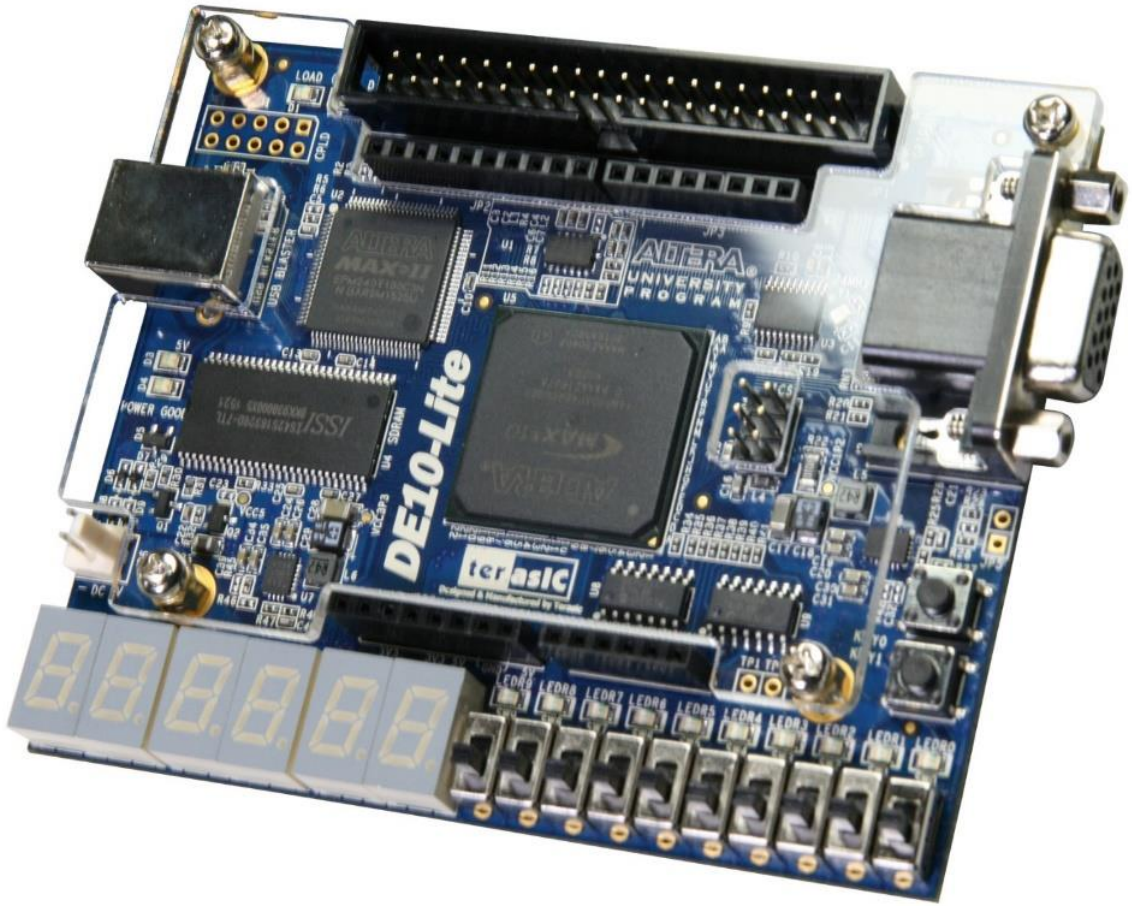
# FPGA Stopwatch




Made by Vitaly Okolelov

Email: okolelovvitaly@gmail.com

GitHub: https://github.com/okolelovv

LinkedIn: https://www.Linkedin.com/in/vitaly-okolelov-715699194/

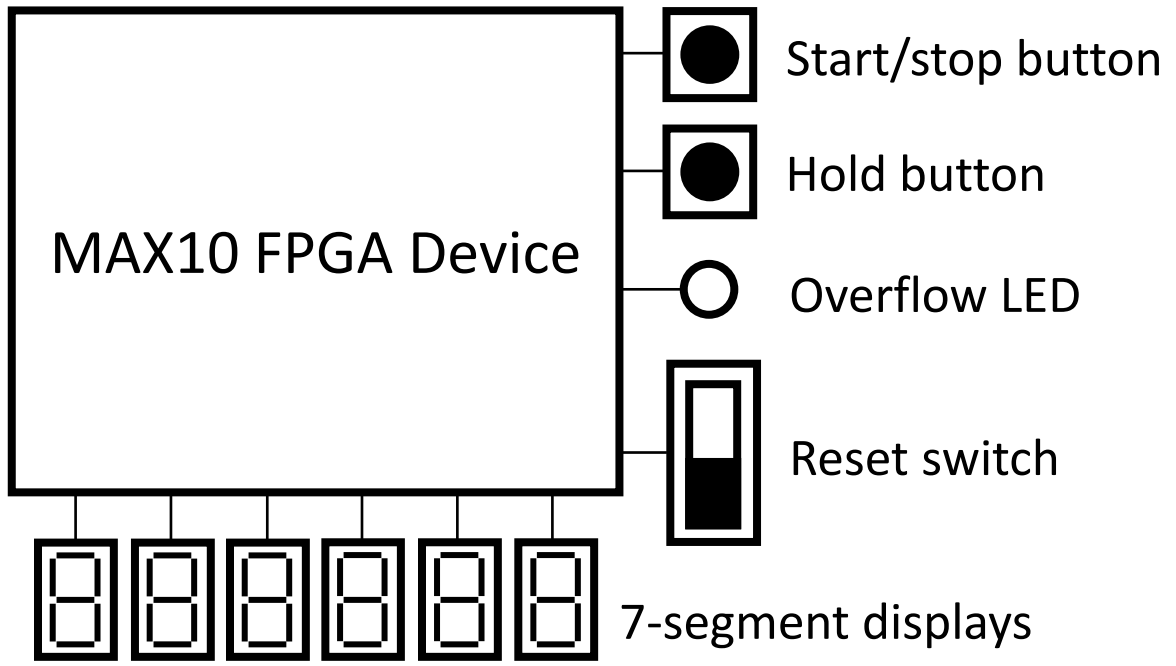
# Functionality

Control signals:

- Start/stop: A falling edge starts the timer, next falling edge stops it
- Hold: Logic 0 pauses the timer; logic 1 enables it
- Reset: Logic 0 pulls the timer to 0 and keeps it there

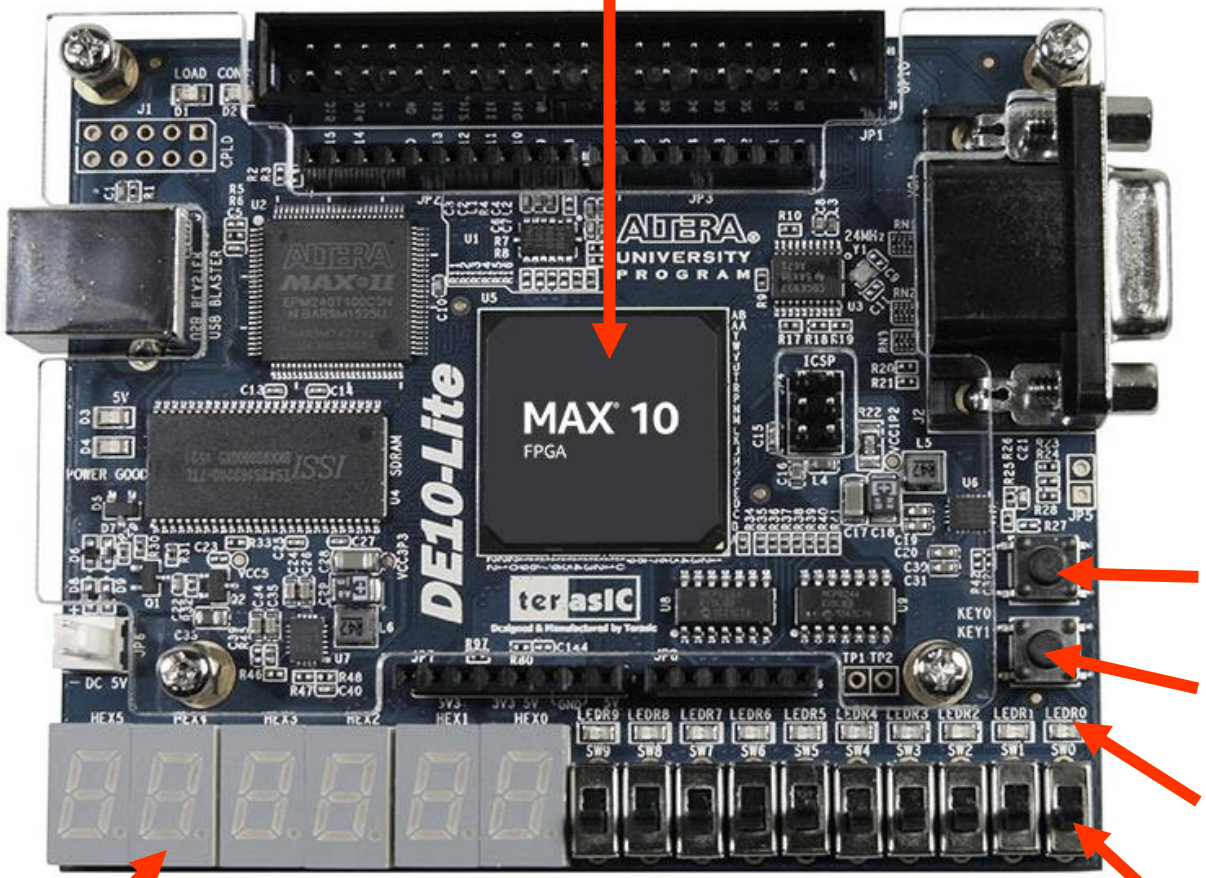
Output:

- Six 7-segment displays outputs minutes, seconds and tenths of seconds
- If the timer goes above its maximum range (99:59:99), the overflow LED is on



# FPGA device

MAX 10 10M50DAF484C7G chip




Start/stop button

Hold button

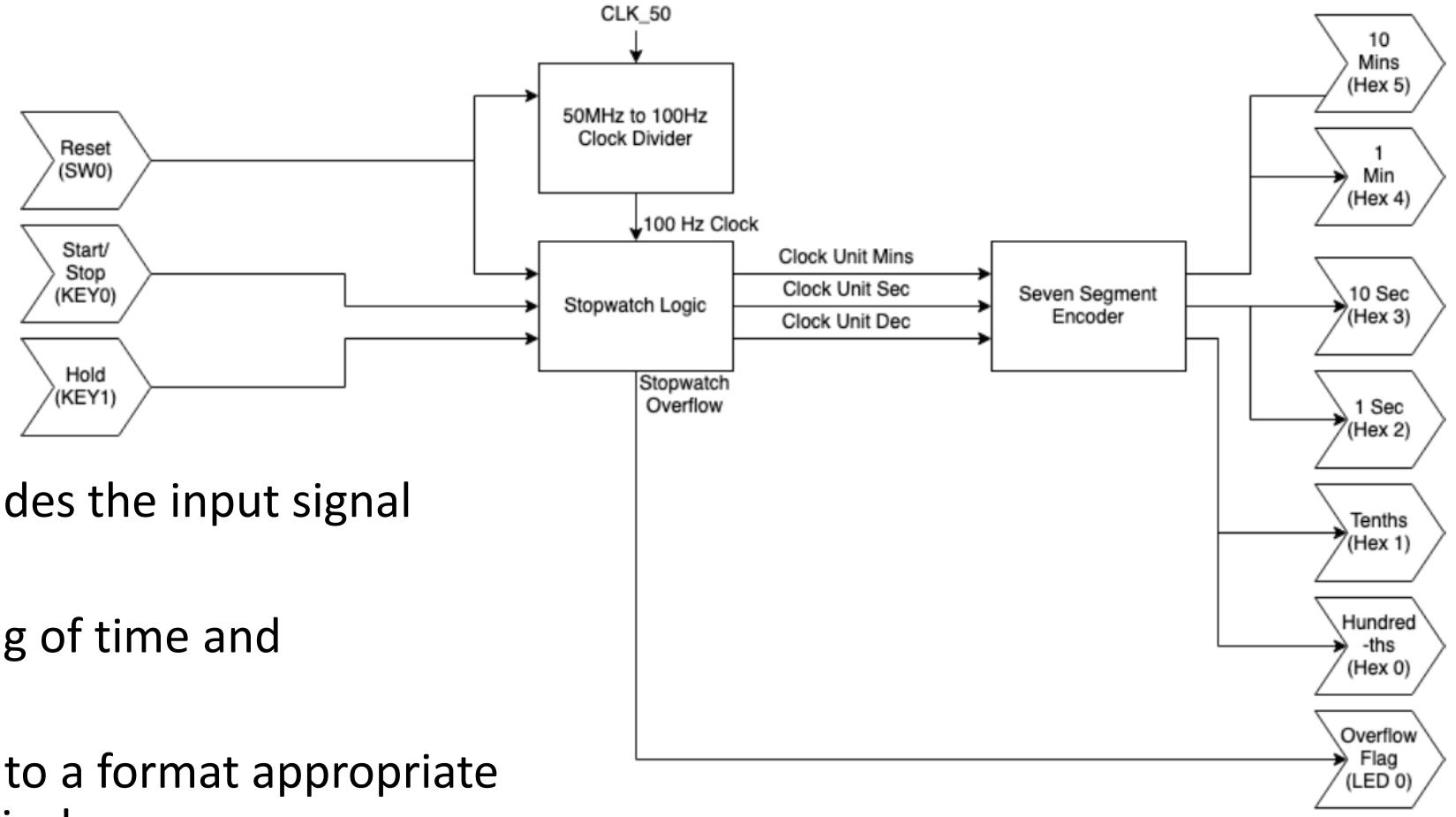
Overflow LED

Reset switch

7-segment displays

DE-10 Lite Board

# Modules and signals




***ClockDivider50MHzTo100Hz*** – divides the input signal frequency by 500,000

***StopwatchLogic*** – handles counting of time and start/stop/reset functionality

***SevenSegEncoder*** – converts time to a format appropriate for display on six seven-segment displays

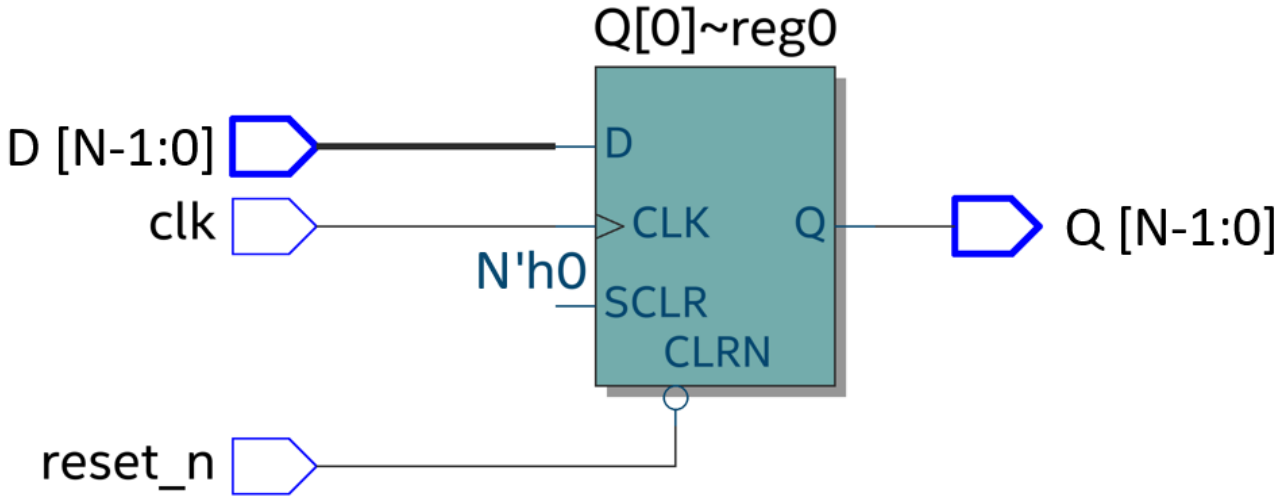
***Stopwatch*** – Top level design

Vitaly Okolelov

# Combination vs behavioural logic

When using behavioural logic, synthesis often results in having flip-flops in the design. The main type of flip-flops used in this project the RS-type flip-flop which is a D-type flip-flop that supports asynchronous set and reset.

Having this module allowed to significantly simplify the design and break it down into purely combinational logic.



# Features of the Verilog code

To make module designs reusable, the following advanced (not yet studied at the university) techniques were used:

- Parameters
- Replication operator
- Generate blocks
- Tasks
- $clog2 function

# Design verification

- ModelSim was used for design verification
- Testbenches written in Verilog
- Different techniques used depending on situation:
  - Directed testing
  - Random testing
  - Constrained random testing

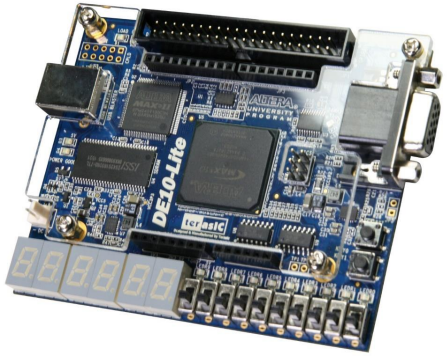
Results:

- Functional correctness was verified
- The accuracy of counting was confirmed down to nanoseconds

# Pin mapping and programming the FPGA Device

Project was completed in Quartus software:

- Pin Planner tool was used for pin assignment
- Then, the design was loaded on FPGA device and tested for correct behaviour




Vitaly Okolelov

# Thank you!