulses from the internal clock source (30 kHz). When the 250 ms window is closed, the counted number is transferred to internal buffer register and after this the 14 bit counter is reset. After this the counted number is given to a decoding table and from the output of this table are given the driving signals for the LED indication.

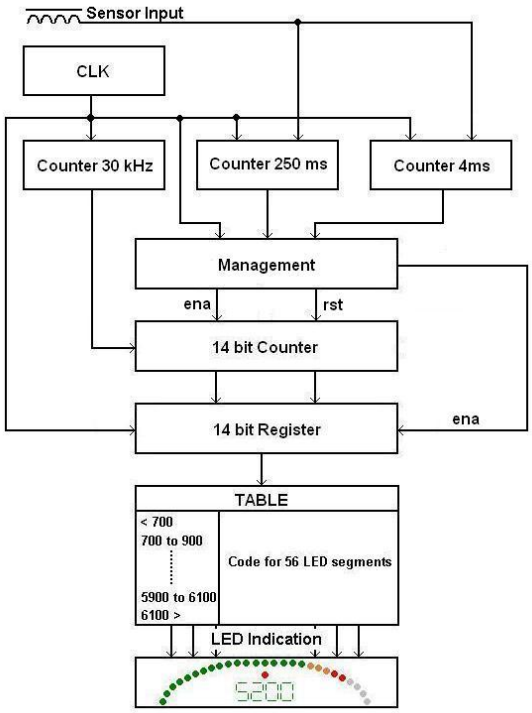

Fig 2. Functional scheme of device's working

Description of the functions of the internal blocks and the signals transferred between them.

3. INSIDE BLOCKS

Management – This is the main control block of the device. It watches the time windows and gives enable signal for the 14 bit counter when both are opened. When a rising edge of the long time window is detected it resets the 14 bit counter. This is also the block that stops the counting when the long time window is closed and enables the transferring of the result to the register which controls the decoding table. This block is designed as a Moore Finite State Machine.

Counter 30kHz – this block divides the main clock of the device (1 MHz) to a 30 KHz clock used for the processing of the input signal.

Counter 250ms – When a rising edge of the sensor signal is detected this block enables the 14 bit counter for the pulses. The block has a 1 bit output which is set to logical high level when the window is opened and is set to logical low level when the window is closed. While the counting is enabled the input signal from the coil sensor is not considered by this block. When the counting ends a new rising edge of the coil sensor signal is waited.

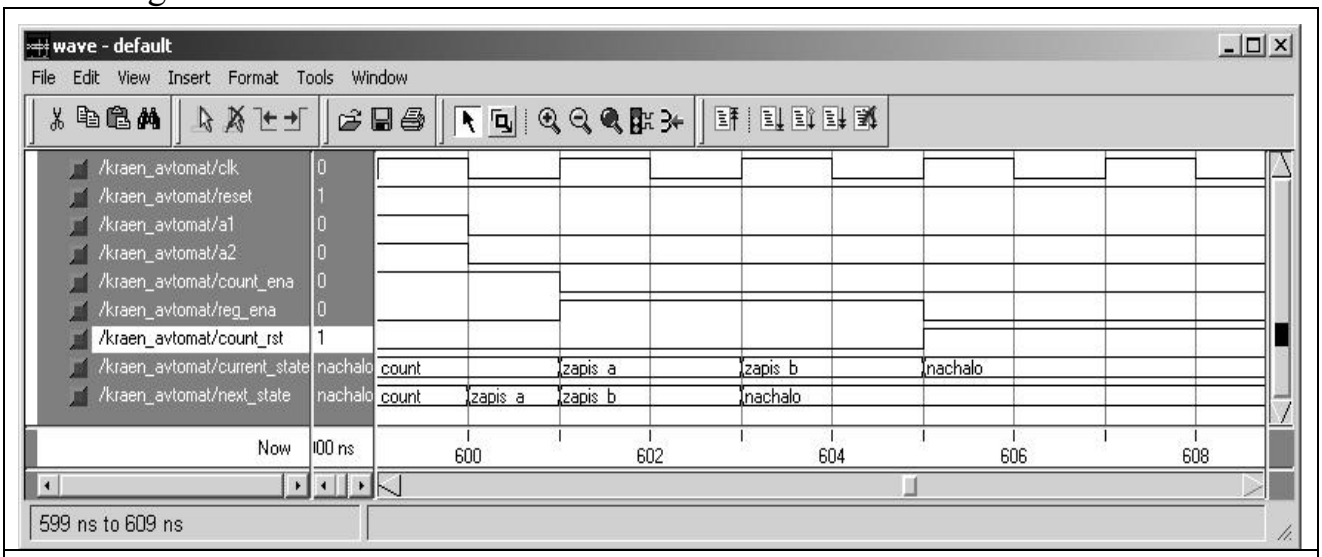

Fig 3. Simulation of block management

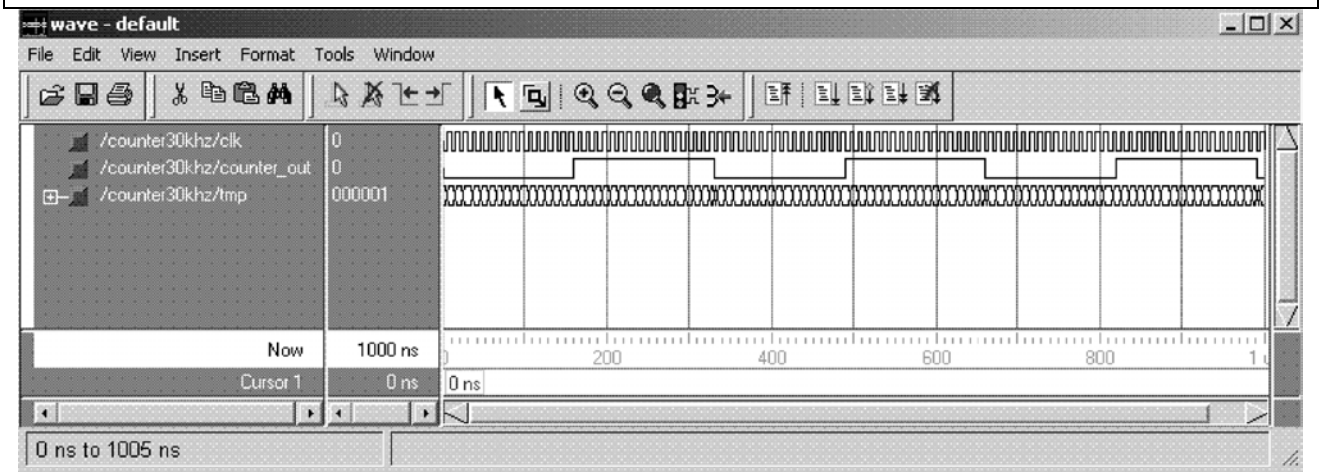

Fig 4. Simulation of block counter 30kHz.

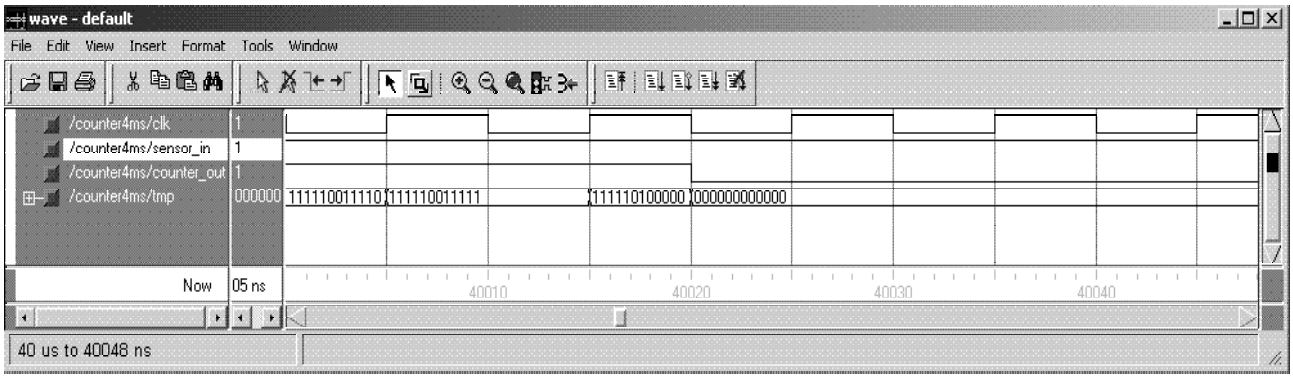

Fig 5. Simulation of block counter 4ms.

Counter 4ms – This block also has a 1 bit output, which is set to logical high with the rising edge of the coil sensor signal and after 4 ms is set to logical low. This counter is enabled with every rising edge of the input coil pulses.

Counter 14 bit – This block counts the clocks with the special internal frequency of 30 KHz when both 4 and 250 ms windows are opened. When the short window is closed this counter waits its new opening ant then it resumes the counting. The enable and reset signals of this counter are generated by the Management block.

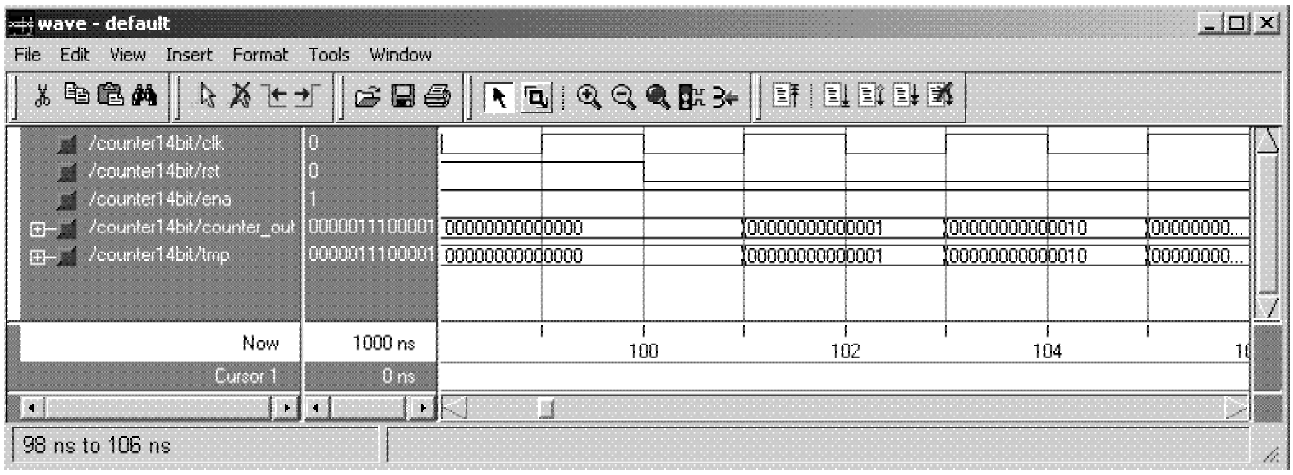

Fig 6. Simulation of block counter 14 bit.

14 bit register - This block stores the result of the 14 bit counter. It's content drives the decoding table. The write enable signal is given by the Management block.

Table – This block contains a list of driving signals for the Led indication. The number which is given as it's input is actually used as an address to fields which contain 56 bit code for the indication. The information from the output of the table is taken out of the scheme by a 56 bit parallel bus by the pins of the CPLD.

4. RESULTS FROM THE SIMULATION OF THE WHOLE DESIGN

After turning on and checking of each component's normal functions of the measuring system, is realized working simulation of the entire device. Fig 7 shows the result of the simulation, where all program components work properly and one another act.

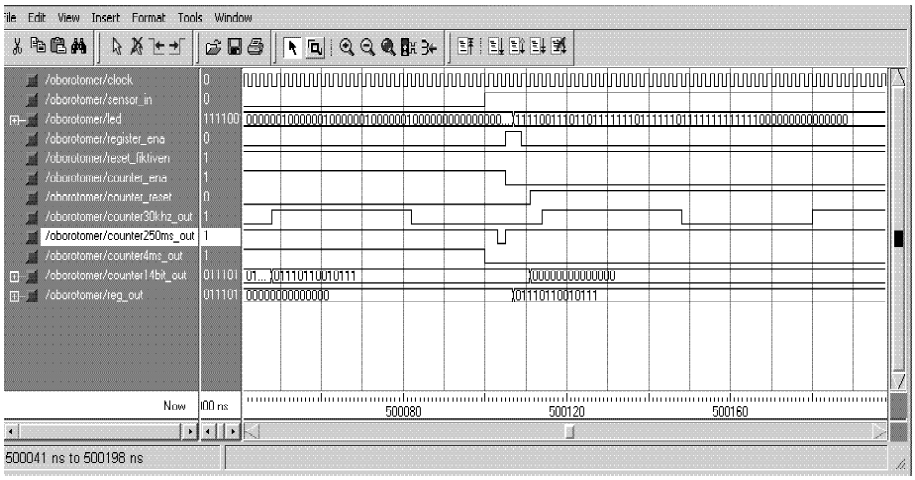

Fig 7. Simulation of the whole design.

The suggested model and the used realization can be used and in other measuring devices, in which the results are need to be indicated after the mathematical treatment of the input impulses' data.

5. REFERENCES

[1] Xilinx Incorporation, "The programmable logic data book," 2002.
[2] WWW.xilinx.com
[3] WWW.altera.com
[4] Guo-Ruey Tsai, Min-Chuan Lin, *FPGA-Based Reconfigurable Measurement Instruments with Functionality Defined by User*, EURASIP Journal on Applied Signal Processing, Volume 2006, Article ID 84340, Pages 1–14