ulm university universität uulm

Institute of Microelectronics

# Project Design of Integrated Systems

## Verilog Design of a Bicycle Computer

Your task is to implement the calculation unit of a bicycle computer, similar to the one you possibly have at your own bicycle. The features of the bicycle computer are described in section 1. Section 2 gives a short task description and section 3 describes the work strategy in more detail.

## 1 Description of the Bicycle Computer

### 1.1 Functionality of the Computer

The display of the bicycle computer is divided into two parts: In the upper part the instantaneous speed is displayed continuously.

In the lower part trip distance, average speed, trip time or maximum speed are displayed alternately. The required range and resolution of the displayed information can be taken from the table below. The wheel circumference is a constant value for the system and is defined in the global toplevel module.

| | abbreviation | range | resolution | unit |
|---|---|---|---|---|
| instantaneous speed | KMH | 0 – 99 | 1 | km / h |
| trip distance | DAY | 0.0 – 999.9 | 0.1 | km |
| average speed | AVS | 0.0 – 99.9 | 0.1 | km / h |
| trip time | TIM | 00:00 – 59:59<br>01:00 – 99:59 | 1<br>1 | min:s<br>h:min |
| maximum speed | MAX | 0 – 99 | 1 | km / h |
| wheel circumference | CIRC | 100 – 255 | 1 | cm |

When the mode-key is pressed, the lower part of the display switches from one information to the next. After pressing the mode-key the corresponding information has to be displayed as soon as possible. For the computation of the average speed the periods in which the bicycle did not move (< 5km/h) have to be left out automatically. The value of the average speed is related to trip distance and trip time. By pressing the reset-key trip distance, average speed, trip time and maximum speed are reset to 0 and the displayed mode is set to trip distance.

To save power (the computer runs on batteries) the displayed values are updated not continuously but only once every second. However, after pressing the mode-key a new value has to be calculated and displayed immediately. Furthermore only the information which is actually displayed has to be re-calculated, if necessary. Handling of overflows and similar exceptional situations/events have to be handled comprehensibly and consistently.

## 1.2 Inputs to the Computer

The bicycle computer has the following input signals (see figure on the next page):

### 1.2.1 Clock Signal

The whole system is clocked with a generated clock signal with frequency $f = 2.048$ kHz.

### 1.2.2 Impulses Coming from the Reed Contact

At this input the computer receives the impulses provided by a Reed contact. Normally the signal has the logic level ‘0’. Once per wheel revolution (when a magnet fixed on a spoke passes the Reed contact) this Reed contact generates one impulse (changing the signal value from ‘0’ to ‘1’ and after one clock period back from ‘1’ to ‘0’).

### 1.2.3 Mode- and Reset-Key

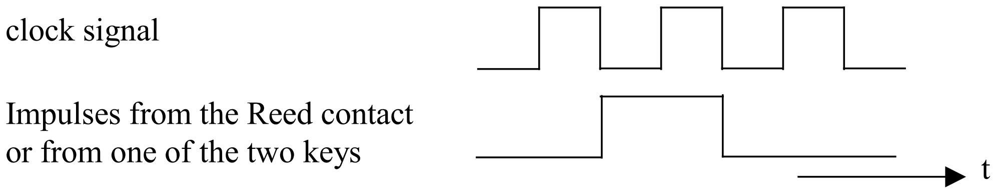
The computer has two keys:

- Mode-Key: The displayed mode switches between trip distance, average speed, trip time and maximum speed.
- Reset-Key: Trip distance, average speed, trip time and maximum speed are reset to 0 and the displayed mode is set to trip distance.

### 1.2.4 Wheel Circumference

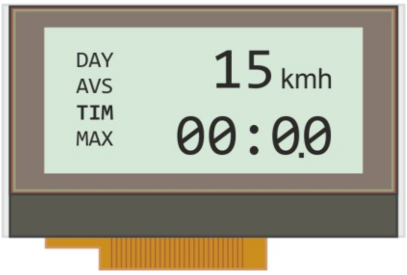
The wheel circumference in cm is defined by assigning the corresponding value to a constant signal inside the design (see table above for range and resolution).

You can assume that the impulses coming from the Reed contact, the mode-key and the reset-key are “ideal” which means: the pulse width is approximately one clock cycle and the signal changes its value always on the falling edge of the clock signal (as shown in the figure below).



## 1.3 Outputs of the Computer

The information calculated by the computer is displayed by an LCD display as shown below:

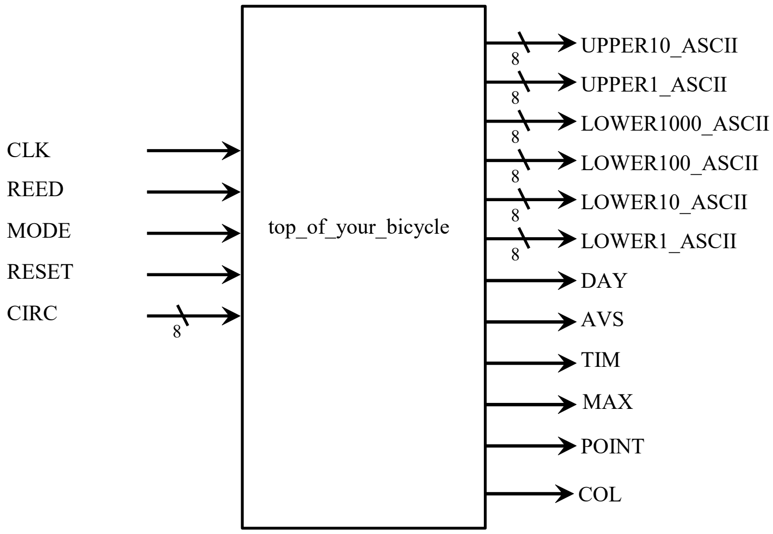


In addition to the values also the information about which mode is displayed currently has to be indicated by setting one of the signals DAY, AVS, TIM or MAX to '1' and all others to '0'. However, if the value of the instantaneous speed exceeds 65 km/h, the three remaining signals are not set to zero but have to blink (0.5 seconds on, 1 second off).

A decimal point (POINT) has to be displayed when showing trip distance or average speed. When the trip time is displayed, a ':' delimiter (COL) toggles every 1.0 seconds (1.0 sec '1', 1.0 sec '0'). The format min:sec is used for trip time values smaller than one hour, and the format hour:min otherwise.

## 1.4 Port Names

Use the following port names for the in- and outputs of your system.



## 1.5 Given Modules

To convert binary numbers to the 8-bit ASCII format, the modules `dual2bcd` and `bcd2ascii` can be used.

Furthermore, as toplevel of the bicycle computer, the module `global_toplevel` is given. Within this module, `pulse_shape`, `clk_divider_bike`, `lcd_controller` and `top_of_your_bicycle` are instantiated. The first module is responsible for shaping the input signals as discussed on page 2, the second one is used to create the 2.048 kHz clock signal, the third contains the display driver, and the latter one contains your code.

## 2 Task Description

The calculation unit of a bicycle computer as described in the previous section has to be developed and modeled with the hardware description language Verilog. The Verilog description has to be verified for correct functionality and synthesizability. The result of the synthesis process using an FPGA library from XILINX has to be simulated again to prove the correct functionality. This self-made bicycle computer is then tested under real time conditions and compared against a commercial product.

## 3 Individual Closing Discussion

After the bicycle computer is finished, a per-group, individual closing discussion about the project together with the supervisors is mandatory. For this discussion, you have to

- update the slide of your presentation outlining your final system architecture
- be able to explain the functionality of your bicycle computer (modules)
- be able to explain the functionality of your source code if being asked
- be able to explain the basics about Verilog and hardware design if being asked

## 4 Strategy

At first, the system architecture has to be developed. Define the functional modules (controller, arithmetic units, counters, etc.) that are needed for the implementation and determine their connections between each other. **The final system should become as small (area consumption) as possible**. Sketch the whole system and shortly describe the functionality of each module. Take into consideration that concurrent events like an impulse from the Reed contact and an impulse from the mode-key may occur.

Each group has to present its developed system architecture with a few slides (5-10 minutes). Groups that fail to present their solution will be excluded from the lab course.

Refine the description of the modules to the register transfer level by using registers, adders, multiplexers etc. Determine the bit-width of all connecting signals according to the speciation.

Describe all modules of the bicycle computer in Verilog and perform simulations to verify their correct functionality. Start your verification with single modules. You can use the stop watch design from the exercises as an example. Generate a bit file for the FPGA and compete your system against a commercial device.

Lastly, you have to pass a final discussion with the supervisors. Update the outline of your system architecture according to your final design and bring it as a printout to the final discussion.

**Note**

- Document the solutions of the tasks and show them to your supervisor.
- Discuss your ideas with your supervisor before you start the description with Verilog!

**Solve all given tasks only together with your partner.**
**Do this with respect to the rules of scientific working of the University of Ulm.**
**Usage of third party material is plagiarism and results in immediate exclusion of the lab course!**

**Good luck for your work!**